

**Aula 00 - (Prof. Paolla  
Ramos e Raphael  
Lacerda)**

*IPM-JP (Analista Previdenciário - Analista  
de Redes e Comunicação)*

*Desenvolvimento de Software - 2024*

*Autor:  
(Pós-Edital)*  
**Felipe Mathias, Paolla Ramos**

20 de Outubro de 2024

# Índice

1) Desenvolvimento de Software - Apresentação do Professor .....	3
2) Apresentação Flashcards .....	5
3) Desenvolvimento Front-End - HTML - Teoria .....	7
4) Desenvolvimento Front-End - HTML - Resumo .....	73
5) Desenvolvimento Front-End - HTML - Questões Comentadas - MULTIBANCAS .....	92
6) Desenvolvimento Front-End - HTML - Lista de Questões - MULTIBANCAS .....	140



## APRESENTAÇÃO

# PROF. PAOLLA RAMOS

FORMADA EM SISTEMAS DE INFORMAÇÃO PELA  
UNIVERSIDADE FEDERAL DE OURO PRETO,  
PÓS-GRADUADA EM SISTEMAS DE INFORMAÇÃO  
DIREITO TRIBUTÁRIO  
DIREITO ADMINISTRATIVO  
AUDITORA FISCAL ESPECIALISTA EM TI.



Olá, pessoal!! Meu nome é Paolla Ramos, sou Auditora Fiscal especialista em TI do ISS-Aracaju. Trabalhar nesse fisco incrível tem sido uma experiência fantástica!!  
Pessoal, eu sou uma pessoa normal, assim como vocês. No início, achava que conquistar a aprovação em um concurso de alto nível era quase impossível, até que provei o contrário! Querem saber qual foi o segredo? Foi o hiper foco, galera! Não existe uma fórmula mágica, e eu nunca fui considerada "superinteligente" ou a primeira aluna na turma. No entanto, sempre fui **MUITO DETERMINADA, PERSISTENTE.**

A equipe de TI e eu estamos aprimorando nossas aulas de forma gradativa para oferecer o melhor conteúdo possível. Sabemos que o estudo pode ser complexo, especialmente por meio de livros eletrônicos, por isso, recomendo estudar em conjunto com as vídeo-aulas.



Além disso, informo que estamos trabalhando na atualização dos cursos neste exato momento! Estamos refazendo a formatação, adicionando questões e diagramas, entre outros aprimoramentos. Gradualmente, os cursos ficaram mais completos e aprofundados. E, para acompanhar as tecnologias mais recentes, novas videoaulas também estão a caminho.

Caso surja alguma demanda, não hesitem em contactar no fórum. Se preferirem, também podem entrar em contato pelo Instagram [@prof.paollaramos](https://www.instagram.com/prof.paollaramos). Eu amo ajudar os alunos e estou disponível para esclarecer quaisquer dúvidas que possam surgir.

A minha missão aqui é dar o meu melhor para ajudar cada um de vocês a conquistar a aprovação também! Podem contar comigo sempre que precisarem.

Então, minha ideia aqui é fazer o meu melhor para que você também consiga ser aprovado! Sempre que precisar, pode contar comigo. Meu instagram é:

 [@prof.paollaramos](https://www.instagram.com/prof.paollaramos)





# ESTRATÉGIA FLASHCARDS

📖 Você tem dificuldade de estudar, memorizar e revisar os conteúdos que estuda em nossas aulas? Então nós temos a ferramenta perfeita para você!

Apresentamos o **Estratégia Cards**: app de flashcards que vai revolucionar sua forma de **estudar** e **revisar** conteúdos de provas de concurso público. Com nossa tecnologia inovadora e interface amigável, você dominará os tópicos mais complexos de maneira eficiente e divertida.

## 🌟 Recursos do Estratégia Cards:

<b>Curadoria de Flashcards</b>	Flashcards criados e revisados por professores especializados em cada área, com qualidade e voltados para concursos públicos.
<b>Flashcards Personalizados</b>	Crie seus próprios flashcards, cobrindo os principais tópicos e matérias dos concursos públicos.
<b>Repetição Espaçada</b>	Técnica de aprendizagem que envolve revisar informações em intervalos crescentes para melhorar a retenção de longo prazo e combater o esquecimento.
<b>Estatísticas Personalizadas</b>	Visualize graficamente o percentual de acertos, erros ou dúvidas dos decks estudados.
<b>Modo Offline</b>	Estude em qualquer lugar, mesmo sem conexão à internet, fazendo o download dos decks.
<b>Estudo por Áudio</b>	<i>Está dirigindo ou fazendo esteira e quer continuar estudando?</i> Basta utilizar a opção “Escutar”.
<b>Decks Favoritos</b>	Você pode escolher decks específicos como favoritos e visualizá-los em uma aba separada do app.
<b>Opções de Estudo</b>	Você poderá estudar todos os cards de um deck; ou apenas os que você errou; ou apenas os que você não estudou ainda; entre outras opções.

## 📱 E como eu consigo baixar?



É muito fácil! Basta pesquisar por “Estratégia Cards” na loja oficial do seu smartphone.

Se você tiver um Android, basta acessar a **Google Play**;



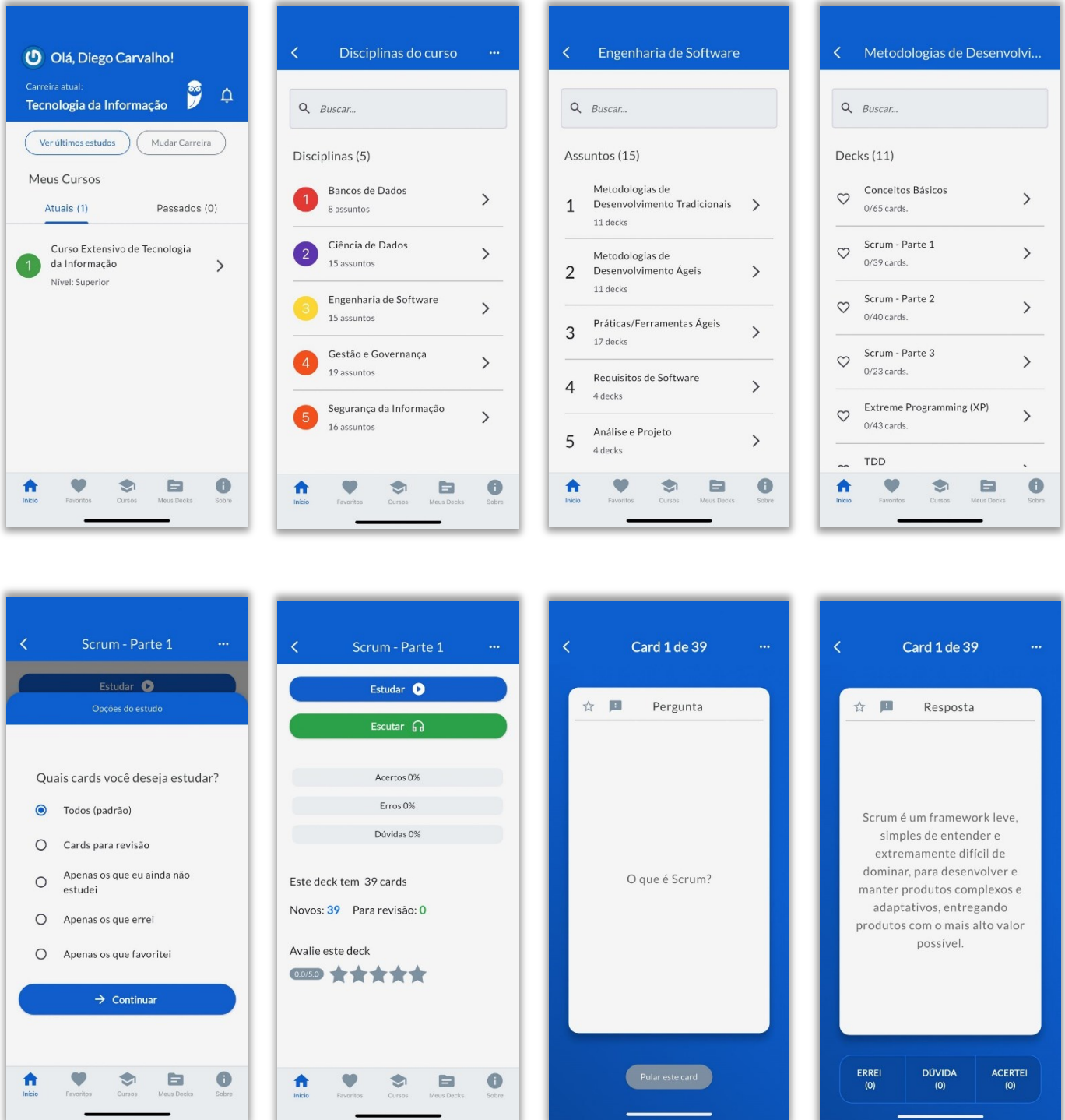
Se for tiver um iPhone, basta acessar a **App Store (iOS)**.



## É para acessar?

Para acessar, basta ter uma conta no Estratégia Concursos. Em seguida, utilize suas credenciais de login e senha para acessar o aplicativo. Por fim, acessa a carreira de Tecnologia da Informação.

## Como utilizar o app:



## Sumário

Apresentação da Aula	3
HTML	4
Conceitos Básicos	5
HTML Básico	8
Elementos HTML	12
Atributos HTML	14
Cabeçalhos HTML	15
Parágrafos HTML	16
Estilos HTML	18
Formatação de Texto HTML	21
Citação em HTML	22
Comentários HTML	24
Cores HTML	25
Links HTML	27
Imagens HTML	29
Tabelas HTML	33
Estilização de Tabelas HTML	36
Listas HTML	37
Listas Não Ordenadas	38
Listas Ordenadas em HTML	41
Bloco HTML e elementos embutidos	44
Elementos de nível de bloco	44
Elementos inline	44
Atributo de classe HTML	47
Formulários HTML	50
Os elementos HTML <form>	53
Principais Características	58
APIs HTML	59
API de geolocalização HTML	59
API de armazenamento da Web em HTML	59



RESUMO	62
REFERÊNCIAS	63
Questões Comentadas	64
Questões Cespe	65
Gabarito	103



# HTML

## Conceitos Básicos

Antes de iniciar a aula, é importante mencionar que vários exemplos dessa aula foram retirados ou inspirados em exemplos do W3Tutorials ([www.w3schools.com/html](http://www.w3schools.com/html)). **Não fizemos isso porque somos preguiçosos, mas por dois motivos: (1) os exemplos são excelentes; (2) essa é uma das fontes de inspiração das bancas.** Além disso, eu sugiro que vocês tenham sempre aberta uma janela com um interpretador online para que vocês possam testar o que veremos. Recomendo esse:

[HTTPS://WWW.W3SCHOOLS.COM/HTML/](https://www.w3schools.com/html/)

```
<!DOCTYPE html>
<html>
<body>

<h1>Meu primeiro título.</h1>

<p>Meu primeiro parágrafo.</p>

</body>
</html>
```

### Meu primeiro título.

Meu primeiro parágrafo.

O que é HTML?

- HTML significa **Hyper Text Markup Language**.
- HTML é a **linguagem de marcação padrão para criar páginas da Web**;
- HTML descreve a estrutura de uma página da Web;
- HTML consiste em uma série de elementos;
- Os elementos HTML informam ao navegador como exibir o conteúdo;
- Os elementos HTML rotulam partes do conteúdo como "isto é um título", "isto é um parágrafo", "isto é um link", etc.

HTML (abreviação para a expressão inglesa **HyperText Markup Language**, que significa: "**Linguagem de Marcação de Hipertexto**") é uma linguagem de marcação utilizada **na construção de páginas na Web**. Documentos HTML podem ser interpretados por navegadores. A tecnologia é fruto da junção entre os padrões **HyTime** e **SGML**<sup>1</sup>.

<sup>1</sup> HyTime é um padrão para a representação estruturada de hipermídia e conteúdo baseado em tempo. Um documento é visto como um conjunto de eventos concorrentes dependentes de tempo (como áudio, vídeo, etc.), conectados por hiperligações (em inglês: hyperlink e link). O padrão é independente de outros padrões de processamento de texto em geral.

SGML é um padrão de formatação de textos. Não foi desenvolvido para hipertexto, mas tornou-se conveniente para transformar documentos em hiper-objetos e para descrever as ligações.



(TCE RJ– 2022) HTML5 é uma linguagem de programação que permite estruturar páginas web e executar comandos como loops de repetição, por exemplo.

**Comentários:** HTML é a linguagem de marcação padrão para criar páginas da Web, ela descreve a estrutura de uma página da Web, porém, não executa comandos como loops de repetição. (Gabarito: Errado).

Vejamos um exemplo de um documento HTML Simples além da visualização da Estrutura da Página HTML gerada.

```
<!DOCTYPE html>
<html>
<body>

<h1>Meu primeiro título.</h1>

<p>Meu primeiro parágrafo.</p>

</body>
</html>
```

## Meu primeiro título.

Meu primeiro parágrafo.

- A declaração `<!DOCTYPE html>` define que este documento é um documento **HTML5**;
- O elemento `<html>` é o elemento raiz de uma página HTML;
- O elemento `<head>` contém metainformações sobre a página HTML;
- O elemento `<title>` especifica um título para a página HTML (que é mostrado na barra de título do navegador ou na guia da página);
- O elemento `<body>` define o corpo do documento e é um recipiente para todos os conteúdos visíveis, como cabeçalhos, parágrafos, imagens, hiperlinks, tabelas, listas, etc.
- O elemento `<h1>` define um título grande;
- O elemento `<p>` define um parágrafo.

(FGV –IBGE – 2016) A declaração `<!DOCTYPE>` permite ao navegador apresentar uma página web corretamente. A declaração correta para uma página em HTML5 é:

- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 5.0 Strict//EN" "http://www.w3.org/TR/html5/strict.dtd">`
- `<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 5.0 Transitional//EN" "http://www.w3.org/TR/html5/loose.dtd">`
- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 5.0">`
- `<!DOCTYPE html>`
- `<!DOCTYPE XML PUBLIC "-//W3C//DTD HTML 5.0">`

**Comentários:** A declaração `<!DOCTYPE html>` define que este documento é um documento HTML5. (Gabarito: Letra D)



O objetivo de um navegador da Web (Chrome, Edge, Firefox, Safari) é ler documentos HTML e exibi-los corretamente. Um navegador não exibe as tags HTML, mas as utiliza para determinar como exibir o documento:

HTML5 é a versão mais recente do HTML e possui vários novos recursos e elementos que permitem aos desenvolvedores da Web criar conteúdo mais interativo e atraente para a Web.

Alguns dos novos recursos do HTML5 incluem:

- Elementos semânticos: o HTML5 apresenta novos elementos com significados específicos, como `<header>`, `<footer>` e `<article>`. Esses elementos tornam mais fácil para os desenvolvedores estruturar seu conteúdo de maneira lógica e significativa.
- Suporte de áudio e vídeo: o HTML5 inclui suporte integrado para reprodução de áudio e vídeo, eliminando a necessidade de plug-ins de terceiros, como o Flash.
- Elemento Canvas: O elemento `<canvas>` permite aos desenvolvedores desenhar gráficos e animações diretamente no navegador, usando JavaScript.
- Armazenamento off-line: o HTML5 fornece uma maneira de os aplicativos da Web armazenarem dados localmente, para que os usuários ainda possam acessar o aplicativo quando estiverem off-line.
- Controles de formulário aprimorados: o HTML5 apresenta novos controles de formulário e tipos de entrada, como data, e-mail e intervalo, o que torna mais fácil para os desenvolvedores criar formulários amigáveis.

O HTML5 foi projetado para ser compatível com as versões anteriores do HTML, portanto, a maioria dos sites existentes continuará a funcionar como antes. No entanto, usando os novos recursos do HTML5, os desenvolvedores podem criar conteúdo mais interativo e atraente para a web.

Os elementos semânticos são elementos HTML usados para adicionar significado à página da Web, em vez de controlar o layout ou a aparência da página. Eles são usados para descrever o conteúdo da página da Web de maneira significativa para humanos e máquinas.

Vejamos uma lista de alguns elementos semânticos comuns em HTML:

- `<header>`: representa o cabeçalho de uma página da web ou seção de uma página da web.
- `<nav>`: representa uma seção da página da Web que contém links de navegação.



- `<main>`: Representa o conteúdo principal da página web.
- `< article >`: representa um conteúdo independente, como uma postagem de blog ou artigo de notícias.
- `<section>`: Representa uma seção da página da web, como um capítulo ou um tema.
- `<aside>`: Representa o conteúdo que está relacionado ao conteúdo principal da página da web, mas que não é essencial para a compreensão do conteúdo principal.
- `<footer>`: representa o rodapé de uma página da web ou seção de uma página da web.

Os elementos semânticos são importantes para melhorar a acessibilidade de uma página da Web, pois fornecem contexto e significado ao conteúdo da página. Eles também ajudam os mecanismos de pesquisa a entender o conteúdo da página da Web e a melhorar sua classificação nos resultados da pesquisa.

É importante observar que os elementos semânticos não devem ser usados para fins de estilo ou layout. Se você deseja controlar o layout ou a aparência da página da Web, use CSS.

```
<html>
  <cabeça>
    <title>Título da página</title>
  </head>
  <corpo>
    <h1>Este é um título</h1>
    <p>Isto é um parágrafo.</p>
    <p>Este é outro parágrafo.</p>
  </body>
</html>
```





**HYPER TEXT MARKUP  
LANGUAGE - HTML**

LINGUAGEM DE MARCAÇÃO PADRÃO PARA CRIAR PÁGINAS DA WEB

DESCREVE A ESTRUTURA DE UMA PÁGINA DA WEB

CONSISTE EM UMA SÉRIE DE ELEMENTOS

ELEMENTOS HTML INFORMAM AO NAVEGADOR COMO EXIBIR O CONTEÚDO E ROTULAM PARTES DO CONTEÚDO COMO "ISTO É UM TÍTULO", "ISTO É UM PARÁGRAFO", "ISTO É UM LINK", ETC.



## HTML Básico

Todos os documentos HTML devem começar com uma declaração de tipo de documento: `<!DOCTYPE html>`. O próprio documento HTML começa `<html>` e termina com `</html>`. A parte visível do documento HTML está entre `<body>` e `</body>`.

```
<!DOCTYPE html>
<html>
<body>

<h1>Meu primeiro título.</h1>

<p>Meu primeiro parágrafo.</p>

</body>
</html>
```

### Meu primeiro título.

Meu primeiro parágrafo.

A declaração `<!DOCTYPE>` representa o tipo de documento e ajuda os navegadores a exibir páginas da Web corretamente. Ele deve aparecer apenas uma vez, no topo da página (antes de qualquer tag HTML). A declaração `<!DOCTYPE>` não diferencia maiúsculas de minúsculas. Vejamos a declaração para HTML5:

```
<!DOCTYPE html>
```

Os cabeçalhos HTML são definidos com as tags `<h1>` a `<h6>`. `<h1>` define o cabeçalho **mais importante**. Enquanto `<h6>` define o título **menos importante**:

```
<h1>Este é o título 1</h1>
<h2>Este é o título 2</h2>
<h3>Este é o título 3</h3>
```

### Este é o título 1

Este é o título 2

Este é o título 3

Os parágrafos HTML são definidos com a tag `<p>`:



<pre>&lt;p&gt;Isto é um parágrafo.&lt;/p&gt; &lt;p&gt;Este é outro parágrafo.&lt;/p&gt;</pre>	Isto é um parágrafo. Este é outro parágrafo.
---	---

Os links HTML são definidos com a tag <a>:

<pre>&lt;a href="https://www.w3schools.com"&gt;Este é um link&lt;/a&gt;</pre>	<a href="https://www.w3schools.com">Este é um link</a>
---	--

O destino do link é especificado no atributo `href`. Os atributos são usados para fornecer informações adicionais sobre os elementos HTML. Em HTML, o atributo "href" é usado para especificar um link para uma página da web ou um local específico dentro de uma página da web. É comumente usado com o elemento <a> para criar um hiperlink para outra página da web ou com o elemento <base> para especificar a URL base para todos os links relativos em uma página da web. Vejamos um exemplo do atributo "href" sendo usado com o elemento <a> para criar um hiperlink:

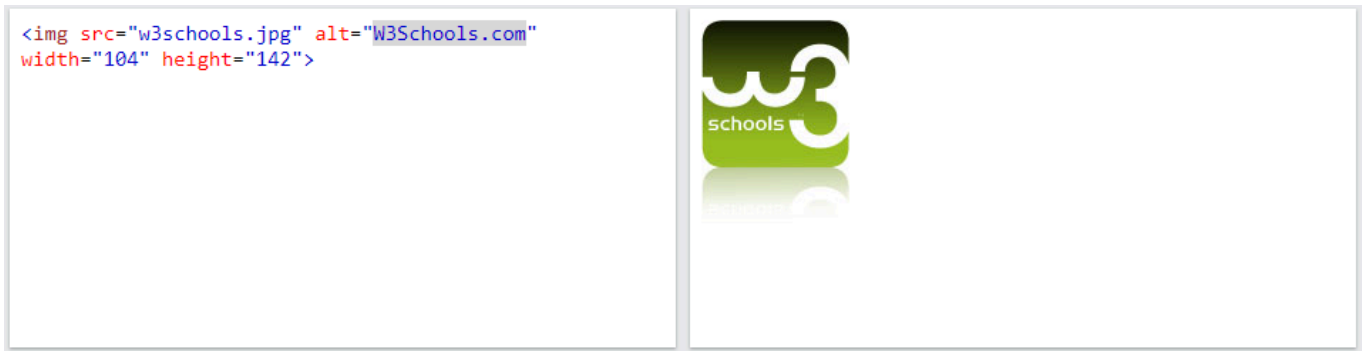
```
<a href="https://www.example.com">This is a link</a>
```

Neste exemplo, o texto "Este é um link" será exibido na página da Web e, ao clicar nele, o usuário será direcionado para a URL "https://www.example.com". Aqui está um exemplo do atributo "href" sendo usado com o elemento <base>:

```
<base href="https://www.example.com/resources/">
```

As imagens HTML são definidas com a tag <img>. O arquivo de origem ( src), texto alternativo ( alt), width e height são fornecidos como atributos:



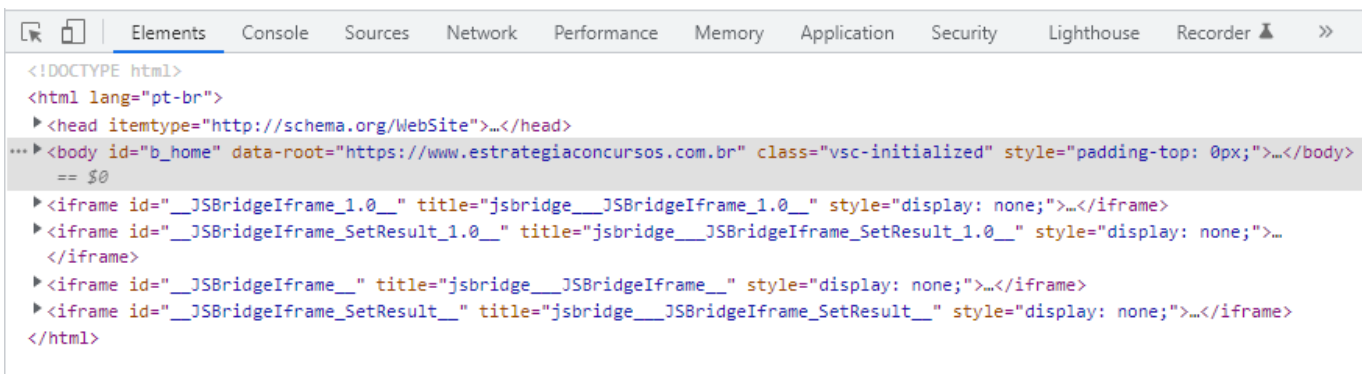


Para verificar o código-fonte de uma página HTML, clique com o botão direito do mouse em uma página HTML e selecione "Exibir fonte da página" (no Chrome) ou "Exibir fonte" (no Edge) ou similar em outros navegadores. Isso abrirá uma janela contendo o código-fonte HTML da página.

```
<!DOCTYPE html>
<html lang="pt-br">
<head itemtype="http://schema.org/WebSite">
<script>
function(f,b,e,v,n,t,s){if(f.fbq)return;
n=f.fbq=function(){n.callMethod? n.callMethod.apply(n,arguments):n.queue.push(arguments)};
if(!f._fbq)f._fbq=n;n.push=n;n.loaded=!0;n.version='2.0';n.queue=[];t=b.createElement(e);t.async=!0;
t.src=v;s=b.getElementsByTagName(e)[0];s.parentNode.insertBefore(t,s)}(window,document,'script','https://connect.facebook.net/en_US/fbevents.js');
fbq('init','1751780821700636');
fbq('track','PageView');
</script>
</script>
<noscript>

</noscript>
<script type="application/ld+json">
{
"@context": "http://schema.org",
"@type": "WebSite",
"■ name": "Estratégia Concursos",
"alternateName": "Estratégia Concursos",
"url": "https://www.estrategiaconcursos.com.br/"
}
</script>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge"><script type="text/javascript">(window.NREUM||(NREUM={})).init={ajax:{deny_list:["bam.nr-data.net"]}};(window.NREUM||(NREUM={
!function(t,e){"object"==typeof exports&&"object"==typeof module?module.exports=e():"function"==typeof define&&define.amd?define([],e):"object"==typeof exports?exports.NRBA=e():t
<meta name="description" content="Estratégia Concursos é referência na preparação de alunos para Concursos Públicos. Estude com nossos Cursos Online para Concursos, 100% focados
```

Por outro lado, para inspecionar um elemento HTML, você deve clicar com o botão direito do mouse em um elemento (ou em uma área em branco) e escolha "Inspecionar" ou "Inspecionar elemento" para ver de que elementos são feitos (você verá o HTML e o CSS). Você também pode editar o HTML ou CSS dinamicamente no painel Elementos ou Estilos que é aberto.



## Elementos HTML

Um elemento HTML é definido por uma tag inicial, algum conteúdo e uma tag final. O elemento HTML é tudo, desde a tag inicial até a tag final:

```
< tagname > O conteúdo vai aqui... < /tagname >  
< h1 > Meu primeiro cabeçalho < /h1 >  
< p > Meu primeiro parágrafo. < /p >
```

Tag inicial	conteúdo do elemento	Tag final
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>
 	<i>none</i>	<i>none</i>

Os elementos HTML podem ser aninhados (isso significa que os elementos podem conter outros elementos). Todos os documentos HTML consistem em elementos HTML aninhados. O exemplo abaixo contém quatro elementos HTML ( , <html>e <body>):<h1><p>

```
<!DOCTYPE html>  
<html>  
<body>  
  
<h1>Meu primeiro título.</h1>  
  
<p>Meu primeiro parágrafo.</p>  
  
</body>  
</html>
```

### Meu primeiro título.

Meu primeiro parágrafo.

O elemento <html> é o elemento raiz e define todo o documento HTML. Ele tem uma tag <html> inicial e uma tag </html> final. Então, dentro do <html>elemento existe um elemento <body> que define o corpo do documento. Ele tem uma tag inicial <body>e uma tag final </body>. Então, no exemplo acima, dentro do elemento <body> existem dois outros elementos: <h1>e <p>. O elemento <h1> define um cabeçalho e possui uma tag inicial <h1>e uma tag final </h1>, assim como ocorre com o elemento <p>. Há elementos html que não precisam da tag final. Mas a boa prática de programação diz: "Nunca pule a tag final!" 😊

Como disse, há elementos que não precisam da tag final! São conhecidos como elementos HTML sem conteúdo ou elementos vazios. A tag <br> define uma **quebra de linha** e é um **elemento vazio sem uma tag de fechamento**. Veja o exemplo a seguir.



```
<!DOCTYPE html>
<html>
<body>

<p>Este é um parágrafo <br> com uma quebra de linha.
</p>

</body>
</html>
```

Este é um parágrafo  
com uma quebra de linha.

HTML não diferencia maiúsculas de minúsculas. As tags HTML não diferenciam maiúsculas de minúsculas: <P>significa o mesmo que <p>.

O padrão HTML **não requer tags em letras minúsculas**, mas é recomendado usar letras minúsculas em HTML e exige letras minúsculas para tipos de documento mais restritos, como XHTML.



## Atributos HTML

Todos os elementos HTML podem ter atributos. Os atributos fornecem informações adicionais sobre os elementos, e são sempre especificados na tag de início. Os atributos geralmente vêm em **pares de nome/valor** como: `name="value"`.

A tag `<a>` define um **hiperlink**. O `href` atributo especifica a URL da página para a qual o link vai:

<pre>&lt;a href="https://www.w3schools.com"&gt;Este é um link&lt;/a&gt;</pre>	<a href="https://www.w3schools.com">Este é um link</a>
---	--

O atributo `preload` é um atributo HTML usado para especificar como um elemento de mídia (como um elemento `<audio>` ou `<video>`) deve ser carregado quando uma página da web é carregada. O atributo `controls` é um atributo HTML usado para exibir os controles padrão do reproduutor de mídia para um elemento de mídia.

O atributo `preload` tem três valores possíveis:

- "auto": A mídia será carregada automaticamente quando a página da web for carregada.
- " metadata ": Somente os metadados (por exemplo, duração, dimensões) da mídia serão carregados quando a página da web for carregada.
- " none ": a mídia não será carregada quando a página da web for carregada.

O atributo `controls` é um atributo booleano, o que significa que não possui um valor. Se o atributo `controls` estiver presente, os controles do reproduutor de mídia serão exibidos e, se não estiver presente, os controles não serão exibidos.

Lembre-se de que os atributos de `preload` e `controls` são **opcionais** e você pode usá-los para personalizar o comportamento do elemento de mídia de acordo com suas necessidades.



## Cabeçalhos HTML

Cabeçalhos HTML são títulos ou subtítulos que você deseja exibir em uma página da web. Os cabeçalhos HTML são definidos com as tags `<h1>` até `<h6>`. Relembrando, `<h1>` define o cabeçalho mais importante. `<h6>` define o título menos importante.

<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;h1&gt;Título 1&lt;/h1&gt; &lt;h2&gt;Título 2&lt;/h2&gt; &lt;h3&gt;Título 3&lt;/h3&gt; &lt;h4&gt;Título 4&lt;/h4&gt; &lt;h5&gt;Título 5&lt;/h5&gt; &lt;h6&gt;Título 6&lt;/h6&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<p><b>Título 1</b></p> <p><b>Título 2</b></p> <p>Título 3</p> <p>Título 4</p> <p>Título 5</p> <p>Título 6</p>
--	---

Os mecanismos de pesquisa usam os cabeçalhos para indexar a estrutura e o conteúdo de suas páginas da web. Os usuários geralmente percorrem uma página por seus títulos. É importante usar cabeçalhos para mostrar a estrutura do documento. Os cabeçalhos `<h1>` devem ser usados para os cabeçalhos **principais**, seguidos pelos cabeçalhos `<h2>`, depois os menos importantes `<h3>` e assim por diante.

Cada título HTML tem um tamanho padrão. No entanto, você pode especificar o tamanho de qualquer cabeçalho com o atributo `style`, usando a propriedade `font-size` CSS:

<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;h1 style="font-size:60px;"&gt;Título 1&lt;/h1&gt;  &lt;p&gt;Você pode alterar o tamanho de um título com o atributo style, usando a propriedade font-size.&lt;/p&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<p><b>Título 1</b></p> <p>Você pode alterar o tamanho de um título com o atributo <code>style</code>, usando a propriedade <code>font-size</code>.</p>
---	--





## Parágrafos HTML

Um parágrafo sempre começa em uma nova linha e geralmente é um bloco de texto. O elemento HTML `<p>` define um parágrafo. Um parágrafo sempre começa em uma nova linha e os navegadores adicionam automaticamente algum espaço em branco (uma margem) antes e depois de um parágrafo.

```
<!DOCTYPE html>
<html>
<body>

<p>Isto é um parágrafo.</p>
<p>Isto é um parágrafo.</p>
<p>Isto é um parágrafo.</p>

</body>
</html>
```

Isto é um parágrafo.

Isto é um parágrafo.

Isto é um parágrafo.

A tag `<hr>` define uma quebra temática em uma página HTML e geralmente é exibida como uma régua horizontal. O elemento `<hr>` é usado para separar o conteúdo (ou definir uma alteração) em uma página HTML. A tag `<hr>` é uma tag vazia, o que significa que não tem tag final.

```
<!DOCTYPE html>
<html>
<body>

<h1>Este é o título 1</h1>
<p>Este é um texto.</p>
<hr>

<h2>Este é o título 2</h2>
<p>Este é outro texto.</p>
<hr>

<h2>Este é o título 2</h2>
<p>Este é outro texto.</p>

</body>
</html>
```

# Este é o título 1

Este é um texto.

---

## Este é o título 2

Este é outro texto.

---

## Este é o título 2

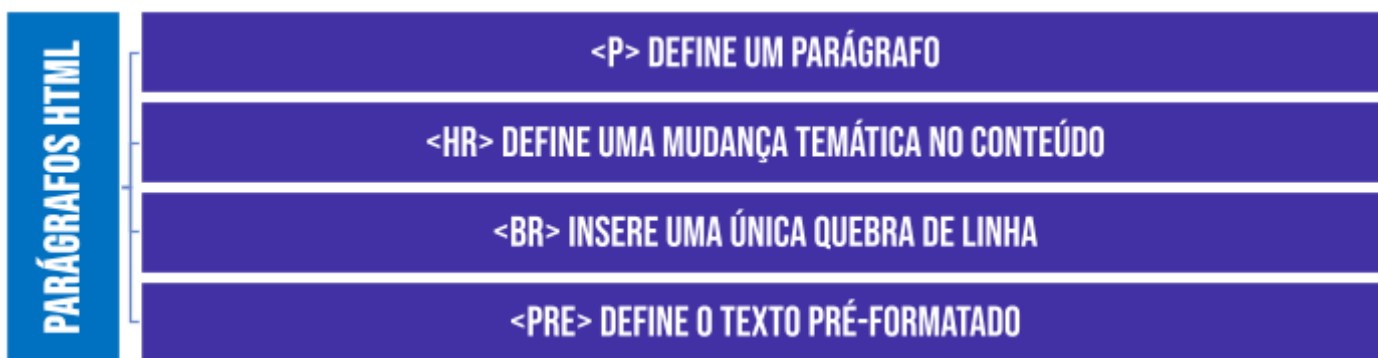
Este é outro texto.



O elemento HTML `<br>` define uma quebra de linha. Use `<br>` se quiser uma quebra de linha (uma nova linha) sem iniciar um novo parágrafo. A tag `<br>` é uma tag vazia, o que significa que não tem tag final.

O elemento HTML `<pre>` define o texto pré-formatado. O texto dentro de um elemento `<pre>` é exibido em uma fonte de largura fixa (geralmente Courier) e preserva espaços e quebras de linha:

<pre>&lt;pre&gt; O que você vai estudar amanhã?  Quantos dias você tem até sua próxima prova?  Qual sua meta de aproveitamento na próxima prova? &lt;/pre&gt;</pre>	<pre>O que você vai estudar amanhã?  Quantos dias você tem até sua próxima prova?  Qual sua meta de aproveitamento na próxima prova?</pre>
---	--



(FCC – SABESP – 2018) Um Técnico está desenvolvendo uma página web com HTML5 e deseja exibir um texto com fonte Courier de largura fixa, preservando os espaços e as quebras de linha.

O texto deverá ser colocado entre as tags

- `<mark>` e `</mark>`
- `<dl>` e `</dl>`
- `<embed>` e `</embed>`
- `<code>` e `</code>`
- `<pre>` e `</pre>`.

**Comentários:** O elemento HTML `<pre>` define o texto pré-formatado. O texto dentro de um elemento `<pre>` é exibido em uma fonte de largura fixa (geralmente Courier) e preserva espaços e quebras de linha. (Gabarito: Letra E)



## Estilos HTML

O atributo HTML style é usado para adicionar estilos a um elemento, como cor, fonte, tamanho e muito mais. A definição do estilo de um elemento HTML pode ser feita com o atributo style.

```
<!DOCTYPE html>
<html>
<body>

<p>Eu sou normal</p>
<p style="color:red;">Eu sou vermelho</p>
<p style="color:blue;">Eu sou azul</p>
<p style="font-size:50px;">Eu sou grande</p>

</body>
</html>
```

Eu sou normal

Eu sou vermelho

Eu sou azul

Eu sou grande

O atributo HTML style tem a seguinte sintaxe<sup>2</sup>:

```
<tagname style="property.value;">
```

A propriedade CSS background-color define a cor de fundo de um elemento HTML.

```
<!DOCTYPE html>
<html>
<body style="background-color:powderblue;">

<h1>Este é um título</h1>
<p>Isso é um parágrafo.</p>

</body>
</html>
```

**Este é um título**

Isso é um parágrafo.

<sup>2</sup> *property* é uma propriedade CSS. *value* é um valor CSS.



```
<!DOCTYPE html>
<html>
<body>

<h1 style="background-color:powderblue;">Este é um título</h1>
<p style="background-color:tomato;">Isso é um parágrafo.</p>

</body>
</html>
```

## Este é um título

Isso é um parágrafo.

Vejamos as outras propriedades style em sequência:

Propriedade CSS	Descrição
Background-color	Define a cor de fundo de um elemento HTML
Color	Define a cor do texto para um elemento HTML
Font-family	Define a fonte a ser usada para um elemento HTML
Font-size	Define o tamanho do texto para um elemento HTML
Text-align	Define o alinhamento horizontal do texto para um elemento HTML

```
<!DOCTYPE html>
<html>
<body>

<h1 style="background-color:powderblue;">Cabeçalho background-color powderblue</h1>
<p style="background-color:tomato;">Parágrafo background-color tomato</p>
<h1 style="color:blue;">Cabeçalho com a fonte blue</h1>
<p style="color:red;">Parágrafo com a fonte red</p>
<h1 style="font-family:verdana;">Cabeçalho com a fonte verdana</h1>
<p style="font-family:courier;">Parágrafo com a fonte courier</p>
<h1 style="font-size:150%;">Cabeçalho tamanho 150%</h1>
<p style="font-size:160%;">Parágrafo tamanho 160%</p>
<h1 style="text-align:center;">Cabeçalho centralizado</h1>
<p style="text-align:center;">Parágrafo centralizado</p>

</body>
</html>
```

## Cabeçalho background-color powderblue

Parágrafo background-color tomato

## Cabeçalho com a fonte blue

Parágrafo com a fonte red

## Cabeçalho com a fonte verdana

Parágrafo com a fonte courier

## Cabeçalho tamanho 150%

Parágrafo tamanho 160%

## Cabeçalho centralizado

Parágrafo centralizado



## PARÁGRAFOS HTML

**STYLE DEFINE ESTILOS PARA ELEMENTOS HTML**

**BACKGROUND-COLOR DEFINE A COR DO FUNDO**

**COLOR DEFINE CORES DO TEXTO**

**FONT-FAMILY DEFINE FONTES DO TEXTO**

**FONT-SIZE DEFINE TAMANHOS DO TEXTO**

**TEXT-ALIGN DEFINE O ALINHAMENTO DO TEXTO**



## Formatação de Texto HTML

O HTML contém vários elementos para definir o texto com um significado especial.

Elementos de Formatação	Descrição
<code>&lt;b&gt;</code>	Texto em negrito, sem nenhuma importância extra.
<code>&lt;strong&gt;</code>	Texto importante. O conteúdo interno geralmente é exibido em negrito.
<code>&lt;i&gt;</code>	Define uma parte do texto em uma voz ou humor alternativo. Texto em itálico. Costuma ser usada para indicar um termo técnico, uma frase de outro idioma, um pensamento, o nome de um navio, etc.
<code>&lt;em&gt;</code>	Texto enfatizado. O conteúdo interno geralmente é exibido em itálico.
<code>&lt;mark&gt;</code>	Texto marcado. Define o texto que deve ser marcado ou destacado
<code>&lt;small&gt;</code>	Texto menor
<code>&lt;del&gt;</code>	Define o texto que foi excluído de um documento. Os navegadores geralmente traçam uma linha através do texto excluído.
<code>&lt;ins&gt;</code>	Texto inserido. Os navegadores geralmente sublinham o texto inserido.
<code>&lt;sub&gt;</code>	Texto subscrito. O texto subscrito aparece meio caractere abaixo da linha normal e, às vezes, é renderizado em uma fonte menor. O texto subscrito pode ser usado para fórmulas químicas, como H <sub>2</sub> O
<code>&lt;sup&gt;</code>	Texto sobrescrito. O texto sobrescrito aparece meio caractere acima da linha normal e, às vezes, é renderizado em uma fonte menor. O texto sobrescrito pode ser usado para notas de rodapé, como WWW <sup>[1]</sup> .

ELEMENTOS DE FORMATAÇÃO HTML	<code>&lt;B&gt;</code> TEXTO EM NEGRITO
	<code>&lt;STRONG&gt;</code> TEXTO IMPORTANTE
	<code>&lt;I&gt;</code> TEXTO EM ITÁLICO
	<code>&lt;EM&gt;</code> TEXTO ENFATIZADO
	<code>&lt;MARK&gt;</code> TEXTO MARCADO
	<code>&lt;SMALL&gt;</code> TEXTO MENOR
	<code>&lt;DEL&gt;</code> TEXTO DELETADO
	<code>&lt;INS&gt;</code> TEXTO INSERIDO
	<code>&lt;SUB&gt;</code> TEXTO SUBSCRITO
	<code>&lt;SUP&gt;</code> TEXTO SOBRESCRITO



## Citação em HTML

O elemento HTML `<blockquote>` define uma seção que é citada de outra fonte. Os navegadores geralmente indentam `<blockquote>` elementos.

```
<!DOCTYPE html>
<html>
<body>

<p>Depoimentos de Alunos: Paolla Ramos - Aprovada no ISS Aracaju</p>

<blockquote cite="https://www.estrategiaconcursos.com.br/assinaturas-platinum/">

Já estudava há um tempo, porém não conseguia ver a minha evolução. Tinha medo de não estar fazendo a coisa certa. Assim que foi lançada a Platinum me inscrevi e fui acompanhada por pessoas incríveis que me ajudaram imensamente do início ao fim! Foi o diferencial para minha aprovação no primeiro concurso para Auditor Fiscal de TI que fiz! Eu aprendi o que fazer, como melhorar meus estudos e consequentemente meu rendimento. Para mim foi essencial poder contar com o apoio dos coaches!
</blockquote>

</body>
</html>
```

Depoimentos de Alunos: Paolla Ramos - Aprovada no ISS Aracaju

Já estudava há um tempo, porém não conseguia ver a minha evolução. Tinha medo de não estar fazendo a coisa certa. Assim que foi lançada a Platinum me inscrevi e fui acompanhada por pessoas incríveis que me ajudaram imensamente do início ao fim! Foi o diferencial para minha aprovação no primeiro concurso para Auditor Fiscal de TI que fiz! Eu aprendi o que fazer, como melhorar meus estudos e consequentemente meu rendimento. Para mim foi essencial poder contar com o apoio dos coaches!

A tag HTML `<q>` define uma citação curta. Os navegadores normalmente inserem aspas ao redor da citação.

```
<!DOCTYPE html>
<html>
<body>

<p>Os navegadores geralmente inserem aspas ao redor do elemento q.</p>

<p> Estratégia Concursos: <q>Conheça o curso online que mais aprova nos maiores concursos do país.</q>
</p>

</body>
</html>
```

Os navegadores geralmente inserem aspas ao redor do elemento q.

Estratégia Concursos: “Conheça o curso online que mais aprova nos maiores concursos do país.”

A tag HTML `<abbr>` define uma abreviação ou um acrônimo, como "HTML", "CSS", "Mr.", "Dr.", "ASAP", "ATM". As abreviações de marcação podem fornecer informações úteis para navegadores, sistemas de tradução e mecanismos de pesquisa.

A tag HTML `<address>` define as informações de contato do autor/proprietário de um documento ou artigo. As informações de contato podem ser um endereço de e-mail, URL, endereço físico, número de telefone, identificador de mídia social, etc. O texto no elemento `<address>` geralmente é renderizado em itálico e os navegadores sempre adicionam uma quebra de linha antes e depois do elemento `<address>`.



A tag HTML <cite> define o título de uma obra criativa (por exemplo, um livro, um poema, uma música, um filme, uma pintura, uma escultura, etc. O texto no elemento <cite> geralmente é renderizado em itálico .

BDO significa Bi-Directional Override. A tag HTML <bdo> é usada para substituir a direção do texto atual:

<b>CITAÇÃO EM HTML</b>	<b>&lt;ABBR&gt; DEFINE UMA ABREVIÇÃO OU ACRÔNIMO</b>
	<b>&lt;ADDRESS&gt; DEFINE AS INFORMAÇÕES DE CONTATO DO AUTOR/PROPRIETÁRIO DE UM DOCUMENTO</b>
	<b>&lt;BDO&gt; DEFINE A DIREÇÃO DO TEXTO</b>
	<b>&lt;BLOCKQUOTE&gt; DEFINE UMA SEÇÃO QUE É CITADA DE OUTRA FONTE</b>
	<b>&lt;CITE&gt; DEFINE O TÍTULO DE UMA OBRA</b>
	<b>&lt;Q&gt; DEFINE UMA CITAÇÃO CURTA EM LINHA</b>





## Comentários HTML

Os comentários HTML não são exibidos no navegador, mas podem ajudar a documentar seu código-fonte HTML. Você pode adicionar comentários à seu código fonte HTML usando a seguinte sintaxe:

```
<!-- Escreva seus comentários aqui -->
```

Observe que há um ponto de exclamação (!) na tag inicial, mas não na tag final. Com comentários, você pode colocar notificações e lembretes em seu código HTML.

```
<!-- Isto é um comentário -->  
  
<p>Isto é um parágrafo.</p>  
  
<!-- Lembre-se de adicionar mais informações aqui -->
```



## Cores HTML

As cores HTML são especificadas com nomes de cores predefinidos ou com valores RGB, HEX, HSL, RGBA ou HSLA. Em HTML, uma cor pode ser especificada usando um nome de cor, a cor de fundo dos elementos HTML, a cor do texto, a cor das bordas, e muito mais. Vejamos os exemplos!

```
<!DOCTYPE html>
<html>
<body>

<h1 style="background-color:Tomato;">Tomato</h1>
<h1 style="background-color:Orange;">Orange</h1>
<h1 style="background-color:DodgerBlue;">DodgerBlue</h1>
<h1 style="background-color:MediumSeaGreen;">MediumSeaGreen</h1>
<h1 style="background-color:Gray;">Gray</h1>
<h1 style="background-color:SlateBlue;">SlateBlue</h1>
<h1 style="background-color:Violet;">Violet</h1>
<h1 style="background-color:LightGray;">LightGray</h1>

</body>
</html>
```

**Tomato**

**Orange**

**DodgerBlue**

**MediumSeaGreen**

**Gray**

**SlateBlue**

**Violet**

**LightGray**

```
<!DOCTYPE html>
<html>
<body>

<h1 style="background-color:DodgerBlue;">Hello World</h1>

<p style="background-color:Tomato;">
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed diam nonummy nibh euismod tincidunt ut laoreet dolore
magna aliquam erat volutpat.
Ut wisi enim ad minim veniam, quis nostrud exerci tation
ullamcorper suscipit lobortis nisl ut aliquip ex ea
commodo consequat.
</p>

</body>
</html>
```

**Hello World**

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.



```
<!DOCTYPE html>
<html>
<body>

<!-- Cor do texto -->
<h3 style="color:Tomato;">Hello World</h3>

<p style="color:DodgerBlue;">Lorem ipsum dolor sit amet,
consectetuer adipiscing elit, sed diam nonummy nibh euismod
tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>

<p style="color:MediumSeaGreen;">Ut wisi enim ad minim veniam,
quis nostrud exerci tation ullamcorper suscipit lobortis nisl
ut aliquip ex ea commodo consequat.</p>

<!-- Cor da borda -->
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>

</body>
</html>
```

## Hello World

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Hello World

Hello World

Hello World

```
<!DOCTYPE html>
<html>
<body>

<p>Same as color name "Tomato":</p>

<h1 style="background-color:rgb(255, 99, 71);">rgb(255, 99, 71)</h1>
<h1 style="background-color:#ff6347;">#ff6347</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">hsl(9, 100%, 64%)</h1>

<p>Same as color name "Tomato", but 50% transparent:</p>
<h1 style="background-color:rgba(255, 99, 71, 0.5);">rgba(255, 99, 71, 0.5)
</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">hsla(9, 100%, 64%, 0.5)</h1>

<p>In addition to the predefined color names, colors can be specified using
RGB, HEX, HSL, or even transparent colors using RGBA or HSLA color values.
</p>

</body>
</html>
```

Same as color name "Tomato":

rgb(255, 99, 71)

#ff6347

hsl(9, 100%, 64%)

Same as color name "Tomato", but 50% transparent:

rgba(255, 99, 71, 0.5)

hsla(9, 100%, 64%, 0.5)

In addition to the predefined color names, colors can be specified using RGB, HEX, HSL, or even transparent colors using RGBA or HSLA color values.



## Links HTML

Links são encontrados em quase todas as páginas da web. Os links permitem que os usuários cliquem de uma página para outra. Links HTML são hiperlinks. Você pode clicar em um link e pular para outro documento. Quando você move o mouse sobre um link, a seta do mouse se transforma em uma mãozinha.

O **texto do link** é a parte que ficará visível para o leitor. Clicar no texto do link enviará o leitor ao endereço de URL especificado. Por padrão, os links aparecerão da seguinte maneira em todos os navegadores:

- Um link não visitado é sublinhado e azul
- Um link visitado é sublinhado e roxo
- Um link ativo é sublinhado e vermelho

Por padrão, a página vinculada será exibida **na janela atual do navegador**. Para alterar isso, você deve especificar outro destino para o link.

O atributo `target` especifica onde abrir o documento vinculado. Ele pode ter um dos seguintes valores:

- `_self`- Predefinição. Abre o documento na mesma janela/guia em que foi clicado
- `_blank`- Abre o documento em uma nova janela ou guia
- `_parent`- Abre o documento no quadro pai
- `_top`- Abre o documento em todo o corpo da janela

Para colocar um endereço de e-mail dentro de um link, você pode utilizar `mailto:` dentro do atributo `href` para criar um link que abra o programa de e-mail do usuário (para permitir que ele envie um novo e-mail):

```
<a href="mailto:professor@estrategia.com">Novo e-mail</a>
```

Para usar um botão HTML como um link, você precisa adicionar algum **código JavaScript**. O JavaScript permite que você **especifique o que acontece em determinados eventos**, como o clique de um botão:

```
<button onclick="document.location='default.asp'">HTML </ button >
```



## Imagens HTML

As imagens podem melhorar o design e a aparência de uma página da web. As imagens não são tecnicamente inseridas em uma página da web; mas sim vinculadas a páginas da web. A tag <img> cria um **espaço de retenção para a imagem referenciada**.

A tag <img> é vazia, ou seja, contém apenas atributos e não possui uma tag de fechamento. Além disso, ela possui dois atributos obrigatórios:

- **src** - Especifica o caminho para a imagem;
- **alt** - Especifica um texto alternativo para a imagem.

A tag <img> também deve conter os atributos **width** e **height**, que especificam a largura e a altura da imagem (em pixels). Quando uma página web carrega, é o navegador, nesse momento, que obtém a imagem de um servidor web e a insere na página. Portanto, certifique-se de que a imagem realmente permaneça no mesmo local em relação à página da Web, caso contrário, seus visitantes receberão um ícone de link quebrado. O ícone de link quebrado e o texto alt são mostrados se o navegador não conseguir encontrar a imagem.

```
<!DOCTYPE html>
<html>
<body>

<h2>0 atributo src</h2>
<p>As imagens HTML são definidas com a tag img e o nome do arquivo da fonte da imagem é especificado no atributo src:</p>



</body>
</html>
```

### O atributo src

As imagens HTML são definidas com a tag `img` e o nome do arquivo da fonte da imagem é especificado no atributo `src`:



Há duas maneiras de especificar a URL no `src` atributo:

1. **URL absoluta** - Links para uma imagem externa que está hospedada em outro site.
2. **URL relativo** - Links para uma imagem hospedada no site. Aqui, a URL não inclui o nome de domínio. Se a URL começar sem uma barra, ela será relativa à página atual. Exemplo: `src="img_girl.jpg"`. Se a URL começar com uma barra, ela será relativa ao domínio. Exemplo: `src="/images/img_girl.jpg"`.

Dica: quase sempre é melhor usar URLs relativos. Eles não vão quebrar se você mudar de domínio.



O atributo alt é obrigatório para a tag <img>. Ele especifica um texto alternativo para uma imagem, se a imagem por algum motivo não puder ser exibida. Isso pode ocorrer devido a uma conexão lenta ou a um erro no atributo src ou se o usuário usar um leitor de tela.

O atributo style é usado para **adicionar estilos a um elemento**, como cor, fonte, tamanho e muito mais. Por exemplo, você pode usar o atributo style para especificar a largura e a altura de uma imagem, especificando em sequência: `style="width:500px;height:600px"`

Os atributos width, height e style são todos válidos em HTML. No entanto, é sugerido o uso do atributo style para evitar que as folhas de estilo alterem o tamanho das imagens.

O atributo alt obrigatório fornece um texto alternativo para uma imagem, se o usuário por algum motivo não puder visualizá-la (devido a uma conexão lenta, um erro no atributo src ou se o usuário usar um leitor de tela). O valor do atributo alt deve descrever a imagem.

```
<!DOCTYPE html>
<html>
<body>

<h2>Alternative text</h2>

<p>O atributo alt deve refletir o conteúdo da imagem,
para que os usuários que não podem ver a imagem entendam
o que ela contém:</p>



</body>
</html>
```

### Alternative text

O atributo alt deve refletir o conteúdo da imagem, para que os usuários que não podem ver a imagem entendam o que ela contém:



O HTML permite GIFs animados. E também permite usar uma imagem como link, para isso, basta colocar a tag <img> dentro da tag <a>

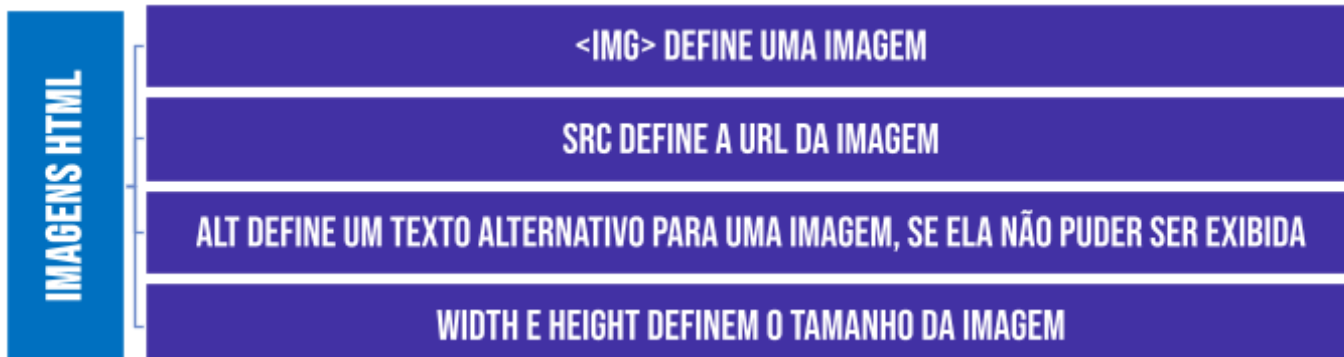
Vejamos os tipos de arquivo de imagem mais comuns, suportados em todos os navegadores (Chrome, Edge, Firefox, Safari, Opera):

Abreviação	Formato do arquivo	Extensão
APNG	Animated Portable Network Graphics	.apng
GIF	Graphics Interchange Format	.gif
ICO	Microsoft Icon	.ico, .cur





JPEG	Joint Photographic Expert Group image	.jpg, .jpeg, .jfif, .pjpeg, .jpp
PNG	Portable Network Graphics	.png
SVG	Scalable Vector Graphics	.svg



Tag	Descrição
<code>&lt;img&gt;</code>	Define uma imagem;
<code>&lt;map&gt;</code>	Define um mapa de imagem;
<code>&lt;area&gt;</code>	Define uma área clicável dentro de um mapa de imagem;
<code>&lt;picture&gt;</code>	Define um contêiner para vários recursos de imagem.



Espera aí, mas o que é um mapa de imagem? Veremos isso agora!

A tag HTML `<map>` define um mapa de imagem. Um mapa de imagem é uma imagem com áreas clicáveis. As áreas são definidas com uma ou mais tags `<area>`.

A ideia por trás de um mapa de imagem é que você deve ser capaz de realizar diferentes ações, dependendo de onde você clicar na imagem.

Para criar um mapa de imagem, você precisa de uma imagem e algum código HTML que descreva as áreas clicáveis.



```
  
<map name="workmap">  
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm"  
>  
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">  
  <area shape="circle" coords="337,300,44" alt="Coffee" href="coffee.htm">  
</map>
```





## Tabelas HTML

As tabelas HTML permitem que os desenvolvedores da Web organizem os dados em linhas e colunas. Uma tabela em HTML consiste em células de tabela dentro de linhas e colunas. Cada célula da tabela é definida por uma tag `<td>` e `</td>`. Por outro lado, cada linha da tabela começa com uma tag `<tr>` e termina com uma `</tr>`.

É possível inserir quantas linhas quiser em uma tabela; apenas certifique-se de que o número de células seja o mesmo em cada linha.

Às vezes, você deseja que suas células sejam células de cabeçalho de tabela. Nesses casos, use a tag `<th>` em vez da tag `<td>`, dessa forma é possível adicionar cabeçalhos de tabela.

```
<!DOCTYPE html>
<html>
<style>
table, th, td {
border:1px solid black;
}
</style>
<body>

<h2>Os elementos TH definem os cabeçalhos da tabela</h2>

<table style="width:100%">
<tr>
<th>Pessoa 1</th>
<th>Pessoa 2</th>
<th>Pessoa 3</th>
</tr>
<tr>
<td>Emil</td>
<td>Tobias</td>
<td>Linus</td>
</tr>
<tr>
<td>16</td>
<td>14</td>
<td>10</td>
</tr>
</table>

<p>Para entender melhor o exemplo, adicionamos bordas à tabela.</p>

</body>
</html>
```

### Os elementos TH definem os cabeçalhos da tabela

Pessoa 1	Pessoa 2	Pessoa 3
Emil	Tobias	Linus
16	14	10

Para entender melhor o exemplo, adicionamos bordas à tabela.



TABELAS HTML	<TABLE> DEFINE UMA TABELA
	<TH> DEFINE UMA CÉLULA DE CABEÇALHO EM UMA TABELA
	<TR> DEFINE UMA LINHA EM UMA TABELA
	<TD> DEFINE UMA CÉLULA EM UMA TABELA
	<CAPTION> DEFINE UMA LEGENDA DE TABELA
	<COLGROUP> ESPECIFICA UM GRUPO DE UMA OU MAIS COLUNAS EM UMA TABELA PARA FORMATAÇÃO
	<COL> ESPECIFICA AS PROPRIEDADES DA COLUNA PARA CADA COLUNA DENTRO DE UM ELEMENTO <COLGROUP>
	<THEAD> AGRUPA O CONTEÚDO DO CABEÇALHO EM UMA TABELA
	<TBODY> AGRUPA O CONTEÚDO DO CORPO EM UMA TABELA
	<TFOOT> AGRUPA O CONTEÚDO DO RODAPÉ EM UMA TABELA

As tabelas HTML podem ter bordas de diferentes estilos e formas. Ao adicionar uma borda a uma tabela, você também adiciona bordas ao redor de cada célula da tabela. Para adicionar uma borda, use a propriedade `border` CSS nos elementos `table`, `th` e `td`.

Se você definir uma cor de fundo para cada célula e atribuir à borda uma cor branca (a mesma do plano de fundo do documento), terá a impressão de uma borda invisível. Com a propriedade `border-style`, você pode definir a aparência da borda.

Os seguintes valores são permitidos:

- dotted
- dashed
- solid
- double
- groove
- ridge
- inset
- outset
- none
- hidden

Com a propriedade `border-color`, você pode definir a cor da borda. As tabelas HTML podem ter tamanhos diferentes para cada coluna, linha ou tabela inteira. Use o atributo `style` com as propriedades `width` ou `height` para especificar o tamanho de uma tabela, linha ou coluna. Para definir a largura da tabela HTML, adicione o atributo `style` ao elemento `<table>`. Para definir o tamanho de uma coluna específica, adicione o atributo `style` em um elemento `<th>` ou `<td>`.



As tabelas HTML podem ter cabeçalhos para cada coluna ou linha ou para muitas colunas/linhas. Os cabeçalhos das tabelas são definidos com elementos th. Cada elemento th representa uma célula da tabela.

Para usar a primeira coluna como cabeçalhos de tabela, defina a primeira célula em cada linha como um elemento <th>. Por padrão, os cabeçalhos das tabelas estão em negrito e centralizados. Para alinhar à esquerda os cabeçalhos da tabela, use a propriedade text-align CSS. Você pode ter um cabeçalho que se estende por duas ou mais colunas. Para fazer isso, use o atributo colspan no <th>elemento. Você pode adicionar uma legenda que sirva como cabeçalho para toda a tabela, para isso, use a tag <caption>.

As tabelas HTML podem ajustar o preenchimento dentro das células e também o espaço entre as células. O preenchimento da célula é o espaço entre as bordas da célula e o conteúdo da célula. Por padrão, o preenchimento é definido como 0. Para adicionar preenchimento nas células da tabela, use a propriedade padding CSS.

Para adicionar preenchimento somente acima do conteúdo, use a propriedade padding-top. E os outros lados com as propriedades padding-bottom, padding-left e padding-right.

Espaçamento celular é o espaço entre cada célula. Por padrão, o espaço é definido como 2 pixels. Para alterar o espaço entre as células da tabela, use a propriedade border-spacing CSS no elemento table.

As tabelas HTML podem ter células que abrangem várias linhas e/ou colunas. Para fazer uma célula abranger várias colunas, use o atributo colspan. Para fazer uma célula abranger várias linhas, use o atributo rowspan.



## Estilização de Tabelas HTML

Use CSS para melhorar a aparência de suas tabelas. Para estilizar todos os outros elementos de linha da tabela, use o seletor `:nth-child(even)` como este. Para fazer listras de zebra verticais, estilize colunas alternadas, em vez de linhas alternadas.

Se você especificar bordas apenas na parte inferior de cada linha da tabela, terá uma tabela com divisores horizontais. Adicione a propriedade `border-bottom` a todos os elementos `tr` para obter divisores horizontais:

O elemento `<colgroup>` é usado para estilizar colunas específicas de uma tabela. Se você deseja estilizar as duas primeiras colunas de uma tabela, use os elementos `<colgroup>` e `<col>`. O elemento `<colgroup>` deve ser usado como um contêiner para as especificações da coluna.

Cada grupo é especificado com um elemento `<col>`. O atributo `span` especifica quantas colunas recebem o estilo. Já, o atributo `style` especifica o estilo a ser dado às colunas.



## Listas HTML

As listas HTML permitem que os desenvolvedores da Web agrupem um conjunto de itens relacionados em listas. Uma lista não ordenada começa com a tag `<ul>`. Cada item da lista começa com a tag `<li>`. Os itens da lista serão marcados com marcadores (pequenos círculos pretos) por padrão:

```
<!DOCTYPE html>
<html>
<body>

<h2>Uma lista HTML não ordenada</h2>

<ul>
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ul>

</body>
</html>
```

### Uma lista HTML não ordenada

- Café
- Chá
- Leite

Uma lista ordenada começa com a tag `<ol>`. Cada item da lista começa com a tag `<li>`. Os itens da lista serão marcados com números por padrão

```
<!DOCTYPE html>
<html>
<body>

<h2>Uma lista HTML ordenada</h2>

<ol>
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ol>

</body>
</html>
```

### Uma lista HTML ordenada

1. Café
2. Chá
3. Leite

O HTML também suporta listas de descrição. Uma lista de descrição é uma lista de termos, com uma descrição de cada termo. A tag `<dl>` define a lista de descrição, a tag `<dt>` define o termo (nome) e a tag `<dd>` descreve cada termo.

```
<!DOCTYPE html>
<html>
<body>

<h2>Uma lista de descrição</h2>

<dl>
  <dt>Café</dt>
  <dd>- bebida quente preta</dd>
  <dt>Leite</dt>
  <dd>- bebida gelada branca</dd>
</dl>

</body>
</html>
```

### Uma lista de descrição

- Café  
- bebida quente preta
- Leite  
- bebida gelada branca



Tag	Descrição
<ul>	Define uma lista não ordenada
<ol>	Define uma lista ordenada
<li>	Definir um item de lista
<dl>	Define uma lista de descrição
<dt>	Define um termo em uma lista de descrição
<dd>	Descreve o termo em uma lista de descrição

(FCC –DPE AM – 2018) Um Técnico Programador deseja fazer um glossário de termos em um site utilizando listas de definições do HTML5. Nestas listas, o termo a ser descrito e a descrição propriamente dita são criados, respectivamente, pelas tags

- a) <dt> e <dd>
- b) <dd> e <tt>
- c) <ul> e <li>
- d) <tt> e <dd>
- e) <dd> e <dt>

### Comentários

O HTML também suporta listas de descrição. Uma lista de descrição é uma lista de termos, com uma descrição de cada termo. A tag <dl> define a lista de descrição, a tag <dt> define o termo (nome) e a tag <dd> descreve cada termo. <dt> define um termo em uma lista de descrição. <dd> descreve o termo em uma lista de descrição. (Gabarito: Letra A).

## Listas Não Ordenadas

Como já vimos, uma **lista não ordenada** começa com a tag <ul> e cada item da lista começa com a tag <li>. Os itens da lista serão marcados com marcadores (pequenos círculos pretos) por padrão.

A propriedade CSS list-style-type é usada para definir o estilo do marcador de item da lista. Pode ter um dos seguintes valores:

Valor	Descrição
disc	Define o marcador de item da lista para um marcador (padrão).
circle	Define o marcador de item da lista para um círculo.
square	Define o marcador de item da lista para um quadrado.
none	Os itens da lista não serão marcados.



```
<!DOCTYPE html>
<html>
<body>

<h2>Lista não ordenada com marcadores circulares</h2>
<ul style="list-style-type:circle;">
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ul>

<h2>Lista não ordenada com marcadores de disco</h2>
<ul style="list-style-type:disc;">
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ul>

<h2>Lista não ordenada com marcadores quadrados</h2>
<ul style="list-style-type:square;">
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ul>

<h2>Lista não ordenada sem marcadores</h2>
<ul style="list-style-type:none;">
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ul>

</body>
</html>
```

### Lista não ordenada com marcadores circulares

- Café
- Chá
- Leite

### Lista não ordenada com marcadores de disco

- Café
- Chá
- Leite

### Lista não ordenada com marcadores quadrados

- Café
- Chá
- Leite

### Lista não ordenada sem marcadores

- Café
- Chá
- Leite

Além de todas as opções já citadas, ainda podemos ter listas aninhadas, ou seja, uma lista dentro da outra lista.

```
<!DOCTYPE html>
<html>
<body>

<h2>Uma lista aninhada</h2>
<p>As listas podem ser aninhadas (lista dentro da lista):</p>

<ul>
  <li>Café</li>
  <li>Chá
    <ul>
      <li>Chá preto</li>
      <li>Chá verde</li>
    </ul>
  </li>
  <li>Leite</li>
</ul>

</body>
</html>
```

### Uma lista aninhada

As listas podem ser aninhadas (lista dentro da lista):

- Café
- Chá
  - Chá preto
  - Chá verde
- Leite

Agora vejamos uma lista horizontal com CSS! As listas HTML podem ser estilizadas de várias maneiras diferentes com CSS e uma maneira popular é **estilizar uma lista horizontalmente**, para criar um **menu de navegação**. Fica bem maneiro, veja:

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333333;
}
```



```
li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 16px;
  text-decoration: none;
}


li a:hover {
  background-color: #111111;
}
</style>
</head>
<body>

<h2>Navegação no menu</h2>
<p>Neste exemplo, usamos CSS para estilizar a lista horizontalmente, para
criar um menu de navegação:</p>
<ul>
  <li><a href="#home"> Página inicial </a></li>
  <li><a href="#news"> Notícias </a></li>
  <li><a href="#contact"> Contato </a></li>
  <li><a href="#about"> Sobre </a></li>
</ul>

</body>
</html>
```

## Navegação no menu

Neste exemplo, usamos CSS para estilizar a lista horizontalmente, para criar um menu de navegação:

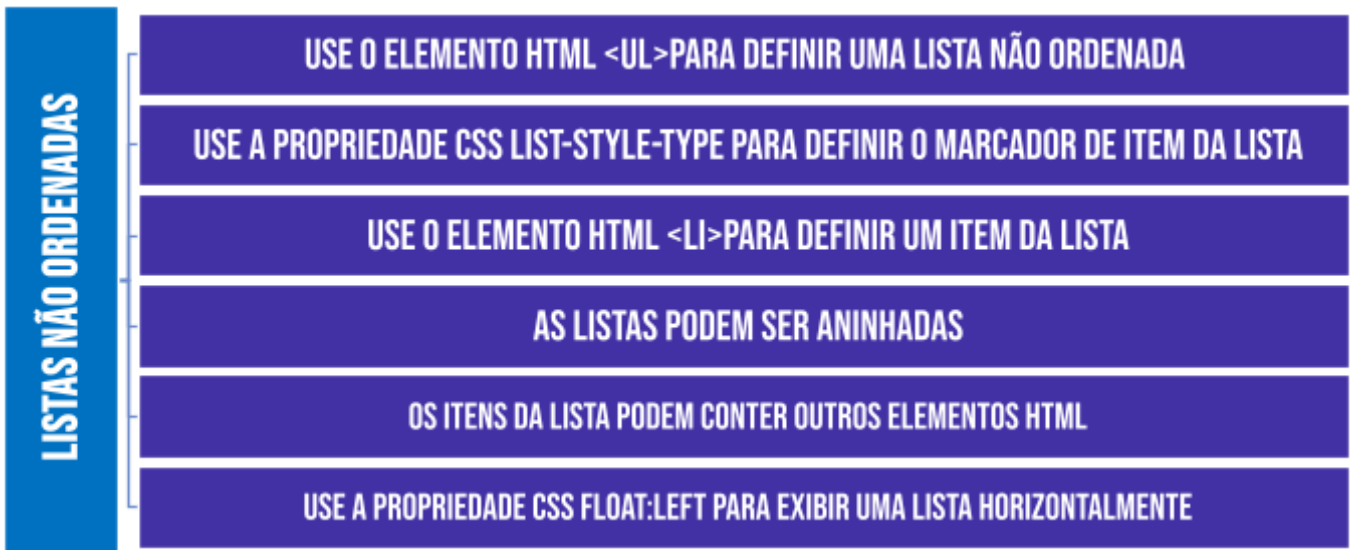


Página inicial   Notícias   Contato   Sobre

Gostaram? Bem bacana não é?







(CESPE – MPC TCE-PA – 2019) O HTML (hypertext markup language) tem amplo uso difundido nas páginas publicadas na Internet. Assinale a opção que corresponde à tag utilizada no caso em que seja necessário utilizar uma lista não ordenada.

- a) <b>
- b) <p>
- c) <tr>
- d) <ul>
- e) <th>

#### Comentários

Como já vimos, uma lista não ordenada começa com a tag <ul> e cada item da lista começa com a tag <li>. Os itens da lista serão marcados com marcadores (pequenos círculos pretos) por padrão. (Gabarito: Letra D)

## Listas Ordenadas em HTML

Uma lista ordenada começa com a tag <ol>. Cada item da lista começa com a tag <li>. Os itens da lista serão marcados com números por padrão:

Em uma Lista ordenada, o atributo Type da tag <ol>, define o tipo de marcador do item da lista.

Type	Descrição
type="1"	Os itens da lista serão numerados com números (padrão)
type="A"	Os itens da lista serão numerados com letras maiúsculas
type="a"	Os itens da lista serão numerados com letras minúsculas
type="I"	Os itens da lista serão numerados com números romanos maiúsculos



`type="i"`

Os itens da lista serão numerados com números romanos minúsculos

Pessoal, a única alteração ocorre na tag `<ol>`. Vocês devem definir o tipo de lista dentro dessa tag da seguinte forma: `<ol type="1">` ou `<ol type="A">`, ou `<ol type="i">`. Enfim, você define o tipo e o resultado será como a imagem logo após o código. Veja:

```
<ol type="1">  
  <li>Café</li>  
  <li>Chá</li>  
  <li>Leite</li>  
</ol>
```

### Lista Ordenada com Números

1. Café
2. Chá
3. Leite

### Lista Ordenada com Letras

- A. Café
- B. Chá
- C. Leite

### Lista Ordenada com Letras Minúsculas

- a. Café
- b. Chá
- c. Leite

### Lista Ordenada com Números Romanos

- I. Café
- II. Chá
- III. Leite

### Lista Ordenada com Números Romanos Minúsculos

- i. Café
- ii. Chá
- iii. Leite

Por padrão, uma lista ordenada começará a contar a partir de 1. Se você quiser começar a contar a partir de um número específico, poderá usar o atributo `start` e definir o valor que deseja iniciar.



```
<!DOCTYPE html>
<html>
<body>

<h2>O atributo inicial</h2>
<p>Por padrão, uma lista ordenada começará a contar a partir de 1. Use o atributo start para começar a contar a partir de um número especificado:</p>

<ol start="50">
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ol>

<ol type="I" start="50">
  <li>Café</li>
  <li>Chá</li>
  <li>Leite</li>
</ol>

</body>
</html>
```

### O atributo inicial

Por padrão, uma lista ordenada começará a contar a partir de 1. Use o atributo start para começar a contar a partir de um número especificado:

50. Café  
51. Chá  
52. Leite

L. Café  
LI. Chá  
LII. Leite

## LISTAS ORDENADAS

USE O ELEMENTO HTML <OL> PARA DEFINIR UMA LISTA ORDENADA

USE O ATRIBUTO HTML TYPE PARA DEFINIR O TIPO DE NUMERAÇÃO

USE O ELEMENTO HTML <LI> PARA DEFINIR UM ITEM DE LISTA

AS LISTAS PODEM SER ANINHADAS

OS ITENS DA LISTA PODEM CONTER OUTROS ELEMENTOS HTML



## Bloco HTML e elementos embutidos

Cada elemento HTML tem um valor de exibição padrão, dependendo do tipo de elemento. Existem dois valores de exibição: bloco e embutido.

### Elementos de nível de bloco

Um elemento de nível de bloco sempre começa em uma nova linha e os navegadores adicionam automaticamente algum espaço (uma margem) antes e depois do elemento. Um elemento de nível de bloco sempre ocupa toda a largura disponível (estende-se para a esquerda e para a direita o máximo possível).

Dois elementos de bloco comumente usados são: `<p>` e `<div>`. O `<p>` elemento define um parágrafo em um documento HTML. E o `<div>` elemento define uma divisão ou uma seção em um documento HTML. Vejamos todos os elementos de nível de bloco HTML:

ELEMENTOS DE NÍVEL DE BLOCO				
<code>&lt;ADDRESS&gt;</code>	<code>&lt;ARTICLE&gt;</code>	<code>&lt;ASIDE&gt;</code>	<code>&lt;BLOCKQUOTE&gt;</code>	<code>&lt;CANVAS&gt;</code>
<code>&lt;DD&gt;</code>	<code>&lt;DIV&gt;</code>	<code>&lt;DL&gt;</code>	<code>&lt;DT&gt;</code>	<code>&lt;FIELDSET&gt;</code>
<code>&lt;FIGCAPTION&gt;</code>	<code>&lt;FIGURE&gt;</code>	<code>&lt;FOOTER&gt;</code>	<code>&lt;FORM&gt;</code>	<code>&lt;H1&gt;</code>
<code>&lt;H6&gt;</code>	<code>&lt;HEADER&gt;</code>	<code>&lt;HR&gt;</code>	<code>&lt;LI&gt;</code>	<code>&lt;MAIN&gt;</code>
<code>&lt;NAV&gt;</code>	<code>&lt;NOSCRIPT&gt;</code>	<code>&lt;OL&gt;</code>	<code>&lt;P&gt;</code>	<code>&lt;PRE&gt;</code>
<code>&lt;SECTION&gt;</code>	<code>&lt;TABLE&gt;</code>	<code>&lt;TFOOT&gt;</code>	<code>&lt;UL&gt;</code>	<code>&lt;VIDEO&gt;</code>

aside

O elemento `<aside>` é um elemento HTML utilizado para representar o conteúdo relacionado ao conteúdo principal da página da Web, mas que não é essencial para o entendimento do conteúdo principal. Geralmente é usado para representar barras laterais, citações e outros conteúdos relacionados tangencialmente.



Lembre-se de que o elemento `<aside>` é um elemento semântico, o que significa que é usado para adicionar significado à página da Web, em vez de controlar o layout ou a aparência da página. Você pode usar CSS para estilizar o elemento `<aside>`, assim como qualquer outro elemento.

### `<header>`

O elemento `<header>` é um elemento HTML usado para representar o cabeçalho de uma página da web ou seção de uma página da web. Normalmente, é usado para conter o título principal, o logotipo e os links de navegação da página da Web.

O elemento `<header>` normalmente é usado no topo da página da web, mas também pode ser usado para representar o cabeçalho de uma seção de uma página da web. Neste caso, deve ser usado dentro de um elemento `<section>`.

Lembre-se de que o elemento `<header>` é diferente do elemento `<head>`, que é usado para conter metadados sobre a página da web. O elemento `<header>` faz parte do conteúdo da página da web e é visível para o usuário, enquanto o elemento `<head>` não é.

### `<main>`

O elemento `<main>` é um elemento HTML usado para representar o conteúdo principal de uma página da web. Ele deve ser usado para agrupar o conteúdo exclusivo da página da Web e que não é compartilhado com outras páginas no mesmo site.

O elemento `<main>` é um elemento semântico, o que significa que é usado para adicionar significado à página da Web, em vez de controlar o layout ou a aparência da página. Você pode usar CSS para estilizar o elemento `<main>`, assim como qualquer outro elemento.

### `<video>`

O elemento `<video>` é um elemento HTML usado para incorporar conteúdo de vídeo em uma página da web. Ele fornece uma maneira padrão de reproduzir arquivos de vídeo nativamente no navegador, sem a necessidade de plug-ins de terceiros, como o Flash.

Você pode usar os atributos de altura e largura para especificar o tamanho do elemento `<video>` na página da web. Os valores desses atributos podem ser especificados em pixels ou como uma porcentagem do espaço disponível.

Lembre-se de que os atributos de altura e largura são opcionais e, se você não os especificar, o elemento `<video>` ajustará automaticamente seu tamanho para caber no



tamanho do arquivo de vídeo. Você também pode usar CSS para estilizar o elemento <video>, assim como qualquer outro elemento.

```
<video controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.webm" type="video/webm">
  Your browser does not support the video tag.
</video>
```

No HTML5, o elemento <video> pode ser usado para incorporar conteúdo de vídeo em uma página da web. O elemento <video> suporta vários formatos de arquivo de vídeo diferentes, incluindo:

- MP4: vídeo codificado usando compressão de vídeo H.264 e áudio codificado usando Advanced Audio Coding (AAC).
- WebM: vídeo codificado usando compressão de vídeo VP8 ou VP9 e áudio codificado usando Vorbis ou Opus.
- Ogg: vídeo codificado usando compressão de vídeo Theora<sup>3</sup> e áudio codificado usando Vorbis.

Em HTML5, o elemento <track> pode ser usado para especificar uma legenda ou arquivo de legenda para um elemento <video>. O elemento <track> permite que você especifique uma trilha de texto que pode ser exibida sobre o vídeo durante a reprodução.

Para usar o elemento <track>, primeiro você precisa criar um arquivo de texto contendo as legendas ou legendas do seu vídeo. O arquivo de texto deve estar em um formato compatível com o elemento <track>, como WebVTT (Web Video Text Tracks).

```
<video controls>
  <source src="my-video.mp4" type="video/mp4">
  <track src="subtitles.vtt" kind="subtitles" srclang="en" label="English">
</video>
```

Quando o vídeo for reproduzido, as legendas serão exibidas sobre o vídeo com base nas informações de tempo no arquivo de texto. O usuário pode ativar ou desativar a exibição das legendas usando os controles fornecidos pelo navegador.

## Elementos inline

Um elemento inline não começa em uma nova linha. Pelo contrário, ocupa apenas a largura necessária.

---

<sup>3</sup> Theora é um codec de vídeo, de compressão com perda de dados



```
<!DOCTYPE html>
<html>
<body>

<p>Este é um elemento <span style="border: 1px solid black">Hello
World</span> dentro de um parágrafo.</p>

<p>O elemento SPAN é um elemento inline e não iniciará em uma nova
linha e ocupará apenas a largura necessária.</p>

</body>
</html>
```

Este é um elemento `<span style="border: 1px solid black">Hello World</span>` dentro de um parágrafo.

O elemento SPAN é um elemento inline e não iniciará em uma nova linha e ocupará apenas a largura necessária.

Vejamos os elementos inline em HTML:

## ELEMENTOS INLINE HTML

&lt;A&gt;

&lt;ABBR&gt;

&lt;ACRONYM&gt;

&lt;B&gt;

&lt;BDO&gt;

&lt;BIG&gt;

&lt;BR&gt;

&lt;BUTTON&gt;

&lt;CITE&gt;

&lt;CODE&gt;

&lt;DFN&gt;

&lt;EM&gt;

&lt;I&gt;

&lt;IMG&gt;

&lt;INPUT&gt;

&lt;KBD&gt;

&lt;LABEL&gt;

&lt;MAP&gt;

&lt;OBJECT&gt;

&lt;OUTPUT&gt;

&lt;Q&gt;

&lt;SAMP&gt;

&lt;SCRIPT&gt;

&lt;SELECT&gt;

&lt;SMALL&gt;

&lt;SPAN&gt;

&lt;STRONG&gt;

&lt;SUB&gt;

&lt;SUP&gt;

&lt;TEXTAREA&gt;

&lt;TIME&gt;

&lt;TT&gt;

&lt;VAR&gt;

O elemento `<div>` geralmente é usado como um contêiner para outros elementos HTML. Ele não tem atributos obrigatórios, mas `style`, `class` e `id` são comuns. Quando usado junto com CSS, o elemento `<div>` pode ser usado para **estilizar blocos de conteúdo**:



```
<!DOCTYPE html>
<html>
<body>

<div style="background-color:black;color:white;padding:20px;">
  <h2>Londres</h2>
  <p>Londres é a capital da Inglaterra. É a cidade mais populosa do Reino Unido, com uma área metropolitana de mais de 13 milhões de habitantes.</p>
  <p>À beira do rio Tâmisa, Londres tem sido um importante assentamento por dois milênios, sua história remonta à sua fundação pelos romanos, que a batizaram de Londinium.</p>
</div>

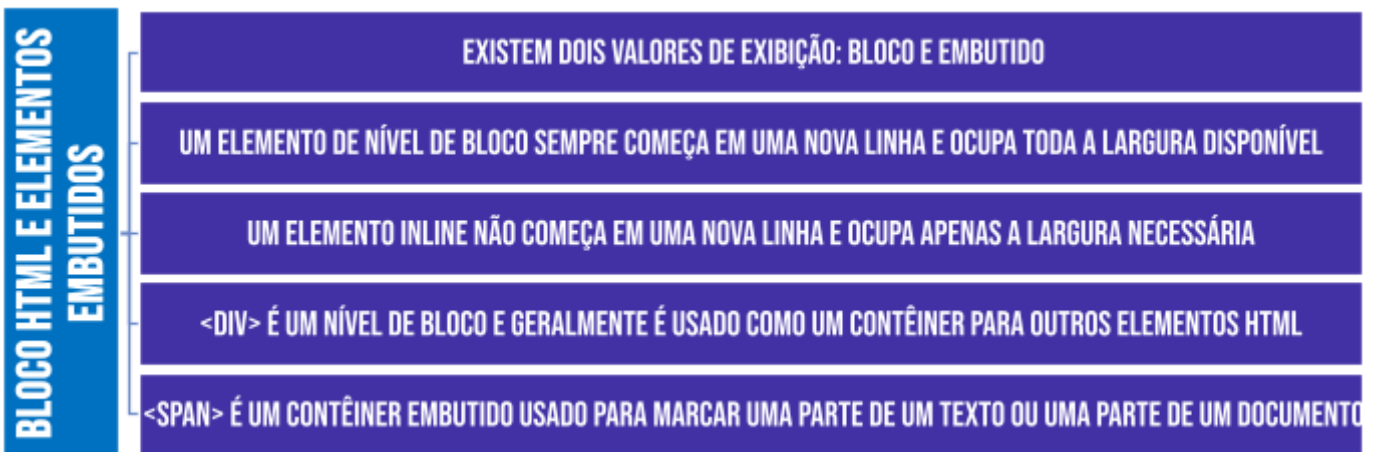
</body>
</html>
```

## Londres

Londres é a capital da Inglaterra. É a cidade mais populosa do Reino Unido, com uma área metropolitana de mais de 13 milhões de habitantes.

À beira do rio Tâmisa, Londres tem sido um importante assentamento por dois milênios, sua história remonta à sua fundação pelos romanos, que a batizaram de Londinium.

O elemento `<span>` é um contêiner embutido usado para marcar uma parte de um texto ou uma parte de um documento. Ele também não tem atributos obrigatórios, mas `style`, `class` e `id` são comuns, assim como no elemento `<div>`. Quando usado junto com CSS, o `<span>` elemento pode ser usado para estilizar partes do texto



Para lembrar, fica esse grande lembrete:

**`<DIV>` DEFINE UMA SEÇÃO EM UM DOCUMENTO (NÍVEL DE BLOCO)**

**`<SPAN>` DEFINE UMA SEÇÃO EM UM DOCUMENTO (EM LINHA)**





## Atributo de classe HTML

O atributo HTML class é usado para especificar uma classe para um elemento HTML.

Vários elementos HTML podem compartilhar a mesma classe.

Usando o atributo de classe

O class atributo geralmente é usado para apontar para um nome de classe em uma folha de estilo. Também pode ser usado por um JavaScript para acessar e manipular elementos com o nome de classe específico.

No exemplo a seguir temos três elementos <div> com um atributo class com o valor "cidade". Todos os três elementos serão estilizados igualmente de acordo com a definição de estilo.

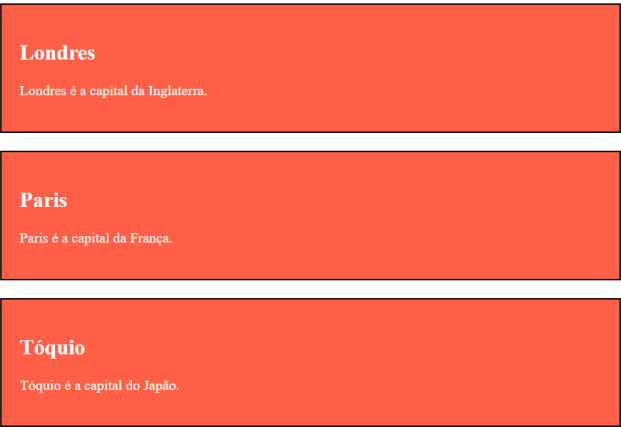
```
<!DOCTYPE html>
<html>
<head>
<style>
.cidade {
background-color: tomato;
color: white;
border: 2px solid black;
margin: 20px;
padding: 20px;
}
</style>
</head>
<body>

<div class="cidade">
<h2>Londres</h2>
<p>Londres é a capital da Inglaterra.</p>
</div>

<div class="cidade">
<h2>Paris</h2>
<p>Paris é a capital da França.</p>
</div>

<div class="cidade">
<h2>Tóquio</h2>
<p>Tóquio é a capital do Japão.</p>
</div>

</body>
</html>
```



Como vimos no exemplo, para criar uma classe; devemos escrever um caractere de ponto (.), seguido de um nome de classe (no exemplo, cidade). Em seguida, defina as propriedades CSS entre chaves {}. Os elementos HTML podem pertencer a mais de uma classe. Para definir várias classes, separe os nomes das classes com um espaço, por exemplo, <div class="city main">. O elemento será estilizado de acordo com todas as classes especificadas.

No exemplo a seguir, o primeiro elemento <h2> pertence tanto à classe city quanto à classe main e obterá os estilos CSS de ambas as classes:



<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt; &lt;style&gt; .city {   background-color: tomato;   color: white;   padding: 10px; }  .main {   text-align: center; } &lt;/style&gt; &lt;/head&gt; &lt;body&gt;  &lt;h2&gt;Múltiplas classes&lt;/h2&gt; &lt;p&gt;Aqui, todos os três elementos h2 pertencem à classe "city". Além disso, Londres também pertence à classe "main", que centraliza o texto.&lt;/p&gt;  &lt;h2 class="city main"&gt;Londres&lt;/h2&gt; &lt;h2 class="city"&gt;Paris&lt;/h2&gt; &lt;h2 class="city"&gt;Tóquio&lt;/h2&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<h3>Múltiplas classes</h3> <p>Aqui, todos os três elementos h2 pertencem à classe "city". Além disso, Londres também pertence à classe "main", que centraliza o texto.</p> <div style="background-color: tomato; color: white; text-align: center; padding: 5px; margin-bottom: 5px;">Londres</div> <div style="background-color: tomato; color: white; text-align: center; padding: 5px; margin-bottom: 5px;">Paris</div> <div style="background-color: tomato; color: white; text-align: center; padding: 5px;">Tóquio</div>
--	--

Para criar uma classe; escreva um caractere de ponto (.), seguido de um nome de classe. Em seguida, defina as propriedades CSS entre chaves {}

Os elementos HTML podem pertencer a mais de uma classe. Para definir várias classes, separe os nomes das classes com um espaço, por exemplo, <div class="city main">. O elemento será estilizado de acordo com todas as classes especificadas. No exemplo a seguir, o primeiro elemento <h2> pertence tanto à classe city quanto à classe main e obterá os estilos CSS de ambas as classes

Diferentes elementos HTML podem apontar para o mesmo nome de classe. <h2> no exemplo a seguir, ambos <p> apontam para a classe "city" e compartilharão o mesmo estilo

<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt; &lt;style&gt; .city {   background-color: tomato;   color: white;   padding: 10px; } &lt;/style&gt; &lt;/head&gt; &lt;body&gt;  &lt;h2&gt;Diferentes elementos podem compartilhar a mesma classe&lt;/h2&gt;  &lt;p&gt;Mesmo que os dois elementos não tenham o mesmo nome de tag, eles podem apontar para a mesma classe e obter o mesmo estilo CSS: &lt;/p&gt;  &lt;h2 class="city"&gt;Paris&lt;/h2&gt; &lt;p class="city"&gt;Paris é a capital da França.&lt;/p&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<h3>Diferentes elementos podem compartilhar a mesma classe</h3> <p>Mesmo que os dois elementos não tenham o mesmo nome de tag, eles podem apontar para a mesma classe e obter o mesmo estilo CSS:</p> <div style="background-color: tomato; color: white; text-align: center; padding: 5px; margin-bottom: 5px;">Paris</div> <div style="background-color: tomato; color: white; text-align: center; padding: 5px;">Paris é a capital da França.</div>
---	--



O nome da classe também pode ser usado pelo JavaScript para executar determinadas tarefas para elementos específicos. O JavaScript pode acessar elementos com um nome de classe específico com o método `getElementsByClassName()`.

```
<!DOCTYPE html>
<html>
<body>

<h2>Uso do atributo class em JavaScript</h2>
<p>Clique no botão para ocultar todos os elementos com o nome de classe "city":</p>

<button onclick="myFunction()">Ocultar elementos</button>

<h2 class="city">Londres</h2>
<p>Londres é a capital da Inglaterra.</p>

<h2 class="city">Paris</h2>
<p>Paris é a capital da França.</p>

<h2 class="city">Tóquio</h2>
<p>Tóquio é a capital do Japão.</p>

<script>
function myFunction() {
  var x = document.getElementsByClassName("city");
  for (var i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
}
</script>
</body>
</html>
```

### Uso do atributo class em JavaScript

Clique no botão para ocultar todos os elementos com o nome de classe "city":

#### Londres

Londres é a capital da Inglaterra.

#### Paris

Paris é a capital da França.

#### Tóquio

Tóquio é a capital do Japão.

## ATRIBUTO DE CLASSE HTML

- O ATRIBUTO HTML CLASS ESPECIFICA UM OU MAIS NOMES DE CLASSE PARA UM ELEMENTO
- CLASSES SÃO USADAS POR CSS E JAVASCRIPT PARA SELECIONAR E ACESSAR ELEMENTOS ESPECÍFICOS
- O ATRIBUTO CLASS PODE SER USADO EM QUALQUER ELEMENTO HTML
- O NOME DA CLASSE DIFERENCIA MAIÚSCULAS DE MINÚSCULAS
- DIFERENTES ELEMENTOS HTML PODEM APONTAR PARA O MESMO NOME DE CLASSE
- JAVASCRIPT PODE ACESSAR ELEMENTOS COM UM NOME DE CLASSE ESPECÍFICO COM O MÉTODO `GETELEMENTSBYCLASSNAME()`



## Formulários HTML

Um formulário HTML é usado para coletar a entrada do usuário. A entrada do usuário geralmente é enviada a um servidor para processamento. O elemento HTML `<form>` é usado para criar um formulário HTML para entrada do usuário. Ele é um contêiner para diferentes tipos de elementos de entrada, como: campos de texto, caixas de seleção, botões de opção, botões de envio, etc.

O elemento HTML `<input>` é o elemento de formulário mais usado. Ele pode ser exibido de várias maneiras, dependendo do atributo `type`.

Type	Descrição
<code>&lt;input type="text"&gt;</code>	Exibe um campo de entrada de texto de linha única
<code>&lt;input type="radio"&gt;</code>	Exibe um botão de opção (para selecionar uma das muitas opções)
<code>&lt;input type="checkbox"&gt;</code>	Exibe uma caixa de seleção (para selecionar zero ou mais opções)
<code>&lt;input type="submit"&gt;</code>	Exibe um botão de envio (para enviar o formulário)
<code>&lt;input type="button"&gt;</code>	Exibe um botão clicável

```
<!DOCTYPE html>
<html>
<body>

<h2>Campos de entrada de texto</h2>

<form>
  <label for="fname">Nome:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Sobrenome:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
</form>

<p>Observe que o formulário em si não está visível.</p>

<p>Observe também que a largura padrão dos campos de entrada de texto é de 20 caracteres.</p>

</body>
</html>
```

### Campos de entrada de texto

Nome:

Sobrenome:

Observe que o formulário em si não está visível.

Observe também que a largura padrão dos campos de entrada de texto é de 20 caracteres.

A tag `<label>` define um rótulo para muitos elementos de formulário. Ela é útil para usuários de leitores de tela, porque o leitor de tela lerá o rótulo em voz alta quando o usuário focar no elemento de entrada.

O elemento `<label>` também ajuda os usuários que têm dificuldade em clicar em regiões muito pequenas (como botões de opção ou caixas de seleção) - porque quando o usuário clica no texto dentro do `<label>` elemento, ele alterna o botão de opção/caixa



de seleção. O atributo `for` da tag `<label>` deve ser igual ao atributo `id` do `<input>` para uni-los.

O `<input type="radio">` define um botão de opção. Os botões de opção permitem que um usuário selecione **UMA** dentre um número limitado de opções.

<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;h2&gt;Botões de opção&lt;/h2&gt;  &lt;p&gt;Escolha seu idioma da Web favorito:&lt;/p&gt;  &lt;form&gt; &lt;input type="radio" id="html" name="fav_language" value="HTML"&gt; &lt;label for="html"&gt;HTML&lt;/label&gt;&lt;br&gt; &lt;input type="radio" id="css" name="fav_language" value="CSS"&gt; &lt;label for="css"&gt;CSS&lt;/label&gt;&lt;br&gt; &lt;input type="radio" id="javascript" name="fav_language" value="JavaScript"&gt; &lt;label for="javascript"&gt;JavaScript&lt;/label&gt; &lt;/form&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<h3>Botões de opção</h3> <p>Escolha seu idioma da Web favorito:</p> <p> <input type="radio"/> HTML  <input type="radio"/> CSS  <input type="radio"/> JavaScript </p>
--	--

O `<input type="checkbox">` define uma caixa de seleção. As caixas de seleção permitem que um usuário selecione **ZERO** ou **MAIS** opções de um número limitado de opções.

<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;h2&gt;Caixas de seleção&lt;/h2&gt; &lt;p&gt;0 &lt;strong&gt;input type="checkbox"&lt;/strong&gt; define uma caixa de seleção:&lt;/p&gt;  &lt;form action="/action_page.php"&gt;   &lt;input type="checkbox" id="vehicle1" name="vehicle1" value="Bike"&gt;   &lt;label for="vehicle1"&gt; Eu tenho uma bicicleta&lt;/label&gt;&lt;br&gt;   &lt;input type="checkbox" id="vehicle2" name="vehicle2" value="Car"&gt;   &lt;label for="vehicle2"&gt; Eu tenho um carro&lt;/label&gt;&lt;br&gt;   &lt;input type="checkbox" id="vehicle3" name="vehicle3" value="Boat"&gt;   &lt;label for="vehicle3"&gt; Eu tenho um barco&lt;/label&gt;&lt;br&gt;&lt;br&gt;   &lt;input type="submit" value="Enviar"&gt; &lt;/form&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<h3>Caixas de seleção</h3> <p>O <code>input type="checkbox"</code> define uma caixa de seleção:</p> <p> <input type="checkbox"/> Eu tenho uma bicicleta  <input type="checkbox"/> Eu tenho um carro  <input type="checkbox"/> Eu tenho um barco </p> <p>Enviar</p>
---	--

O `<input type="submit">` define um botão para enviar os dados do formulário para um manipulador de formulário. O manipulador de formulário geralmente é um arquivo no servidor com um script para processar dados de entrada. O manipulador de formulário é especificado no atributo `action` do formulário.



```
<!DOCTYPE html>
<html>
<body>

<h2>Formulários HTML</h2>

<form action="/action_page.php">
  <label for="fname">Nome:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Sobrenome:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="enviar" value="Enviar">
</form>

<p>Se você clicar no botão "Enviar", os dados do formulário serão enviados para uma página chamada "/action_page.php".</p>

</body>
</html>
```

### Formulários HTML

Nome:

Sobrenome:

Se você clicar no botão "Enviar", os dados do formulário serão enviados para uma página chamada "/action\_page.php".

Observe que cada campo de entrada deve ter um atributo name a ser submetido. Se o atributo name for omitido, o valor do campo de entrada não será enviado.

Já o atributo action define a ação a ser executada quando o formulário é enviado. Normalmente, os dados do formulário são enviados para um arquivo no servidor quando o usuário clica no botão enviar. No exemplo abaixo, os dados do formulário são enviados para um arquivo chamado "action\_page.php". Este arquivo contém um script do lado do servidor que manipula os dados do formulário:

O atributo target especifica onde exibir a resposta recebida após o envio do formulário. Ele pode ter um dos seguintes valores, sendo que o valor padrão é `_self` o que significa que a resposta será aberta na janela atual.

Valor	Descrição
<code>_blank</code>	A resposta é exibida em uma nova janela ou guia;
<code>_self</code>	A resposta é exibida na janela atual;
<code>_parent</code>	A resposta é exibida no quadro pai;
<code>_top</code>	A resposta é exibida em todo o corpo da janela;
<code>framename</code>	A resposta é exibida em um iframe nomeado.

O atributo do **método** especifica o método HTTP a ser usado ao enviar os dados do formulário. Os dados do formulário podem ser enviados como variáveis de URL (com `method="get"`) ou como pós-transação HTTP (com `method="post"`). O método HTTP padrão ao enviar dados de formulário é **GET**.

Método	Descrição
Get	Anexa os dados do formulário ao URL, em pares de nome/valor; NUNCA use GET para enviar dados confidenciais! (os dados do formulário enviado são visíveis na URL!);



	<p>O comprimento de um URL é limitado (2048 caracteres); Útil para envios de formulários em que um usuário deseja marcar o resultado como favorito;</p> <p>GET é bom para dados não seguros, como strings de consulta no Google.</p>
<b>Post</b>	<ul style="list-style-type: none"> <li>- Anexa os dados do formulário dentro do corpo da solicitação HTTP (os dados do formulário enviado não são mostrados na URL);</li> <li>- O POST não tem limitações de tamanho e pode ser usado para enviar grandes quantidades de dados;</li> <li>- Os envios de formulário com POST não podem ser marcados.</li> </ul>

Vejam os atributos dos formulários HTML.

Atributo de Formulários	Descrição
accept-charset	Especifica as codificações de caracteres usadas para envio de formulário
action	Especifica para onde enviar os dados do formulário quando um formulário é enviado
autocomplete	Especifica se um formulário deve ter o preenchimento automático ativado ou desativado
enctype	Especifica como os dados do formulário devem ser codificados ao enviá-los ao servidor (somente para method="post")
method	Especifica o método HTTP a ser usado ao enviar dados de formulário
name	Especifica o nome do formulário
novalidate	Especifica que o formulário não deve ser validado quando enviado
rel	Especifica a relação entre um recurso vinculado e o documento atual
target	Especifica onde exibir a resposta recebida após o envio do formulário

## Os elementos HTML <form>

Vimos os atributos agora vamos ver os elementos dos formulários HTML! O elemento <form> pode conter um ou mais dos seguintes elementos de formulário:

Elementos de Formulários	Descrição
<form>	Define um formulário HTML para entrada do usuário
<input>	Define um controle de entrada
<textarea>	Define um controle de entrada multilinha (área de texto)
<label>	Define um rótulo para um elemento <input>





<code>&lt;fieldset&gt;</code>	Agrupa elementos relacionados em um formulário
<code>&lt;legend&gt;</code>	Define uma legenda para um elemento <code>&lt;fieldset&gt;</code>
<code>&lt;selecionar&gt;</code>	Define uma lista suspensa
<code>&lt;optgroup&gt;</code>	Define um grupo de opções relacionadas em uma lista suspensa
<code>&lt;option&gt;</code>	Define uma opção em uma lista suspensa
<code>&lt;button&gt;</code>	Define um botão clicável
<code>&lt;datalist&gt;</code>	Especifica uma lista de opções predefinidas para controles de entrada
<code>&lt;output&gt;</code>	Define o resultado de um cálculo

Vejamos agora os diferentes tipos de entrada que você pode usar em HTML

Tipos de entrada de Formulários	Descrição
<code>&lt;input type="button"&gt;</code>	Define um botão
<code>&lt;input type="checkbox"&gt;</code>	Define uma caixa de seleção. As caixas de seleção permitem que um usuário selecione ZERO ou MAIS opções de um número limitado de opções.
<code>&lt;input type="color"&gt;</code>	É usado para campos de entrada que devem conter uma cor. Dependendo do suporte do navegador, um seletor de cores pode aparecer no campo de entrada.
<code>&lt;input type="date"&gt;</code>	É usado para campos de entrada que devem conter uma data. Dependendo do suporte do navegador, um seletor de data pode aparecer no campo de entrada.
<code>&lt;input type="datetime-local"&gt;</code>	Especifica um campo de entrada de data e hora, sem fuso horário. Dependendo do suporte do navegador, um seletor de data pode aparecer no campo de entrada.
<code>&lt;input type="email"&gt;</code>	É usado para campos de entrada que devem conter um endereço de e-mail. Dependendo do suporte do navegador, o endereço de e-mail pode ser validado automaticamente quando enviado.
<code>&lt;input type="file"&gt;</code>	Define um campo de seleção de arquivo e um botão "Procurar" para uploads de arquivos.
<code>&lt;input type="hidden"&gt;</code>	Define um campo de entrada oculto (não visível para um usuário). Um campo oculto permite que os desenvolvedores da web incluam dados que não podem ser vistos ou modificados pelos usuários quando um formulário é enviado. Um campo oculto geralmente armazena o registro do banco de dados que precisa ser atualizado quando o formulário é enviado.
<code>&lt;input type="image"&gt;</code>	Define uma imagem como um botão de envio. O caminho para a imagem é especificado no <code>src</code> atributo.
<code>&lt;input type="month"&gt;</code>	Permite ao usuário selecionar um mês e um ano. Dependendo do suporte do navegador, um seletor de data pode aparecer no campo de entrada.





<code>&lt;input type="number"&gt;</code>	Define um campo de entrada numérica. Você também pode definir restrições sobre quais números são aceitos.
<code>&lt;input type="password"&gt;</code>	Define um campo de senha
<code>&lt;input type="radio"&gt;</code>	Define um botão de opção. Os botões de opção permitem que um usuário selecione apenas uma dentre um número limitado de opções.
<code>&lt;input type="range"&gt;</code>	Define um controle para inserir um número cujo valor exato não é importante (como um controle deslizante). O intervalo padrão é de 0 a 100. No entanto, você pode definir restrições sobre quais números são aceitos com os atributos <code>min</code> , <code>max</code> e <code>step</code>
<code>&lt;input type="reset"&gt;</code>	Define um botão de redefinição que redefinirá todos os valores do formulário para seus valores padrão
<code>&lt;input type="search"&gt;</code>	É usado para campos de pesquisa (um campo de pesquisa se comporta como um campo de texto normal).
<code>&lt;input type="submit"&gt;</code>	Define um botão para enviar dados de formulário para um manipulador de formulário
<code>&lt;input type="tel"&gt;</code>	É usado para campos de entrada que devem conter um número de telefone.
<code>&lt;input type="text"&gt;</code>	Define um campo de entrada de texto de linha única
<code>&lt;input type="time"&gt;</code>	Permite ao usuário selecionar um horário (sem fuso horário). Dependendo do suporte do navegador, um seletor de tempo pode aparecer no campo de entrada.
<code>&lt;input type="url"&gt;</code>	É usado para campos de entrada que devem conter um endereço URL. Dependendo do suporte do navegador, o campo url pode ser validado automaticamente quando enviado.
<code>&lt;input type="week"&gt;</code>	Permite ao usuário selecionar uma semana e um ano. Dependendo do suporte do navegador, um seletor de data pode aparecer no campo de entrada.





## APIs HTML

O HTML5 apresenta várias novas APIs (Application Programming Interfaces) que permitem aos desenvolvedores da Web criar aplicativos da Web mais poderosos e interativos. Aqui estão alguns exemplos:

- A API de geolocalização permite que aplicativos da web acessem as informações de localização do usuário, com a permissão do usuário.
- A API Web Storage fornece uma maneira de armazenar dados localmente em um navegador da web, permitindo que os aplicativos da web funcionem offline ou armazenem dados para uso futuro.
- A API Web Workers permite que aplicativos da Web executem tarefas em segundo plano, como processamento de dados, sem interromper a experiência do usuário.
- A API WebSockets permite a comunicação full-duplex entre o navegador da web e um servidor, permitindo a troca de dados em tempo real.
- A API Canvas permite que os desenvolvedores da Web desenhem gráficos e animações em uma página da Web usando JavaScript.
- A API de áudio da Web permite que os desenvolvedores da Web criem e manipulem áudio em uma página da Web, como sintetizar som ou processar entradas de áudio.
- API Server-Sent Events (SSE) é uma tecnologia que permite que um servidor da Web envie dados para um navegador da Web em tempo real, sem a necessidade de o navegador fazer uma solicitação. Isso facilita a criação de aplicativos em tempo real, como salas de bate-papo, feeds de notícias e outros tipos de aplicativos que exigem atualizações frequentes.

Estes são apenas alguns exemplos das APIs disponíveis no HTML5. Existem muitos mais, cada um com seu próprio conjunto específico de recursos e capacidades.

### API de geolocalização HTML

A API de geolocalização HTML é usada para localizar a posição de um usuário. Como isso pode comprometer a privacidade, a posição não está disponível a menos que o usuário aprove. a API de geolocalização permite que aplicativos da web acessem as informações de localização do usuário, com a permissão do usuário. Isso pode ser útil para uma ampla variedade de aplicativos, como pesquisa baseada em localização, publicidade com reconhecimento de localização e jogos baseados em localização.



É importante observar que as informações de localização do usuário são confidenciais e os aplicativos da Web devem obter a permissão do usuário antes de acessá-las. O usuário normalmente será solicitado a permitir ou negar o acesso às suas informações de localização.

O método `getCurrentPosition()` é usado para retornar a posição do usuário. O exemplo abaixo retorna a latitude e longitude da posição do usuário:

```
<script>
var x = document.getElementById("demo");
function getLocation() {
  if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(showPosition);
  } else {
    x.innerHTML = "Geolocation is not supported by this browser.";
  }
}

function showPosition(position) {
  x.innerHTML = "Latitude: " + position.coords.latitude +
  "<br>Longitude: " + position.coords.longitude;
}
</script>
```

Não entendeu nada? Vamos lá, o exemplo acima executa as seguintes ações:

- Verifica se a geolocalização é suportada
- Se suportado, executa o método `getCurrentPosition()`.
- Se não, exibir uma mensagem para o usuário
- Se o método `getCurrentPosition()` for bem-sucedido, ele retornará um objeto de coordenadas para a função especificada no parâmetro (`showPosition`)
- A função `showPosition()` gera a Latitude e Longitude

A geolocalização também é muito útil para informações específicas do local, como:

- Informações locais atualizadas;
- Mostrar pontos de interesse perto do usuário;
- Navegação curva a curva (GPS).

## API de armazenamento da Web em HTML

Para falar sobre API de armazenamento da Web em HTML primeiro devemos saber o que é armazenamento na Web em HTML? Com o armazenamento na web, os aplicativos da web podem **armazenar dados localmente no navegador do usuário**.

Antes do HTML5, os dados do aplicativo precisavam ser armazenados em **cookies**, incluídos em **todas as solicitações do servidor**. O armazenamento na Web é mais seguro



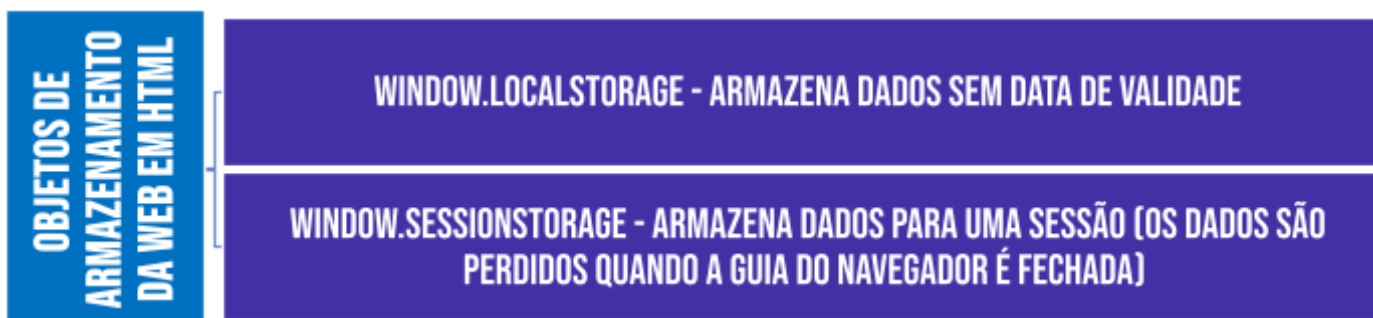
e grandes quantidades de dados podem ser armazenadas localmente, sem afetar o desempenho do site.

Ao contrário dos cookies, o **limite de armazenamento é muito maior** (pelo menos 5 MB) e as informações nunca são transferidas para o servidor. O armazenamento na Web é por origem (por domínio e protocolo). Todas as páginas, de uma origem, podem armazenar e acessar os mesmos dados.

O armazenamento da web HTML fornece dois objetos para armazenar dados no cliente:

- **window.localStorage** - armazena dados sem data de validade; os dados não serão excluídos quando o navegador for fechado e estarão disponíveis no próximo dia, semana ou ano.
- **window.sessionStorage** - armazena dados para uma sessão. Os dados são excluídos quando o usuário fecha a guia específica do navegador.

Antes de usar o armazenamento na web, é necessário verificar o suporte do navegador para localStorage e sessionStorage.



(CESPE – PGE RJ – 2022) Julgue o item que se segue, relacionado a desenvolvimento de sistemas.

No HTML 5, sessionStorage pode ser utilizado para armazenamento local de dados, tendo como característica o armazenamento de dados restritos à aba em funcionamento.

#### Comentários

É possível usar dois objetos para armazenar dados no cliente: window.localStorage e sessionStorage. O localStorage - armazena dados sem data de validade; os dados não serão excluídos quando o navegador for fechado e estarão disponíveis no próximo dia, semana ou ano. Por outro lado, o window.sessionStorage - armazena dados para uma sessão. Os dados são excluídos quando o usuário fecha a guia específica do navegador. A questão erra ao afirmar que sessionStorage pode ser utilizado para armazenamento local de dados quando na verdade é o window.localStorage que o faz. (Gabarito: Errado)



## XHTML

XHTML (Extensible HyperText Markup Language) é uma linguagem de marcação que é uma variante do HTML (HyperText Markup Language). É uma versão mais rígida e limpa do HTML, com uma sintaxe mais rígida e com foco na conformidade com os padrões.

Assim como o HTML, o XHTML é usado para estruturar e formatar o conteúdo de páginas da Web, incluindo texto, imagens e outros elementos multimídia. No entanto, o XHTML segue as regras do XML (eXtensible Markup Language), o que significa que deve ser bem formado e obedecer a certas regras de sintaxe. Isso facilita a análise e o processamento e pode ser manipulado por uma ampla gama de ferramentas e aplicativos.

XHTML foi projetado para ser uma ponte entre HTML e XML, e é frequentemente usado em combinação com outras tecnologias, como CSS (Cascading Style Sheets) e JavaScript, para criar páginas da Web dinâmicas e interativas. É suportado por todos os navegadores da web modernos e é comumente usado no desenvolvimento web.

Existem várias diferenças entre HTML e XHTML:

- **Sintaxe:** XHTML tem uma sintaxe mais estrita e rígida do que HTML. Ele segue as regras do XML, o que significa que deve ser bem formado e obedecer a certas regras de sintaxe, como fechar todas as tags e usar letras minúsculas. O HTML, por outro lado, é mais flexível e tolerante em termos de sintaxe, e não é necessário seguir as regras do XML.
- **Extensibilidade:** XHTML é projetado para ser extensível, o que significa que pode ser estendido com novos elementos e atributos. Isso é feito usando os mecanismos do XML, como namespaces. O HTML, por outro lado, não é tão extensível e é limitado aos elementos e atributos definidos na especificação HTML.
- **Compatibilidade:** XHTML é compatível com uma ampla gama de ferramentas e aplicativos, pois segue as regras do XML. O HTML é menos compatível, pois não segue as regras do XML.
- **Tipo de documento:** XHTML usa uma declaração de tipo de documento (DTD) diferente do HTML. A DTD especifica as regras para o documento e determina como o documento deve ser interpretado por navegadores da Web e outros aplicativos.
- **Tipo MIME:** XHTML usa um tipo MIME (Multipurpose Internet Mail Extensions) diferente do HTML. O tipo MIME informa aos navegadores da Web e outros aplicativos como lidar com o documento.



Em geral, o XHTML é uma versão mais moderna e compatível com os padrões do HTML, e é frequentemente usado em combinação com outras tecnologias, como CSS e JavaScript, para criar páginas da Web dinâmicas e interativas. No entanto, o HTML ainda é amplamente usado e suportado por todos os navegadores modernos.



## Acessibilidade HTML

O HTML5 inclui vários recursos que podem **melhorar a acessibilidade do conteúdo da Web** para usuários com deficiências. Alguns desses recursos incluem:

- **Elementos semânticos:** o HTML5 apresenta novos elementos semânticos, como `<header>`, `<footer>` e `<nav>`, que podem ajudar a melhorar a estrutura e o significado do conteúdo da web. Isso pode tornar mais fácil para as tecnologias assistivas, como leitores de tela, interpretar e navegar pelo conteúdo.
- **Atributos ARIA:** A especificação Accessible Rich Internet Applications (ARIA) define um conjunto de atributos que podem ser usados para fornecer informações adicionais sobre a finalidade e o comportamento dos elementos em uma página da web. Esses atributos podem ser usados para melhorar a acessibilidade de elementos interativos, como botões e controles de formulário.
- **Legendas e legendas de vídeo:** O elemento `<track>` permite que você especifique uma trilha de texto que pode ser exibida como legendas ou legendas para um elemento `<video>`. Isso pode ajudar a tornar o conteúdo de vídeo mais acessível para usuários surdos ou com deficiência auditiva.
- **Controles de formulário:** o HTML5 apresenta novos controles de formulário, como os tipos de entrada de data e intervalo, que podem melhorar a usabilidade e a acessibilidade dos formulários da web.

Ao usar esses e outros recursos do HTML5, você pode ajudar a tornar seu conteúdo da Web mais acessível e fácil de usar para uma ampla gama de usuários.

O atributo `lang` é um atributo HTML que **pode ser usado para especificar o idioma de um elemento e seu conteúdo**. É um importante **recurso de acessibilidade**, pois permite que tecnologias assistivas, como leitores de tela, interpretem e pronunciem corretamente o conteúdo de uma página da web.

É uma boa ideia usar o atributo `lang` sempre que estiver usando um idioma diferente do idioma padrão da página da web. Isso ajuda a **garantir que o conteúdo seja interpretado e pronunciado corretamente por tecnologias assistivas**.

Vale a pena notar que o atributo `lang` não está limitado ao HTML5 – ele faz parte do HTML há muitos anos. No entanto, é um importante recurso de acessibilidade que vale a pena considerar em qualquer documento HTML.

```
<html lang="en">  
  <!-- resto do documento vai aqui -->  
</html>
```





## Erros HTTP

Vamos falar sobre os erros HTTP é um assunto que costuma aparecer em provas então é bom estar bem preparado! Erros HTTP podem parecer complicados à primeira vista, mas são realmente importantes para entender como os servidores e navegadores se comunicam. Em termos simples, esses códigos ajudam a identificar o que deu errado quando uma página da web não carrega como esperado.

Vamos começar com os códigos mais comuns. Por exemplo, o **404 Não encontrado** é um dos mais conhecidos. Isso acontece quando o servidor não consegue localizar o recurso solicitado. E o **500 Erro interno do servidor** indica que algo deu errado no servidor, mas não especifica exatamente o que.

Outros códigos notáveis incluem o **200 OK**, que significa que a solicitação foi bem-sucedida, e o **403 Proibido**, que indica que o servidor entendeu a solicitação, mas se recusa a atendê-la. Cada um desses códigos oferece uma visão diferente do que está acontecendo com a comunicação entre cliente e servidor.

ERROS HTTP	<b>1XX: INFORMATIVO - O SERVIDOR ESTÁ PROCESSANDO A SOLICITAÇÃO.</b>
	<b>2XX: SUCESSO - A SOLICITAÇÃO FOI BEM-SUCEDIDA.</b>
	<b>3XX: REDIRECIONAMENTO - O SERVIDOR SOLICITA UMA NOVA AÇÃO OU REDIRECIONA.</b>
	<b>4XX: ERRO DO CLIENTE - PROBLEMA COM A SOLICITAÇÃO FEITA PELO CLIENTE.</b>
	<b>5XX: ERRO DO SERVIDOR - PROBLEMA NO SERVIDOR AO PROCESSAR A SOLICITAÇÃO.</b>

Tag	Descrição
1xx Informativa	Indica que a requisição foi recebida e entendida. A resposta é provisória enquanto o processamento continua. Serve para alertar o cliente a esperar pela resposta final. Apenas o status-line e cabeçalhos opcionais são incluídos; encerrada por uma linha vazia. Servidores HTTP/1.0 não devem enviar uma resposta 1xx, exceto sob condições experimentais.
100	Continuar: O servidor recebeu os cabeçalhos da solicitação e o cliente deve enviar o corpo do pedido.
101	Mudando protocolos: O solicitante pediu ao servidor para mudar os protocolos e este está reconhecendo que irá fazê-lo.
102	Processamento (WebDAV): O servidor recebeu e está processando o pedido, mas ainda não há resposta disponível.



122	Pedido-URI muito longo: URI é maior do que o máximo de 2083 caracteres. (Padrão IE7).
<b>2xx Sucesso</b>	A ação solicitada foi recebida, compreendida, aceita e processada com êxito.
200	OK: A resposta depende do método de solicitação. Em GET, a resposta conterá o recurso solicitado. Em POST, a resposta conterá a descrição da ação realizada.
201	Criado: O pedido foi cumprido e resultou na criação de um novo recurso.
202	Aceito: O pedido foi aceito para processamento, mas o tratamento ainda não foi concluído.
203	Não-autorizado: O servidor retornou informações que podem ser de outra fonte.
204	Nenhum conteúdo: O servidor processou a solicitação, mas não há necessidade de resposta.
205	Reset: O servidor processou a solicitação com sucesso, mas o cliente deve redefinir a exibição de documento.
206	Conteúdo parcial: O servidor está entregando apenas parte do recurso devido a um "cabeçalho intervalo" enviado pelo cliente.
207	Status Multi (WebDAV): O corpo da mensagem é um XML contendo vários códigos de resposta para sub-pedidos.
3xx Redirecionamento	O cliente deve tomar medidas adicionais para completar o pedido.
300	Múltipla escolha: Várias opções estão disponíveis para o recurso.
301	Movido: Todas as solicitações futuras devem ser direcionadas para o URL fornecido.
302	Encontrado: O recurso foi movido temporariamente. (Comportamento histórico similar ao 303).
303	Consulte Outros: A resposta pode ser encontrada em outro URI usando o método GET.
304	Não modificado: O recurso não foi modificado desde o último pedido.
305	Use Proxy: Muitos clientes HTTP podem não tratar corretamente esse código por razões de segurança.
306	Proxy Switch: Mudança de proxy (não utilizado atualmente).
307	Redirecionamento temporário: O pedido deve ser repetido com outro URI, mas futuras solicitações podem usar o URI original.
308	Redirecionamento permanente: O recurso foi movido para um novo URI permanente. Todas as solicitações futuras devem usar o URI retornado.
4xx Erro de cliente	Indica que o cliente cometeu um erro.
400	Requisição inválida: O pedido não pôde ser entregue devido à sintaxe incorreta.
401	Não autorizado: A autenticação é possível, mas não fornecida ou falhou.



402	Pagamento necessário: Reservado para uso futuro; não utilizado normalmente.
403	Proibido: O pedido é reconhecido, mas o servidor se recusa a executá-lo.
404	Não encontrado: O recurso requisitado não foi encontrado, mas pode estar disponível no futuro.
405	Método não permitido: O método de pedido não é compatível com o recurso.
406	Não aceitável: O recurso solicitado não pode gerar conteúdos aceitáveis de acordo com os cabeçalhos Accept enviados.
407	Autenticação de proxy necessária: A autenticação é necessária para acessar o proxy.
408	Tempo de requisição esgotou: O cliente não apresentou o pedido dentro do tempo esperado.
409	Conflito geral: A solicitação não pode ser processada devido a um conflito.
410	Gone: O recurso não está mais disponível e não estará novamente.
411	Comprimento necessário: O pedido não especifica o comprimento do conteúdo exigido.
412	Pré-condição falhou: O servidor não cumpriu uma condição colocada na solicitação.
413	Entidade de solicitação muito grande: A solicitação é maior do que o servidor pode processar.
414	Pedido-URI muito longo: O URI fornecido é muito longo para ser processado.
415	Tipo de mídia não suportado: O tipo de mídia do pedido não é suportado.
416	Solicitação de faixa não satisfatória: O cliente solicitou uma parte do arquivo que o servidor não pode fornecer.
417	Falha na expectativa: O servidor não pode cumprir as exigências do cabeçalho "Expect".
418	Eu sou um bule de chá: Código de brincadeira de 1º de abril, não implementado por servidores reais.
422	Entidade improcessável (WebDAV): O pedido foi bem formado, mas não pode ser seguido devido a erros de semântica.
423	Fechado (WebDAV): O recurso está bloqueado.
424	Falha de Dependência (WebDAV): A solicitação falhou devido à falha de uma solicitação anterior.
425	Coleção não ordenada: Definido em WebDAV Protocolos de Coleções Avançadas, não presente em WebDAV Protocolo de Coleções Ordenadas.
426	Upgrade Obrigatório: O cliente deve mudar para outro protocolo, como TLS/1.0.
429	Pedidos em excesso: O usuário enviou muitas solicitações em um determinado período de tempo.



444	Sem resposta: Extensão Nginx que fecha a conexão sem retornar informação.
449	Repetir: Extensão Microsoft; o pedido deve ser repetido após ação apropriada.
450	Bloqueados pelo Controle de Pais do Windows: Extensão Microsoft que bloqueia o acesso devido a controles parentais.
451	Indisponível por razões legais: O acesso ao recurso foi negado por uma ordem legal.
499	Cliente fechou pedido: Extensão Nginx que indica que o cliente fechou a conexão antes da resposta.
<b>5xx Erros do servidor</b>	Indica erros do servidor ao processar a solicitação.
500	Erro interno do servidor: Erro genérico indicando falha no servidor ao processar a solicitação.
501	Não implementado: O servidor ainda não suporta a funcionalidade solicitada.
502	Bad Gateway: Erro na configuração entre servidores de back-end.
503	Serviço indisponível: O servidor está em manutenção ou sobrecarregado.
504	Gateway Time-Out: O servidor não recebeu uma resposta a tempo de um servidor upstream.
505	HTTP Version not supported: O servidor não suporta a versão do protocolo HTTP utilizada na solicitação.



## RESUMO

```
<!DOCTYPE html>
<html>
<body>

<h1>Meu primeiro título.</h1>

<p>Meu primeiro parágrafo.</p>

</body>
</html>
```

### Meu primeiro título.

Meu primeiro parágrafo.

O que é HTML?

- HTML significa **Hyper Text Markup Language**.
- HTML é a **linguagem de marcação padrão para criar páginas da Web**;
- HTML descreve a estrutura de uma página da Web;
- HTML consiste em uma série de elementos;
- Os elementos HTML informam ao navegador como exibir o conteúdo;
- Os elementos HTML rotulam partes do conteúdo como "isto é um título", "isto é um parágrafo", "isto é um link", etc.

```
<html>
```

```
<cabeça>
```

```
<title>Título da página</title>
```

```
</head>
```

```
<corpo>
```

```
<h1>Este é um título</h1>
```

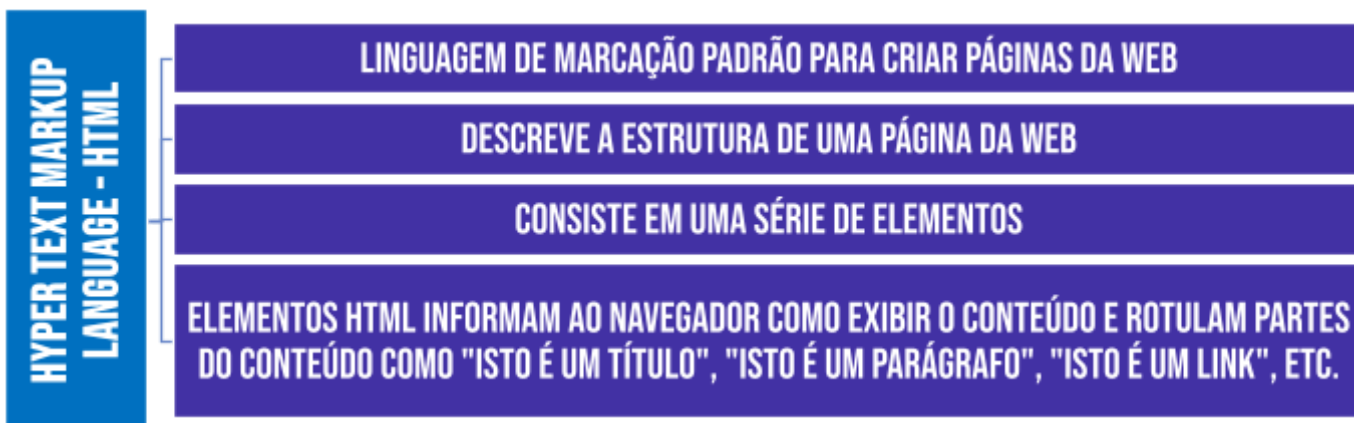
```
<p>Isto é um parágrafo.</p>
```

```
<p>Este é outro parágrafo.</p>
```

```
</body>
```

```
</html>
```





## HTML Básico

<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;h1&gt;Meu primeiro título.&lt;/h1&gt;  &lt;p&gt;Meu primeiro parágrafo.&lt;/p&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<p><b>Meu primeiro título.</b></p> <p>Meu primeiro parágrafo.</p>
--	---

```
<!DOCTYPE html>
```

Os cabeçalhos HTML são definidos com as tags <h1>to <h6>. <h1>define o cabeçalho mais importante. Enquanto <h6>define o título menos importante:

<pre>&lt;h1&gt;Este é o título 1&lt;/h1&gt; &lt;h2&gt;Este é o título 2&lt;/h2&gt; &lt;h3&gt;Este é o título 3&lt;/h3&gt;</pre>	<p><b>Este é o título 1</b></p> <p><b>Este é o título 2</b></p> <p><b>Este é o título 3</b></p>
---	---

## Elementos HTML

Um elemento HTML é definido por uma tag inicial, algum conteúdo e uma tag final. O elemento HTML é tudo, desde a tag inicial até a tag final:

```
< tagname > O conteúdo vai aqui... < /tagname >  
< h1 > Meu primeiro cabeçalho < /h1 >  
< p > Meu primeiro parágrafo. < /p >
```

Tag inicial	conteúdo do elemento	Tag final
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>
 	<i>none</i>	<i>none</i>

```
<!DOCTYPE html>  
<html>  
<body>  
  
<h1>Meu primeiro título.</h1>  
  
<p>Meu primeiro parágrafo.</p>  
  
</body>  
</html>
```

**Meu primeiro título.**  
Meu primeiro parágrafo.

## Atributos HTML

Todos os elementos HTML podem ter atributos. Os atributos fornecem informações adicionais sobre os elementos, e são sempre especificados na tag de início. Os atributos geralmente vêm em **pares de nome/valor** como: `name="value"`.

A tag `<a>` define um **hiperlink**. O `href` atributo especifica a URL da página para a qual o link vai:

```
<a href="https://www.w3schools.com">Este é um link</a>
```

[Este é um link](https://www.w3schools.com)

## Cabeçalhos HTML

Cabeçalhos HTML são títulos ou subtítulos que você deseja exibir em uma página da web. Os cabeçalhos HTML são definidos com as tags `<h1>` até `<h6>`. Lembrando, `<h1>` define o cabeçalho mais importante. `<h6>` define o título menos importante.



<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;h1&gt;Título 1&lt;/h1&gt; &lt;h2&gt;Título 2&lt;/h2&gt; &lt;h3&gt;Título 3&lt;/h3&gt; &lt;h4&gt;Título 4&lt;/h4&gt; &lt;h5&gt;Título 5&lt;/h5&gt; &lt;h6&gt;Título 6&lt;/h6&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<h1>Título 1</h1> <h2>Título 2</h2> <h3>Título 3</h3> <h4>Título 4</h4> <h5>Título 5</h5> <h6>Título 6</h6>
--	---

## Parágrafos HTML

Um parágrafo sempre começa em uma nova linha e geralmente é um bloco de texto. O elemento HTML `<p>` define um parágrafo. Um parágrafo sempre começa em uma nova linha e os navegadores adicionam automaticamente algum espaço em branco (uma margem) antes e depois de um parágrafo.

<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;p&gt;Isto é um parágrafo.&lt;/p&gt; &lt;p&gt;Isto é um parágrafo.&lt;/p&gt; &lt;p&gt;Isto é um parágrafo.&lt;/p&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<p>Isto é um parágrafo.</p> <p>Isto é um parágrafo.</p> <p>Isto é um parágrafo.</p>
--	---

A tag `<hr>` define uma quebra temática em uma página HTML e geralmente é exibida como uma régua horizontal. O elemento `<hr>` é usado para separar o conteúdo (ou definir uma alteração) em uma página HTML. A tag `<hr>` é uma tag vazia, o que significa que não tem tag final.

## Estilos HTML

O atributo HTML `style` é usado para adicionar estilos a um elemento, como cor, fonte, tamanho e muito mais. A definição do estilo de um elemento HTML pode ser feita com o atributo `style`.





<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;p&gt;Eu sou normal&lt;/p&gt; &lt;p style="color:red;"&gt;Eu sou vermelho&lt;/p&gt; &lt;p style="color:blue;"&gt;Eu sou azul&lt;/p&gt; &lt;p style="font-size:50px;"&gt;Eu sou grande&lt;/p&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<p>Eu sou normal</p> <p>Eu sou vermelho</p> <p>Eu sou azul</p> <h1>Eu sou grande</h1>
---	--

O atributo HTML style tem a seguinte sintaxe<sup>1</sup>:

```
<tagname style="property.value;">
```

<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;body&gt;  &lt;h1&gt;Este é o título 1&lt;/h1&gt; &lt;p&gt;Este é um texto.&lt;/p&gt; &lt;hr&gt;  &lt;h2&gt;Este é o título 2&lt;/h2&gt; &lt;p&gt;Este é outro texto.&lt;/p&gt; &lt;hr&gt;  &lt;h2&gt;Este é o título 2&lt;/h2&gt; &lt;p&gt;Este é outro texto.&lt;/p&gt;  &lt;/body&gt; &lt;/html&gt;</pre>	<h1>Este é o título 1</h1> <p>Este é um texto.</p> <hr/> <h2>Este é o título 2</h2> <p>Este é outro texto.</p> <hr/> <h2>Este é o título 2</h2> <p>Este é outro texto.</p>
---	--

<pre>&lt;pre&gt;   O que você vai estudar amanhã?    Quantos dias você tem até sua próxima prova?    Qual sua meta de aproveitamento na próxima prova? &lt;/pre&gt;</pre>	<p>O que você vai estudar amanhã?</p> <p>Quantos dias você tem até sua próxima prova?</p> <p>Qual sua meta de aproveitamento na próxima prova?</p>
---	--

<sup>1</sup> *property* é uma propriedade CSS. *value* é um valor CSS.



PARÁGRAFOS HTML

<P> DEFINE UM PARÁGRAFO

<HR> DEFINE UMA MUDANÇA TEMÁTICA NO CONTEÚDO

<BR> INSERE UMA ÚNICA QUEBRA DE LINHA

<PRE> DEFINE O TEXTO PRÉ-FORMATADO

| Propriedade CSS  | Descrição  |
|------------------|--|
| Background-color | Define a cor de fundo de um elemento HTML                      |
| Color            | Define a cor do texto para um elemento HTML                    |
| Font-family      | Define a fonte a ser usada para um elemento HTML               |
| Font-size        | Define o tamanho do texto para um elemento HTML                |
| Text-align       | Define o alinhamento horizontal do texto para um elemento HTML |

## Formatação de Texto HTML

ELEMENTOS DE FORMATAÇÃO HTML

<B> TEXTO EM NEGRITO

<STRONG> TEXTO IMPORTANTE

<I> TEXTO EM ITÁLICO

<EM> TEXTO ENFATIZADO

<MARK> TEXTO MARCADO

<SMALL> TEXTO MENOR

<DEL> TEXTO DELETADO

<INS> TEXTO INSERIDO

<SUB> TEXTO SUBSCRITO

<SUP> TEXTO SOBRESCRITO





Vejamos inicialmente a Referência completa do HTML, incluindo todas as tags.

| Tag          | Descrição   |
|--------------|---|
| <!--...-->   | Define um comentário  |
| <!DOCTYPE>   | Define o tipo de documento  |
| <a>          | Define um hiperlink   |
| <abbr>       | Define uma abreviação ou um acrônimo  |
| <acronym>    | Não suportado em HTML5. Em vez disso, use <abbr>  |
| <endereço>   | Define as informações de contato do autor/proprietário de um documento  |
| <applet>     | Não suportado em HTML5. Use <embed>   |
| <area>       | Define uma área dentro de um mapa de imagem   |
| <article>    | Define um artigo  |
| <aside>      | Define o conteúdo além do conteúdo da página  |
| <audio>      | Define o conteúdo de som incorporado  |
| <b>          | Define o texto em negrito   |
| <base>       | Especifica a URL/alvo base para todas as URLs relativas em um documento   |
| <basefont>   | Não suportado em HTML5. Em vez disso, use CSS. Especifica uma cor, tamanho e fonte padrão para todo o texto em um documento |
| <bdi>        | Isola uma parte do texto que pode estar formatada em uma direção diferente de outro texto fora dele                         |
| <bdo>        | Substitui a direção do texto atual  |
| <big>        | Não suportado em HTML5. Em vez disso, use CSS. Define texto grande  |
| <blockquote> | Define uma seção que é citada de outra fonte  |
| <body>       | Define o corpo do documento   |
| <br>         | Define uma única quebra de linha  |
| <button>     | Define um botão clicável  |
| <canvas>     | Usado para desenhar gráficos, em tempo real, via script (geralmente JavaScript)   |
| <caption>    | Define uma legenda de tabela  |
| <center>     | Não suportado em HTML5. Em vez disso, use CSS. Define o texto centrado  |
| <cite>       | Define o título de uma obra   |
| <code>       | Define um pedaço de código de computador  |
| <col>        | Especifica as propriedades da coluna para cada coluna dentro de um elemento <colgroup>                                      |
| <colgroup>   | Especifica um grupo de uma ou mais colunas em uma tabela para formatação  |
| <data>       | Adiciona uma tradução legível por máquina de um determinado conteúdo  |
| <datalist>   | Especifica uma lista de opções predefinidas para controles de entrada   |



|                                 |   |
|---------------------------------|---|
| <code>&lt;dd&gt;</code>         | Define uma descrição/valor de um termo em uma lista de descrição  |
| <code>&lt;del&gt;</code>        | Define o texto que foi excluído de um documento   |
| <code>&lt;details&gt;</code>    | Define detalhes adicionais que o usuário pode visualizar ou ocultar   |
| <code>&lt;dfn&gt;</code>        | Especifica um termo que será definido dentro do conteúdo  |
| <code>&lt;dialog&gt;</code>     | Define uma caixa de diálogo ou janela   |
| <code>&lt;dir&gt;</code>        | Não suportado em HTML5. Em vez disso, use <code>&lt;ul&gt;</code>   |
| <code>&lt;div&gt;</code>        | Define uma seção em um documento  |
| <code>&lt;dl&gt;</code>         | Define uma lista de descrição   |
| <code>&lt;dt&gt;</code>         | Define um termo/nome em uma lista de descrição  |
| <code>&lt;em&gt;</code>         | Define o texto enfatizado   |
| <code>&lt;embed&gt;</code>      | Define um contêiner para um aplicativo externo  |
| <code>&lt;fieldset&gt;</code>   | Agrupa elementos relacionados em um formulário  |
| <code>&lt;figcaption&gt;</code> | Define uma legenda para um elemento <code>&lt;figure&gt;</code>   |
| <code>&lt;figure&gt;</code>     | Especifica o conteúdo independente  |
| <code>&lt;font&gt;</code>       | Não suportado em HTML5. Em vez disso, use CSS. Define a fonte, a cor e o tamanho do texto                     |
| <code>&lt;footer&gt;</code>     | Define um rodapé para um documento ou seção   |
| <code>&lt;form&gt;</code>       | Define um formulário HTML para entrada do usuário   |
| <code>&lt;frame&gt;</code>      | Não suportado em HTML5. Define uma janela (um quadro) em um conjunto de quadros                               |
| <code>&lt;frameset&gt;</code>   | Não suportado em HTML5. Define um conjunto de quadros   |
| <code>&lt;h1&gt;</code>         | a <code>&lt;h6&gt;</code>   |
| <code>&lt;head&gt;</code>       | Contém metadados/informações para o documento   |
| <code>&lt;header&gt;</code>     | Define um cabeçalho para um documento ou seção  |
| <code>&lt;hr&gt;</code>         | Define uma mudança temática no conteúdo   |
| <code>&lt;html&gt;</code>       | Define a raiz de um documento HTML  |
| <code>&lt;i&gt;</code>          | Define uma parte do texto em uma voz ou humor alternativo   |
| <code>&lt;iframe&gt;</code>     | Define um quadro embutido. Uma página dentro de outra página  |
| <code>&lt;img&gt;</code>        | Define uma imagem   |
| <code>&lt;input&gt;</code>      | Define um controle de entrada   |
| <code>&lt;ins&gt;</code>        | Define um texto que foi inserido em um documento  |
| <code>&lt;kbd&gt;</code>        | Define a entrada do teclado   |
| <code>&lt;label&gt;</code>      | Define um rótulo para um elemento <code>&lt;input&gt;</code>  |
| <code>&lt;legend&gt;</code>     | Define uma legenda para um elemento <code>&lt;fieldset&gt;</code>   |
| <code>&lt;li&gt;</code>         | Define um item de lista   |
| <code>&lt;link&gt;</code>       | Define o relacionamento entre um documento e um recurso externo (mais usado para vincular a folhas de estilo) |



|                           |  |
|---------------------------|--|
| <b>&lt;main&gt;</b>       | Especifica o conteúdo principal de um documento  |
| <b>&lt;map&gt;</b>        | Define um mapa de imagem   |
| <b>&lt;mark&gt;</b>       | Define o texto marcado/destacado   |
| <b>&lt;meta&gt;</b>       | Define metadados sobre um documento HTML   |
| <b>&lt;meter&gt;</b>      | Define uma medição escalar dentro de um intervalo conhecido (um medidor)                     |
| <b>&lt;nav&gt;</b>        | Define links de navegação  |
| <b>&lt;noframes&gt;</b>   | Não suportado em HTML5. Define um conteúdo alternativo para usuários que não suportam frames |
| <b>&lt;noscript&gt;</b>   | Define um conteúdo alternativo para usuários que não suportam scripts do lado do cliente     |
| <b>&lt;object&gt;</b>     | Define um contêiner para um aplicativo externo   |
| <b>&lt;ol&gt;</b>         | Define uma lista ordenada  |
| <b>&lt;optgroup&gt;</b>   | Define um grupo de opções relacionadas em uma lista suspensa                                 |
| <b>&lt;option&gt;</b>     | Define uma opção em uma lista suspensa   |
| <b>&lt;output&gt;</b>     | Define o resultado de um cálculo   |
| <b>&lt;p&gt;</b>          | Define um parágrafo  |
| <b>&lt;param&gt;</b>      | Define um parâmetro para um objeto   |
| <b>&lt;picture&gt;</b>    | Define um contêiner para vários recursos de imagem   |
| <b>&lt;pre&gt;</b>        | Define o texto pré-formatado   |
| <b>&lt;progress&gt;</b>   | Representa o progresso de uma tarefa   |
| <b>&lt;q&gt;</b>          | Define uma citação curta   |
| <b>&lt;rp&gt;</b>         | Define o que mostrar em navegadores que não suportam anotações ruby                          |
| <b>&lt;rt&gt;</b>         | Define uma explicação/pronúncia de caracteres (para tipografia do Leste Asiático)            |
| <b>&lt;ruby&gt;</b>       | Define uma anotação rubi (para tipografia do Leste Asiático)                                 |
| <b>&lt;s&gt;</b>          | Define o texto que não está mais correto   |
| <b>&lt;samp&gt;</b>       | Define a saída de amostra de um programa de computador                                       |
| <b>&lt;script&gt;</b>     | Define um script do lado do cliente  |
| <b>&lt;section&gt;</b>    | Define uma seção em um documento   |
| <b>&lt;selecionar&gt;</b> | Define uma lista suspensa  |
| <b>&lt;small&gt;</b>      | Define um texto menor  |
| <b>&lt;source&gt;</b>     | Define vários recursos de mídia para elementos de mídia (<vídeo>)                            |
| <b>&lt;span&gt;</b>       | Define uma seção em um documento   |
| <b>&lt;strike&gt;</b>     | Não suportado em HTML5. Use <del>  |
| <b>&lt;estilo&gt;</b>     | Define informações de estilo para um documento   |
| <b>&lt;sub&gt;</b>        | Define o texto subscrito   |



|                         |   |
|-------------------------|---|
| <b>&lt;summary&gt;</b>  | Define um cabeçalho visível para um elemento <details>                                |
| <b>&lt;sup&gt;</b>      | Define o texto sobrescrito  |
| <b>&lt;svg&gt;</b>      | Define um contêiner para gráficos SVG   |
| <b>&lt;table&gt;</b>    | Define uma tabela   |
| <b>&lt;tbody&gt;</b>    | Agrupar o conteúdo do corpo em uma tabela   |
| <b>&lt;td&gt;</b>       | Define uma célula em uma tabela   |
| <b>&lt;template&gt;</b> | Define um contêiner para o conteúdo que deve ser ocultado quando a página é carregada |
| <b>&lt;textarea&gt;</b> | Define um controle de entrada multilinha (área de texto)                              |
| <b>&lt;tfoot&gt;</b>    | Agrupar o conteúdo do rodapé em uma tabela  |
| <b>&lt;th&gt;</b>       | Define uma célula de cabeçalho em uma tabela  |
| <b>&lt;thead&gt;</b>    | Agrupar o conteúdo do cabeçalho em uma tabela   |
| <b>&lt;time&gt;</b>     | Define uma hora específica (ou datettime)   |
| <b>&lt;title&gt;</b>    | Define um título para o documento   |
| <b>&lt;tr&gt;</b>       | Define uma linha em uma tabela  |
| <b>&lt;track&gt;</b>    | Define trilhas de texto para elementos de mídia (<video>)                             |
| <b>&lt;tt&gt;</b>       | Não suportado em HTML5. Em vez disso, use CSS. Define o texto do teletipo             |
| <b>&lt;u&gt;</b>        | Define algum texto que não é articulado e tem um estilo diferente do texto normal     |
| <b>&lt;ul&gt;</b>       | Define uma lista não ordenada   |
| <b>&lt;var&gt;</b>      | Define uma variável   |
| <b>&lt;video&gt;</b>    | Define o conteúdo de vídeo incorporado  |
| <b>&lt;wbr&gt;</b>      | Define uma possível quebra de linha   |



## Mais cobrados em provas

### <iframe>

Um iframe (abreviação de "frame inline") é um elemento HTML usado para incorporar uma página da Web em outra página da Web. Ele fornece uma maneira de exibir o conteúdo de outra página da Web na página da Web atual e é comumente usado para incorporar vídeos, mapas e outros conteúdos interativos.

Para criar um iframe, você pode usar o elemento <iframe> e especificar o local da página a ser incorporada usando o atributo src. Você também pode usar os atributos de largura e altura para especificar o tamanho do iframe na página da web.

O elemento <iframe> possui vários atributos que podem ser usados para personalizar seu comportamento, como rolagem, frameborder e allowfullscreen. Você também pode usar JavaScript para controlar o elemento iframe programaticamente, usando os métodos e propriedades do elemento iframe.

Lembre-se de que os iframes podem ser um risco de segurança se você incorporar conteúdo de fontes não confiáveis, pois o conteúdo incorporado tem acesso à página da Web principal e pode executar códigos maliciosos. É importante ser cauteloso ao usar iframes e incorporar apenas conteúdo de fontes confiáveis.

### Símbolos HTML

| Tag | Número | Entidade | Descrição                  |
|-----|--------|----------|----------------------------|
| ©   | &#169  | &copy    | Sinal de direitos autorais |
| ®   | &#174  | &reg     | Assinatura registrada      |
| €   | &#8364 | &euro    | Sinal do euro              |
| ™   | &#8482 | &trade   | Marca comercial            |
| ←   | &#8592 | &larr    | Seta para a esquerda       |
| ↑   | &#8593 | &uarr    | Seta para cima             |
| →   | &#8594 | &rarr    | Seta para a direita        |
| ↓   | &#8595 | &darr    | Seta para baixo            |
| ♠   | &#9824 | &spades  | Espadas preto              |
| ♣   | &#9827 | &clubs   | Club preto                 |
| ♥   | &#9829 | &hearts  | Coração negro              |
| ♦   | &#9830 | &diams   | Diamante negro             |





## getElementById

Em HTML, a função `getElementById` é um método do objeto de documento que pode ser usado para recuperar um elemento do documento por seu identificador exclusivo (ID). A função `getElementById` retorna uma referência ao elemento com o ID especificado ou `null` se tal elemento não existir.

Vejamos um exemplo de como a função `getElementById` pode ser usada:

```
<div id="myDiv"> Este é um elemento div.</div>

<script>
  var myDiv = document.getElementById('myDiv');
  console.log(myDiv.innerHTML); // Outputs: " Este é um elemento div."
</script>
```

Neste exemplo, a função `getElementById` é usada para recuperar o elemento `<div>` com um ID de "myDiv". A propriedade `innerHTML` do elemento é registrada no console.

A função `getElementById` é uma maneira útil de acessar elementos no DOM (Document Object Model) e manipular suas propriedades e estilos. Geralmente é usado em conjunto com outros métodos e propriedades DOM, como `innerHTML` e `estilo`, para criar páginas da Web dinâmicas e interativas.

## <meta>

A tag `<meta>` define metadados sobre um documento HTML. Metadados são dados (informações) sobre dados. As tags `<meta>` sempre ficam dentro do elemento `<head>` e normalmente são usadas para especificar o conjunto de caracteres, a descrição da página, as palavras-chave, o autor do documento e as configurações da viewport. Vejamos os atributos, os valores e suas respectivas descrições.

| Atributo                | Valor   | Descrição   |
|-------------------------|---|---|
| <code>charset</code>    | <code>character_set</code>  | Especifica a codificação de caracteres para o documento HTML                          |
| <code>content</code>    | <code>text</code>   | Especifica o valor associado ao atributo <code>http-equiv</code> ou <code>name</code> |
| <code>http-equiv</code> | <code>content-security-policy</code><br><code>content-type</code><br><code>default-style</code><br><code>refresh</code> | Fornece um cabeçalho HTTP para as informações/valor do atributo de conteúdo           |
| <code>name</code>       | <code>application-name</code><br><code>author</code>  | Especifica um nome para os metadados  |



|  |             |  |
|--|-------------|--|
|  | description |  |
|  | generator   |  |
|  | keywords    |  |
|  | viewport    |  |

Os metadados são usados por navegadores (como exibir conteúdo ou recarregar a página), mecanismos de pesquisa (palavras-chave) e outros serviços da web. Vejamos alguns exemplos:

Define palavras-chave para motores de busca:

```
<meta name="keywords" content="HTML, CSS, JavaScript">
```

Define uma descrição da sua página da web:

```
<meta name="description" content="Free Web tutorials for HTML and CSS">
```

Define o autor de uma página:

```
<meta name="author" content="John Doe">
```

Atualize o documento a cada 30 segundos:

```
<meta http-equiv="refresh" content="30">
```

Configurando a janela de visualização para que seu site tenha uma boa aparência em todos os dispositivos:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

A viewport é a área visível do usuário de uma página da web. Varia de acordo com o dispositivo - será menor em um telefone celular do que na tela do computador. Você deve incluir o seguinte elemento <meta> em todas as suas páginas da web:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

## Manifesto de cache

Para tratar um site offline com HTML5, você pode usar o recurso de armazenamento de cache do HTML5, que permite que você armazene arquivos em cache no navegador do usuário. Isso significa que, quando o usuário visita o site pela primeira vez, os arquivos



são armazenados em cache e, em visitas posteriores, o site é carregado a partir do cache, em vez de ser baixado da internet.

Para habilitar o armazenamento de cache, você precisa criar um arquivo chamado "manifesto de cache" e vinculá-lo ao seu HTML usando a tag `<html>`. O manifesto de cache é um arquivo de texto simples que contém uma lista de arquivos que devem ser armazenados em cache. Aqui está um exemplo de um manifesto de cache:

```
CACHE MANIFEST
# versão 1.0
index.html
css/styles.css
js/scripts.js
images/logo.png
```

Em seguida, você precisa adicionar a tag `<html>` ao seu arquivo HTML e definir o atributo `manifest` como o caminho para o seu arquivo de manifesto de cache:

```
<html manifest="/cache.manifest">
```

Isso permitirá que o navegador armazene em cache os arquivos especificados no manifesto de cache. Quando o usuário visita o site pela primeira vez, os arquivos serão baixados e armazenados em cache. Em visitas posteriores, o site será carregado a partir do cache, permitindo que ele funcione offline.

## Erros HTTP

<b>ERROS HTTP</b>	<b>1XX: INFORMATIVO - O SERVIDOR ESTÁ PROCESSANDO A SOLICITAÇÃO.</b>
	<b>2XX: SUCESSO - A SOLICITAÇÃO FOI BEM-SUCEDIDA.</b>
	<b>3XX: REDIRECIONAMENTO - O SERVIDOR SOLICITA UMA NOVA AÇÃO OU REDIRECIONA.</b>
	<b>4XX: ERRO DO CLIENTE - PROBLEMA COM A SOLICITAÇÃO FEITA PELO CLIENTE.</b>
	<b>5XX: ERRO DO SERVIDOR - PROBLEMA NO SERVIDOR AO PROCESSAR A SOLICITAÇÃO.</b>

Tag	Descrição
1xx Informativa	Indica que a requisição foi recebida e entendida. A resposta é provisória enquanto o processamento continua. Serve para alertar o cliente a esperar pela resposta final. Apenas o status-line e cabeçalhos opcionais são incluídos; encerrada por uma linha vazia. Servidores HTTP/1.0 não devem enviar uma resposta 1xx, exceto sob condições experimentais.



100	Continuar: O servidor recebeu os cabeçalhos da solicitação e o cliente deve enviar o corpo do pedido.
101	Mudando protocolos: O solicitante pediu ao servidor para mudar os protocolos e este está reconhecendo que irá fazê-lo.
102	Processamento (WebDAV): O servidor recebeu e está processando o pedido, mas ainda não há resposta disponível.
122	Pedido-URI muito longo: URI é maior do que o máximo de 2083 caracteres. (Padrão IE7).
2xx Sucesso	A ação solicitada foi recebida, compreendida, aceita e processada com êxito.
200	OK: A resposta depende do método de solicitação. Em GET, a resposta conterá o recurso solicitado. Em POST, a resposta conterá a descrição da ação realizada.
201	Criado: O pedido foi cumprido e resultou na criação de um novo recurso.
202	Aceito: O pedido foi aceito para processamento, mas o tratamento ainda não foi concluído.
203	Não-autorizado: O servidor retornou informações que podem ser de outra fonte.
204	Nenhum conteúdo: O servidor processou a solicitação, mas não há necessidade de resposta.
205	Reset: O servidor processou a solicitação com sucesso, mas o cliente deve redefinir a exibição de documento.
206	Conteúdo parcial: O servidor está entregando apenas parte do recurso devido a um "cabeçalho intervalo" enviado pelo cliente.
207	Status Multi (WebDAV): O corpo da mensagem é um XML contendo vários códigos de resposta para sub-pedidos.
3xx Redirecionamento	O cliente deve tomar medidas adicionais para completar o pedido.
300	Múltipla escolha: Várias opções estão disponíveis para o recurso.
301	Movido: Todas as solicitações futuras devem ser direcionadas para o URL fornecido.
302	Encontrado: O recurso foi movido temporariamente. (Comportamento histórico similar ao 303).
303	Consulte Outros: A resposta pode ser encontrada em outro URI usando o método GET.
304	Não modificado: O recurso não foi modificado desde o último pedido.
305	Use Proxy: Muitos clientes HTTP podem não tratar corretamente esse código por razões de segurança.
306	Proxy Switch: Mudança de proxy (não utilizado atualmente).
307	Redirecionamento temporário: O pedido deve ser repetido com outro URI, mas futuras solicitações podem usar o URI original.



<b>308</b>	Redirecionamento permanente: O recurso foi movido para um novo URI permanente. Todas as solicitações futuras devem usar o URI retornado.
<b>4xx Erro de cliente</b>	Indica que o cliente cometeu um erro.
<b>400</b>	Requisição inválida: O pedido não pôde ser entregue devido à sintaxe incorreta.
<b>401</b>	Não autorizado: A autenticação é possível, mas não fornecida ou falhou.
<b>402</b>	Pagamento necessário: Reservado para uso futuro; não utilizado normalmente.
<b>403</b>	Proibido: O pedido é reconhecido, mas o servidor se recusa a executá-lo.
<b>404</b>	Não encontrado: O recurso requisitado não foi encontrado, mas pode estar disponível no futuro.
<b>405</b>	Método não permitido: O método de pedido não é compatível com o recurso.
<b>406</b>	Não aceitável: O recurso solicitado não pode gerar conteúdos aceitáveis de acordo com os cabeçalhos Accept enviados.
<b>407</b>	Autenticação de proxy necessária: A autenticação é necessária para acessar o proxy.
<b>408</b>	Tempo de requisição esgotou: O cliente não apresentou o pedido dentro do tempo esperado.
<b>409</b>	Conflito geral: A solicitação não pode ser processada devido a um conflito.
<b>410</b>	Gone: O recurso não está mais disponível e não estará novamente.
<b>411</b>	Comprimento necessário: O pedido não especifica o comprimento do conteúdo exigido.
<b>412</b>	Pré-condição falhou: O servidor não cumpriu uma condição colocada na solicitação.
<b>413</b>	Entidade de solicitação muito grande: A solicitação é maior do que o servidor pode processar.
<b>414</b>	Pedido-URI muito longo: O URI fornecido é muito longo para ser processado.
<b>415</b>	Tipo de mídia não suportado: O tipo de mídia do pedido não é suportado.
<b>416</b>	Solicitação de faixa não satisfatória: O cliente solicitou uma parte do arquivo que o servidor não pode fornecer.
<b>417</b>	Falha na expectativa: O servidor não pode cumprir as exigências do cabeçalho "Expect".
<b>418</b>	Eu sou um bule de chá: Código de brincadeira de 1º de abril, não implementado por servidores reais.
<b>422</b>	Entidade improcessável (WebDAV): O pedido foi bem formado, mas não pode ser seguido devido a erros de semântica.
<b>423</b>	Fechado (WebDAV): O recurso está bloqueado.



424	Falha de Dependência (WebDAV): A solicitação falhou devido à falha de uma solicitação anterior.
425	Coleção não ordenada: Definido em WebDAV Protocolos de Coleções Avançadas, não presente em WebDAV Protocolo de Coleções Ordenadas.
426	Upgrade Obrigatório: O cliente deve mudar para outro protocolo, como TLS/1.0.
429	Pedidos em excesso: O usuário enviou muitas solicitações em um determinado período de tempo.
444	Sem resposta: Extensão Nginx que fecha a conexão sem retornar informação.
449	Repetir: Extensão Microsoft; o pedido deve ser repetido após ação apropriada.
450	Bloqueados pelo Controle de Pais do Windows: Extensão Microsoft que bloqueia o acesso devido a controles parentais.
451	Indisponível por razões legais: O acesso ao recurso foi negado por uma ordem legal.
499	Cliente fechou pedido: Extensão Nginx que indica que o cliente fechou a conexão antes da resposta.
<b>5xx Erros do servidor</b>	Indica erros do servidor ao processar a solicitação.
500	Erro interno do servidor: Erro genérico indicando falha no servidor ao processar a solicitação.
501	Não implementado: O servidor ainda não suporta a funcionalidade solicitada.
502	Bad Gateway: Erro na configuração entre servidores de back-end.
503	Serviço indisponível: O servidor está em manutenção ou sobrecarregado.
504	Gateway Time-Out: O servidor não recebeu uma resposta a tempo de um servidor upstream.
505	HTTP Version not supported: O servidor não suporta a versão do protocolo HTTP utilizada na solicitação.



## REFERÊNCIAS

1. W3Schools - HTML  
*W3Schools HTML Reference*  
<https://www.w3schools.com/html/>  
Um guia completo sobre HTML, incluindo tutoriais e exemplos para todos os aspectos da linguagem.
2. Wikipedia - Lista de códigos de estado HTTP  
*Lista de códigos de estado HTTP*  
[https://pt.wikipedia.org/wiki/Lista\\_de\\_c%C3%B3digos\\_de\\_estado\\_HTTP](https://pt.wikipedia.org/wiki/Lista_de_c%C3%B3digos_de_estado_HTTP)
3. MDN Web Docs - HTML  
*MDN Web Docs HTML*  
<https://developer.mozilla.org/pt-BR/docs/Web/HTML>
4. HTML.com - Recursos e Tutoriais  
*HTML.com*  
<https://html.com/>
5. W3C - HTML Specification  
*W3C HTML Specification*  
A especificação oficial do HTML pelo World Wide Web Consortium (W3C).  
<https://html.spec.whatwg.org/multipage/>
6. CSS-Tricks - Guia de Referência HTML  
*CSS-Tricks HTML Reference*  
Guia de referência para HTML, incluindo dicas e truques úteis.
7. HTML5 Rocks - Tutoriais e Exemplos  
*HTML5 Rocks*  
Recursos e tutoriais focados em HTML5 e suas novas funcionalidades.
8. Can I Use - Suporte a HTML e CSS  
*Can I Use*  
Verifique a compatibilidade de recursos HTML e CSS em diferentes navegadores.





## QUESTÕES COMENTADAS

### Questões Cespe

1. (CESPE – TCE RJ– 2022). Quanto ao desenvolvimento de sistemas web, julgue o item seguinte.

HTML5 é uma linguagem de programação que permite estruturar páginas web e executar comandos como loops de repetição, por exemplo.

#### Comentários:

HTML é a linguagem de marcação padrão para criar páginas da Web, ela descreve a estrutura de uma página da Web, porém, não executa comandos como loops de repetição.

Gabarito: Errado

2. (CESPE –DP DF– 2022) Julgue o item seguinte, a respeito da formatação de dados.

A tag `<meta charset="UTF-8">` define o conjunto de caracteres usados na página, nesse caso, o UTF-8, que é o padrão para HTML5.

#### Comentários:

O atributo `charset` especifica a codificação de caracteres para o documento HTML. A especificação HTML5 incentiva os desenvolvedores da Web a usar o conjunto de caracteres UTF-8, que abrange quase todos os caracteres e símbolos do mundo!

Gabarito: Correto

3. (CESPE – PGE RJ – 2022) Julgue o item que se segue, relacionado a desenvolvimento de sistemas.

No HTML 5, `sessionStorage` pode ser utilizado para armazenamento local de dados, tendo como característica o armazenamento de dados restritos à aba em funcionamento.

#### Comentários:

É possível usar dois objetos para armazenar dados no cliente: `window.localStorage` e `sessionStorage`. O `localStorage` - armazena dados sem data de validade; os dados não serão excluídos quando o navegador for fechado e estarão disponíveis no próximo dia, semana ou ano. Por outro lado, o `window.sessionStorage` - armazena dados para uma sessão. Os dados são excluídos quando o usuário fecha a guia específica do navegador. A questão erra ao afirmar que





sessionStorage pode ser utilizado para armazenamento local de dados quando na verdade é o window.localStorage que o faz.

Gabarito: Errado

---

4. (CESPE – APEX – 2021) Em HTML5, considerando-se o contexto de geolocalização e acesso ao dispositivo, para se obter retorno mais rápido e de baixa precisão sobre a localização de um dispositivo, deve-se
- a) invocar um método para o objeto PositionOptions da API com o parâmetro "0".
  - b) configurar a função watchPosition() da API de geolocalização.
  - c) testar a existência do objeto navigator.geolocation no dispositivo.
  - d) chamar o método getCurrentPosition() da API de geolocalização.

Comentários:

Pessoal, uma das APIs que vimos em aula foi a API de geolocalização HTML. Nela vimos o método getCurrentPosition() que é usado para retornar a posição do usuário. Portanto nosso gabarito é a letra D, para se obter retorno mais rápido e de baixa precisão sobre a localização de um dispositivo, deve-se chamar o método getCurrentPosition() da API de geolocalização.

Gabarito: Letra D

---

5. (CESPE – PGDF – 2021) Acerca de linguagens de marcação utilizadas para formatação de dados, julgue o item a seguir.

HTML e XML são equivalentes, pois ambas possuem uma semântica de apresentação predefinida.

Comentários:

HTML e XML são equivalentes aqui já encerramos o raciocínio! HTML é a linguagem de marcação padrão para páginas da Web. Já, XML significa eXtensible Markup Language e foi projetado para armazenar e transportar dados.

Gabarito: Errado

---

6. (CESPE – SEFAZ CE – 2021) Com relação à arquitetura de desenvolvimento de software, julgue o item a seguir.

Um link de navegação compartilhado por diversas páginas é incluído no elemento <main> de uma página HTML5.

Comentários:



Pessoal, errada questão. na verdade, o Elemento HTML de Navegação <nav> representa uma seção de uma página que aponta para outras páginas ou para outras áreas da página, ou seja, uma seção com links de navegação. Portanto, o correto seria: Um link de navegação compartilhado por diversas páginas é incluído no elemento <nav>.

Gabarito: Errado

7. (CESPE – PGDF – 2021) Julgue o item a seguir, referente a linguagens de scripts.

Considere o código seguinte, em HTML e JavaScript.

```
<html>
<input type='text' id='a' value='5'>
<br>
<input type='text' id='b' value='2'>
<br>
<input type='text' id='c' >
<script>
document.getElementById('c').value=(document.
getElementById('a').value+document.getElement
ById('b').value);
</script>
</html>
```

Em um navegador Internet com JavaScript habilitado, esse código apresentará o resultado a seguir

5
2
52

Comentários:

Pessoal, perfeita questão. O script realiza a concatenação dos valores id a e b e insere em c.



```
<html>
<input type='text' id='a' value='5'>
<br>
<input type='text' id='b' value='2'>
<br>
<input type='text' id='c' >
<script>
document.getElementById('c').value=
(document.getElementById('a').value+document.getE
lementById('b').value);
</script>
</html>
```

5
2
52

Gabarito: Correto

8. (CESPE – TJ PA – 2020) A respeito do tratamento off-line de um sítio no HTML 5, assinale a opção correta.

- a) Esse tratamento pode ser usado para a criação de dados em momento anterior ao acesso à aplicação.
- b) Na sessão cache do arquivo manifesto, devem estar relacionados todos os arquivos que o navegador deve copiar para que estejam disponíveis para uso off-line.
- c) É necessário que os arquivos PHP estejam listados na sessão cache.
- d) Na sessão network do arquivo manifesto, devem estar relacionados os arquivos que precisam ser substituídos por outros no retorno da conexão.
- e) Na sessão fallback do arquivo manifesto, devem estar relacionados os arquivos que não são utilizados para o processamento off-line.

Comentários:

Para tratar um site offline com HTML5, você pode usar o recurso de armazenamento de cache do HTML5, que permite que você armazene arquivos em cache no navegador do usuário. Isso significa que, quando o usuário visita o site pela primeira vez, os arquivos são armazenados em cache e, em visitas posteriores, o site é carregado a partir do cache, em vez de ser baixado da internet. Para habilitar o armazenamento de cache, você precisa criar um arquivo chamado "manifesto de cache" e vinculá-lo ao seu HTML usando a tag <html>. O manifesto de cache é um arquivo de texto simples que contém uma lista de arquivos que devem ser armazenados em cache.

Gabarito: Letra B

9. (CESPE – MPE CE – 2020) Acerca de JSON e HTML 5, julgue o item subsequente.

No HTML 5, localStorage é um recurso de armazenamento local que usa objetos JavaScript e que permite manter dados sem data de expiração prévia.



### Comentários:

HTML5 localStorage é um recurso de armazenamento na web que permite que um site armazene dados no navegador da web de um usuário. É semelhante a um cookie, mas é armazenado em uma área separada do computador do usuário e tem uma capacidade de armazenamento muito maior (até 5 MB). localStorage é útil para armazenar dados que precisam ser acessados no lado do cliente, como preferências do usuário ou dados de formulário. Os dados armazenados no localStorage são persistentes, ou seja, não são excluídos quando o usuário fecha o navegador ou desliga o computador. Para usar localStorage, você pode definir e recuperar valores usando JavaScript.

Gabarito: Correto

---

10. (CESPE – ProTI ME – 2020) Acerca de desenvolvimento de sistemas web, julgue o item a seguir:

No HTML 5, os novos campos para formulários, como email, search e range, e os atributos, como placeholder, pattern e required, reduzem a necessidade de utilização de plugins para auxiliar a formatação dos elementos.

### Comentários:

Pessoal, o HTML5 modifica a forma de como escrevemos código e organizamos a informação na página. Seria mais semântica com menos código; mais interatividade sem a necessidade de instalação de plugins e perda de performance. É a criação de código interoperável, pronto para futuros dispositivos e que facilita a reutilização da informação de diversas formas. Assim, de fato a questão está correta porque os novos campos para formulários, como email, search e range, assim como os atributos, como placeholder, pattern e required, reduzem a necessidade de utilização de plugins para auxiliar a formatação dos elementos.

Gabarito: Correto

---

11. (CESPE – ProTI ME – 2020) Acerca de desenvolvimento de sistemas web, julgue o item a seguir:

No HTML 5, as tags de link e script usadas para referenciar arquivos de CSS e JavaScript não precisam informar o atributo type, porque, na sua ausência, o navegador assume que o arquivo é do tipo text/css ou text/javascript.

### Comentários:



De fato, no HTML 5, nas tags link e script, utilizadas para referenciar arquivos CSS e JavaScript, respectivamente, não é mais necessário informar o atributo type="text/css" ou text="text/javascript", como era feito na HTML 4.1. Para exemplificar, no HTML 4.1 era assim:

```
<link rel="stylesheet" type="text/css" href="arquivo.css">
<script type="text/javascript" src="arquivo.js"></script>
```

Já, no HTML 5 é assim:

```
<link rel="stylesheet" href="arquivo.css">
<script src="arquivo.js"></script>
```

Gabarito: Correto

12. (CESPE – ProTI ME – 2020) Acerca de desenvolvimento de sistemas web, julgue o item a seguir.

A instrução DOCTYPE do HTML 5 é mais simples que a das versões anteriores HTML 4 ou XHTML 1.

Comentários:

Pessoal, como vimos, é bem mais simples! Vamos lembrar como é a instrução DOCTYPE do HTML 5:

```
<!DOCTYPE html>
```

Gabarito: Correto

13. (CESPE – TJ PA – 2020)

```
<!DOCTYPE html>
<html>
<p id="prova"></p>
<script>
meuObjeto = { "nome":"Caca", "Idade":52,
"carro":null };
for (x in meuObjeto) {
document.getElementById("prova").innerHTML += x;
}
</script>
</body>
</html>
```

O código html precedente, ao ser executado em um navegador de Internet, produzirá o seguinte resultado

- a) Caca52null
- b) nomeldadecarro
- c) [object Object][object Object][object Object]



- d) Caca+52+null
- e) nome+Idade+carro.

#### Comentários:

Ao carregar a página, o script percorrerá o objeto meuObjeto e adicionará o nome de cada propriedade ao elemento HTML com o ID "prova". No final, o conteúdo do elemento de prova será "nomeldadecarro". Se você quiser incluir o valor das propriedades também, deverá alterar a linha: `document.getElementById("prova").innerHTML += x;` para: `document.getElementById("prova").innerHTML += x + ": " + meuObjeto[x] + "<br>;` Isso incluirá o nome da propriedade e o valor dela na saída. Por exemplo: nome: Caca. Idade: 52. carro: null. Vejamos o resultado da execução do código.

<pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;p id="prova"&gt;&lt;/p&gt; &lt;script&gt; meuObjeto = { "nome":"Caca", "Idade":52, "carro":null }; for (x in meuObjeto) {   document.getElementById("prova").innerHTML += x; } &lt;/script&gt; &lt;/body&gt; &lt;/html&gt;</pre>	nomeldadecarro
--	----------------

Gabarito: Letra B

14. (CESPE – TJ PA – 2020) Na linguagem HTML 5, geralmente considera-se determinado elemento como o ponto central do conteúdo do documento, o qual pode ser, por exemplo, um post.

Esse elemento, que representa um conteúdo independente e altamente relevante, é o

- a) aside.
- b) canvas.
- c) embed.
- d) article.
- e) figure.

#### Comentários:



Alguns dos novos recursos do HTML5 incluem: Elementos semânticos: o HTML5 apresenta novos elementos com significados específicos, como <header>, <footer> e <article>. Esses elementos tornam mais fácil para os desenvolvedores estruturar seu conteúdo de maneira lógica e significativa. aside: Trata-se de conteúdo relacionado à seção próxima. Análogo às barras laterais em conteúdo impresso. O elemento HTML <canvas> é usado para desenhar gráficos em uma página da web. A tag <embed> define um contêiner para um recurso externo, como uma página da Web, uma imagem, um reprodutor de mídia ou um aplicativo de plug-in. A tag <article> define um conteúdo independente de outras partes do site e altamente relevante. É autocontido. Exemplo: comentário enviado por usuário, post de um blog, artigo de uma revista. A tag <figure> especifica o conteúdo independente, como ilustrações, diagramas, fotos, listagens de código, etc. Portanto, nosso gabarito é a tag < article >

Gabarito: Letra D

15. (CESPE –SLU DF – 2019) Com relação a desenvolvimento de software, julgue o item a seguir.

De acordo com o trecho de código a seguir, escrito em HTML5, novos valores de date e time são válidos como atributos de elementos de formulário, e apenas o campo data é de preenchimento obrigatório.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>HTML5</title>
</head>
<body>
  <form method="post" action="teste.html">
    <label for="data">Data: </label>
    <input id="data" type="date" required name="data"/>
    <br />
    <label for="hora">Hora: </label>
    <input id="hora" type="time" name="hora"/>

    <input type="submit" value="Enviar"/>
  </form>
</body>
</html>
```

### Comentários:

Pessoal, perfeita questão! O atributo data possui um "required" que o torna obrigatório. O código cria um formulário HTML simples com dois campos de entrada: um para data e outro para hora. O atributo type é usado para especificar o tipo de entrada de dados que o campo deve aceitar. No caso do campo de data, o tipo é date, o que significa que





o usuário só poderá inserir uma data válida nesse campo. O campo de hora tem o tipo `time`, o que significa que o usuário só poderá inserir uma hora válida nesse campo. O atributo `required` é usado para indicar que o campo de data é obrigatório e o usuário deve inserir um valor antes de enviar o formulário. O atributo `name` é usado para dar um nome ao campo de entrada, que pode ser usado posteriormente para acessar o valor inserido pelo usuário. O botão de envio é criado usando o elemento `<input type="submit">`. Quando o usuário clica no botão, o formulário é enviado para o arquivo `teste.html`, que é especificado no atributo `action` do elemento `<form>`.

Gabarito: Correto

16. (CESPE – MPC TCE-PA – 2019) O HTML (hypertext markup language) tem amplo uso difundido nas páginas publicadas na Internet. Assinale a opção que corresponde à tag utilizada no caso em que seja necessário utilizar uma lista não ordenada.

- a) `<b>`
- b) `<p>`
- c) `<tr>`
- d) `<ul>`
- e) `<th>`

Comentários:

Como já vimos, uma lista não ordenada começa com a tag `<ul>` e cada item da lista começa com a tag `<li>`. Os itens da lista serão marcados com marcadores (pequenos círculos pretos) por padrão. a) `<b>` é um elemento HTML obsoleto que define o texto em negrito. Recomenda-se usar a propriedade CSS "font-weight" para estilizar o texto como negrito. b) `<p>` é um elemento HTML que define um parágrafo de texto. c) `<tr>` é um elemento HTML que define uma linha em uma tabela HTML. d) `<ul>` é um elemento HTML que define uma lista não ordenada (uma lista com marcadores). e) `<th>` é um elemento HTML que define uma célula como um cabeçalho de tabela em uma tabela HTML. Normalmente é usado para exibir o cabeçalho de um grupo de células da tabela.

Gabarito: Letra D

17. (CESPE – TJ AM – 2019) Acerca do desenvolvimento web mediante o uso do HTML 5, do JavaScript, do XML e do CSS, julgue o item subsequente.

O HTML 5 define como os navegadores web devem lidar com marcações antigas como `<font>`, `<center>` e outras tags de apresentação.

Comentários:





Pessoal, apesar dessas tags serem obsoletas, o HTML5 ainda reconhece o funcionamento dessas tags. E o HTML 5 define como os navegadores web devem lidar com elas.

Gabarito: Correto

18. (CESPE – TJ AM – 2019) Acerca do desenvolvimento web mediante o uso do HTML 5, do JavaScript, do XML e do CSS, julgue o item subsequente.

O HTML 5 especifica novas API (application program interface) para o modelo de objeto de documento (DOM — document object model) referente a arrastar e soltar eventos enviados pelo servidor, como desenhos, vídeos e similares.

Comentários:

Pessoal, o Document Object Model (DOM) é uma interface de programação para documentos HTML e XML. Ele representa a estrutura de um documento como uma árvore de objetos, com cada objeto representando um elemento do documento. No HTML5, o DOM é usado para manipular o conteúdo e a estrutura de um documento HTML. Por exemplo, você pode usar o DOM para adicionar, excluir ou modificar elementos em um documento HTML ou para alterar os atributos de um elemento. Assim, é possível arrastar e soltar eventos enviados pelo servidor, como desenhos, vídeos e similares.

▪

Gabarito: Correto

19. (CESPE – TJ AM – 2019) Acerca do desenvolvimento web mediante o uso do HTML 5, do JavaScript, do XML e do CSS, julgue o item subsequente.

A XHTML 5 é uma serialização XML que tem as mesmas características e sintaxes do HTML 5.

Comentários:

Na verdade, XHTML (Extensible HyperText Markup Language) é uma linguagem de marcação que é uma variante do HTML (HyperText Markup Language). É uma versão mais rígida e limpa do HTML, com uma sintaxe mais rígida e com foco na conformidade com os padrões. XHTML foi projetado para ser uma ponte entre HTML e XML, e é frequentemente usado em combinação com outras tecnologias, como CSS (Cascading Style Sheets) e JavaScript, para criar páginas da Web dinâmicas e interativas. É suportado por todos os navegadores da web modernos e é comumente usado no desenvolvimento web. Existem várias diferenças entre HTML e XHTML como sintaxe, extensibilidade, compatibilidade, tipo de documento. Portanto, não tem as mesmas características e sintaxes do HTML 5.



20. (CESPE – MPE PI – 2018) Julgue o próximo item, relativo a lógica de programação e linguagens de programação.

```
<html>
<body>
<h2>JavaScript </h2>
<p id="demo"></p>
<script>
var colors = ["Blue", "Red", "White"];
document.getElementById("demo").innerHTML = colors;
</script>
</body>
</html>
```

A execução do código JavaScript anteriormente apresentado retornará o seguinte resultado:

```
JavaScript
Blue, Red, White
```

#### Comentários:

Pessoal, está perfeita a questão! Apresentará o título que consiste no código `<h2>JavaScript </h2>` além do valor que está em `colors` no seguinte trecho: `var colors = ["Blue", "Red", "White"];`  
`document.getElementById("demo").innerHTML = colors;`

21. (CESPE – TJ STJ – 2018) Julgue o item seguinte, a respeito de Maven, desenvolvimento web, servidor web, servidor de aplicação e criptografia.

No HTML5, o atributo `autofocus` possibilita que qualquer elemento `<input>` seja automaticamente focado quando do carregamento da página.

#### Comentários:

O atributo `autofocus` é um atributo HTML5 que pode ser usado para especificar que um elemento de formulário deve ser focado automaticamente quando a página é carregada. Isso é útil para garantir que a atenção do usuário seja imediatamente atraída para um determinado campo de formulário, como uma caixa de pesquisa ou um formulário de



login. O erro da questão está na grafia da palavra, o que é no mínimo uma maldade do examinador,

**Gabarito:** Errado

---

22. (CESPE – STM – 2018) Julgue o item subsequente, a respeito de programação web.

Em HTML5, a tag <output> fornece uma indicação ao usuário do que pode ser inserido no campo.

**Comentários:**

O elemento <output> é um elemento HTML5 que representa o resultado de um cálculo ou ação do usuário. Ele pode ser usado para exibir o resultado de um cálculo de formulário ou para exibir conteúdo dinâmico atualizado com base na entrada do usuário. A questão refere-se a tag <input>. O elemento <input> é um elemento HTML usado para criar controles de formulário para entrada do usuário. Ele pode ser usado para criar uma variedade de controles de formulário, incluindo campos de texto, caixas de seleção, botões de opção e muito mais.

**Gabarito:** Errado

---

23. (CESPE – TRE BA – 2017) Entre os novos elementos do HTML5, o elemento:

- a) <mark> é o ponto de parada do cursor em qualquer parte da página HTML.
- b) <progress> define o progresso de uma tarefa.
- c) <meter> interpreta medições meteorológicas.
- d) <figcaption> captura figuras.
- e) <main> define a estrutura principal da linguagem C dentro da página HTML

**Comentários:**

a) <mark> Define o texto marcado/destacado; b) <progress> Representa o progresso de uma tarefa; c) <meter> Define uma medição escalar dentro de um intervalo conhecido (um medidor); d) <figcaption> Define uma legenda para um elemento <figure>; e) <main> Especifica o conteúdo principal de um documento. Portanto, nosso gabarito é a letra B.

**Gabarito:** Letra B

---

24. (CESPE – TCE-PA – 2016) Acerca dos conceitos e das técnicas necessários à construção de sítios web em que se utilizam CSS e HTML, julgue o item que se segue.

HTML é uma linguagem de programação utilizada na construção de páginas na Web.



**Comentários:**

HTML significa Hyper Text Markup Language, é uma linguagem de marcação padrão para criar páginas da Web. Ele é usado para criar páginas da Web e dar estrutura e significado ao conteúdo da Web. O erro está em dizer: HTML é uma linguagem de programação, quando na verdade é uma linguagem de marcação.

**Gabarito:** Errado

---

25. (CESPE –TCE-PA – 2016) Julgue o item que se segue, relativamente a desenvolvimento de sistemas web.

Após a incorporação do jQuery ao HTML5, o desenvolvimento de funcionalidades por meio dessa biblioteca JavaScript ficou limitado a aplicações para dispositivos móveis.

**Comentários:**

jQuery é uma biblioteca JavaScript rápida, pequena e rica em recursos que facilita a manipulação do DOM (Document Object Model), manipula eventos e anima elementos em uma página da web. Foi criado em 2006 e ainda hoje é amplamente utilizado. As principais funcionalidades do jQuery são: Resolução da incompatibilidade entre os navegadores; Redução de código; Reutilização do código através de plugins; Utilização de uma vasta quantidade de plugins criados por outros desenvolvedores; Trabalha com AJAX e DOM; Implementação segura de recursos do CSS1, CSS2 e CSS3. O HTML5 é a versão mais recente do HTML e apresenta vários novos elementos e recursos que facilitam a criação de conteúdo da Web interativo e atraente. Embora o jQuery não esteja especificamente relacionado ao HTML5, ele pode ser usado em conjunto com o HTML5 para criar aplicativos da Web interativos. Por exemplo, você pode usar jQuery para selecionar e manipular elementos HTML5, como os novos elementos semânticos introduzidos no HTML5 (por exemplo, <header>, <footer> e <article>). A questão limita a funcionalidade do jQuery além de citar a incorporação do jQuery ao HTML5, o que está errado.

**Gabarito:** Errado

---

26. (CESPE –TCE-PA – 2016) Julgue o item que se segue, relativamente a desenvolvimento de sistemas web.

O elemento <canvas> do HTML5 especifica uma forma padrão para inserir um vídeo em uma página da Web.

**Comentários:**

Na verdade, o elemento HTML `<canvas>` é usado para desenhar gráficos em uma página da web.

Gabarito: Errado

---

27. (CESPE – FUNPRES-EXE – 2016) Acerca da tecnologia Java, julgue o próximo item.

Em HTML5, o atributo `title` da tag `<img>` pode ser usado para se adicionar um texto fixo a ser sempre apresentado imediatamente acima de uma imagem.

Comentários:

A tag `<img>` é vazia, ou seja, contém apenas atributos e não possui uma tag de fechamento. Além disso, ela possui dois atributos obrigatórios: `src` - Especifica o caminho para a imagem e `alt` - Especifica um texto alternativo para a imagem. Portanto, o atributo que pode ser usado para adicionar um texto fixo é o `alt`.

Gabarito: Errado

---

28. (CESPE – FUNPRES-JUD – 2016) A respeito das tecnologias relacionadas ao desenvolvimento web em Java, julgue o item a seguir.

No HTML 5, a tag `<rp>` é usada para informar o que deve ser exibido nos browsers que não suportam anotações ruby.

Comentários:

Na referência do HTML 5 temos que `<rp>` define o que mostrar em navegadores que não suportam anotações ruby.

Gabarito: Correto

---

29. (CESPE – TRT 8 – 2016) Assinale a opção que apresenta a tag incluída na especificação do HTML5 que permite a reprodução de arquivos que contenham som ou música.

- a) `<phonic>`
- b) `<img>`
- c) `<sound>`
- d) `<music>`
- e) `<audio>`.

Comentários:



O elemento <audio> é um elemento HTML5 que permite incorporar arquivos de áudio em uma página da web. Ele fornece uma maneira padrão de reproduzir arquivos de áudio nativamente no navegador, sem a necessidade de plug-ins de terceiros, como o Flash.

Gabarito: Letra E

30. (CESPE – TRE PI – 2016) A respeito de páginas web desenvolvidas utilizando-se HTML 5, assinale a opção correta.

- a) Para a visualização de vídeos incluídos na página web, é necessária a presença de plug-ins adequados aos formatos de mídia utilizados.
- b) A indefinição dos parâmetros altura e largura dos vídeos pode gerar problemas de renderização.
- c) Para se adicionar vídeos, o uso do atributo preload exige a presença do atributo controls.
- d) O elemento <iframe> permite a inclusão de outra página web na página que esteja sendo construída.
- e) O elemento <header> é usado exclusivamente no início de uma página para determinar o seu cabeçalho.

#### Comentários:

Nosso gabarito é a letra D. O elemento <iframe> é um elemento HTML usado para incorporar um documento dentro de outro documento. Ele fornece uma maneira de exibir o conteúdo de outra página da Web na página da Web atual e é comumente usado para incorporar vídeos, mapas e outros conteúdos interativos. Ademais, vejamos os erros das demais alternativas. A alternativa A erra ao dizer que é necessária a presença de plug-ins adequados aos formatos de mídia utilizados. Na verdade, é possível usar a tag <source>, por exemplo, que define vários recursos de mídia para elementos de mídia (<video>). A letra B também está incorreta, porque a indefinição dos parâmetros altura e largura dos vídeos não causará problemas porque os atributos de altura e largura são opcionais e, se você não os especificar, o elemento <video> ajustará automaticamente seu tamanho para caber no tamanho do arquivo de vídeo. A letra C também está errada. Ambos os atributos **preload** e **controls** são **opcionais** e você pode usá-los para personalizar o comportamento do elemento de mídia de acordo com suas necessidades, além disso, preload não exige a presença do atributo controls. Por fim, o elemento <header> normalmente é usado no topo da página da web, mas também pode ser usado para representar o cabeçalho de uma seção de uma página da web. Neste caso, deve ser usado dentro de um elemento <section>. Portanto a questão erra ao dizer que ele é usado exclusivamente no início de uma página.

Gabarito: Letra D



31. (CESPE – TCE-PA – 2016) Julgue o item seguinte no que se refere à construção de formulários eletrônicos.

O elemento `label` funciona como um indicador de caminho a seguir. Muitos browsers renderizam o conteúdo daquela tag como uma área clicável a fim de levar o foco para o campo relacionado.

#### Comentários:

O elemento `<label>` é um elemento HTML usado para rotular um controle de formulário. Ele fornece uma maneira de associar um rótulo de texto a um controle de formulário, tornando mais fácil para os usuários entenderem a finalidade do controle. O elemento `<label>` tem vários benefícios como ajudar a melhorar a acessibilidade de seus formulários, pois o software leitor de tela pode usar o texto do rótulo para descrever a finalidade do controle de formulário para usuários com deficiências. Tornar mais fácil para os usuários entender a finalidade do controle de formulário, especialmente quando o controle de formulário tem uma finalidade não óbvia (por exemplo, uma caixa de seleção usada para aceitar um boletim informativo). Ajudar a melhorar a usabilidade de seus formulários, pois os usuários podem clicar no texto do rótulo para focar o controle do formulário. O elemento `<label>` é um elemento importante para a criação de formulários amigáveis e acessíveis, e é amplamente utilizado no desenvolvimento web moderno. A utilização do `<label>` é mais comum quando trabalhamos com formulários. Para cada campo é inserido um `label` para associar visualmente com o campo a ser preenchido. Por isso, está correta a questão ao dizer que o elemento `label` funciona como um indicador de caminho a seguir.

Gabarito: Correto

32. (CESPE – TCE-PA – 2016) Julgue o item seguinte no que se refere à construção de formulários eletrônicos.

É vedada a utilização de `FIELDSET` para agrupar qualquer variedade de elementos `input` de formulários.

#### Comentários:

O elemento `<fieldset>` é um elemento HTML usado para agrupar controles de formulário relacionados. Ele fornece uma maneira de organizar visual e semanticamente os controles de formulário, tornando mais fácil para os usuários entender e usar o formulário. Portanto, como a questão está dizendo o oposto, nosso gabarito é: Errado.

Gabarito: Errado





33. (CESPE – TCE-PA – 2016) Julgue o item seguinte no que se refere à construção de formulários eletrônicos.

A tag <label> pode ser aplicada a todos os elementos de formulário, até mesmo a elementos button.

#### Comentários:

Pessoal, novamente cobrando a tag <label>. O erro da questão está em dizer que a tag pode ser aplicada a todos os elementos de formulário, até mesmo a elementos button.

Gabarito: Errado

34. (CESPE – TCE-PA – 2016) Julgue o item seguinte no que se refere à construção de formulários eletrônicos.

É possível agrupar inputs de um formulário e, ainda, as opções de uma tag <select> usando-se a tag <fieldset>.

#### Comentários:

Pessoal, é super possível! Inclusive uma das formas é utilizando a tag <fieldset>.

```
<form>
  <fieldset>
    <legend>Personal Information</legend>
    <label for="name">Name:</label>
    <input type="text" id="name" name="name"><br>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email"><br>
  </fieldset>
  <fieldset>
    <legend>Shipping Information</legend>
    <label for="address">Address:</label>
    <input type="text" id="address" name="address"><br>
    <label for="city">City:</label>
    <input type="text" id="city" name="city"><br>
  </fieldset>
</form>
```

Gabarito: Correto





## Questões FCC

35. (FCC – PGE AM – 2022) No cabeçalho de uma página HTML deseja-se indicar ao navegador o conjunto de caracteres recomendado pela linguagem HTML5, que abrange a maioria dos caracteres e símbolos utilizados na maior parte dos idiomas, inclusive acentos existentes no Português. Para isso deve-se utilizar a instrução
- a) `<meta charset="UTF-8">`
  - b) `<meta charset="ISO-8959-1">`
  - c) `<meta charset="Windows-8859">`
  - d) `<meta charset="ASC-II">`
  - e) `<meta charset="PT-BR">`

### Comentários:

Pessoal, nosso gabarito é a letra A: `<meta charset="UTF-8">`. O elemento `<meta>` é um elemento HTML usado para fornecer metadados sobre uma página da web. O atributo `charset` é usado para especificar a codificação de caracteres da página da web e é usado para garantir que os caracteres na página da web sejam exibidos corretamente no navegador. O atributo `charset` é particularmente importante quando a página da Web contém caracteres de vários idiomas ou scripts, pois diferentes codificações de caracteres podem ser necessárias para exibir os caracteres corretamente.

**Gabarito:** Letra A

36. (FCC – TRT 23 – 2022) Para criar sites responsivos usando HTML5 é aconselhável fornecer ao navegador instruções sobre como controlar as dimensões e a escala da página por meio da definição da viewport da página, utilizando o comando
- a) `<viewport initial-scale="1.0" max-width="100%" />`
  - b) `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
  - c) `<viewport content="width=device-width, initial-scale=1.0" />`
  - d) `<meta type="viewport" initial-scale="100%" max-width="100%" />`
  - e) `<meta type="viewport" screen="width=100%, initial-scale=1.0">`

### Comentários:

Um site responsivo é um site projetado para ajustar automaticamente seu layout e conteúdo para se adequar ao tamanho do dispositivo e da tela do usuário. Sites responsivos usam uma combinação de grades flexíveis, imagens e consultas de mídia para criar uma experiência de usuário perfeita e intuitiva em uma ampla variedade de dispositivos, desde computadores desktop até smartphones e tablets.

Aqui estão alguns dos principais recursos dos sites responsivos: Layouts fluidos: sites responsivos usam grades flexíveis que se ajustam ao tamanho da tela, para que o



conteúdo seja exibido em um layout de fácil leitura, independentemente do dispositivo usado. Imagens responsivas: sites responsivos usam imagens que ajustam automaticamente seu tamanho para caber na largura da tela, para que as imagens tenham uma boa aparência em todos os dispositivos. Consultas de mídia: sites responsivos usam consultas de mídia para detectar o tamanho da tela e aplicar estilos diferentes com base no tamanho da tela. Isso permite que o site ajuste seu layout e conteúdo com base no dispositivo que está sendo usado. Um exemplo de um site responsivo simples:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <style>
      /* Use a fluid grid to make the website responsive */
      .container {
        display: grid;
        grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
        grid-gap: 20px;
      }

      /* Use media queries to adjust the layout based on the screen size */
      @media (max-width: 600px) {
        .container {
          grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
        }
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div>Content goes here</div>
    </div>
  </body>
</html>
```

Podemos observar que o exemplo apresenta a tag `<meta name="viewport" content="width=device-width, initial-scale=1.0">`. Assim, nosso gabarito é a letra B.

**Gabarito:** Letra B

37. (FCC – TRT22 – 2022) Uma Analista deseja escrever no rodapé da página web HTML5 “Copyright ©”, sem aspas, indicando que a página possui direitos autorais.

Uma das maneiras corretas de fazer isso é por meio da instrução

- a) `<p>Copyright &copy;</p>`
- b) `<p>Copyright &szlig;</p>`
- c) `<p>Copyright &circledR;</p>`



- d) `<p>Copyright &copyright;</p>`  
e) `<p>Copyright &cpsymbol;</p>`  
e) `<p>Copyright &cpsymbol;</p>`

#### Comentários:

O sinal © é criado pela instrução `&copy` que significa sinal de direitos autorais.

Gabarito: Letra A

38. (FCC – TJ SC – 2021) Em uma situação hipotética, um profissional de TI deseja inserir um vídeo institucional do Tribunal de Justiça de Santa Catarina que está no Youtube, no site do Tribunal.

Para isso, obteve o código personalizado abaixo no site do Youtube.

```
<...I... width="560" height="315" src="https://www.youtube.com/embed/-u33KrrhlpU"
frameborder="0" allow=" accelerometer; autoplay; encrypted-media; gyroscope; picture-in-
picture" allowfullscreen></...I... >
```

Tal fragmento de código permite concluir que o comando HTML5 que deve ser utilizado na lacuna I é

- a) video
- b) movie
- c) media
- d) frame
- e) iframe

#### Comentários:

The `<iframe>` element is an HTML element that is used to embed a document within another document. It provides a way to display content from another webpage within the current webpage, and is commonly used to embed videos, maps, and other interactive content.

Gabarito: Letra E

39. (FCC – TJ MA – 2019) Um Analista que está desenvolvendo a página de abertura de um site deseja fazer com que as páginas HTML referentes aos links do menu sejam direcionadas e exibidas em um contêiner criado pela tag `section` no interior desta página. Nesse contêiner, para permitir a abertura das páginas, deve-se utilizar uma tag:

- a) aside
- b) main
- c) dialog



- d) iframe
- e) article

#### Comentários:

Vejamos a definição de cada tag. a) aside: O elemento <aside> é um elemento HTML utilizado para representar o conteúdo relacionado ao conteúdo principal da página da Web, mas que não é essencial para o entendimento do conteúdo principal. Geralmente é usado para representar barras laterais, citações e outros conteúdos relacionados tangencialmente. b) main: O elemento <main> é um elemento HTML usado para representar o conteúdo principal de uma página da web. Ele deve ser usado para agrupar o conteúdo exclusivo da página da Web e que não é compartilhado com outras páginas no mesmo site. c) dialog: O elemento <dialog> é um elemento HTML usado para representar uma caixa de diálogo ou janela. Ele pode ser usado para criar caixas de diálogo modais que exigem que o usuário interaja com elas antes de retornar à página da Web principal. d) iframe: O elemento <iframe> é um elemento HTML usado para incorporar um documento dentro de outro documento. Ele fornece uma maneira de exibir o conteúdo de outra página da Web na página da Web atual e é comumente usado para incorporar vídeos, mapas e outros conteúdos interativos. e) article: A tag <article> define um conteúdo independente de outras partes do site e altamente relevante. É autocontido. Exemplo: comentário enviado por usuário, post de um blog, artigo de uma revista. Pessoal, a única tag que permite a abertura de outra página (documento) é a tag <iframe>. Dessa forma, nosso gabarito é a letra D.

**Gabarito:** Letra D

40. (FCC – AFAP – 2019) Em uma página HTML 5 de abertura de um site, um Analista de Informática deseja definir uma área no centro onde outras páginas HTML poderão ser carregadas a partir de cliques nos links do menu principal. Ao abrir a página de abertura, um arquivo HTML já poderá ser exibido nessa área, cujo conteúdo poderá mudar na medida que se clica nos links do menu. Para que seja possível o comportamento descrito, essa área deverá ser definida por meio da tag

- a) div.
- b) section.
- c) iframe.
- d) main.
- e) core.

#### Comentários:

Um iframe (abreviação de "frame inline") é um elemento HTML usado para incorporar uma página da Web em outra página da Web. Ele fornece uma maneira de exibir o



conteúdo de outra página da Web na página da Web atual e é comumente usado para incorporar vídeos, mapas e outros conteúdos interativos. Portanto é o nosso gabarito.

Gabarito: Letra C

41. (FCC – Pref Manaus – 2019) Em um parágrafo de uma página web desenvolvida com HTML5, um programador está usando palavras longas e está com medo do navegador quebrar as palavras ou linhas em locais incorretos. Para indicar os locais desejados para possíveis quebras de linha, quando a largura da janela do navegador mudar, o programador deverá usar a tag

- a) `</br>`.
- b) `<break>`.
- c) `<\n>`.
- d) `<wbr>`.
- e) `<br/>`.

Comentários:

Para indicar os locais desejados para possíveis quebras de linha, quando a largura da janela do navegador mudar, o programador deverá usar a tag `<wbr>`. De acordo com a definição, `<wbr>` define uma possível quebra de linha.

Gabarito: Letra D

42. (FCC – TRF 4 – 2019) HTML5 incluiu um conjunto de tags semânticas para substituir algumas construções que eram feitas com HTML4. Por exemplo, as instruções `<div id="header">` `</div>` e `<div id="menu">` `</div>` feitas em HTML4, são feitas na HTML5, usando as tags semânticas, através das instruções:

- a) `<area id="header">` `</area>` e `<área id="menu">` `</área>`
- b) `<header>` `</header>` e `<nav>` `</nav>`
- c) `<head>` `</head>` e `<menu>` `</menu>`
- d) `<header>` `</header>` e `<navigation>` `</navigation>`
- e) `<header>` `</header>` e `<menuarea>` `</menuarea>`

Comentários:

Alguns dos novos recursos do HTML5 incluem elementos semânticos: o HTML5 apresenta novos elementos com significados específicos, como `<header>`: representa o cabeçalho de uma página da web ou seção de uma página da web. `<nav>`: representa uma seção da página da Web que contém links de navegação. Esses elementos tornam mais fácil para os desenvolvedores estruturar seu conteúdo de maneira lógica e significativa.



Gabarito: Letra B

43. (FCC – TJ MA – 2019) Um Técnico Judiciário que utiliza HTML5 está criando um parágrafo em uma página web que tem muitas palavras longas e, para evitar que o navegador quebre linha no lugar errado, utilizou uma tag para especificar onde, em um parágrafo, será aceitável uma quebra de linha. A tag correta para isso é:
- a) <break>
  - b) <wbr>
  - c) <bl>
  - d) <rt>
  - e) <br>

Comentários:

Pessoal, é exatamente o mesmo caso da questão da Pref Manaus de 2019! Deve ser utilizada a tag <wbr> que define uma possível quebra de linha.

Gabarito: Letra B

44. (FCC – TRF 3 – 2019) Considere o fragmento abaixo de uma página HTML5 que utiliza elementos semânticos.

```
<figure>
  
</figure>
```

Para inserir a legenda "Figura 1 – Foto das dependências do TRF3" na imagem, após  deve-se colocar a instrução

- a) <label>Figura 1 – Foto das dependências do TRF3</label>
- b) <title>Figura 1 – Foto das dependências do TRF3</title>
- c) <legend>Figura 1 – Foto das dependências do TRF3</legend>
- d) <figcaption>Figura 1 – Foto das dependências do TRF3</figcaption>
- e) <figcaption>Figura 1 – Foto das dependências do TRF3.</figcaption>

Comentários:

Pessoal, o elemento <figcaption> define uma legenda para um elemento <figure>.

Gabarito: Letra E

45. (FCC – Pref Manaus – 2019) Um programador precisa utilizar uma tag semântica incorporada a versão 5 da linguagem HTML, cujo objetivo é definir algum conteúdo



adicional colocado normalmente à direita, mas relacionado com um conteúdo que o circunda colocado normalmente à esquerda. Deverá escolher, nesse caso, a tag:

- a) <article>
- b) <section>
- c) <aside>
- d) <figcaption>
- e) <summary>

#### Comentários:

O examinador vai dando dicas: definir algum conteúdo adicional; colocado normalmente à direita; mas relacionado com um conteúdo que o circunda colocado normalmente à esquerda. Quando o examinador diz: " mas relacionado com um conteúdo que o circunda" já se infere a tag <aside> que é o nosso gabarito. Para complementar, vejamos as demais alternativas. A tag <article> define um conteúdo independente de outras partes do sítio e altamente relevante. É autocontido. Exemplo: comentário enviado por usuário, post de um blog, artigo de uma revista. O elemento <section> é um elemento HTML usado para representar uma seção da página da Web, como um capítulo ou um tema. É um elemento semântico, o que significa que é usado para adicionar significado à página da Web, em vez de controlar o layout ou a aparência da página. <figcaption> define uma legenda para um elemento <figure>. Por fim, o elemento <summary> é um elemento HTML usado para representar um resumo ou legenda para um elemento <details>. Normalmente é usado para fornecer uma breve visão geral ou resumo do conteúdo dentro do elemento <details>, que pode ser expandido ou reduzido pelo usuário.

**Gabarito:** Letra C

46. (FCC – TRF 3 – 2019) Um programador está criando o menu principal da página de abertura de um site, que conterà o bloco principal de links de navegação. Considerando os elementos semânticos da linguagem HTML5, estes links devem ficar imediatamente no interior da tag

- a) <mark>
- b) <progress>
- c) <choose>
- d) <nav>
- e) <navigation>

#### Comentários:





Se deseja criar o menu principal, vai utilizar a tag nav. Vejamos a definição: o elemento <nav> representa uma seção de uma página que contém links de navegação. É usado para agrupar links que levam a outras páginas ou a diferentes partes da mesma página.

Gabarito: Letra D

47. (FCC – SANASA – 2019) Um Desenvolvedor de software precisa inserir uma instrução no cabeçalho de uma página HTML que fará referência a um arquivo chamado a001.css a ser aplicado apenas quando a página for aberta em dispositivos com tela de até 600 pixels. A instrução correta que deverá ser inserida é

- a) <@import URL(a001.css) only screen and (max-width: 600px)>
- b) <link rel="media" device="only screen with (max-width: 600px)" href="a001.css">
- c) <link rel="stylesheet" media="screen and (max-width: 600px)" href="a001.css">
- d) <inport file="a001.css" media="screen and (max-width: 600px)">
- e) <style>@media only screen and (min-width: 600px) URL(a001.css) <style>.

Comentários:

O elemento <link> com o atributo "rel" definido como "stylesheet" informa ao navegador que o documento vinculado é uma folha de estilo e fornece os meios para aplicar os estilos definidos na folha de estilo a uma página da web. O atributo "media" especifica o tipo de mídia para o qual a folha de estilo se destina, como uma tela ou impressão. Neste caso, o atributo "media" é definido como "screen and (max-width: 600px)", o que significa que os estilos na folha de estilo serão aplicados apenas a telas com largura máxima de 600 pixels. O atributo "href" especifica a URL da folha de estilo. Nesse caso, o valor do atributo "href" é "a001.css", o que indica que a folha de estilo está localizada na URL "a001.css". Este elemento <link> seria colocado no cabeçalho de um documento HTML para aplicar os estilos da folha de estilo "a001.css" à página da Web em telas com largura máxima de 600 pixels.

Gabarito: Letra C

48. (FCC – SABESP – 2018) Um Técnico está desenvolvendo uma página web com HTML5 e deseja exibir um texto com fonte Courier de largura fixa, preservando os espaços e as quebras de linha.

O texto deverá ser colocado entre as tags

- a) <mark> e </mark>
- b) <dl> e </dl>
- c) <embed> e </embed>
- d) <code> e </code>
- e) <pre> e </pre>.





### Comentários:

Vamos relembrar o que foi falado durante a aula? O elemento HTML `<pre>` define o texto pré-formatado. O texto dentro de um elemento `<pre>` é exibido em uma fonte de largura fixa (geralmente Courier) e preserva espaços e quebras de linha.

Gabarito: Letra E

49. (FCC –DPE AM – 2018) Um Técnico Programador deseja fazer um glossário de termos em um site utilizando listas de definições do HTML5. Nestas listas, o termo a ser descrito e a descrição propriamente dita são criados, respectivamente, pelas tags

- a) `<dt>` e `<dd>`
- b) `<dd>` e `<tt>`
- c) `<ul>` e `<li>`
- d) `<tt>` e `<dd>`
- e) `<dd>` e `<dt>`

### Comentários:

O HTML também suporta listas de descrição. Uma lista de descrição é uma lista de termos, com uma descrição de cada termo. A tag `<dl>` define a lista de descrição, a tag `<dt>` define o termo (nome) e a tag `<dd>` descreve cada termo. `<dt>` define um termo em uma lista de descrição. `<dd>` descreve o termo em uma lista de descrição.

Gabarito: Letra A

50. (FCC – TST – 2017) Considere, hipoteticamente, que um Programador deseja publicar um vídeo institucional no site do Tribunal Superior do Trabalho. Para isso, no local da página HTML5 onde deseja colocar o vídeo, utilizou o bloco de código abaixo.

```
<video autoplay>  
  <source src="[nome_arquivo.extensão]" type="video/..I..">  
</video>
```

Considere que `[nome_arquivo.extensão]` representa o nome de um arquivo de vídeo válido. Nas últimas versões dos principais navegadores, os tipos de arquivos de vídeo válidos para preencher a lacuna I são

- a) avi, mpeg e mov
- b) mkv, ogg e flv
- c) mp4, webm e ogg
- d) wmv, rmvb e mp4
- e) mp4, avi e mov.



### Comentários:

O elemento <video> suporta vários formatos de arquivo de vídeo diferentes, incluindo: MP4, WebM e Ogg.

Gabarito: Letra C

51. (FCC – TRF 5 – 2017) Uma das recomendações que consta nas Web Content Accessibility Guidelines – WCAG é fornecer legendas e outras alternativas para conteúdo multimídia.

Nesse contexto, considere o código abaixo, que disponibiliza um vídeo de orientação ao cidadão em um site do Governo.

```
<video controls>  
  <source src="orienta_cidadao.mp4" type="video/mp4">  
  ...  
</video>
```

Para fornecer um arquivo de legenda em português chamado legenda\_br.vtt para o vídeo utiliza-se, na lacuna I, o comando

- a) <embed src="legenda\_br.vtt" kind="track" srclang="pt" label="Português">
- b) <subtitle src="legenda\_br.vtt" kind="text/media" srclang="pt" label="Português">
- c) <caption src="legenda\_br.vtt" kind="media-query" srclang="pt-br" caption="Português">
- d) <caption href="legenda\_br.vtt" kind="subtitles" srclang="pt" label="Português">
- e) <track src="legenda\_br.vtt" kind="subtitles" srclang="pt" label="Português">

### Comentários:

Em HTML5, o elemento <track> pode ser usado para especificar uma legenda ou arquivo de legenda para um elemento <video>. O elemento <track> permite que você especifique uma trilha de texto que pode ser exibida sobre o vídeo durante a reprodução. Portanto, o nosso gabarito é a letra E.

Gabarito: Letra E

52. (FCC – TRT 24 – 2017) Preocupado com a acessibilidade de um site que está sendo desenvolvido para o Tribunal Regional do Trabalho da 24ª Região, um Técnico recomendou o uso de recursos da linguagem HTML, versão 5, para identificar o idioma principal da página. Para isso deve-se utilizar:



- a) o atributo lang na tag <html>.
- b) o atributo lang na tag <body>.
- c) o atributo language na tag <!DOCTYPE>.
- d) a propriedade language na tag <meta>.
- e) a tag <lang> no interior da tag <head>

#### Comentários:

O atributo lang é um atributo HTML que pode ser usado para especificar o idioma de um elemento e seu conteúdo. É um importante recurso de acessibilidade, pois permite que tecnologias assistivas, como leitores de tela, interpretem e pronunciem corretamente o conteúdo de uma página da web. É usado na tag html: <html lang="en">, portanto, nosso gabarito é a letra a.

**Gabarito:** Letra A

---

53. (FCC – ALMS – 2016) A viewport é a área visível do usuário de uma página web e pode variar de acordo com o dispositivo. HTML5 introduziu um método para deixar os web designers terem controle sobre a viewport através da tag

- a) <viewport>
- b) <meta>
- c) <grid>
- d) <page>
- e) <scale>

#### Comentários:

O elemento meta com um atributo viewport é um recurso HTML5 que pode ser usado para controlar o layout de uma página da Web em diferentes dispositivos. O atributo viewport permite especificar o tamanho e a escala da viewport, ou a área visível de uma página da web, em um navegador da web. O elemento meta com um atributo viewport é frequentemente usado para tornar as páginas da Web responsivas, o que significa que elas podem ajustar seu layout e aparência para caber no tamanho e na resolução do dispositivo que está sendo usado para visualizá-las.

**Gabarito:** Letra B

---

54. (FCC – ELETROSUL – 2016) A uma página HTML5 deseja-se vincular um arquivo chamado arq1.css e um arquivo chamado arq2.js.

Para isso devem-se utilizar as instruções



- a) `<link rel="stylesheet" type="file/css" src="arq1.css">` e `<script src="arq2.js">`  
`</script>` no corpo da página.
- b) `<import type="text/css" href="arq1.css">` e `<script file="arq2.js">` `</script>` no cabeçalho da página.
- c) `<link type="css" style="external" href="arq1.css">` e `<script src="arq2.js">`  
`</script>` no cabeçalho da página.
- d) `<include type="text/css" href="arq1.css">` e `<script src="arq2.js">` `</script>` no corpo da página.
- e) `<link rel="stylesheet" type="text/css" href="arq1.css">` e `<script src="arq2.js">`  
`</script>` no cabeçalho da página.

### Comentários:

O elemento `<link>` com o atributo "rel" definido como "stylesheet" informa ao navegador que o documento vinculado é uma folha de estilo e fornece os meios para aplicar os estilos definidos na folha de estilo a uma página da web. O atributo "type" especifica o tipo do documento vinculado, que neste caso é "text/css" para uma folha de estilo CSS. O atributo "href" especifica a URL da folha de estilo. Neste caso, o valor do atributo "href" é "arq1.css", o que indica que a folha de estilo está localizada na URL "arq1.css". O elemento `<script>` com o atributo "src" especifica a URL de um script a ser executado. Neste caso, o valor do atributo "src" é "arq2.js", o que indica que o script está localizado na URL "arq2.js". Esses elementos seriam colocados no cabeçalho do documento HTML para aplicar os estilos na folha de estilo "arq1.css" e executar o script em "arq2.js" na página da web.

**Gabarito:** Letra E

55. (FCC – ALMS – 2016) Considere o fragmento de código HTML5 a seguir.

```
<body>
  <audio controls>
    ..I..
  </audio>
</body>
```

Para disponibilizar na página o áudio chamado beethoven.mp3 a lacuna I deve ser corretamente preenchida por

- a) `<sound src="beethoven.mp3" type="mpeg/ogg/wav" play="on">`
- b) `<source target="beethoven.mp3" media_type="sound/mpeg">`
- c) `<source src="beethoven.mp3" type="audio/mpeg">`
- d) `<sound source="beethoven.mp3" media="audio/mp3">`
- e) `<source src="beethoven.mp3" media="sound/mpeg" play="on">`

### Comentários:



No HTML5, o elemento `<source>` é usado para especificar várias fontes de mídia para os elementos `<video>` e `<audio>`. O atributo `src` do elemento `<source>` especifica a URL do arquivo de mídia a ser usado. A sintaxe é: `<source src="my-video.mp4" type="video/mp4">` no caso da questão ficaria `<source src="beethoven.mp3" type="audio/mpeg">`

Gabarito: Letra C

56. (FCC – SEMF Teresina – 2016) Em uma página HTML há um parágrafo vazio criado pela tag `<p id="local"></p>`. Em um bloco JavaScript da mesma página, para inserir neste parágrafo a palavra Teresina, utiliza-se o comando

- a) `document.getElementById("local").innerHTML = "Teresina";`
- b) `document.getElementById("p#demo").innerHTML = "Teresina";`
- c) `document.p["#local"].value = "Teresina";`
- d) `document.demo.value = "Teresina";`
- e) `document.getElementsByName("local").innerHTML = "Teresina";`

Comentários:

Em HTML, a função `getElementById` é um método usado para recuperar um elemento do documento por seu identificador exclusivo (ID). A função `getElementById` retorna uma referência ao elemento com o ID especificado ou `null` se tal elemento não existir. Para inserir o elemento `id="local"`, deve usar o método `document.getElementById` e usar o `id "local"` e inserir com `innerHTML = "Teresina"`.

Gabarito: Letra A

57. (FCC – TRT 23 – 2016) De acordo com as recomendações da W3C para a linguagem HTML5, a instrução

- a) `<meta charset="utf-8" />` está incorreta, pois tem uma barra no final.
- b) `<!doctype html>` está incorreta, pois deveria estar em letra maiúscula.
- c) `<table class=table striped>` está correta.
- d) `<html lang="en-US">` está correta.
- e) `<script src="myscript.js">` está incorreta, pois faltou o atributo `language="javascript"`

Comentários:

Pessoal, vejamos cada item. a) `<meta charset="utf-8" />` está incorreta, de fato, o correto seria: `<meta charset="UTF-8">`. b) `<!doctype html>` está incorreta, de fato, o correto seria: `<!DOCTYPE html>`. A letra c) também está incorreta. O elemento `<table>` em HTML é usado para criar uma tabela. O atributo `class` é usado para especificar um ou mais nomes de classe para um elemento, que pode ser usado para aplicar estilos ao elemento usando CSS. E possui a sintaxe `<table class="table striped">` portanto, faltou



inserir as aspas duplas. Nosso gabarito é a letra d. `<html lang="en-US">` está correta. Por fim, a letra e está errada. O correto seria `<script src="myscripts.js"></script>`

Gabarito: Letra D

---

58. (FCC –TRF3 – 2016) Em um site que utiliza HTML5 Application Cache há um arquivo chamado `dados.appcache` que define que uma página poderá ter o conteúdo acessado mesmo estando offline. Nessa página, para apontar para esse arquivo deve-se utilizar a instrução

- a) `<html manifest="dados.appcache">`
- b) `<cache manifest="dados.appcache">`
- c) `<html src="dados.inf" cached>`
- d) `<cache src="dados.inf">`
- e) `<!MANIFEST cache="demo.appcache">`

#### Comentários:

No HTML5, o atributo `manifest` é usado para especificar a localização de um arquivo de manifesto do cache do aplicativo. Um manifesto de cache de aplicativo é um arquivo que define os recursos que devem ser armazenados em cache pelo navegador da Web para que possam ser usados offline.

A sintaxe usada é a seguinte: `<html manifest="dados.appcache">`, portanto o gabarito é a letra A.

Gabarito: Letra A

---



## Questões FGV

59. (FGV –TJ TO– 2022) No HTML5, o evento que é disparado quando um objeto é carregado é:

- a) begin;
- b) load;
- c) ready;
- d) run;
- e) start.

### Comentários:

No HTML5, o evento load é disparado quando um recurso e seus recursos dependentes terminam de carregar. O evento load pode ser usado com elementos como <img>, <audio> e <video> para detectar quando os recursos aos quais eles se referem terminaram de ser carregados. Portanto nosso gabarito é a letra b. Ademais, vejamos a definição das demais alternativas. O atributo begin é usado para especificar um deslocamento de tempo para o elemento <track>. O evento ready é acionado quando o documento está totalmente carregado e pronto para manipulação. O atributo run é usado para especificar se um script deve ou não ser executado assim que estiver disponível, em vez de esperar que todo o documento seja carregado. O atributo start é usado para especificar o ponto inicial padrão para uma trilha de texto, como subtítulos ou legendas.

**Gabarito:** Letra B

60. (FGV –MPE SC– 2022) Sobre elementos block-level (nível de bloco) e elementos inline (em linha) no HTML, analise as afirmativas a seguir.

- I. Um elemento block-level sempre começa numa nova linha.
- II. <p> é um elemento inline.
- III. <div> é um elemento block-level.

Está correto somente o que se afirma em:

- a) I;
- b) II;
- c) I e II;
- d) I e III;
- e) II e III..

### Comentários:



Pessoal, elementos de nível de bloco são elementos usados para criar blocos de conteúdo formatados independentemente de outros blocos. Os elementos de nível de bloco geralmente começam em uma nova linha e ocupam toda a largura de seu elemento pai. Portanto, a assertiva está errada. A segunda assertiva está errada, lembre-se que dois elementos de bloco comumente usados são: `<p>` e `<div>`. Por fim, a assertiva III está correta: `<div>` é um elemento block-level. Assim, ficamos com as assertivas I e III.

Gabarito: Letra D

61. (FGV –MPE GO – 2022) Num documento HTML5, assinale o elemento que contém meta informações sobre a página.
- a) `<!DOCTYPE html>`
  - b) `<body>` (
  - c) `<head>`
  - d) `<html>`
  - e) `<script>`.

#### Comentários:

Meta tags são linhas de código HTML ou "etiquetas" que, entre outras coisas, descrevem o conteúdo do seu site para os buscadores. É nelas que você vai inserir as palavras-chave que facilitarão a vida do usuário na hora de te encontrar, por exemplo. Por meio delas, você pode também "assinar" seu site, declarando sua autoria sobre o código fonte. As meta tags devem ser incluídas no seu código HTML, dentro da tag `<head>`, portanto, nosso gabarito é a letra C.

Gabarito: Letra C

62. (FGV –TJ TO– 2022) Observe o código HTML e JavaScript a seguir:





```
<!DOCTYPE html>
<html>
<body>
  <p id="Tejota"></p>
  <script>
    const tjProcesso = {ID:"2022.1", envolvidos:
      ["João", "Maria"]};
    document.getElementById("Tejota").innerHTML
      = tjProcesso.envolvidos[0];
  </script>
</body>
</html>
```

Após a execução do código, o resultado exibido será:

- a) Maria
- b) João
- c) 2022.1
- d) Tejota
- e) tjProcesso

#### Comentários:

Pessoal, vejamos o que ocorre nesse caso. O método getElementById acessa o elemento 0 da seguinte forma: `tjProcesso.envolvidos[0] = João` e o insere na página. Portanto, nosso gabarito é a letra B.

**Gabarito: Letra B**

63. (FGV –Sefaz AM – 2022) A linguagem de marcação HTML é amplamente utilizada na construção de páginas da Internet.

O elemento HTML, que permite reproduzir um arquivo de vídeo no formato MP4 em uma página da web, é

- a) <clip>
- b) <embedded>
- c) <media>
- d) <movie>
- e) <video>.

#### Comentários:

No HTML5, usamos o elemento <video> para incorporar conteúdo de vídeo em uma página da web. Os elementos <clip>, <embedded>, <media> e <movie> não são



elementos HTML. Não fazem parte da especificação HTML e não são reconhecidos pelos navegadores da web.

Gabarito: Letra E

64. (FGV –IMBEL – 2021) Analise a imagem a seguir, produzida numa página Web.

A	C		
B	<table border="1"> <tr> <td>F</td> <td>G</td> </tr> </table>	F	G
F	G		

```
<style>
  th { border: 1px solid black; }
</style>
<table>
...
</table>
```

que substitui corretamente a linha com pontos.

- a) `<tr><th>A</th><th>C</th></tr>`  
`<tr><td>F</td><td><table>`  
`<tr><td>G</td><td>G</td></tr>`  
`</table></th></tr>`
- b) `<tr><th>A</th><th>C</th></tr>`  
`<tr><td>B</td><td><table>`  
`<tr><th>F</th><th>G</th></tr>`  
`</table></th></tr>`
- c) `<tr><th>A</th><th>C</th></tr>`  
`<tr><td>B</td><td>`  
`<tr><td>F</td><td>G</td></tr>`  
`</th></tr>`
- d) `<tr><th>A</th><th>C</th></tr>`  
`<tr><td>B</td><td></table><tr><td>F</td><td>G</td></tr>`  
`<table></th></tr>`
- e) `<tr><td>A</td><td>C</td></tr>`  
`<tr><td>B</td><td><table>`  
`<tr><td>F</td><td>G</td></tr>`  
`</table></th></tr>`.

Gabarito:

Comentários:

A alternativa B cria uma tabela HTML com três linhas e duas colunas. A primeira linha contém dois cabeçalhos de tabela (TH) com o texto "A" e "C". A segunda linha contém duas células (TD) com o texto "B" e uma tabela interna com uma linha e dois cabeçalhos de tabela "F" e "G", sendo este nosso gabarito. Por outro lado, a letra A tem três linhas



e duas colunas. A primeira linha contém dois cabeçalhos de tabela "A" e "C". A segunda linha contém duas células. A primeira célula contém o texto "F" e a segunda célula contém uma tabela interna com uma linha e duas células "G" e "G". A letra C apresenta erros: este código HTML é inválido. A tabela interna não está fechada corretamente, pois não há tag `</table>` de fechamento. Além disso, a segunda linha da tabela interna não está aninhada corretamente na segunda célula da tabela externa. Assim como a letra D e a letra E, que possuem códigos inválidos.

Gabarito: Letra B

65. (FGV – TJ RO– 2021) No contexto do HTML, a sintaxe correta para um comentário, delimitando um trecho que NÃO deve ser exibido pelo browser, é:
- `<-- Texto do comentário -->`
  - `!--Texto do comentário >`
  - `<-Texto do comentário ->`
  - `<!--Texto do comentário -->`
  - `<--Texto do comentário --/>`

#### Comentários:

Em HTML, a sintaxe `<!-- -->` é usada para criar um comentário. Os Comentários: são usados para adicionar notas e explicações ao código HTML e não são exibidos na página da web. O nosso gabarito é a letra D: `<!--Texto do comentário -->`

Gabarito: Letra D

66. (FGV – IMBEL– 2021) Analise o trecho HTML a seguir.

```
<table>
<tr><th>A</th><th>C</th></tr>
<tr><th>B</th><th>D</th></tr>
</table>
```

Na exibição da página Web, esse script produz uma tabela

- com duas linhas e duas colunas.
- com quatro linhas e uma coluna.
- com quatro colunas e uma linha.
- com uma linha e uma coluna.
- com linhas e colunas desalinhadas.

#### Comentários:



Pessoal, o terço HTML gera uma linha com as colunas A e C e outra coluna B e D. Vamos lembrar das tags tr e th? Cada célula da tabela é definida por uma tag <td> e </td>. Por outro lado, cada linha da tabela começa com uma tag <tr> e termina com uma </tr>. Além disso, o elemento <th> é usado para criar uma célula de cabeçalho de tabela. O elemento <th> é usado dentro de um elemento <tr> (linha da tabela) para definir uma célula que contém informações de cabeçalho para uma tabela.

Gabarito: Letra A

---

67. (FGV –MPE AL – 2018) O HTML 5 introduziu diversos elementos semânticos, gráficos e de multimídia.

Assinale o elemento que não é uma novidade nessa versão.

- a) <article>
- b) <div>
- c) <footer>
- d) <header>
- e) <section>.

Comentários:

O HTML5 apresenta novos elementos com significados específicos, como <header>, <footer> e <article>. Além desses, temos o <section> que representa uma seção da página da web, como um capítulo ou um tema. Dentre os novos elementos semânticos <header>, <footer> <article> e <section>, a única opção que não é uma novidade no HTML 5 é o <div>.

Gabarito: Letra B

---

68. (FGV –ALERO – 2018) No contexto dos elementos introduzidos pelo HTML 5, analise as afirmativas a seguir.

- I. Os elementos "header", "section" e "footer" são elementos block-level.
- II. O elemento "<wbr>" é funcionalmente idêntico ao já conhecido elemento "<br>".
- III. Além dos atributos height e width, o elemento "<canvas>" permite o uso do atributo depth, para uso em gráficos tridimensionais.

Está correto o que se afirma em

- a) I, somente.
- b) II, somente.
- c) III, somente.
- d) I e II, somente.



e) II e III, somente..

#### Comentários:

Pessoal, elementos de nível de bloco são elementos usados para criar blocos de conteúdo formatados independentemente de outros blocos. Os elementos de nível de bloco geralmente começam em uma nova linha e ocupam toda a largura de seu elemento pai. O item I está correto! Já o item II está incorreto: a tag `<wbr>` define uma possível quebra de linha. Algumas das diferenças entre o `<wbr>` e `<br>` são: `<wbr>` é usado para agrupar palavras longas ou URLs para melhorar a legibilidade, enquanto `<br>` é usado para criar uma quebra de linha em um bloco de texto. `<wbr>` é um elemento inline, enquanto `<br>` é um elemento de nível de bloco. Isso significa que `<wbr>` pode ser usado dentro de um bloco de texto, enquanto `<br>` cria um novo bloco. `<wbr>` nem sempre é respeitado pelo navegador, pois depende do layout e do espaço disponível. `<br>`, por outro lado, sempre cria uma quebra de linha. `<wbr>` não é suportado em todos os navegadores, enquanto `<br>` é amplamente suportado. O uso do atributo `depth` no "`<canvas>`" não tem por finalidade gráficos tridimensionais. A `depth` no "`<canvas>`" refere-se ao número de bits usados para representar a cor de cada pixel na tela. A profundidade é normalmente expressa como o número de bits por pixel (bpp). Por exemplo, uma tela com profundidade de 8 bpp pode representar até 256 cores diferentes, enquanto uma tela com profundidade de 24 bpp pode representar até 16 milhões de cores diferentes.

Gabarito: Letra A

69. (FGV –ALERO – 2018) Assinale a opção que melhor descreve a utilização da tag `canvas` no HTML5.

- a) Definir mídias de áudio e vídeo.
- b) Desenhar elementos gráficos por meio de scripts.
- c) Dividir a tela em áreas horizontais de largura variável, que funcionam como painéis.
- d) Dividir a tela em áreas verticais de altura variável, que funcionam como painéis.
- e) Desenhar a entrada de dados e as caixas de diálogo..

#### Comentários:

O elemento `<canvas>` é um elemento HTML que foi introduzido no HTML5 e é usado para desenhar gráficos, animações e jogos usando JavaScript. É uma tela de bitmap dependente de resolução que pode ser usada para renderizar gráficos, gráficos de jogos ou outras imagens visuais em tempo real.

Gabarito: Letra B

70. (FGV –MPE AL – 2018) Observe o código a seguir.



```
<button onmouseover="this.innerHTML='ON' "  
onmouseout="this.innerHTML='OFF' ">Aqui</button>
```

Supondo que o cursor do mouse inicialmente esteja fora da área de botão, assinale o texto exibido no botão quando a página que o contém estiver carregada.

- a) "Abrir" inicialmente, "on" quando o cursor do mouse entra na área do botão, e "off" quando o cursor do mouse sai da área do botão, assim mudando a cada movimentação para dentro ou para fora.
- b) "Abrir" inicialmente, "on" quando o cursor do mouse entra na área do botão, e assim permanece.
- c) "Abrir" inicialmente, "on" quando o cursor do mouse entra na área do botão, e "off" quando o cursor do mouse sai da área do botão, e assim permanece mesmo que o mouse seja movimentado sobre o botão.
- d) "Abrir" inicialmente, "on" quando o cursor do mouse entra na área do botão, e "Abrir" quando o cursor do mouse sai da área do botão, assim mudando a cada movimentação para dentro ou para fora.
- e) "Abrir" inicialmente, "on" quando o cursor do mouse entra na área do botão, e "off" quando o cursor do mouse sai da área do botão pela primeira vez. Nas demais saídas, mudar para "Abrir".

#### Comentários:

O código cria um botão HTML que exibe o texto "ON" quando o mouse está sobre ele e "OFF" quando o mouse não está sobre ele. Isso pode ser obtido usando os atributos onmouseover e onmouseout, que permitem especificar o código JavaScript a ser executado quando o mouse entra ou sai do elemento. Portanto, nosso gabarito é a letra a: o texto exibido no botão quando a página que o contém estiver carregada vai "Abrir" inicialmente, "on" quando o cursor do mouse entra na área do botão, e "off" quando o cursor do mouse sai da área do botão, assim mudando a cada movimentação para dentro ou para fora.

**Gabarito:** Letra A

71. (FGV –ALERO – 2018) Analise o código a seguir.



```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"
></script>
<script>
$(document).ready(function()
{
  $("button").click(function()
  {
    $(this)[0].innerText
      = parseInt($(this)[0].innerText) + 1;
  });
});
</script>
</head>
<body>
<button id="xpto">1</button>
</body>
</html>
```

Sabe-se que com essa página exibida, o operador clicou no botão por três vezes. Assinale o valor exibido no botão após esses cliques.

- a) 0
- b) 1
- c) 1111
- d) 1234
- e) 4.

#### Comentários:

O código consiste em uma página HTML com um botão que incrementa um contador quando clicado. A biblioteca jQuery está sendo carregada de uma CDN (Content Delivery Network) para habilitar o uso de funções jQuery no script. O script usa a função `$(document).ready()` para agrupar o código que será executado quando a página for carregada e a função `$("#button").click()` para vincular uma função ao evento click do botão elemento com o ID "xpto".

Dentro da função de clique, a expressão `$(this)[0].innerText` é usada para obter o conteúdo de texto do elemento de botão, que é analisado como um número inteiro e incrementado em 1. Por fim, o valor atualizado é atribuído de volta ao propriedade `innerText` do elemento de botão. Assim, como o operador clicou no botão por três vezes o valor exibido no botão após esses cliques é 4.





72. (FGV – BANESTES– 2018) HTML é uma linguagem utilizada para construção de páginas na Internet. O comando de formatação da linguagem HTML para indicar que um determinado texto deve ser apresentado sublinhado é:
- a) <b></b>
  - b) <s></s>
  - c) <i></i>
  - d) <p></p>
  - e) <u></u>.

Comentários:

Vejamos o significado de cada tag: A tag <b> representa um trecho de texto que deve ser renderizado em negrito. A tag <s> representa um trecho de texto que não é mais preciso ou relevante. Geralmente é usado para riscar o texto para indicar que ele foi removido ou substituído. A tag <i> representa um trecho de texto que deve ser renderizado em uma fonte em itálico. A tag <p> representa um parágrafo de texto. A tag <u> representa um trecho de texto que deve ser renderizado como sublinhado.

Gabarito: Letra E

73. (FGV –BANESTES – 2018) Ao desenvolver uma aplicação cliente Web, o programador precisa mostrar um aviso quando a página requisitada tem o acesso proibido.

Assim, ele tem que tratar o erro de resposta HTTP:

- a) 200;
- b) 403;
- c) 404;
- d) 500;
- e) 501.

Comentários:

Vejamos o significado de cada código de erro: 200: Este é um código de status de sucesso, indicando que a solicitação foi bem-sucedida e o recurso solicitado foi retornado. 403: Este é um código de erro que ocorre quando a página requisitada tem o acesso proibido, indicando que o servidor entendeu a solicitação, mas se recusou a autorizá-la. Isso pode ocorrer quando o usuário não possui as permissões necessárias para acessar o recurso solicitado. 404: Este é um código de erro não encontrado, indicando que o servidor não pode encontrar o recurso solicitado. Isso pode ocorrer





quando a URL do recurso está incorreta ou o recurso foi movido ou excluído. 500: Este é um código de erro do servidor, indicando que o servidor encontrou uma condição inesperada que o impediu de atender à solicitação. 501: Este é um código de erro não implementado, indicando que o servidor não suporta a funcionalidade necessária para atender à solicitação. Isso pode ocorrer quando o servidor não reconhece o método de solicitação (por exemplo, GET, POST, etc.) ou não suporta o recurso solicitado.

Gabarito: Letra B

74. (FGV –IBGE – 2017) HTML5 (Hypertext Markup Language, versão 5) é uma linguagem utilizada para estruturação e apresentação de conteúdo na Internet.

Analise a página HTML5 a seguir:

```
<!DOCTYPE html>
<html>
<head><script>
function clickCounter() {
  var A = localStorage.clickcount;
  var B = sessionStorage.clickcount;
  if (A) {A = Number(A)+1;} else {A= 1;}
  if (B) {B = Number(B) + 1;} else {B= 1;}

  document.getElementById("result").innerHTML =
    "Você apertou " + A + B + " vez(es).";
  sessionStorage.clickcount = B;
  localStorage.clickcount = A;
}
</script></head>
<body>

<p><button id="B" onclick="clickCounter()" type="button">
Aperte aqui!</button></p>
<div id="result"></div>
</body></html>
```

Considere que você está utilizando um navegador web que suporta HTML5 Local Storage e ao acessar essa página pela primeira vez você apertou três vezes o botão indicado com id="B" e então você fechou o navegador.

Posteriormente, você utiliza o mesmo navegador para acessar novamente essa mesma página.

Após apertar duas vezes o mesmo botão, será apresentada a seguinte mensagem no corpo da página:

- a) Você apertou 5 vezes.
- b) Você apertou 7 vezes.
- c) Você apertou 10 vezes.
- d) Você apertou 52 vezes.
- e) Você apertou 55 vezes.



## Comentários:

O código cria uma página HTML com um botão que conta o número de cliques. A contagem de cliques é armazenada no armazenamento local e da sessão e exibida na página usando a propriedade innerHTML de um elemento div com o ID "resultado". O objeto localStorage é usado para armazenar dados que persistem mesmo quando o navegador é fechado, enquanto o objeto sessionStorage é usado para armazenar dados que estão disponíveis apenas para a sessão atual da página. O botão conta o número de cliques e exibe a contagem total na página. Se você acessou a página pela primeira vez e clicou no botão três vezes, o valor da variável localStorage.clickcount seria definido como 4, pois começa em 1 e é incrementado em 1 toda vez que o botão é clicado. O valor da variável sessionStorage.clickcount também seria definido como 4, pois começa em 1 e é incrementado em 1 toda vez que o botão é clicado. sessionStorage -> os dados ficam armazenados na sessão do navegador, ou seja, se fechar, perderá esses dados. Por outro lado, no local storage o browser, sendo apagado ao desligar ou ao reiniciar. Portanto, o elemento document.getElementById("result").innerHTML = "Você apertou " + A + B + " vez(es)."; apresentará como Saída: Você apertou 52 vez(es).

Gabarito: Letra D

75. (FGV –ALERJ – 2017) Analise a estrutura básica da página HTML a seguir:

```
<html>
  <head>
    <title>Título da página</title>
  </head>
  <body bgcolor="black"
    alink="green"
    vlink="yellow"
    link="red"
    text="blue">
```

Corpo do documento.

```
  <a href="#">Click aqui!</a>
</body>
</html>
```

Esse código define que a cor do texto da página e dos links quando clicados são, respectivamente:

- a) preta e vermelha;
- b) azul e amarela;
- c) vermelha e verde;
- d) azul e verde;



e) preta e amarela.

#### Comentários:

O elemento <body> é usado para definir o conteúdo principal de um documento HTML. O atributo "bgcolor" especifica a cor de fundo da página da web. Nesse caso, o valor do atributo "bgcolor" é "black", o que tornaria a cor de fundo da página da Web preta. Os atributos "alink", "vlink" e "link" especificam as cores dos links na página da web. O atributo "alink" especifica a cor de um link quando ele é clicado. O atributo "vlink" especifica a cor de um link visitado. O atributo "link" especifica a cor de um link não visitado. Nesse caso, os valores dos atributos "alink", "vlink" e "link" são "green", "yellow" e "red", respectivamente, o que tornaria a cor de um link clicado verde, o a cor de um link visitado é amarela e a cor de um link não visitado é vermelha. O atributo "texto" especifica a cor do texto na página da web. Nesse caso, o valor do atributo "texto" é "azul", o que tornaria a cor do texto azul. Portanto, a cor do texto da página e dos links quando clicados são, respectivamente azul e verde.

Gabarito: Letra D

76. (FGV –MPE BA – 2017) A HTML é uma linguagem de marcação utilizada na construção de páginas na Web.

O comando de formatação da linguagem para definir um hyperlink indicando que o recurso linkado deve ser aberto em uma nova janela do navegador é:

- a) <a href = "url" target = "\_top">meu link</a>
- b) <a href = "url".new>meu link</a>
- c) <a href = "url" target="\_blank">meu link</a>
- d) <a href = "url" target = "\_parent">meu link</a>
- e) <a href = "url" target = "\_self">meu link</a>.

#### Comentários:

O atributo target especifica onde abrir o documento vinculado. Ele pode ter um dos seguintes valores: \_self- Predefinição. Abre o documento na mesma janela/guia em que foi clicado. \_blank- Abre o documento em uma nova janela ou guia. \_parent- Abre o documento no quadro pai. \_top- Abre o documento em todo o corpo da janela. Portanto, nosso gabarito é a letra c: \_blank- Abre o documento em uma nova janela ou guia.

Gabarito: Letra C

77. (FGV –IBGE – 2017) O HTML 5 introduziu um método para permitir que o designer controle a área de visualização de um site através da tag <meta>. A sintaxe correta



para que a largura de uma página siga a largura da tela de um dispositivo, com uma escala inicial de 1X no momento em que ela é carregada é:

- a) `<meta name="viewport" content="device, scale=1.0">`;
- b) `<meta type="viewport" width="device" scale="1.0">`;
- c) `<meta name="viewport" width="device-width, initial-scale=1.0">`;
- d) `<meta name="viewport" content="width=device-width, initial-scale=1.0">`;
- e) `<meta type="viewport" content="width=device-width" scale="initial-scale-1">`.

#### Comentários:

Questão importante! Foi cobrada no concurso FCC – TRT 23 – 2022! A sintaxe correta para que a largura de uma página siga a largura da tela de um dispositivo, com uma escala inicial de 1X no momento em que ela é carregada é `<meta name="viewport" content="width=device-width, initial-scale=1.0">`. A tag meta com um atributo name "viewport" é usada para controlar o layout de uma página da Web em um dispositivo móvel. O atributo de conteúdo é usado para definir a largura da viewport para a largura do dispositivo e para definir a escala inicial para 1,0, o que significa que a página da Web será exibida em uma escala de 1:1 no dispositivo. É necessário saber a sintaxe correta.

**Gabarito:** Letra D

78. (FGV – IBGE– 2016) A sigla HTML significa Hyper Text Markup Language, o que pode ser traduzido como Linguagem de Marcação de Hipertexto. Uma linguagem de marcação pode ser definida como um sistema para:

- a) marcar um documento indicando a sua estrutura lógica e hierárquica especificamente para a transmissão e exibição eletrônicas;
- b) definir o comportamento visual em meio eletrônico do conteúdo textual de um documento, incluindo tipografia, cor e tamanho dos caracteres;
- c) marcar um documento indicando a ordem em que o conteúdo deve ser apresentado em meio eletrônico;
- d) definir as ligações entre diferentes documentos a partir de palavras-chave específicas;
- e) criar documentos específicos para transmissão eletrônica através da Internet.

#### Comentários:

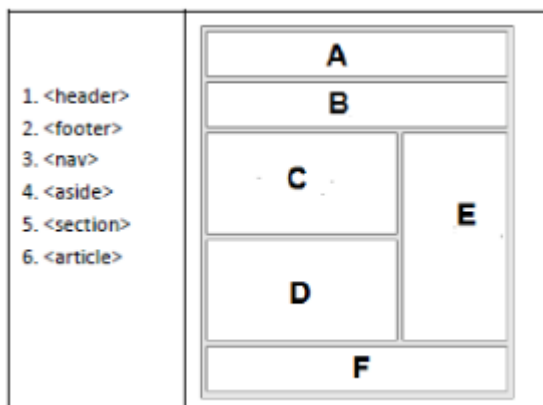
HTML (Hypertext Markup Language) é uma linguagem de marcação usada para estruturar e formatar conteúdo na web. É usado para marcar um documento, indicando sua estrutura lógica e hierárquica, para que os navegadores da web possam interpretar e exibir o conteúdo corretamente. Portanto, nosso gabarito é: Uma linguagem de marcação pode ser definida como um sistema para marcar um documento indicando a



sua estrutura lógica e hierárquica especificamente para a transmissão e exibição eletrônicas

Gabarito: Letra A

79. (FGV –IBGE – 2016) Um desenvolvedor Web mobile pretende utilizar os novos elementos semânticos disponíveis no HTML5 em suas páginas. Associe corretamente os elementos HTML5 enumerados com o posicionamento na ilustração que representa conceitualmente as partes de uma página HTML:



A sequência correta é:

- a) A=1, B=3, C=5, D=6, E=4 e F=2;
- b) A=2, B=5, C=4, D=3, E=6 e F=1;
- c) A=1, B=4, C=6, D=5, E=3 e F=2;
- d) A=2, B=3, C=6, D=4, E=5 e F=3;
- e) A=1, B=6, C=4, D=3, E=5 e F=2.

Comentários:

Vamos lá: 1) header: Este elemento representa o cabeçalho de uma seção ou página. Ele normalmente contém o logotipo, a navegação do site e outras informações importantes sobre a página ou o site. Este é o nosso elemento A. 2) footer: Este elemento representa o rodapé de uma seção ou página. Ele normalmente contém informações de direitos autorais, links para documentos relacionados e outras informações que não fazem parte do conteúdo principal da página. Esse é representado pelo elemento F. 3) nav: Este elemento representa uma seção da página que contém links de navegação para outras páginas ou para outras partes da mesma página. Este é o elemento B. 4) aside: Este elemento representa uma seção da página que está relacionada ao conteúdo principal, mas separada dele. Geralmente é usado para exibir barras laterais ou citações extraíveis, este é o elemento E. 5) section: Este elemento representa uma seção genérica de um documento, como um capítulo, uma seção de um jornal ou uma seção de um site, este é o elemento C. 6) article: Este elemento representa um conteúdo independente que pode



ser distribuído ou reutilizado independentemente, como uma postagem de blog ou um artigo de notícias, este é o elemento D.

A=1, B=3, C=5, D=6, E=4 e F=2

Gabarito: Letra A

80. (FGV – IBGE– 2016) Com a introdução do HTML5, diversas novas APIs Javascript (Application Programming Interfaces) foram disponibilizadas, aumentando consideravelmente a quantidade de recursos disponíveis para a produção de páginas web. São APIs exclusivas do HTML5:

- a) múltiplas colunas de texto, transformações 2D/3D e RWD (Responsive Web Design);
- b) armazenamento em nuvem, suporte a telas de toque e SSL (Secure Sockets Layer);
- c) acesso a câmeras em dispositivos móveis, suporte a streaming de vídeo e SSE (Streaming SIMD Extensions);
- d) armazenamento local, geolocalização e SSE (Server-Sent Events);
- e) redimensionamento dinâmico de imagens, detecção de resolução de tela e RWD (Responsive Web Display)

Comentários:

O HTML5 apresenta várias novas APIs (Application Programming Interfaces) que permitem aos desenvolvedores da Web criar aplicativos da Web mais poderosos e interativos. Algumas APIs exclusivas do HTML5 são: API de geolocalização, API Web Storage, API Web Workers, API WebSockets, API Canvas, API de áudio da Web, API Server-Sent Events (SSE).

Gabarito: Letra D

81. (FGV –IBGE – 2016) A declaração <!DOCTYPE> permite ao navegador apresentar uma página web corretamente. A declaração correta para uma página em HTML5 é:

- a) <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 5.0 Strict//EN" "http://www.w3.org/TR/html5/strict.dtd">
- b) <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 5.0 Transitional//EN" "http://www.w3.org/TR/html5/loose.dtd">
- c) <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 5.0">
- d) <!DOCTYPE html>
- e) <!DOCTYPE XML PUBLIC "-//W3C//DTD HTML 5.0">

Comentários:

A declaração <!DOCTYPE html> define que este documento é um documento **HTML5**.







## LISTAS QUESTÕES

### Questões Cespe

1. (CESPE – TCE RJ– 2022). Quanto ao desenvolvimento de sistemas web, julgue o item seguinte.

HTML5 é uma linguagem de programação que permite estruturar páginas web e executar comandos como loops de repetição, por exemplo.

2. (CESPE –DP DF– 2022) Julgue o item seguinte, a respeito da formatação de dados.

A tag `<meta charset="UTF-8">` define o conjunto de caracteres usados na página, nesse caso, o UTF-8, que é o padrão para HTML5.

3. (CESPE – PGE RJ – 2022) Julgue o item que se segue, relacionado a desenvolvimento de sistemas.

No HTML 5, `sessionStorage` pode ser utilizado para armazenamento local de dados, tendo como característica o armazenamento de dados restritos à aba em funcionamento.

4. (CESPE – APEX – 2021) Em HTML5, considerando-se o contexto de geolocalização e acesso ao dispositivo, para se obter retorno mais rápido e de baixa precisão sobre a localização de um dispositivo, deve-se

- a) invocar um método para o objeto `PositionOptions` da API com o parâmetro "0".
- b) configurar a função `watchPosition()` da API de geolocalização.
- c) testar a existência do objeto `navigator.geolocation` no dispositivo.
- d) chamar o método `getCurrentPosition()` da API de geolocalização.

5. (CESPE – PGDF – 2021) Acerca de linguagens de marcação utilizadas para formatação de dados, julgue o item a seguir.

HTML e XML são equivalentes, pois ambas possuem uma semântica de apresentação predefinida.

6. (CESPE – SEFAZ CE – 2021) Com relação à arquitetura de desenvolvimento de software, julgue o item a seguir.

Um link de navegação compartilhado por diversas páginas é incluído no elemento `<main>` de uma página HTML5.





7. (CESPE – PGDF – 2021) Julgue o item a seguir, referente a linguagens de scripts.

Considere o código seguinte, em HTML e JavaScript.

```
<html>
<input type='text' id='a' value='5'>
<br>
<input type='text' id='b' value='2'>
<br>
<input type='text' id='c' >
<script>
document.getElementById('c').value=(document.
getElementById('a').value+document.getElement
ById('b').value);
</script>
</html>
```

Em um navegador Internet com JavaScript habilitado, esse código apresentará o resultado a seguir

5
2
52

8. (CESPE – TJ PA – 2020) A respeito do tratamento off-line de um sítio no HTML 5, assinale a opção correta.

- a) Esse tratamento pode ser usado para a criação de dados em momento anterior ao acesso à aplicação.
- b) Na sessão cache do arquivo manifesto, devem estar relacionados todos os arquivos que o navegador deve copiar para que estejam disponíveis para uso off-line.
- c) É necessário que os arquivos PHP estejam listados na sessão cache.
- d) Na sessão network do arquivo manifesto, devem estar relacionados os arquivos que precisam ser substituídos por outros no retorno da conexão.
- e) Na sessão fallback do arquivo manifesto, devem estar relacionados os arquivos que não são utilizados para o processamento off-line.

9. (CESPE – MPE CE – 2020) Acerca de JSON e HTML 5, julgue o item subsecutivo.

No HTML 5, localStorage é um recurso de armazenamento local que usa objetos JavaScript e que permite manter dados sem data de expiração prévia.



10. (CESPE – ProTI ME – 2020) Acerca de desenvolvimento de sistemas web, julgue o item a seguir:

No HTML 5, os novos campos para formulários, como email, search e range, e os atributos, como placeholder, pattern e required, reduzem a necessidade de utilização de plugins para auxiliar a formatação dos elementos.

11. (CESPE – ProTI ME – 2020) Acerca de desenvolvimento de sistemas web, julgue o item a seguir:

No HTML 5, as tags de link e script usadas para referenciar arquivos de CSS e JavaScript não precisam informar o atributo type, porque, na sua ausência, o navegador assume que o arquivo é do tipo text/css ou text/javascript.

12. (CESPE – ProTI ME – 2020) Acerca de desenvolvimento de sistemas web, julgue o item a seguir.

A instrução DOCTYPE do HTML 5 é mais simples que a das versões anteriores HTML 4 ou XHTML 1.

13. (CESPE – TJ PA – 2020)

```
<!DOCTYPE html>
<html>
<p id="prova"></p>
<script>
meuObjeto = { "nome":"Caca", "Idade":52,
"carro":null };
for (x in meuObjeto) {
document.getElementById("prova").innerHTML += x;
}
</script>
</body>
</html>
```

O código html precedente, ao ser executado em um navegador de Internet, produzirá o seguinte resultado

- a) Caca52null
- b) nomeladecarro
- c) [object Object][object Object][object Object]
- d) Caca+52+null
- e) nome+Idade+carro.



14. (CESPE – TJ PA – 2020) Na linguagem HTML 5, geralmente considera-se determinado elemento como o ponto central do conteúdo do documento, o qual pode ser, por exemplo, um post.

Esse elemento, que representa um conteúdo independente e altamente relevante, é o

- a) aside.
- b) canvas.
- c) embed.
- d) article.
- e) figure.

15. (CESPE –SLU DF – 2019) Com relação a desenvolvimento de software, julgue o item a seguir.

De acordo com o trecho de código a seguir, escrito em HTML5, novos valores de date e time são válidos como atributos de elementos de formulário, e apenas o campo data é de preenchimento obrigatório.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>HTML5</title>
</head>
<body>
  <form method="post" action="teste.html">
    <label for="data">Data: </label>
    <input id="data" type="date" required name="data"/>
    <br />
    <label for="hora">Hora: </label>
    <input id="hora" type="time" name="hora"/>

    <input type="submit" value="Enviar"/>
  </form>
</body>
</html>
```

16. (CESPE – MPC TCE-PA – 2019) O HTML (hypertext markup language) tem amplo uso difundido nas páginas publicadas na Internet. Assinale a opção que corresponde à tag utilizada no caso em que seja necessário utilizar uma lista não ordenada.

- a) <b>
- b) <p>
- c) <tr>
- d) <ul>
- e) <th>



17. (CESPE – TJ AM – 2019) Acerca do desenvolvimento web mediante o uso do HTML 5, do JavaScript, do XML e do CSS, julgue o item subsequente.

O HTML 5 define como os navegadores web devem lidar com marcações antigas como <font>, <center> e outras tags de apresentação.

18. (CESPE – TJ AM – 2019) Acerca do desenvolvimento web mediante o uso do HTML 5, do JavaScript, do XML e do CSS, julgue o item subsequente.

O HTML 5 especifica novas API (application program interface) para o modelo de objeto de documento (DOM — document object model) referente a arrastar e soltar eventos enviados pelo servidor, como desenhos, vídeos e similares.

19. (CESPE – TJ AM – 2019) Acerca do desenvolvimento web mediante o uso do HTML 5, do JavaScript, do XML e do CSS, julgue o item subsequente.

A XHTML 5 é uma serialização XML que tem as mesmas características e sintaxes do HTML 5.

20. (CESPE – MPE PI – 2018) Julgue o próximo item, relativo a lógica de programação e linguagens de programação.

```
<html>
<body>
<h2>JavaScript </h2>
<p id="demo"></p>
<script>
var colors = ["Blue", "Red", "White"];
document.getElementById("demo").innerHTML = colors;
</script>
</body>
</html>
```

A execução do código JavaScript anteriormente apresentado retornará o seguinte resultado:

```
JavaScript
Blue,Red,White
```

21. (CESPE – TJ STJ – 2018) Julgue o item seguinte, a respeito de Maven, desenvolvimento web, servidor web, servidor de aplicação e criptografia.



No HTML5, o atributo autofocus possibilita que qualquer elemento <input> seja automaticamente focado quando do carregamento da página.

22. (CESPE – STM – 2018) Julgue o item subsequente, a respeito de programação web.

Em HTML5, a tag <output> fornece uma indicação ao usuário do que pode ser inserido no campo.

23. (CESPE – TRE BA – 2017) Entre os novos elementos do HTML5, o elemento:

- a) <mark> é o ponto de parada do cursor em qualquer parte da página HTML.
- b) <progress> define o progresso de uma tarefa.
- c) <meter> interpreta medições meteorológicas.
- d) <figcaption> captura figuras.
- e) <main> define a estrutura principal da linguagem C dentro da página HTML

24. (CESPE – TCE-PA – 2016) Acerca dos conceitos e das técnicas necessários à construção de sítios web em que se utilizam CSS e HTML, julgue o item que se segue.

HTML é uma linguagem de programação utilizada na construção de páginas na Web.

25. (CESPE –TCE-PA – 2016) Julgue o item que se segue, relativamente a desenvolvimento de sistemas web.

Após a incorporação do jQuery ao HTML5, o desenvolvimento de funcionalidades por meio dessa biblioteca JavaScript ficou limitado a aplicações para dispositivos móveis.

26. (CESPE –TCE-PA – 2016) Julgue o item que se segue, relativamente a desenvolvimento de sistemas web.

O elemento <canvas> do HTML5 especifica uma forma padrão para inserir um vídeo em uma página da Web.

27. (CESPE – FUNPRESP-EXE – 2016) Acerca da tecnologia Java, julgue o próximo item.

Em HTML5, o atributo title da tag <img> pode ser usado para se adicionar um texto fixo a ser sempre apresentado imediatamente acima de uma imagem.



28. (CESPE – FUNPRESP-JUD – 2016) A respeito das tecnologias relacionadas ao desenvolvimento web em Java, julgue o item a seguir.

No HTML 5, a tag `<rp>` é usada para informar o que deve ser exibido nos browsers que não suportam anotações ruby.

29. (CESPE – TRT 8 – 2016) Assinale a opção que apresenta a tag incluída na especificação do HTML5 que permite a reprodução de arquivos que contenham som ou música.

- a) `<phonic>`
- b) `<img>`
- c) `<sound>`
- d) `<music>`
- e) `<audio>`.

30. (CESPE – TRE PI – 2016) A respeito de páginas web desenvolvidas utilizando-se HTML 5, assinale a opção correta.

- a) Para a visualização de vídeos incluídos na página web, é necessária a presença de plug-ins adequados aos formatos de mídia utilizados.
- b) A indefinição dos parâmetros altura e largura dos vídeos pode gerar problemas de renderização.
- c) Para se adicionar vídeos, o uso do atributo `preload` exige a presença do atributo `controls`.
- d) O elemento `<iframe>` permite a inclusão de outra página web na página que esteja sendo construída.
- e) O elemento `<header>` é usado exclusivamente no início de uma página para determinar o seu cabeçalho.

31. (CESPE – TCE-PA – 2016) Julgue o item seguinte no que se refere à construção de formulários eletrônicos.

O elemento `label` funciona como um indicador de caminho a seguir. Muitos browsers renderizam o conteúdo daquela tag como uma área clicável a fim de levar o foco para o campo relacionado.

32. (CESPE – TCE-PA – 2016) Julgue o item seguinte no que se refere à construção de formulários eletrônicos.

É vedada a utilização de `FIELDSET` para agrupar qualquer variedade de elementos input de formulários.

Comentários:



O elemento <fieldset> é um elemento HTML usado para agrupar controles de formulário relacionados. Ele fornece uma maneira de organizar visual e semanticamente os controles de formulário, tornando mais fácil para os usuários entender e usar o formulário. Portanto, como a questão está dizendo o oposto, nosso gabarito é: Errado.

Gabarito: Errado

---

33. (CESPE – TCE-PA – 2016) Julgue o item seguinte no que se refere à construção de formulários eletrônicos.

A tag <label> pode ser aplicada a todos os elementos de formulário, até mesmo a elementos button.

34. (CESPE – TCE-PA – 2016) Julgue o item seguinte no que se refere à construção de formulários eletrônicos.

É possível agrupar inputs de um formulário e, ainda, as opções de uma tag <select> usando-se a tag <fieldset>.





## Questões FCC

35. (FCC – PGE AM – 2022) No cabeçalho de uma página HTML deseja-se indicar ao navegador o conjunto de caracteres recomendado pela linguagem HTML5, que abrange a maioria dos caracteres e símbolos utilizados na maior parte dos idiomas, inclusive acentos existentes no Português. Para isso deve-se utilizar a instrução

- a) `<meta charset="UTF-8">`
- b) `<meta charset="ISO-8959-1">`
- c) `<meta charset="Windows-8859">`
- d) `<meta charset="ASC-II">`
- e) `<meta charset="PT-BR">`

36. (FCC – TRT 23 – 2022) Para criar sites responsivos usando HTML5 é aconselhável fornecer ao navegador instruções sobre como controlar as dimensões e a escala da página por meio da definição da viewport da página, utilizando o comando

- a) `<viewport initial-scale="1.0" max-width="100%" />`
- b) `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
- c) `<viewport content="width=device-width, initial-scale=1.0" />`
- d) `<meta type="viewport" initial-scale="100%" max-width="100%" />`
- e) `<meta type="viewport" screen="width=100%, initial-scale=1.0">`

37. (FCC – TRT22 – 2022) Uma Analista deseja escrever no rodapé da página web HTML5 "Copyright ©", sem aspas, indicando que a página possui direitos autorais.

Uma das maneiras corretas de fazer isso é por meio da instrução

- a) `<p>Copyright &copy;</p>`
- b) `<p>Copyright &szlig;</p>`
- c) `<p>Copyright &circledR;</p>`
- d) `<p>Copyright &copyright;</p>`
- e) `<p>Copyright &cpsymbol;</p>`

38. (FCC – TJ SC – 2021) Em uma situação hipotética, um profissional de TI deseja inserir um vídeo institucional do Tribunal de Justiça de Santa Catarina que está no Youtube, no site do Tribunal.

Para isso, obteve o código personalizado abaixo no site do Youtube.

```
<...1... width="560" height="315" src="https://www.youtube.com/embed/-u33KrrhlpU"
frameborder="0" allow=" accelerometer; autoplay; encrypted-media; gyroscope; picture-in-
picture" allowfullscreen></...1... >
```





Tal fragmento de código permite concluir que o comando HTML5 que deve ser utilizado na lacuna I é

- a) video
- b) movie
- c) media
- d) frame
- e) iframe

39. (FCC – TJ MA – 2019) Um Analista que está desenvolvendo a página de abertura de um site deseja fazer com que as páginas HTML referentes aos links do menu sejam direcionadas e exibidas em um contêiner criado pela tag section no interior desta página. Nesse contêiner, para permitir a abertura das páginas, deve-se utilizar uma tag:

- a) aside
- b) main
- c) dialog
- d) iframe
- e) article

40. (FCC – AFAP – 2019) Em uma página HTML 5 de abertura de um site, um Analista de Informática deseja definir uma área no centro onde outras páginas HTML poderão ser carregadas a partir de cliques nos links do menu principal. Ao abrir a página de abertura, um arquivo HTML já poderá ser exibido nessa área, cujo conteúdo poderá mudar na medida que se clica nos links do menu. Para que seja possível o comportamento descrito, essa área deverá ser definida por meio da tag

- a) div.
- b) section.
- c) iframe.
- d) main.
- e) core.

41. (FCC – Pref Manaus – 2019) Em um parágrafo de uma página web desenvolvida com HTML5, um programador está usando palavras longas e está com medo do navegador quebrar as palavras ou linhas em locais incorretos. Para indicar os locais desejados para possíveis quebras de linha, quando a largura da janela do navegador mudar, o programador deverá usar a tag

- a) </br>.
- b) <break>.
- c) <\n>.



- d) `<wbr>`.
- e) `<br/>`.

42. (FCC – TRF 4 – 2019) HTML5 incluiu um conjunto de tags semânticas para substituir algumas construções que eram feitas com HTML4. Por exemplo, as instruções `<div id="header"> </div>` e `<div id="menu"></div>` feitas em HTML4, são feitas na HTML5, usando as tags semânticas, através das instruções:

- a) `<area id="header"></area>` e `<área id="menu"></área>`
- b) `<header></header>` e `<nav></nav>`
- c) `<head></head>` e `<menu></menu>`
- d) `<header></header>` e `<navigation></navigation>`
- e) `<header></header>` e `<menuarea></menuarea>`

43. (FCC – TJ MA – 2019) Um Técnico Judiciário que utiliza HTML5 está criando um parágrafo em uma página web que tem muitas palavras longas e, para evitar que o navegador quebre linha no lugar errado, utilizou uma tag para especificar onde, em um parágrafo, será aceitável uma quebra de linha. A tag correta para isso é:

- a) `<break>`
- b) `<wbr>`
- c) `<bl>`
- d) `<rt>`
- e) `<br>`

44. (FCC – TRF 3 – 2019) Considere o fragmento abaixo de uma página HTML5 que utiliza elementos semânticos.

```
<figure>
  
</figure>
```

Para inserir a legenda "Figura 1 – Foto das dependências do TRF3" na imagem, após `` deve-se colocar a instrução

- a) `<label>Figura 1 – Foto das dependências do TRF3</label>`
- b) `<title>Figura 1 – Foto das dependências do TRF3</title>`
- c) `<legend>Figura 1 – Foto das dependências do TRF3</legend>`
- d) `<figlabel>Figura 1 – Foto das dependências do TRF3</figlabel>`
- e) `<figcaption>Figura 1 – Foto das dependências do TRF3.</figcaption>`

45. (FCC – Pref Manaus – 2019) Um programador precisa utilizar uma tag semântica incorporada a versão 5 da linguagem HTML, cujo objetivo é definir algum conteúdo



adicional colocado normalmente à direita, mas relacionado com um conteúdo que o circunda colocado normalmente à esquerda. Deverá escolher, nesse caso, a tag:

- a) <article>
- b) <section>
- c) <aside>
- d) <figcaption>
- e) <summary>

46. (FCC – TRF 3 – 2019) Um programador está criando o menu principal da página de abertura de um site, que conterà o bloco principal de links de navegação. Considerando os elementos semânticos da linguagem HTML5, estes links devem ficar imediatamente no interior da tag

- a) <mark>
- b) <progress>
- c) <choose>
- d) <nav>
- e) <navigation>

47. (FCC – SANASA – 2019) Um Desenvolvedor de software precisa inserir uma instrução no cabeçalho de uma página HTML que fará referência a um arquivo chamado a001.css a ser aplicado apenas quando a página for aberta em dispositivos com tela de até 600 pixels. A instrução correta que deverá ser inserida é

- a) <@import URL(a001.css) only screen and (max-width: 600px)>
- b) <link rel="media" device="only screen with (max-width: 600px)" href="a001.css">
- c) <link rel="stylesheet" media="screen and (max-width: 600px)" href="a001.css">
- d) <inport file="a001.css" media="screen and (max-width: 600px)">
- e) <style>@media only screen and (min-width: 600px) URL(a001.css) <style>.

48. (FCC – SABESP – 2018) Um Técnico está desenvolvendo uma página web com HTML5 e deseja exibir um texto com fonte Courier de largura fixa, preservando os espaços e as quebras de linha.

O texto deverá ser colocado entre as tags

- a) <mark> e </mark>
- b) <dl> e </dl>
- c) <embed> e </embed>
- d) <code> e </code>
- e) <pre> e </pre>.



49. (FCC –DPE AM – 2018) Um Técnico Programador deseja fazer um glossário de termos em um site utilizando listas de definições do HTML5. Nestas listas, o termo a ser descrito e a descrição propriamente dita são criados, respectivamente, pelas tags
- a) <dt> e <dd>
  - b) <dd> e <tt>
  - c) <ul> e <li>
  - d) <tt> e <dd>
  - e) <dd> e <dt>

50. (FCC – TST – 2017) Considere, hipoteticamente, que um Programador deseja publicar um vídeo institucional no site do Tribunal Superior do Trabalho. Para isso, no local da página HTML5 onde deseja colocar o vídeo, utilizou o bloco de código abaixo.

```
<video autoplay>  
  <source src="[nome_arquivo.extensão]" type="video/..I..">  
</video>
```

Considere que [nome\_arquivo.extensão] representa o nome de um arquivo de vídeo válido. Nas últimas versões dos principais navegadores, os tipos de arquivos de vídeo válidos para preencher a lacuna I são

- a) avi, mpeg e mov
  - b) mkv, ogg e flv
  - c) mp4, webm e ogg
  - d) wmv, rmvb e mp4
  - e) mp4, avi e mov.
51. (FCC – TRF 5 – 2017) Uma das recomendações que consta nas Web Content Accessibility Guidelines – WCAG é fornecer legendas e outras alternativas para conteúdo multimídia.

Nesse contexto, considere o código abaixo, que disponibiliza um vídeo de orientação ao cidadão em um site do Governo.

```
<video controls>  
  <source src="orienta_cidadao.mp4" type="video/mp4">  
  ..I..  
</video>
```

Para fornecer um arquivo de legenda em português chamado legenda\_br.vtt para o vídeo utiliza-se, na lacuna I, o comando

- a) <embed src="legenda\_br.vtt" kind="track" srclang="pt" label="Português">



- b) `<subtitle src="legenda_br.vtt" kind="text/media" srclang="pt" label="Português">`
- c) `<caption src="legenda_br.vtt" kind="media-query" srclang="pt-br" caption="Português">`
- d) `<caption href="legenda_br.vtt" kind="subtitles" srclang="pt" label="Português">`
- e) `<track src="legenda_br.vtt" kind="subtitles" srclang="pt" label="Português">`

52. (FCC – TRT 24 – 2017) Preocupado com a acessibilidade de um site que está sendo desenvolvido para o Tribunal Regional do Trabalho da 24ª Região, um Técnico recomendou o uso de recursos da linguagem HTML, versão 5, para identificar o idioma principal da página. Para isso deve-se utilizar:

- a) o atributo lang na tag `<html>`.
- b) o atributo lang na tag `<body>`.
- c) o atributo language na tag `<!DOCTYPE>`.
- d) a propriedade language na tag `<meta>`.
- e) a tag `<lang>` no interior da tag `<head>`

53. (FCC – ALMS – 2016) A viewport é a área visível do usuário de uma página web e pode variar de acordo com o dispositivo. HTML5 introduziu um método para deixar os web designers terem controle sobre a viewport através da tag

- a) `<viewport>`
- b) `<meta>`
- c) `<grid>`
- d) `<page>`
- e) `<scale>`

54. (FCC – ELETROSUL – 2016) A uma página HTML5 deseja-se vincular um arquivo chamado `arq1.css` e um arquivo chamado `arq2.js`.

Para isso devem-se utilizar as instruções

- a) `<link rel="stylesheet" type="file/css" src="arq1.css">` e `<script src="arq2.js">` `</script>` no corpo da página.
- b) `<import type="text/css" href="arq1.css">` e `<script file="arq2.js">` `</script>` no cabeçalho da página.
- c) `<link type="css" style="external" href="arq1.css">` e `<script src="arq2.js">` `</script>` no cabeçalho da página.
- d) `<include type="text/css" href="arq1.css">` e `<script src="arq2.js">` `</script>` no corpo da página.
- e) `<link rel="stylesheet" type="text/css" href="arq1.css">` e `<script src="arq2.js">` `</script>` no cabeçalho da página.

55. (FCC – ALMS – 2016) Considere o fragmento de código HTML5 a seguir.



```
<body>
  <audio controls>
    I
    ...
  </audio>
</body>
```

Para disponibilizar na página o áudio chamado beethoven.mp3 a lacuna I deve ser corretamente preenchida por

- `<sound src="beethoven.mp3" type="mpeg/ogg/wav" play="on">`
- `<source target="beethoven.mp3" media_type="sound/mpeg">`
- `<source src="beethoven.mp3" type="audio/mpeg">`
- `<sound source="beethoven.mp3" media="audio/mp3">`
- `<source src="beethoven.mp3" media="sound/mpeg" play="on">`

56. (FCC – SEMF Teresina – 2016) Em uma página HTML há um parágrafo vazio criado pela tag `<p id="local"></p>`. Em um bloco JavaScript da mesma página, para inserir neste parágrafo a palavra Teresina, utiliza-se o comando

- `document.getElementById("local").innerHTML = "Teresina";`
- `document.getElementById("p#demo").innerHTML = "Teresina";`
- `document.p["#local"].value = "Teresina";`
- `document.demo.value = "Teresina";`
- `document.getElementsByName("local").innerHTML = "Teresina";`

57. (FCC – TRT 23 – 2016) De acordo com as recomendações da W3C para a linguagem HTML5, a instrução

- `<meta charset="utf-8" />` está incorreta, pois tem uma barra no final.
- `<!doctype html>` está incorreta, pois deveria estar em letra maiúscula.
- `<table class=table striped>` está correta.
- `<html lang="en-US">` está correta.
- `<script src="myscript.js">` está incorreta, pois faltou o atributo `language="javascript"`

58. (FCC –TRF3 – 2016) Em um site que utiliza HTML5 Application Cache há um arquivo chamado dados.appcache que define que uma página poderá ter o conteúdo acessado mesmo estando offline. Nessa página, para apontar para esse arquivo deve-se utilizar a instrução

- `<html manifest="dados.appcache">`
- `<cache manifest="dados.appcache">`
- `<html src="dados.inf" cached>`



d) `<cache src="dados.inf">`

e) `<!MANIFEST cache="demo.appcache">`



## Questões FGV

59. (FGV –TJ TO– 2022) No HTML5, o evento que é disparado quando um objeto é carregado é:
- a) begin;
  - b) load;
  - c) ready;
  - d) run;
  - e) start.
60. (FGV –MPE SC– 2022) Sobre elementos block-level (nível de bloco) e elementos inline (em linha) no HTML, analise as afirmativas a seguir.
- I. Um elemento block-level sempre começa numa nova linha.
  - II. <p> é um elemento inline.
  - III. <div> é um elemento block-level.
- Está correto somente o que se afirma em:
- a) I;
  - b) II;
  - c) I e II;
  - d) I e III;
  - e) II e III..
61. (FGV –MPE GO – 2022) Num documento HTML5, assinale o elemento que contém meta informações sobre a página.
- a) <!DOCTYPE html>
  - b) <body> (
  - c) <head>
  - d) <html>
  - e) <script>.

62. (FGV –TJ TO– 2022) Observe o código HTML e JavaScript a seguir:





```
<!DOCTYPE html>
<html>
<body>
  <p id="Tejota"></p>
  <script>
    const tjProcesso = {ID:"2022.1", envolvidos:
      ["João", "Maria"]};
    document.getElementById("Tejota").innerHTML
      = tjProcesso.envolvidos[0];
  </script>
</body>
</html>
```

Após a execução do código, o resultado exibido será:

- a) Maria
- b) João
- c) 2022.1
- d) Tejota
- e) tjProcesso

63. (FGV –Sefaz AM – 2022) A linguagem de marcação HTML é amplamente utilizada na construção de páginas da Internet.

O elemento HTML, que permite reproduzir um arquivo de vídeo no formato MP4 em uma página da web, é

- a) <clip>
- b) <embedded>
- c) <media>
- d) <movie>
- e) <video>.

64. (FGV –IMBEL – 2021) Analise a imagem a seguir, produzida numa página Web.

A	C
B	F/G

```
<style>
  th { border: 1px solid black; }
</style>
<table>
...
</table>
```



que substitui corretamente a linha com pontos.

a) `<tr><th>A</th><th>C</th></tr>`

`<tr><td>F</td><td><table>`

`<tr><td>G</td><td>G</td></tr>`

`</table></th></tr>`

b) `<tr><th>A</th><th>C</th></tr>`

`<tr><td>B</td><td><table>`

`<tr><th>F</th><th>G</th></tr>`

`</table></th></tr>`

c) `<tr><th>A</th><th>C</th></tr>`

`<tr><td>B</td><td>`

`<tr><td>F</td><td>G</td></tr>`

`</th></tr>`

d) `<tr><th>A</th><th>C</th></tr>`

`<tr><td>B</td><td></table><tr><td>F</td><td>G</td></tr>`

`<table></th></tr>`

e) `<tr><td>A</td><td>C</td></tr>`

`<tr><td>B</td><td><table>`

`<tr><td>F</td><td>G</td></tr>`

`</table></th></tr>`.

65. (FGV – TJ RO– 2021) No contexto do HTML, a sintaxe correta para um comentário, delimitando um trecho que NÃO deve ser exibido pelo browser, é:

a) `<-- Texto do comentário -->`

b) `!--Texto do comentário >`

c) `<-Texto do comentário ->`

d) `<!--Texto do comentário -->`

e) `<--Texto do comentário --/>`.

66. (FGV – IMBEL– 2021) Analise o trecho HTML a seguir.

`<table>`

`<tr><th>A</th><th>C</th></tr>`

`<tr><th>B</th><th>D</th></tr>`

`</table>`

Na exibição da página Web, esse script produz uma tabela

a) com duas linhas e duas colunas.



- b) com quatro linhas e uma coluna.
- c) com quatro colunas e uma linha.
- d) com uma linha e uma coluna.
- e) com linhas e colunas desalinhadas.

67. (FGV –MPE AL – 2018) O HTML 5 introduziu diversos elementos semânticos, gráficos e de multimídia.

Assinale o elemento que não é uma novidade nessa versão.

- a) <article>
- b) <div>
- c) <footer>
- d) <header>
- e) <section>.

68. (FGV –ALERO – 2018) No contexto dos elementos introduzidos pelo HTML 5, analise as afirmativas a seguir.

- I. Os elementos "header", "section" e "footer" são elementos block-level.
- II. O elemento "<wbr>" é funcionalmente idêntico ao já conhecido elemento "<br>".
- III. Além dos atributos height e width, o elemento "<canvas>" permite o uso do atributo depth, para uso em gráficos tridimensionais.

Está correto o que se afirma em

- a) I, somente.
- b) II, somente.
- c) III, somente.
- d) I e II, somente.
- e) II e III, somente..

69. (FGV –ALERO – 2018) Assinale a opção que melhor descreve a utilização da tag canvas no HTML5.

- a) Definir mídias de áudio e vídeo.
- b) Desenhar elementos gráficos por meio de scripts.
- c) Dividir a tela em áreas horizontais de largura variável, que funcionam como painéis.
- d) Dividir a tela em áreas verticais de altura variável, que funcionam como painéis.
- e) Desenhar a entrada de dados e as caixas de diálogo..



70. (FGV –MPE AL – 2018) Observe o código a seguir.

```
<button onmouseover="this.innerHTML='ON' "  
onmouseout="this.innerHTML='OFF' ">Aqui</button>
```

Supondo que o cursor do mouse inicialmente esteja fora da área de botão, assinale o texto exibido no botão quando a página que o contém estiver carregada.

- a) "Abrir" inicialmente, "on" quando o cursor do mouse entra na área do botão, e "off" quando o cursor do mouse sai da área do botão, assim mudando a cada movimentação para dentro ou para fora.
- b) "Abrir" inicialmente, "on" quando o cursor do mouse entra na área do botão, e assim permanece.
- c) "Abrir" inicialmente, "on" quando o cursor do mouse entra na área do botão, e "off" quando o cursor do mouse sai da área do botão, e assim permanece mesmo que o mouse seja movimentado sobre o botão.
- d) "Abrir" inicialmente, "on" quando o cursor do mouse entra na área do botão, e "Abrir" quando o cursor do mouse sai da área do botão, assim mudando a cada movimentação para dentro ou para fora.
- e) "Abrir" inicialmente, "on" quando o cursor do mouse entra na área do botão, e "off" quando o cursor do mouse sai da área do botão pela primeira vez. Nas demais saídas, mudar para "Abrir".

---

71. (FGV –ALERO – 2018) Analise o código a seguir.



```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"
></script>
<script>
$(document).ready(function( )
{
$("button").click(function( )
{
$(this)[0].innerText
= parseInt($(this)[0].innerText) + 1;
});
});
</script>
</head>
<body>
<button id="xpto">1</button>
</body>
</html>
```

Sabe-se que com essa página exibida, o operador clicou no botão por três vezes. Assinale o valor exibido no botão após esses cliques.

- a) 0
- b) 1
- c) 1111
- d) 1234
- e) 4.

72. (FGV – BANESTES– 2018) HTML é uma linguagem utilizada para construção de páginas na Internet. O comando de formatação da linguagem HTML para indicar que um determinado texto deve ser apresentado sublinhado é:

- a) <b></b>
- b) <s></s>
- c) <i></i>
- d) <p></p>
- e) <u></u>.

73. (FGV –BANESTES – 2018) Ao desenvolver uma aplicação cliente Web, o programador precisa mostrar um aviso quando a página requisitada tem o acesso proibido.



Assim, ele tem que tratar o erro de resposta HTTP:

- a) 200;
- b) 403;
- c) 404;
- d) 500;
- e) 501.

---

74. (FGV –IBGE – 2017) HTML5 (Hypertext Markup Language, versão 5) é uma linguagem utilizada para estruturação e apresentação de conteúdo na Internet.

Analise a página HTML5 a seguir:

```
<!DOCTYPE html>
<html>
<head><script>
function clickCounter() {
  var A = localStorage.clickcount;
  var B = sessionStorage.clickcount;
  if (A) {A = Number(A)+1;} else {A= 1;}
  if (B) {B = Number(B) + 1;} else {B= 1;}

  document.getElementById("result").innerHTML =
    "Você apertou " + A + B + " vez(es).";
  sessionStorage.clickcount = B;
  localStorage.clickcount = A;
}
</script></head>
<body>

<p><button id="B" onclick="clickCounter()" type="button">
Aperte aqui!</button></p>
<div id="result"></div>
</body></html>
```

Considere que você está utilizando um navegador web que suporta HTML5 Local Storage e ao acessar essa página pela primeira vez você apertou três vezes o botão indicado com id="B" e então você fechou o navegador.

Posteriormente, você utiliza o mesmo navegador para acessar novamente essa mesma página.

Após apertar duas vezes o mesmo botão, será apresentada a seguinte mensagem no corpo da página:

- a) Você apertou 5 vezes.
- b) Você apertou 7 vezes.
- c) Você apertou 10 vezes.
- d) Você apertou 52 vezes.



e) Você apertou 55 vezes.

75. (FGV –ALERJ – 2017) Analise a estrutura básica da página HTML a seguir:

```
<html>
  <head>
    <title>Título da página</title>
  </head>
  <body bgcolor="black"
    alink="green"
    vlink="yellow"
    link="red"
    text="blue">
```

Corpo do documento.

```
  <a href="#">Click aqui!</a>
</body>
</html>
```

Esse código define que a cor do texto da página e dos links quando clicados são, respectivamente:

- a) preta e vermelha;
- b) azul e amarela;
- c) vermelha e verde;
- d) azul e verde;
- e) preta e amarela.

76. (FGV –MPE BA – 2017) A HTML é uma linguagem de marcação utilizada na construção de páginas na Web.

O comando de formatação da linguagem para definir um hyperlink indicando que o recurso linkado deve ser aberto em uma nova janela do navegador é:

- a) <a href = "url" target = "\_top">meu link</a>
- b) <a href = "url".new>meu link</a>
- c) <a href = "url" target="\_blank">meu link</a>
- d) <a href = "url" target = "\_parent">meu link</a>
- e) <a href = "url" target = "\_self">meu link</a>.



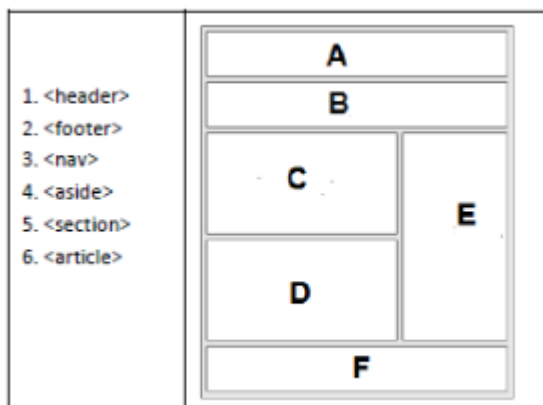
77. (FGV –IBGE – 2017) O HTML 5 introduziu um método para permitir que o designer controle a área de visualização de um site através da tag <meta>. A sintaxe correta para que a largura de uma página siga a largura da tela de um dispositivo, com uma escala inicial de 1X no momento em que ela é carregada é:

- a) <meta name="viewport" content="device, scale=1.0">;
- b) <meta type="viewport" width="device" scale="1.0">;
- c) <meta name="viewport" width="device-width, initial-scale=1.0">;
- d) <meta name="viewport" content="width=device-width, initial-scale=1.0">;
- e) <meta type="viewport" content="width=device-width" scale="initial-scale-1">.

78. (FGV – IBGE– 2016) A sigla HTML significa Hyper Text Markup Language, o que pode ser traduzido como Linguagem de Marcação de Hipertexto. Uma linguagem de marcação pode ser definida como um sistema para:

- a) marcar um documento indicando a sua estrutura lógica e hierárquica especificamente para a transmissão e exibição eletrônicas;
- b) definir o comportamento visual em meio eletrônico do conteúdo textual de um documento, incluindo tipografia, cor e tamanho dos caracteres;
- c) marcar um documento indicando a ordem em que o conteúdo deve ser apresentado em meio eletrônico;
- d) definir as ligações entre diferentes documentos a partir de palavras-chave específicas;
- e) criar documentos específicos para transmissão eletrônica através da Internet.

79. (FGV –IBGE – 2016) Um desenvolvedor Web mobile pretende utilizar os novos elementos semânticos disponíveis no HTML5 em suas páginas. Associe corretamente os elementos HTML5 enumerados com o posicionamento na ilustração que representa conceitualmente as partes de uma página HTML:



A sequência correta é:





- a) A=1, B=3, C=5, D=6, E=4 e F=2;
- b) A=2, B=5, C=4, D=3, E=6 e F=1;
- c) A=1, B=4, C=6, D=5, E=3 e F=2;
- d) A=2, B=3, C=6, D=4, E=5 e F=3;
- e) A=1, B=6, C=4, D=3, E=5 e F=2.

---

80. (FGV – IBGE– 2016) Com a introdução do HTML5, diversas novas APIs Javascript (Application Programming Interfaces) foram disponibilizadas, aumentando consideravelmente a quantidade de recursos disponíveis para a produção de páginas web. São APIs exclusivas do HTML5:

- a) múltiplas colunas de texto, transformações 2D/3D e RWD (Responsive Web Design);
- b) armazenamento em nuvem, suporte a telas de toque e SSL (Secure Sockets Layer);
- c) acesso a câmeras em dispositivos móveis, suporte a streaming de vídeo e SSE (Streaming SIMD Extensions);
- d) armazenamento local, geolocalização e SSE (Server-Sent Events);
- e) redimensionamento dinâmico de imagens, detecção de resolução de tela e RWD (Responsive Web Display)

---

81. (FGV –IBGE – 2016) A declaração <!DOCTYPE> permite ao navegador apresentar uma página web corretamente. A declaração correta para uma página em HTML5 é:

- a) <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 5.0 Strict//EN" "http://www.w3.org/TR/html5/strict.dtd">
- b) <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 5.0 Transitional//EN" "http://www.w3.org/TR/html5/loose.dtd">
- c) <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 5.0">
- d) <!DOCTYPE html>
- e) <!DOCTYPE XML PUBLIC "-//W3C//DTD HTML 5.0">



## GABARITO

- |             |             |             |
|-------------|-------------|-------------|
| 1. Errado   | 28. Correto | 55. Letra C |
| 2. Correto  | 29. Letra E | 56. Letra A |
| 3. Errado   | 30. Letra D | 57. Letra D |
| 4. Letra D  | 31. Correto | 58. Letra A |
| 5. Errado   | 32. Errado  | 59. Letra B |
| 6. Errado   | 33. Errado  | 60. Letra D |
| 7. Correto  | 34. Correto | 61. Letra C |
| 8. Letra B  | 35. Letra A | 62. Letra B |
| 9. Correto  | 36. Letra B | 63. Letra E |
| 10. Correto | 37. Letra A | 64. Letra B |
| 11. Correto | 38. Letra E | 65. Letra D |
| 12. Correto | 39. Letra D | 66. Letra A |
| 13. Letra B | 40. Letra C | 67. Letra B |
| 14. Letra D | 41. Letra D | 68. Letra A |
| 15. Correto | 42. Letra B | 69. Letra B |
| 16. Letra D | 43. Letra B | 70. Letra A |
| 17. Correto | 44. Letra E | 71. Letra E |
| 18. Correto | 45. Letra C | 72. Letra E |
| 19. Errado  | 46. Letra D | 73. Letra B |
| 20. Correto | 47. Letra C | 74. Letra D |
| 21. Errado  | 48. Letra E | 75. Letra D |
| 22. Errado  | 49. Letra A | 76. Letra C |
| 23. Letra B | 50. Letra C | 77. Letra D |
| 24. Errado  | 51. Letra E | 78. Letra A |
| 25. Errado  | 52. Letra A | 79. Letra A |
| 26. Errado  | 53. Letra B | 80. Letra D |
| 27. Errado  | 54. Letra E | 81. Letra D |



82.





# ESSA LEI TODO MUNDO CONHECE: PIRATARIA É CRIME.

Mas é sempre bom revisar o porquê e como você pode ser prejudicado com essa prática.



**1** Professor investe seu tempo para elaborar os cursos e o site os coloca à venda.



**2** Pirata divulga ilicitamente (grupos de rateio), utilizando-se do anonimato, nomes falsos ou laranjas (geralmente o pirata se anuncia como formador de "grupos solidários" de rateio que não visam lucro).



**3** Pirata cria alunos fake praticando falsidade ideológica, comprando cursos do site em nome de pessoas aleatórias (usando nome, CPF, endereço e telefone de terceiros sem autorização).



**4** Pirata compra, muitas vezes, clonando cartões de crédito (por vezes o sistema anti-fraude não consegue identificar o golpe a tempo).



**5** Pirata fere os Termos de Uso, adultera as aulas e retira a identificação dos arquivos PDF (justamente porque a atividade é ilegal e ele não quer que seus fakes sejam identificados).



**6** Pirata revende as aulas protegidas por direitos autorais, praticando concorrência desleal e em flagrante desrespeito à Lei de Direitos Autorais (Lei 9.610/98).



**7** Concurseiro(a) desinformado participa de rateio, achando que nada disso está acontecendo e esperando se tornar servidor público para exigir o cumprimento das leis.



**8** O professor que elaborou o curso não ganha nada, o site não recebe nada, e a pessoa que praticou todos os ilícitos anteriores (pirata) fica com o lucro.



Deixando de lado esse mar de sujeira, aproveitamos para agradecer a todos que adquirem os cursos honestamente e permitem que o site continue existindo.