

## **Aula 00**

*SEFAZ-AM (Analista de Tecnologia da  
Informação) Desenvolvimento de  
Software*

Autor:  
**Diego Carvalho**

06 de Julho de 2024

# Índice

1) JavaScript DEV - Teoria .....	3
2) JavaScript DEV- Questões Comentadas .....	14
3) JavaScript DEV - Lista de Questões .....	39
4) JavaScript Moderno (Ecma 2021) - Teoria .....	54
5) JavaScript Moderno (Ecma 2021) - Questões Comentadas .....	58
6) JavaScript Moderno (Ecma 2021) - Lista de Questões .....	63
7) JQuery - Teoria .....	68
8) JQuery - Questões Comentadas .....	73
9) JQuery - Lista de Questões .....	95
10) AJAX - Teoria .....	107
11) AJAX - Questões Comentadas .....	112
12) AJAX - Lista de Questões .....	129
13) Single Page Application (SPA) - Teoria .....	141



# JAVASCRIPT

## Conceitos Básicos

INCIDÊNCIA EM PROVA: ALTÍSSIMA

A tecnologia Javascript nasceu a partir de um contexto maior, logo inicialmente eu vou precisar fazer uma grande contextualização, para que o assunto faça sentido para vocês. A linguagem de marcação HTML, surgiu na década de 1990 e destina-se a estruturar uma página web, não sendo empregada para adicionar estilos ou apresentação visual aos elementos que constituem a página, sendo tais tarefas função das folhas de estilo em cascata.

**Ela também não possui funcionalidades que permitam adicionar interatividade avançada à página, sendo tal tarefa função das linguagens de programação.** Então grave o seguinte:

*A linguagem HTML não é empregada para adicionar estilos ou apresentação visual aos elementos que constituem a página, sendo tais tarefas função das folhas de estilo em cascata. **Ela também não possui funcionalidades que permitam adicionar interatividade avançada à página, sendo tal tarefa função das linguagens de programação.***

**PROFESSOR, MAS A AULA NÃO É SOBRE JAVASCRIPT, ECMA, TYPESCRIPT, JQUERY E AJAX?**



**Calma jovem, lembre-se que estamos contextualizando. Como assim, professor? O Javascript foi criado pela Netscape em parceria com a Sun Microsystems, com a finalidade de fornecer um**



**meio de adicionar interatividade a uma página web.** A primeira versão, denominada JavaScript 1.0, foi lançada em 1995 e implementada em março de 1996 no navegador Netscape Navigator 2.0 quando o mercado era dominado pela Netscape.

Logo a seguir, veio a época da chamada guerra dos browsers, cujos efeitos nocivos se fazem sentir até os dias atuais. Para não fugir à regra, a Microsoft, em resposta à Netscape, criou a linguagem JScript baseada em Visual Basic cuja primeira versão denominada JScript 1.0 e foi lançada com o navegador Internet Explorer 3.0. **Não há como fazer funcionar um formulário HTML com o uso de elementos HTML.**

A HTML limita-se a criar os rótulos e campos de um formulário para serem preenchidos pelo usuário e nada mais. Com HTML, não conseguimos processar os dados nem mesmo enviá-los ao servidor ou a outra máquina qualquer. **Para cumprir essas tarefas, é necessário utilizar um programa que consiga manipular e processar os dados. Entendido?**

Um desenvolvedor web moderno deve saber três linguagens de programação: HTML, para definir o conteúdo das páginas; CSS para especificar o layout das páginas; e Javascript para programar o comportamento das páginas. **Trata-se de uma linguagem extremamente abrangente, funcionando em computadores, servidores, laptops, tablets, smartphones, entre outros<sup>1</sup>.**

Existe um conceito muito importante chamado Document Object Model (DOM), que é uma especificação W3C para representar e interagir com objetos em HTML, XHTML e XML. **Pois é, a primeira grande vantagem do Javascript é a sua capacidade de manipular o DOM de Páginas HTML por meio de diversos métodos.** Ele torna possível criar, alterar, deletar e copiar vários elementos HTML.

*Como ele faz isso? Ele é capaz de modificar valores de atributos de um Documento HTML.* Ademais, ele é capaz de modificar Folhas de Estilos do CSS. *Incrível, não?* Ele também é bastante utilizado para validar entradas de dados do usuário. *Como assim, professor?* Eventualmente, um formulário pede sua Data de Nascimento! Ele é capaz de receber seus dados e verificar se está no formato correto (dd/mm/aaaa).



**Podemos dizer que HTML é para conteúdo; CSS é para apresentação; e Javascript é para Interatividade!** *Professor, ouvi dizer que Javascript faz parte da Plataforma Java! Ouvi errado!* O primeiro é uma linguagem de scripts de propósito específico, dinamicamente tipada, baseada em protótipos, com escopo de função, que não precisa ser compilada. O segundo é uma linguagem de programação de propósito geral, estaticamente tipada, baseada em classes, com escopo de bloco, que precisa ser compilada.

<sup>1</sup> Javascript é a linguagem de script padrão de todos os navegadores modernos e do HTML5. Apesar disso, ela não é utilizada apenas em Páginas HTML.

## IMPORTANTE

JAVASCRIPT É UMA LINGUAGEM DE USO ESPECÍFICO! ELA FOI DESENVOLVIDA PARA APRIMORAR A EXPERIÊNCIA DOS USUÁRIOS NOS NAVEGADORES. ELA NUNCA FOI UMA LINGUAGEM DE PROPÓSITO GERAL (EX: JAVA, CAPAZ DE CRIAR APLICAÇÕES INTEIRAS). NO ENTANTO, RECENTEMENTE O JAVASCRIPT TEM SE ENVEREDADO PARA ESSE PROPÓSITO, POR MEIO DAS JSLIBS. PERCEBA: "JSLIBS IS A STANDALONE JAVASCRIPT DEVELOPMENT RUNTIME ENVIRONMENT FOR USING JAVASCRIPT AS A GENERAL-PURPOSE SCRIPTING LANGUAGE"!

**Há uma frase em inglês que eu acredito que defina bem a diferença entre ambas as linguagens: "Java and Javascript are as similar as Car and Carpet".** Isso quer dizer que as linguagens são completamente diferentes e uma das poucas coisas em comum é o prefixo do seu nome (Java), e só! *Captaram a mensagem?* Pois é, não caiam em pegadinha de banca alguma agora.

JavaScript é uma linguagem client-side (desenvolvida para rodar no lado do cliente), i.e., a interpretação e o funcionamento da linguagem dependem de funcionalidades hospedadas no navegador do usuário (ele contém um interpretador JavaScript). **Com o JavaScript, pode-se manipular conteúdo e apresentação; manipular o navegador; interagir com formulários; e interagir com outras linguagens dinâmicas.**

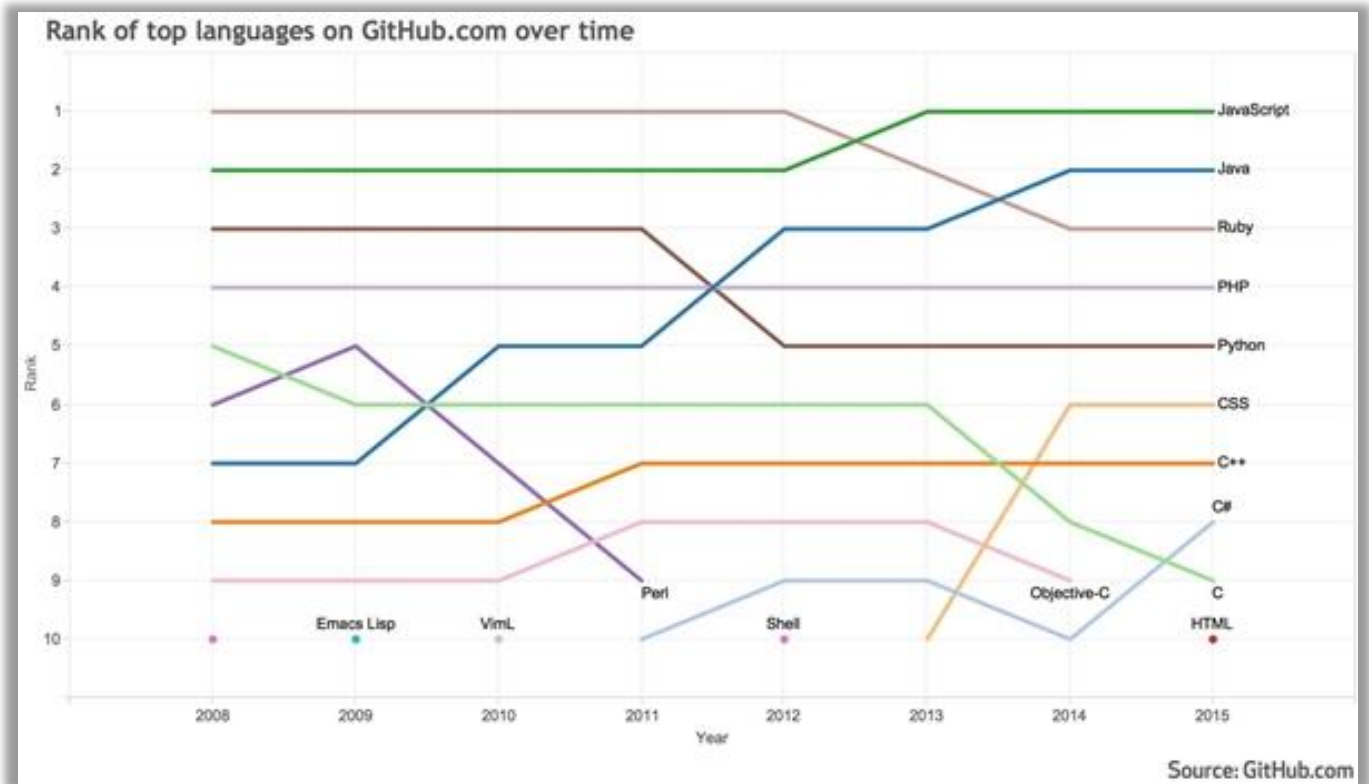
Por fim, um detalhe interessante: JavaScript é baseado em ECMAScript. *Professor, o que é ECMAScript?* É uma especificação para linguagens de scripts que é implementada pelo JavaScript, entre outras linguagens (vamos ver mais sobre ela ainda nessa aula). Ela se encontra na Versão 2021 e **trouxe a maior mudança para a linguagem JavaScript desde a sua criação, há cerca de 20 anos.**

**O principal objetivo da nova versão especificação foi tornar a linguagem mais flexível, enxuta e fácil de se aprender e trabalhar,** tornando-a mais próxima a outras linguagens orientadas a objeto, como Java e Python. Dentre as principais mudanças, temos: criação de novos tipos de dados (Map, WeakMap, Set, WeakSet); novas maneiras de iterar objetos e coleções; declaração de variáveis com `let` e `const`.

Além dessas, temos: modularização e estrutura de classes; geradores e símbolos; operadores `rest` e `spread`. Nos últimos anos, o JavaScript ganhou muita força na comunidade de desenvolvedores através do "grande boom" do ecossistema das tecnologias Node.js e npm. **Desde então, a flexibilidade da linguagem tem sido usada a favor não somente do lado do cliente, mas também do lado do servidor.**

Galera, isso era algo impensável até alguns anos atrás. **Tudo isso fez com que não somente a comunidade, mas também grandes empresas de tecnologia como o Facebook, reconhecessem a força da linguagem e finalmente a levasse a sério.** Esse interesse na linguagem se refletiu na criação de uma imensa diversidade de bibliotecas e frameworks, como: ReactJS, Angular2, Meteor, Electron, etc;





## JS: ESCOPO

Para escrever código Javascript, não há necessidade de instalar um software especial. Escrevemos com um editor de textos simples e visualizamos o resultado em um navegador. **Arquivos escritos com o uso dessa linguagem devem ser separados com a extensão *.js* e são criados para serem executados dentro de um Arquivo HTML.** O código deve sempre vir entre as tags `<script>` e `</script>`:

```
<script>
function minhaFuncao() {
    document.getElementById("demo").innerHTML = "Minha primeira função JavaScript!";
}
</script>
```

Existem três maneiras de inserir Javascript em HTML:

**Externo:** escrever o script em um arquivo externo com extensão `.js` e inserir com um link na tag `<script>` dentro da seção `<body>` ou `<head>` do documento

```
<!DOCTYPE html>

<html>
<body>
    <script src="meuScript.js"></script>
</body>
</html>
```



- **Incorporado:** também conhecida como interna, insere-se o script na seção <head> do documento.

```
<!DOCTYPE html>

<html>
<head>
  <script>
    function myFunction() {
      Document.getElementById("demo").innerHTML = "Parágrafo modificado.";
    }
  </script>
</head>

<body>

  <h1>A Web Page</h1>
  <p id="demo">Um Parágrafo</p>
  <button type="button" onclick="myFunction()">Experimente!</button>

</body>
</html>
```

- **Inline:** insere-se o script diretamente na seção <body> do documento. Não é recomendada, pelo princípio da separação das camadas.

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

<h1>Uma Página Web</h1>
<p id="demo">Um Parágrafo</p>
<button type="button" onclick="myFunction()">Experimente!</button>

<script>
function myFunction() {
  document.getElementById("demo").innerHTML = "Parágrafo Modificado.";
}
</script>

</body>
</html>
```

**As modernas práticas de desenvolvimento preconizam o uso de scripts externos a serem linkados ao documento.** Cria-se um ou mais arquivos contendo os scripts e gravados com a extensão .js e usa-se o elemento script na seção head do documento para linká-lo à página. Essa técnica permite linkar o script a várias páginas do site, facilitando a manutenção.

**Inserir script inline é uma prática que pertence ao passado e deve ser evitada pelo desenvolvedor comprometido com os Padrões Web.** Ao escrevermos scripts dentro da marcação



HTML, estaremos misturando as camadas de marcação e comportamento, dificultando a manutenção e o entendimento dos códigos. *Entendido até agora?*

### JS: CASE SENSITIVE E COMENTÁRIOS

O Javascript é sensível ao tamanho da caixa (Case-Sensitive). Portanto, **os nomes de variáveis, funções e demais identificadores são diferenciados quando escritos com letras maiúsculas ou minúsculas**. Por exemplo: as variáveis *total*, *Total*, *toTal* e *TOTAL* são diferentes. O método `write()` deve ser escrito em minúscula, pois escrever `Write()` ou `WRITE()` causará um erro no script.

**Comentários são pequenos textos que o desenvolvedor insere ao longo do script com a finalidade de facilitar o entendimento e a manutenção do código.** A linguagem Javascript admite dois tipos de marcadores para comentários: comentários de linha única e comentários de múltiplas linhas. Abaixo, podemos ver como funcionam cada uma.

Comentários de linha única:

```
// Modifica o cabeçalho:  
document.getElementById("meuHeader").innerHTML = "Minha primeira página";  
// Modifica o Parágrafo:  
document.getElementById("myP").innerHTML = "Meu primeiro parágrafo.";
```

Comentários de múltiplas linhas:

```
/*  
O código abaixo modificará  
o cabeçalho com id = "meuHeader"  
e o parágrafo com id = "meuParagrafo"  
em minha página web:  
*/  
  
document.getElementById("meuHeader").innerHTML = "Minha primeira página";  
document.getElementById("meuParagrafo").innerHTML = "Meu primeiro parágrafo.";
```

### JS: DECLARAÇÕES

Um script consiste em uma série de instruções escritas segundo uma sintaxe própria e rígida. As instruções determinam a realização de tarefas com a finalidade de obter um resultado final. **Cada uma das instruções de um script constitui uma declaração independente e existem duas sintaxes para separar as declarações.** Separe-as com ponto-e-vírgula ou coloque cada declaração em uma linha separada.

```
a = 5;  
b = 6;  
c = a + b;
```

É o mesmo que...

```
a = 5; b = 6; c = a + b;
```





A sintaxe determina que o ponto-e-vírgula seja obrigatório para separar declarações em uma mesma linha e facultativo para separar declarações em linhas diferentes, contudo é uma boa prática usar ponto-e-vírgula para separar declarações em linhas diferentes. **Estou passando rapidamente por esses aspectos, porque isso não é nada diferente da maioria das linguagens de programação.**

#### JS: ESPAÇO EM BRANCO E QUEBRAS DE LINHA

**Quebras de linhas e espaços em branco, quando inseridos entre nomes de variáveis, nomes de funções, números e entidades similares da linguagem, são ignorados na sintaxe Javascript.** Contudo, para strings e expressões regulares, tais espaçamentos são considerados. No exemplo abaixo, ignoram-se os espaços extras fora da string! Se for dentro da string, não se ignoram os espaços.

```
var person = "Diego";  
var person="Diego";
```

Nesse exemplo, quebra-se a linha de código dentro de uma string:

```
document.getElementById("demo".innerHTML = "Oi \\  
Diego!");
```

**Na terminologia Javascript, a palavra literal designa qualquer dado – valor fixo (não variável) – que se insere no código do script.** Podem ser de diversos tipos, tais como: inteiros (Ex: 250), decimais (Ex: 3.1415), booleanos (Ex: true e false), strings<sup>2</sup> (Ex: "Edson Arantes do Nascimento"), arrays (Ex: frutas = ["laranja", "pera", "goiaba", "morango"]) e objetos (Ex: Carro).

#### JS: VETORES (ARRAYS)

Os literais arrays, na sintaxe Javascript, são os conjuntos de zero ou mais valores, separados por vírgula e envolvidos por colchetes ([ ]). Os valores contidos em um array recebem um índice sequencial começando com zero. A sintaxe é:

```
var nome-array = [item1, item2, ...]
```

Um exemplo concreto seria:

```
var carros = ["Volks", "Ford", "BMW"];
```

Mas essa não é a única maneira de se criar um vetor, podemos também utilizar o operador *new*:

```
var carros = new Array("Volks", "Ford", "BMW");
```

<sup>2</sup> Strings podem vir com aspas simples ou duplas.



## JS: SAÍDA DE DADOS

Professor, como eu faço para manipular os elementos HTML? Para acessar, devemos utilizar o método `document.getElementById(id)`. **O atributo `id` deve ser usado para identificar o elemento HTML e `innerHTML` deve ser usado para se referir ao conteúdo do elemento.** Observem a tag `<script>` no exemplo abaixo! Ela busca pelo elemento HTML cujo `id="demo"`.

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

<h1>Minha primeira Página Web</h1>
<p>Meu primeiro parágrafo</p>

<p id="demo"></p>

<script>
    document.getElementById("demo").innerHTML = "Parágrafo modificado";
</script>

</body>
</html>
```

Trata-se do elemento `<p>`. Ele pega o conteúdo interno desse elemento e modifica o valor de seu atributo `id` para "*Parágrafo modificado*". É possível também escrever diretamente no Documento HTML por meio do método `write()`. **É recomendável utilizá-lo apenas para realizar testes, na medida em que ele sobrescreve todos os elementos HTML.** *Bacana?*

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

<h1>Minha primeira Página Web</h1>
<p>Meu primeiro parágrafo.</p>

<script>
    document.write("Meu segundo parágrafo");
</script>

</body>
</html>
```

## JS: EVENTOS

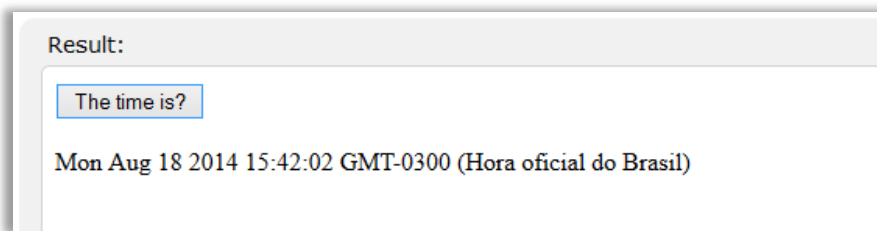
O Javascript é bastante utilizado com eventos. *Como assim, professor?* **Um evento é uma ação capaz de disparar uma reação.** Clicar em um link ou colocar o ponteiro do mouse sobre um elemento podem ser considerados eventos. Ao clicar em um link, desencadeamos uma reação que poderá ser a de abrir uma nova página. Ao colocar o ponteiro do mouse sobre um elemento, podemos alterar sua opacidade.



Eventos são bastante utilizados em Javascript e viabilizam a interatividade em uma página web. Na verdade, eventos viabilizam a própria existência do Javascript. Sem eles, não teríamos como fazer funcionar os scripts. Há uma gama imensa de possibilidades de eventos<sup>3</sup>! **Ele permite a execução de funções quando eventos são detectados. Vamos ver como ele funciona:**

```
<button onclick='getElementById("demo").innerHTML=Date()>The Time is?</button>
```

No exemplo acima, quando o usuário clicar no botão, será informada a data e hora atual. O resultado é mostrado abaixo:



Observem que o Javascript modifica o conteúdo do elemento com *id="demo"*.

**Por fim, vamos falar rapidamente do Unobstrusive JavaScript.** Antes do CSS, colocavam-se elementos de apresentação dentro dos elementos HTML. O CSS veio e mudou esse paradigma, separando a apresentação da estruturação, em um arquivo separado. Da mesma forma, antes do JavaScript, colocavam-se eventos e outros elementos de integração dentro dos elementos HTML. O Unobstrusive JavaScript veio e mudou esse paradigma, separando a integração da estruturação, em um arquivo separado.

**Dessa forma, é possível separar o comportamento da marcação, reutilizar código, realizar melhoria contínua, melhorou a detecção do código JavaScript e o suporte a cache.** Dessa forma, desenvolvedores podem escrever códigos mais fáceis de manter e depurar.

Há uma biblioteca JavaScript que **ajuda a criar interfaces de usuário ricas e responsivas relacionadas a um modelo de dados subjacente: chama-se Knockout!** Ele permite associar facilmente elementos DOM com um modelo de dados usando uma sintaxe concisa e legível. Ademais, quando qualquer coisa muda o estado de seu modelo de dados, a interface de usuário é atualizada automaticamente.

Ele permite também configurar implicitamente cadeias de relacionamentos entre dados do modelo para transformá-los e combiná-los; e é capaz de gerar rapidamente sofisticadas interfaces de usuário aninhadas em função de seu modelo de dados. **Galera, ele implementa o padrão MVC, é livre, bem documentado, pequeno, leve e suportado pelos navegadores mais populares.**

---

<sup>3</sup> Eventos podem ser disparados pelo usuário (Ex: *onclick* – usuário clica em um elemento qualquer da página) ou independente de interferência do usuário (Ex: *onload* – quando um documento é carregado).



*Eu sei, isso é tudo muito abstrato. Vamos ver um exemplo para vocês entenderem melhor: imagine um carrinho de compras de uma página de e-commerce (Ex: Netshoes). **Quando o usuário deleta um item do carrinho, você tem que remover o item do modelo de dados (back-end), remover o elemento HTML associado no carrinho de compras (front-end), além de atualizar o preço total.***

Podemos escrever event handlers e event listeners para rastrear essas dependências ou simplesmente podemos utilizar o Knockout. **Ele fornece uma maneira simples e conveniente para gerenciar esse tipo complexo de interface orientada a dados.** Em vez de rastrear mudanças no estado do modelo de dados manualmente, cada elemento do HTML automaticamente atualiza o DOM.

Apesar de ter muitas similaridades com tecnologias como jQuery, Prototype, MooTools e Ajax, seu objeto não é apresentar animações, manipular eventos ou criar funcionalidades assíncronas. **Seu foco principal está em projetar interfaces de usuário escaláveis e orientadas a dados, em que eu posso manipular modelo de dados e visão sem perda de referências.** Chega de plug-ins...

Galera, assim como outras linguagens de programação, a imensa maioria das questões de JavaScript pergunta sobre tags e é inviável passar na aula dezenas e dezenas de tags. Logo, não vou passá-las aqui antes, vou passá-las durante os exercícios. **Dessa forma, vocês aprendem na prática e descobrem quais são as mais frequentes.** *Bacana?*

▪ **(BANESTES – 2021)** Considere o código javascript a seguir:

```
let txt = "";
function funcao(value, index, array) {

if (index % 2 == 0) {txt += value};

}

function xpto (x) {

x.forEach(funcao);

return txt;

}

alert (xpto([0, 1, 1, 2, 3, 5]));
```

A execução desse código provoca a exibição de:

- a) 011235
- b) 013
- c) 125
- d) 532110



## e) null

**Comentários:** pessoal, estamos utilizando na execução do código o index, então este é concatenado através do método forEach em cada elemento do array. Toda vez que invocamos a função, essa recebe um valor como parâmetro, o índice e a lista. O primeiro elemento que temos é:

```
value = 0;  
index = 0;  
array = [0,1,1,2,3,5]  
if(0 % 2 == 0){ Zero dividido por 2 = 0  
txt += 0 Temos a string txt concatenada com o  
  
}
```

No próximo elemento, chamamos a função de novo:

```
value = 1;  
index = 1;  
array = [0,1,1,2,3,5]  
  
if(1 % 2 == 0){ 1/2 = 0.5, logo não entramos na seleção, já que a divisão é diferente de zero.  
}
```

Próximo elemento:

```
value = 1;  
index = 2;  
array = [0,1,1,2,3,5]  
  
if(2 % 2 == 0){ 2/2, o resto da divisão por 2 = 0, logo txt += 1 (a string txt é concatenada com 1, ficando: 01
```

Quando realizamos de novo com index = 3, teremos: 013. Logo, a Letra B é a correta (Letra B).

**(BANESTES – 2021)** Considere o código javascript a seguir:

```
document.getElementById('demo').innerHTML = Date()
```

Numa página web na qual esse código seja aplicado, o elemento que é compatível com a estrutura do comando para receber a data corrente é:

- a) <p "demo">H</p>
- b) <p class="demo">H</p>
- c) <p id="demo">H</p>
- d) <p js="demo">H</p>
- e) <p onclick="demo">H</p>

**Comentários:** para receber a data corrente devemos passar o parâmetro recebido no GET:

```
document.getElementById('demo').innerHTML = Date()
```

então o correto é:

```
<p id="demo">H</p>
```

(Letra C).



## QUESTÕES COMENTADAS – JS

1. (ESAF/CGU - 2012) Variáveis Javascript contêm:

- a) um identificador, escopo e um tipo de dados específico.
- b) um contextualizador, localizador e um tipo de dados específico.
- c) um identificador, parâmetro e um tipo de escopo.
- d) um delimitador, escopo e um referenciador.
- e) um identificador, extensão e um ponteiro.

### Comentários:

Em JavaScript, as variáveis têm identificador, escopo e tipo. *Qual a pegadinha?* A pegadinha é que o tipo não é obrigatório – JavaScript é fracamente tipada.

**Gabarito:** Letra A

2. (FGV / SENADO – 2008) Em relação ao JavaScript, não é correto afirmar que:

- a) é uma linguagem script que só pode ser usada em páginas HTML.
- b) proporciona aos desenvolvedores de páginas web uma linguagem de programação.
- c) é uma linguagem interpretada.
- d) pode ser usada para criar cookies.
- e) pode adicionar interatividade nas páginas html.

### Comentários:

(a) Errado, ela pode ser utilizada com outras linguagens; (b) Correto, é mais uma alternativa para os desenvolvedores; (c) Correto, Javascript é uma linguagem de script e interpretada; (d) Correto, pode criar, ler e deletar cookies; (e) Correto, ele ajuda bastante a melhorar a interatividade das páginas HTML.

**Gabarito:** Letra A

3. (FGV / DPE-RO – 2015) As linguagens de programação utilizadas em programação frontend e back-end são, respectivamente:

- a) Javascript e Java;
- b) PHP e Javascript;
- c) Python e Javascript;
- d) PHP e Java;
- e) C++ e PHP.



### Comentários:

Grosso modo, o front-end trata da interface de interação com o usuário e o back-end trata das regras de negócio do sistema. Portanto, vamos julgar: (a) Front-end e Back-end; (b) Back-end e Front-end; (c) Back-end e Front-end; (d) Back-end e Back-end; (e) Back-end e Back-end.

**Gabarito:** Letra A

4. (CONSULPLAN / TRF 2 – 2017) Brendan Eich desenvolveu a primeira versão do JavaScript para o browser Mozilla, em 1995. A ideia era que a linguagem tivesse uma sintaxe parecida com Java, utilizando até mesmo alguns objetos e métodos com nomes iguais. Dessa forma, a sintaxe correta para a inicialização de um array em Javascript corresponde à questão:

- a) `var array = [16, 34, 36, 42, 50, 58].`
- b) `Int[ ] array = [16, 34, 36, 42, 50, 58].`
- c) `int array[ ] = [16, 34, 36, 42, 50, 58].`
- d) `var array = int[16, 34, 36, 42, 50, 58].`

### Comentários:

Para declarar arrays explicitamente, a sintaxe utilizada é `var nome_do_array = [item_1, item_2, ...].`

**Gabarito:** Letra A

5. (CONSULPLAN/ TRF2 – 2017) Em javascript, eventos são chamadas de código que ocorrem quando o usuário ou o browser executam determinadas ações. Existem eventos para quando o usuário clicar em algum lugar, para quando mover o ponteiro do mouse sobre uma região ou quando o ponteiro do mouse sair dessa região. Os eventos que compreendem carregamento de janela e alteração em um campo são representados respectivamente pelos comandos:

- a) `onload` e `onclick`
- b) `onload` e `onchange`
- c) `onchange` e `onreload`
- d) `onclick` e `onexchange`

### Comentários:

PROPRIEDADES	DESCRIÇÃO
<code>onload</code>	Indica o carregamento da janela.
<code>onclick</code>	Indica um clique em algum objeto.
<code>onchange</code>	Indica alteração em um campo de um objeto.



onreload	Não existe.
onexchange	Não existe.

Conforme a tabela acima, Onload indica o carregamento da janela e Onclick indica um clique em algum objeto.

**Gabarito:** Letra B

6. (FGV / IBGE – 2016) Uma página contém o seguinte código JavaScript:

```
function teste() {  
  var letras = [];  
  grupoInicial: {  
    letras.push('A');  
    letras.push('B');  
    break grupoInicial;  
    letras.push('C');  
  }  
  letras.push('D');  
  letras.push('E');  
  return letras.join(',');  
}
```

O resultado esperado é:

- a) A,B
- b) A,B,C
- c) A,B,D,E
- d) A,B,C,D,E
- e) D,E

#### Comentários:

O comando break termina a execução do escopo atual e o código continua a ser executado em seu escopo mais externo. Nesse caso, a linha `letras.push('C');` não será executada.

**Gabarito:** Letra C

7. (FGV / IBGE – 2016) Um website foi programado para exibir o nome do usuário no canto da tela através do seguinte código HTML:

```
<p id="cliente">usuário</p>
```

Considerando que o nome está armazenado na variável "meuNome", a sintaxe correta em Javascript para trocar a palavra "usuário" pelo conteúdo da variável é:





- a) `document.getElementById(cliente).innerHTML = "meuNome"`
- b) `document.getElementById("cliente").innerHTML = meuNome`
- c) `document.getElementsByName(cliente).innerHTML = meuNome`
- d) `document.getElementById("cliente").innerHTML = meuNome`
- e) `document.getElementById(meuNome).innerHTML = "cliente"`

### Comentários:

Para obtermos uma referência a partir de um id, utilizamos o método `getElementById` do objeto `document` e, para acessar seu conteúdo html, usamos a propriedade `innerHTML`. Como o valor que queremos está na variável `meuNome`, o correto será:

```
document.getElementById("cliente").innerHTML = meuNome (Sem as aspas!)
```

**Gabarito:** Letra D

**8. (FGV / IBGE – 2016)** Para a produção de um website responsivo com grande quantidade de textos e imagens, as técnicas mais eficientes se baseiam na aplicação de:

- a) grades fixas e dimensionadas em unidades absolutas, imagens dimensionadas em unidades absolutas e posicionadas de modo fluido, e JavaScript para alterar as regras de estilo com base na resolução de tela
- b) grades fluidas e dimensionadas em unidades absolutas, imagens flexíveis limitadas por elementos de contenção e posicionadas em unidades absolutas, e JavaScript para alterar as regras de estilo com base na resolução de tela
- c) grades fluidas e proporcionais, imagens flexíveis limitadas por elementos de contenção e dimensionadas em unidades relativas, e *media queries* para alterar as regras de estilo com base na resolução de tela
- d) grades fixas e proporcionais, imagens fixas e dimensionadas por Javascript, e *media queries* para alterar as regras de estilo com base na resolução de tela
- e) grades fluidas, imagens flexíveis limitadas por elementos de contenção, e JavaScript para posicionar todo o conteúdo de acordo com a resolução de tela.

### Comentários:

Um website responsivo deve adaptar-se a qualquer tamanho de tela, desde um monitor de muitas polegadas até um celular. Para isso, podemos destacar algumas propriedades de um website



responsivo. Grades (divs e outros elementos) fluidas e proporcionais, distribuindo o conteúdo de acordo com o tamanho da tela.

Imagens de tamanho relativo para que não fiquem nem muito pequenas e ilegíveis, e nem muito grandes e ultrapassem o tamanho da tela. *Media queries*, que permitem o uso de diferentes regras de estilo para cada tamanho de tela.

**Gabarito:** Letra C

---

9. (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:

Diversas APIs foram disponibilizadas com o HTML5, dentre elas. Armazenamento local, onde possível armazenar variáveis no navegador; Geolocalização, para localização do cliente; e SSE (Server-Sent Events), onde é possível que a página receba atualizações de eventos do servidor.

**Gabarito:** Letra D

---

10. (FGV / TCM-SP – 2015) Em Javascript, considere o trecho de código a seguir:



```
function base(x)
{
    return function produto(y)
    {
        return x * y;
    }
}

var f = base(2);
var g = base(-1);
```

Após as duas atribuições, supondo que os valores de f e g não mudem, a avaliação da expressão  $f(2) + g(-1)$  produzirá o valor:

- a) 5;
- b) 4;
- c) 3;
- d) 2;
- e) 1.

#### Comentários:

Podemos pensar da seguinte maneira:  $f = \text{base}(2)$  retorna a função  $\text{produto}(y) \{ \text{return } 2*y; \}$ , logo  $f(y) = 2*y$  e  $f(2) = 4$ . De forma análoga temos:

$g = \text{base}(-1)$  retorna a função  $\text{produto}(y) \{ \text{return } -1*y \}$ , logo  $g(y) = -y$  e  $g(-1) = 1$ .

Com isso concluímos facilmente que  $f(2) + g(1) = 5$ .

**Gabarito:** Letra A

**11. (IESES / IFC-SC – 2015)** Analise as afirmativas a seguir sobre o JavaScript 1.8 e assinale a alternativa correta:

I. A maneira mais comum de se declarar uma classe em JavaScript com o nome "Cachorro" seria da seguinte forma: **class Cachorro { }**

II. A maneira mais comum de se instanciar um objeto em JavaScript a partir da classe "Cachorro" seria da seguinte forma: **Cachorro cachorro = new Cachorro;**



III. Tanto comentários de uma linha com barras duplas (`// ...`) quanto comentários em blocos com barra e asterisco (`/* ... */`) são suportados em JavaScript.

- a) As afirmativas I, II e III são verdadeiras.
- b) Somente a afirmativa I é verdadeira.
- c) Somente as afirmativas I e III são verdadeiras.
- d) Somente a afirmativa III é verdadeira

### Comentários:

(I) Errado, a diretriz `class` não existe no JavaScript. A maneira mais comum seria `var Cachorro = function() {}`; (II) Errado, a declaração de qualquer tipo é realizada utilizando `var`. A maneira mais comum seria `var cachorro = new Cachorro()`; (III) Correto, o item está perfeito – ambos são suportados.

**Gabarito:** Letra D

**12. (IESES / IFSC – 2015)** Uma determinada linha de código JavaScript 1.8 se lê da seguinte forma:

```
var novoCachorro = new Cachorro();
```

Assinale a alternativa que apresenta um trecho de código que deve ser inserido antes da linha demonstrada, de forma que o código seja sintática e semanticamente válido:

- a) `function(e) { e.class(Cachorro); }`
- b) `var Cachorro = function() { }`
- c) `class Cachorro { function(e) { e.instanceof(Cachorro); } }`
- d) `class Cachorro { }`

### Comentários:

A linha apresentada no enunciado mostra uma instanciação da classe `Cachorro`. Para que isso seja possível, precisamos que a classe seja declarada. A maneira correta de declarar uma classe em Javascript é `var Cachorro = function() {}`.

**Gabarito:** Letra B

**13. (IESES / IFSC – 2015)** Observe o código javascript abaixo:



```
<script type="text/javascript">
  var cidades = ["São Paulo", "Criciúma", "Curitiba", "Itajaí", "Rio do Sul", "Erechim"];
  for(var i = 0; i < cidades.length; i++){
    str = cidades[i];
    if(str.length == 8){
      document.write(str);
      document.write("<br>");
    }
  }
</script>
```

O resultado da execução desse script é:

- a) Criciúma, Curitiba
- b) Criciúma, Curitiba
- c) São Paulo Criciúma Curitiba Itajaí Rio do Sul Erechim
- d) Criciúma Curitiba

#### Comentários:

O script acima simplesmente percorre todos os itens do Array cidades e escreve no documento aqueles em que o tamanho do item é igual a 8. Vale notar que ele também adiciona uma quebra de linha (<br>) entre cada escrita.

**Gabarito:** Letra A

14. (FGV / TCE-SE – 2015) Em um programa Javascript, encontra-se o seguinte comando:

```
var x = 3 + "4";
```

Após a execução deste comando, a variável x conterà:

- a) a cadeia "34";
- b) número 34;
- c) a cadeia "7";
- d) número 7;
- e) o mesmo de antes da execução do comando, devido a um erro referente à conversão de tipos

#### Comentários:

Em JavaScript, o operador + é utilizado para concatenação de *strings* e, se algum dos operandos for uma *string*, ele irá converter qualquer outro operando em *string*. Isso se chama conversão implícita.

**Gabarito:** Letra A



15. (FGV / TCE-SP – 2015) Em uma página HTML, é declarado o seguinte formulário:

```
<form id="dados" action="script.php"
      method="POST">
  <input type="submit" name="Submit"
        value="ENVIAR"/>
  <input type="hidden" id="tamanho"
        name="TAMANHO"
        value="" />
</form>
```

O valor do campo TAMANHO deverá ser preenchido por uma função Javascript, que terá acesso a ele através do construtor:

- a) document.getElementById("dados").tamanho.value;
- b) document.getElementById("hidden").value;
- c) document.getElementById("tamanho").value;
- d) document.getElementById("input").tamanho.value;
- e) document.getElementById("form").tamanho.value.

#### Comentários:

Para obtermos uma instância de um objeto a partir de seu id, utilizamos o método

```
document.getElementById("id")
```

e, para acessar seu valor, utilizamos o atributo value. Logo, para acessarmos o valor do campo TAMANHO utilizamos

```
document.getElementById("tamanho").value;
```

**Gabarito:** Letra C

16. (NUCEPE / SEFAZ-PI – 2015) Laços são importantes construções em linguagens de programação. Qual tipo de laço (loop) NÃO possui suporte em JavaScript?

- a) For.
- b) For/in.
- c) Repeat/until.
- d) While.
- e) Do/while.



## Comentários:

A sintaxe do JavaScript não suporta laços do tipo Repeat/until. Todos das outras alternativas são suportados. Podemos ver alguns exemplos de utilização dos laços abaixo:

```
for (int a=0; a<10; a++) { ... }  
for (element in array) { ... }  
while (a == true) { ... }  
do { ... } while (a < 10);
```

Gabarito: Letra C

17. (MPE-RS / MPE-RS – 2015) Dado o código HTML abaixo:

```
<!DOCTYPE html>  
<html>  
  
<body>  
<p>UM</p>  
<p>DOIS</p>  
<p>TRES</p>  
  
<script type="text/javascript">  
window.onload = function() {  
  
.....  
}  
  
</script>  
</body>  
</html>
```

Quais dos comandos Javascript listados abaixo podem ser colocados no espaço indicado pelos pontos para que a página, ao ser carregada, altere somente a cor do texto "UM" para azul?

1. `document.getElementsByTagName("p")[0].style.color = "#0000FF";`
  2. `document.getElementsByTagName("p")[0].color = "#0000FF";`
  3. `document.getElementsByTagName("p").item(0).style.color = "#0000FF";`
- a) Apenas 1.  
b) Apenas 2.  
c) Apenas 3.



- d) Apenas 1 e 3.
- e) 1, 2 e 3.

### Comentários:

O método `document.getElementsByTagName("p")` retorna uma coleção de objetos do tipo `HTMLCollection`. Para acessarmos o primeiro elemento da coleção, podemos utilizar tanto a sintaxe de acesso por índices, `[0]`, quanto o método `item(0)`. Uma vez que obtivermos o objeto, para mudar sua cor é necessário acessar primeiro o atributo `style` e depois o atributo `color`, `.style.color`.

**Gabarito:** Letra D

18. (MPE-RS / MPE-RS – 2015) Dado o código Javascript abaixo:

```
<script type="text/javascript">
var numeros =
["Um", "Dois", "Tres", "Quatro"];
numeros.pop();
numeros.push("Dez");
numeros.shift();
document.write(numeros);
</script>
```

Quando o bloco for inserido no corpo de uma página HTML e executado no navegador, o que será exibido na tela?

- a) Dois,Tres,Dez
- b) Quatro,Dez,Dois,Tres
- c) Dois,Tres,Dez,Um
- d) Dez,Dois,Tres
- e) Um,Dois,Tres,Dez,Dez

### Comentários:

Vejamos a função de cada método utilizado:

- `pop()` remove o último item do array ("Quatro");
  - o **Resultado:** "Um", "Dois", "Tres".
- `push()` adiciona um item ao final do array (Dez); e
  - o **Resultado:** "Um", "Dois", "Tres", "Dez".
- `shift()` remove o primeiro element do array (Um).





**Resultado:** "Dois", "Tres", "Dez".

**Gabarito:** Letra A

19.(MPE-RS / MPE-RS – 2015) Dado o código Javascript abaixo:

```
<script type="text/javascript">
function Aluno() {
  var nome = "Ze";
  var dados = {
    idade : 18,
    peso : 75
  };
  this.getDados = function() {
    return dados;
  };
}
var x = new Aluno();
var y = x.getDados();
document.write(x.nome + " ");
document.write(y.idade + " ");
y.idade += 2;
document.write(x.getDados().idade);
</script>
```

Quando o bloco for inserido no corpo de uma página HTML e executado no navegador, o que será exibido na tela?

- a) undefined undefined 20
- b) undefined 18 20
- c) Ze 18 18
- d) Ze 18 20
- e) undefined 18 18

### Comentários:

Vamos analisar as linhas onde há escrita:

```
document.write(x.nome + " ");
```

- O objeto x, da classe Aluno, não possui a propriedade nome (sua variável dados, sim), portanto essa linha imprime undefined.

```
document.write(y.idade + " ");
```

- Até aqui é bem simples: y tem o valor de dados, dentro de x. Portanto y.idade é igual a 18.

```
document.write(x.getDados().idade);
```



- Aqui está a verdadeira questão, o fato de mudar o valor de `y.idade` afetará o valor de `x.dados.idade`? A resposta é sim, e vamos ver o porque.

Para isso, precisamos separar em dois casos de atribuição com o operador `=`:

1) Com valores primitivos, o operador `=` funciona por valor. Vejamos um exemplo:

```
var name = "Carlos"; var firstName = name;  
name = "Carla";  
console.log(name); // "Carla"  
console.log(firstName); // "Carlos"
```

Nesse caso, a variável `firstName` recebe uma cópia de `name`, portanto alterar o valor de `name` não irá alterar o valor de `firstName`.

2) Com objetos, o operador `=` funciona por referência. Vejamos um exemplo:

```
var myName = { firstName: "Carlos" }; var identity = myName;  
myName.firstName = "Carla";  
console.log(myName.firstName); // "Carla"  
console.log(identity.firstName); // "Carla"
```

Nesse caso, a variável `firstName` recebe uma referência de `name`, portanto alterar o valor de `name` irá alterar o valor de `firstName`.

Voltando para nossa questão, como se trata de um objeto, a atribuição é feita por referência, ou seja, quando alteramos o valor de `y.idade` também alteramos o valor de `x.getDados().idade`.

**Gabarito:** Letra B

**20. (MPE-RS / MPE-RS – 2015)** Quais são os valores de saída do código Javascript abaixo?



```
<script type="text/javascript">
<!--
var x = 1;
while (x < 20)
{
  if (x == 5){
    break;
  }
  x = x + 1;
  document.write( x + "<br />");
}
//-->
</script>
```

- a) 2-3-4-5
- b) 2-3-4
- c) 1-2-3-4-5
- d) 1-2-3-4
- e) 0-1-2-3-4

### Comentários:

Pelo *loop* principal, `while (x < 20)`, a variável `x` percorreria os valores de 1 a 19, porém quando `x` chega a 5 o código sai do loop devido ao comando `break`. Logo, os valores de `x` são 1, 2, 3 e 4. Como o valor de `x` é incrementado antes da impressão, os valores impressos são 2, 3, 4 e 5. Vale ressaltar que as linhas `<!--e // -->` não alteram em nada o código. Além disso, como há uma impressão da tag `<br />`, a saída seria como os números linha a linha, e não lado a lado.

**Gabarito:** Letra A

21. (MPE-RS / MPE-RS – 2015) Dado o seguinte código JSON com Javascript:

```
var text = '{ "employees" : [ ' +
'{"firstName":"John" , "lastName":"Doe" },' +
'{"firstName":"Anna" , "lastName":"Smith" },' +
'{"firstName":"Peter" , "lastName":"Jones" } ]}';
```

É correto afirmar que o código apresentado acima:

- a) concatena três string formando uma classe e atribui dados a ela.
- b) concatena três objetos formando uma classe e atribui dados a ela.
- c) divide um objeto em três strings e atribui dados a elas.
- d) concatena três arrays em um objeto e atribui dados a ele.



e) cria um array com três objetos e atribui dados a eles

### Comentários:

ENTRADA	SAÍDA
<pre>var text = '{"employees" : [' + '{ "firstName":"John", "lastName":"Doe" },' + '{ "firstName":"Anna", "lastName":"Smith" },' + '{ "firstName":"Peter", "lastName":"Jones" } ]}';</pre>	<pre>{"employees" : [   { "firstName":"John", "lastName":"Doe" },   { "firstName":"Anna", "lastName":"Smith" },   { "firstName":"Peter", "lastName":"Jones" } ]}</pre>

Analisando a entrada com *syntax highlighting*, podemos ver facilmente que se trata de uma concatenação de três strings, porém não forma uma classe, e sim um array employees com três objetos que possuem os atributos firstName e lastName.

Vale notar que a variável text é apenas uma string, e que para que seja transformada no que a banca queria seria necessário usar JSON.parse(text) para obter-se o array.

**Gabarito:** Letra E

22. (UFG / AL-GO – 2015) Leia o trecho em Javascript a seguir:

```
var a={},  
b={key:'b'},  
c={key:'c'};  
  
a[b]=123;  
a[c]=456;  
  
console.log(a[b]);
```

Qual é a saída da execução desse trecho de código Javascript?

- a) 123
- b) 456
- c) 156
- d) 423

### Comentários:

Para responder essa questão, precisamos entender como funcionam os índices de arrays. Quando o índice é um objeto, o JavaScript irá primeiro converter o índice para uma string, e depois utilizá-lo. Vejamos um exemplo:



```
var a = {};  
var b = [1, 2, 3];  
a[b] = 'oi';
```

No exemplo, `a[b]` retorna `'oi'`, `a[[1, 2, 3]]` retorna `'oi'` e `a[1, 2, 3]` retorna `'oi'`. Isso porque todos os argumentos quando transformados em string retornam a mesma coisa: `b.toString() = '1, 2, 3'`, `[1, 2, 3].toString() = '1, 2, 3'` e `'1, 2, 3'.toString() = '1, 2, 3'`.

No código da questão acima, `b.toString() = '[object Object]'` e `c.toString() = '[object Object]'`, logo para o array `a`, os índices `b` e `c` são iguais, como podemos ver no output do console abaixo:

```
> a[b] = 123;  
> a  
< {[object Object]: 123}  
> a[c] = 456;  
> a  
< {[object Object]: 456}
```

Logo, chamar `a[b]` ou `a[c]` seria o mesmo que chamar `a['[object Object]']` e portanto ambos retornariam `456` ao final do código.

**Gabarito:** Letra B

**23. (FGV / BANESTES – 2021)** Considere o código javascript a seguir:

```
let txt = "";  
function funcao(value, index, array) {  
  if (index % 2 == 0) {txt += value};  
}  
  
function xpto (x) {  
  x.forEach(funcao);  
  return txt;  
}  
  
alert (xpto([0, 1, 1, 2, 3, 5]));
```

A execução desse código provoca a exibição de:

- a) 011235
- b) 013
- c) 125



- d) 532110
- e) null

### Comentários:

Pessoal, estamos utilizando na execução do código o index, então este é concatenado através do método forEach em cada elemento do array. Toda vez que invocamos a função, essa recebe um valor como parâmetro, o índice e a lista. O primeiro elemento que temos é:

```
value = 0;  
index = 0;  
array = [0,1,1,2,3,5]  
if(0 % 2 == 0){ Zero dividido por 2 = 0  
txt += 0 Temos a string txt concatenada com 0  
}
```

No próximo elemento, chamamos a função de novo:

```
value = 1;  
index = 1;  
array = [0,1,1,2,3,5]  
if(1 % 2 == 0){ 1/2 = 0.5, logo não entremos na seleção, já que a divisão é diferente de zero.  
}
```

Próximo elemento:

```
value = 1;  
index = 2;  
array = [0,1,1,2,3,5]  
if(2 % 2 == 0){ 2/2, o resto da divisão por 2 = 0, logo txt += 1 (a string txt é concatenada com 1,  
ficando: 01
```

Quando realizamos de novo com index = 3, teremos: 013. Logo, a Letra B é a correta.

**Gabarito:** Letra B

**24. (FGV / BANESTES – 2021)** Considere o código javascript a seguir:

```
document.getElementById('demo').innerHTML = Date()
```

Numa página web na qual esse código seja aplicado, o elemento que é compatível com a estrutura do comando para receber a data corrente é:

- a) <p "demo">H</p>



- b) `<p class="demo">H</p>`
- c) `<p id="demo">H</p>`
- d) `<p js="demo">H</p>`
- e) `<p onclick="demo">H</p>`

### Comentários:

Galera, para receber a data corrente devemos passar o parâmetro recebido no GET:

```
document.getElementById('demo').innerHTML = Date()
```

Então, o correto é:

```
<p id="demo">H</p>
```

**Gabarito:** Letra C

25. (CESPE / PGDF - 2021) Julgue o item a seguir, referente a linguagem de scripts.

```
<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.getEment
ById('b').value);
</script>
</html>
```

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

5
2
52

### Comentários:

Pessoal, os inputs são texto e valores, teremos então os textos sendo concatenados e três inputs de valores, como o terceiro input é vazio, concatena-se com os anteriores, 5 e 2, por isso apresenta texto 52. Mas tem uma indução a erro na questão, que é apresentar o resultado em forma de tabela, todavia a banca desconsiderou esse fato e entendeu como certa a apresentação dos respectivos resultados (5,2, 52) em formato de tabela.



26. (CESPE / SEED-PR - 2021) resultado = (x for x in [5..2])

Assinale a opção que apresenta o resultado da execução do código CoffeeScript precedente.

- a) 5,5
- b) 5,4,3,2
- c) 2,2,2,2,2
- d) 5,3,1
- e) 4,3

#### Comentários:

CoffeeScript é uma linguagem compilada do JS, então o que aprendemos com JS se aplica aqui, ok?

Temos de forma bem simples um array, onde o resultado é equivalente a (x para todo x dentro do intervalo de (5 a 2), logo termos: 5,4,3,2, dentro do intervalo 5 a 2. Simples, não é verdade?

Gabarito: Letra B

27. (CESPE / SEED-PR - 2021) JavaScript trabalha com números usando os operadores aritméticos fornecidos pela própria linguagem. No entanto, a linguagem aceita operações matemáticas mais complexas por meio de um conjunto de funções e constantes definidas como propriedades do objeto Math.

#### Comentários:

Perfeito! O objeto Math define as funções e constante para operações dentro da linguagem JS.

Gabarito: Correto

28. (CESPE / SEED-PR - 2021) A opção Math.ceil () apresenta a função que permite realizar, em JS, a operação matemática de arredondar para cima o número 1,17, obtendo-se o valor 2.

#### Comentários:

Perfeito! Quando utilizamos o Math.ceil arredondamos para cima o valor do número 1,17, resultando em 2. Se fosse utilizado o Math.round () teríamos o arredondamento para o inteiro mais próximo, ou seja, 2.0.





**Gabarito:** Correto

---

**29. (CESPE / ME - 2020)** Nos padrões web, as camadas deverão ser separadas de acordo com o objetivo para o qual elas foram desenvolvidas. Criar páginas em camadas lógicas é uma boa prática: xHTML, folhas de estilo CSS e JavaScript são voltadas, respectivamente, para as camadas de conteúdo, de apresentação visual do conteúdo e de comportamento dos elementos.

**Comentários:**

Pessoal, não é objeto dessa aula tratar de xHTML e CSS, porém a questão abordou, logo temos que xHTML é responsável pelo conteúdo, CSS pela apresentação do conteúdo e o JS cuidará da interação ou comportamento do conteúdo da página web. Dito isso, é uma boa prática utilizar xHTML, CSS e JS nas respectivas funções apresentadas.

**Gabarito:** Correto

---

**30. (CESPE / TJ-AM - 2019)** JavaScript é uma linguagem de programação orientada ao desenvolvimento da interface para aplicações web, cujo código-fonte é compilado pelo servidor antes de sua execução.

**Comentários:**

JS é uma linguagem interpretada, assim como PHP e Python. Exemplo de linguagem compilada seria o C++.

**Gabarito:** Errado

---

**31. (CESPE / SLU-DF - 2019)** Uma função JavaScript é um bloco de código utilizado para executar tarefas repetidas e é definida pela palavra-chave public seguida por um nome seguido por parênteses ( ).

**Comentários:**

Pessoal, quando apresentamos a função JS em código para execução de tarefas repetidas (loop), demonstramos a estrutura:

```
<script>
function minhaFuncao() {
document.getElementById("demo").innerHTML = "Minha primeira função JavaScript!";
}
</script>
```



Ou seja, a função se inicia pelo function e não pela palavra public, com argumento (no exemplo temos minhaFuncao), seguido de parenteses.

**Gabarito:** Errado

---

**32. (CESPE / SLU-DF - 2019)** O JSX (JavaScript Syntax Extension) é de uso obrigatório no React e permite inserir a interface do usuário no código JavaScript.

#### Comentários:

O React (também denominado React.js ou ReactJS) é uma biblioteca JavaScript de código aberto com foco em criar interfaces de usuário (frontend) em páginas web. É mantido pelo Facebook, Instagram, outras empresas e uma comunidade de desenvolvedores individuais. Dito isso, qual o problema da questão: O JSX (JavaScript Syntax Extension) NÃO é de uso obrigatório no React, ou seja, o React NÃO requer o uso do JavaScript Syntax Extension. Dificilmente vocês verão uma questão escrito "é obrigatório" e estará correta.

**Gabarito:** Errado

---

**33. (FCC / TRT 11 - 2005)** Em JavaScript, para declarar nomes de identificadores válidos utiliza-se:

- a) somente número no primeiro caractere.
- b) somente letra ou underscore no primeiro caractere.
- c) qualquer tipo de caractere a partir do segundo caractere.
- d) espaço entre caracteres a partir do segundo caractere.
- e) uma palavra reservada do JavaScript

#### Comentários:

Para declarar nomes de identificadores, utiliza-se apenas letra ou underscore no primeiro caractere.

**Gabarito:** Letra B

---

**34. (FCC / TRE-SE - 2007)** O cliente JavaScript, quando se soltar um botão do mouse pressionado, invocará o evento:

- a) MouseOver.
- b)MouseDown.
- c) MouseUp.
- d) MouseMove.
- e) MouseOut.

#### Comentários:



PROPRIEDADES	DESCRIÇÃO
onmouseover	O ponteiro é movido a um elemento ou a seus filhos.
onmousedown	O usuário pressiona o botão do mouse em um elemento.
onmouseup	O usuário solta o botão sobre um elemento.
onmousemove	O ponteiro se move quando se está sobre um elemento.
onmouseout	O usuário retira o ponteiro de um elemento ou de seus filhos.

**Gabarito:** Letra C

**35. (FCC / MPE-SE - 2008)** Uma função embutida na linguagem JavaScript que calcula o conteúdo de uma string denomina-se:

- a) string.
- b) eval.
- c) number.
- d) parseInt.
- e) parseFloat

**Comentários:**

“The eval() function evaluates or executes an argument. If the argument is an expression, eval() evaluates the expression. If the argument is one or more JavaScript statements, eval() executes the statements”. Portanto, ela avalia e executa códigos dentro de uma string e apresenta o resultado, como podemos ver abaixo:

```
var x = 10;  
var y = 20;  
var a = eval("x * y") + "<br>";  
var b = eval("2 + 2") + "<br>";  
var c = eval("x + 17") + "<br>";  
  
var res = a + b + c;
```

O Resultado será: 200, 4 e 27.

**Gabarito:** Letra B

**36. (FCC / TRT-1 - 2014)** O trecho de programa a seguir foi elaborado na linguagem HTML com JavaScript e será aberto por um navegador que as suporte.

```
<html>  
<body>  
<h1>Página Principal</h1>
```



```
<form>  
comando y  
</form>  
</body>  
</html>
```



O comando que deve ser colocado no lugar do comando y, de modo que, ao se pressionar o botão Nova Página, seja aberta a página P1.html, é:

- a) `<input type = "button" text="Nova Página" click="window.location='P1.html'">`
- b) `<input type="button" text="Nova Página" onclick = "window.location='P1.html'">`
- c) `<input type="onbutton" text="Nova Página" onclick = "window.location='P1.html'">`
- d) `<input type="button" value="Nova Página" onclick = "window.location='P1.html'">`
- e) `<input type="onbutton" value="Nova Página" click = "window.location='P1.html'">`

### Comentários:

Bem, basta saber usar a tag `<input>`! Ela serve para receber dados. O atributo com valor `type="button"` especifica um botão cujo nome é `value="Nova Página"` e o evento `onclick = "window.location='P1.html'"` leva o usuário que clica nesse botão à página P1.html

**Gabarito:** Letra D

**37. (FCC / TRT-18 - 2008)** Na linguagem JavaScript, os métodos de interface com o usuário que fazem parte do objeto window são apenas os denominados:

- a) select, submit e confirm.
- b) open, alert e write.
- c) click, select e write.
- d) open, submit e close.
- e) alert, confirm e prompt.

### Comentários:

Trata-se da última opção: alert, confirm e prompt.



MÉTODOS	DESCRIÇÃO
alert( )	Exibe uma janela de alerta com um Botão Ok.
confirm ( )	Exibe uma janela de diálogo com uma mensagem e os botões Ok e Cancel.
prompt ( )	Exibe uma janela de diálogo à espera de uma entrada do usuário.

**Gabarito:** Letra E

**38.(FCC / DPE-SP - 2010)** As tecnologias Web utilizam linguagem de scripting JavaScript, linguagem de scripting ASP e applets Java para incluir aplicações, respectivamente, no lado:

- a) servidor, cliente e servidor.
- b) cliente, servidor e cliente.
- c) servidor, servidor e cliente.
- d) cliente, cliente e servidor.
- e) cliente, cliente e cliente.

**Comentários:**

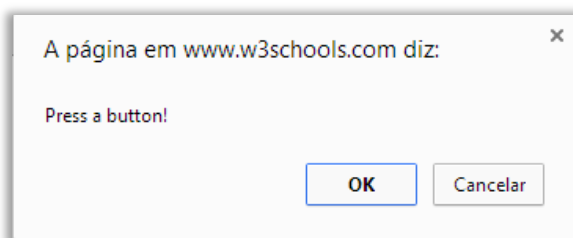
JavaScript e Applets rodam no cliente e Scripting ASP roda no servidor.

**Gabarito:** Letra B

**39.(FCC / TRT-20 - 2010)** No JavaScript, a caixa de mensagem "Confirm":

- a) permite que o usuário insira um nome e/ou um número dentro da caixa de texto.
- b) Exibe apenas uma informação para o usuário.
- c) Permite que o usuário insira um nome dentro da caixa de texto.
- d) Permite que o usuário insira um número dentro da caixa de texto.
- e) Solicita uma confirmação do usuário, positiva ou negativa.

**Comentários:**



As caixas de diálogo de confirmação visam apenas exibir uma mensagem e pedir uma confirmação positiva ou negativa

**Gabarito:** Letra E



40.(FCC – TRT-20 - 2010) É um método do objeto History:

- a) Case.
- b) Left.
- c) Open.
- d) Forward
- e) Hostname

#### Comentários:

O objeto History contém as URLs visitadas por um usuário.

MÉTODOS	DESCRIÇÃO
back()	Carrega a URL anterior do histórico.
forward()	Carrega a próxima URL do histórico.
go()	Carrega uma URL específica do histórico.

**Gabarito:** Letra D



## LISTA DE QUESTÕES – JS

- (ESAF/CGU - 2012)** Variáveis Javascript contêm:
  - um identificador, escopo e um tipo de dados específico.
  - um contextualizador, localizador e um tipo de dados específico.
  - um identificador, parâmetro e um tipo de escopo.
  - um delimitador, escopo e um referenciador.
  - um identificador, extensão e um ponteiro.
- (FGV / SENADO – 2008)** Em relação ao JavaScript, não é correto afirmar que:
  - é uma linguagem script que só pode ser usada em páginas HTML.
  - proporciona aos desenvolvedores de páginas web uma linguagem de programação.
  - é uma linguagem interpretada.
  - pode ser usada para criar cookies.
  - pode adicionar interatividade nas páginas html.
- (FGV / DPE-RO – 2015)** As linguagens de programação utilizadas em programação frontend e back-end são, respectivamente:
  - Javascript e Java;
  - PHP e Javascript;
  - Python e Javascript;
  - PHP e Java;
  - C++ e PHP.
- (CONSULPLAN / TRF 2 – 2017)** Brendan Eich desenvolveu a primeira versão do JavaScript para o browser Mozilla, em 1995. A ideia era que a linguagem tivesse uma sintaxe parecida com Java, utilizando até mesmo alguns objetos e métodos com nomes iguais. Dessa forma, a sintaxe correta para a inicialização de um array em Javascript corresponde à questão:
  - `var array = [16, 34, 36, 42, 50, 58].`
  - `Int[ ] array = [16, 34, 36, 42, 50, 58].`
  - `int array[ ] = [16, 34, 36, 42, 50, 58].`
  - `var array = int[16, 34, 36, 42, 50, 58].`
- (CONSUPLAN/ TRF2 – 2017)** Em javascript, eventos são chamadas de código que ocorrem quando o usuário ou o browser executam determinadas ações. Existem eventos para quando o usuário clicar em algum lugar, para quando mover o ponteiro do mouse sobre uma região ou quando o ponteiro do mouse sair dessa região. Os eventos que compreendem carregamento de janela e alteração em um campo são representados respectivamente pelos comandos:



- a) onload e onclick
- b) onload e onchange
- c) onchange e onreload
- d) onclick e onexchange

6. (FGV / IBGE – 2016) Uma página contém o seguinte código JavaScript:

```
function teste() {  
  var letras = [];  
  grupoInicial: {  
    letras.push('A');  
    letras.push('B');  
    break grupoInicial;  
    letras.push('C');  
  }  
  letras.push('D');  
  letras.push('E');  
  return letras.join(',');  
}
```

O resultado esperado é:

- a) A,B
  - b) A,B,C
  - c) A,B,D,E
  - d) A,B,C,D,E
  - e) D,E
7. (FGV / IBGE – 2016) Um website foi programado para exibir o nome do usuário no canto da tela através do seguinte código HTML:

```
<p id="cliente">usuário</p>
```

Considerando que o nome está armazenado na variável "meuNome", a sintaxe correta em Javascript para trocar a palavra "usuário" pelo conteúdo da variável é:

- a) document.getElementById(cliente).innerHTML = "meuNome"
  - b) document.getElement("cliente").innerHTML = meuNome
  - c) document.getElementByName(cliente).innerHTML = meuNome
  - d) document.getElementById("cliente").innerHTML = meuNome
  - e) document.getElementById(meuNome).innerHTML = "cliente"
8. (FGV / IBGE – 2016) Para a produção de um website responsivo com grande quantidade de textos e imagens, as técnicas mais eficientes se baseiam na aplicação de:





- a) grades fixas e dimensionadas em unidades absolutas, imagens dimensionadas em unidades absolutas e posicionadas de modo fluido, e JavaScript para alterar as regras de estilo com base na resolução de tela
- b) grades fluidas e dimensionadas em unidades absolutas, imagens flexíveis limitadas por elementos de contenção e posicionadas em unidades absolutas, e JavaScript para alterar as regras de estilo com base na resolução de tela
- c) grades fluidas e proporcionais, imagens flexíveis limitadas por elementos de contenção e dimensionadas em unidades relativas, e *media queries* para alterar as regras de estilo com base na resolução de tela
- d) grades fixas e proporcionais, imagens fixas e dimensionadas por Javascript, e *media queries* para alterar as regras de estilo com base na resolução de tela
- e) grades fluidas, imagens flexíveis limitadas por elementos de contenção, e JavaScript para posicionar todo o conteúdo de acordo com a resolução de tela.

9. (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*)

10. (FGV / TCM-SP – 2015) Em Javascript, considere o trecho de código a seguir:



```
function base(x)
{
    return function produto(y)
    {
        return x * y;
    }
}

var f = base(2);
var g = base(-1);
```

Após as duas atribuições, supondo que os valores de f e g não mudem, a avaliação da expressão  $f(2) + g(-1)$  produzirá o valor:

- a) 5;
- b) 4;
- c) 3;
- d) 2;
- e) 1.

11. (IESES / IFC-SC – 2015) Analise as afirmativas a seguir sobre o JavaScript 1.8 e assinale a alternativa correta:

I. A maneira mais comum de se declarar uma classe em JavaScript com o nome “Cachorro” seria da seguinte forma: **class Cachorro { }**

II. A maneira mais comum de se instanciar um objeto em JavaScript a partir da classe “Cachorro” seria da seguinte forma: **Cachorro cachorro = new Cachorro;**

III. Tanto comentários de uma linha com barras duplas (`// ...`) quanto comentários em blocos com barra e asterisco (`/* ... */`) são suportados em JavaScript.

- a) As afirmativas I, II e III são verdadeiras.
- b) Somente a afirmativa I é verdadeira.
- c) Somente as afirmativas I e III são verdadeiras.
- d) Somente a afirmativa III é verdadeira

12. (IESES / IFSC – 2015) Uma determinada linha de código JavaScript 1.8 se lê da seguinte forma:

```
var novoCachorro = new Cachorro();
```



Assinale a alternativa que apresenta um trecho de código que deve ser inserido antes da linha demonstrada, de forma que o código seja sintática e semanticamente válido:

- a) `function(e) { e.class(Cachorro); }`
- b) `var Cachorro = function() { }`
- c) `class Cachorro { function(e) { e.instanceof(Cachorro); } }`
- d) `class Cachorro { }`

13. (IESES / IFSC – 2015) Observe o código javascript abaixo:

```
<script type="text/javascript">
  var cidades = ["São Paulo", "Criciuma", "Curitiba", "Itajaí", "Rio do Sul", "Erechim"];
  for(var i = 0; i < cidades.length; i++){
    str = cidades[i];
    if(str.length == 8){
      document.write(str);
      document.write("<br>");
    }
  }
</script>
```

O resultado da execução desse script é:

- a) Criciuma, Curitiba
- b) Criciuma, Curitiba
- c) São Paulo Criciuma Curitiba Itajaí Rio do Sul Erechim
- d) Criciuma Curitiba

14. (FGV / TCE-SE – 2015) Em um programa Javascript, encontra-se o seguinte comando:

```
var x = 3 + "4";
```

Após a execução deste comando, a variável x conterà:

- a) a cadeia "34";
- b) número 34;
- c) a cadeia "7";
- d) número 7;
- e) o mesmo de antes da execução do comando, devido a um erro referente à conversão de tipos

15. (FGV / TCE-SP – 2015) Em uma página HTML, é declarado o seguinte formulário:



```
<form id="dados" action="script.php"
      method="POST">
  <input type="submit" name="Submit"
        value="ENVIAR"/>
  <input type="hidden" id="tamanho"
        name="TAMANHO"
        value="" />
</form>
```

O valor do campo TAMANHO deverá ser preenchido por uma função Javascript, que terá acesso a ele através do construtor:

- a) document.getElementById("dados").tamanho.value;
- b) document.getElementById("hidden").value;
- c) document.getElementById("tamanho").value;
- d) document.getElementById("input").tamanho.value;
- e) document.getElementById("form").tamanho.value.

**16.(NUCEPE / SEFAZ-PI – 2015)** Laços são importantes construções em linguagens de programação. Qual tipo de laço (loop) NÃO possui suporte em JavaScript?

- a) For.
- b) For/in.
- c) Repeat/until.
- d) While.
- e) Do/while.

**17. (MPE-RS / MPE-RS – 2015)** Dado o código HTML abaixo:



```
<!DOCTYPE html>
<html>

<body>
<p>UM</p>
<p>DOIS</p>
<p>TRES</p>

<script type="text/javascript">
window.onload = function() {

.....

}

</script>
</body>
</html>
```

Quais dos comandos Javascript listados abaixo podem ser colocados no espaço indicado pelos pontos para que a página, ao ser carregada, altere somente a cor do texto "UM" para azul?

1. document.getElementsByTagName("p")[0].style.color = "#0000FF";
2. document.getElementsByTagName("p")[0].color = "#0000FF";
3. document.getElementsByTagName("p").item(0).style.color = "#0000FF";

- a) Apenas 1.
- b) Apenas 2.
- c) Apenas 3.
- d) Apenas 1 e 3.
- e) 1, 2 e 3.

18. (MPE-RS / MPE-RS – 2015) Dado o código Javascript abaixo:

```
<script type="text/javascript">
var numeros =
["Um", "Dois", "Tres", "Quatro"];
numeros.pop();
numeros.push("Dez");
numeros.shift();
document.write(numeros);
</script>
```



Quando o bloco for inserido no corpo de uma página HTML e executado no navegador, o que será exibido na tela?

- a) Dois,Tres,Dez
- b) Quatro,Dez,Dois,Tres
- c) Dois,Tres,Dez,Um
- d) Dez,Dois,Tres
- e) Um,Dois,Tres,Dez,Dez

19.(MPE-RS / MPE-RS – 2015) Dado o código Javascript abaixo:

```
<script type="text/javascript">
function Aluno() {
  var nome = "Ze";
  var dados = {
    idade : 18,
    peso : 75
  };
  this.getDados = function() {
    return dados;
  };
}
var x = new Aluno();
var y = x.getDados();
document.write(x.nome + " ");
document.write(y.idade + " ");
y.idade += 2;
document.write(x.getDados().idade);
</script>
```

Quando o bloco for inserido no corpo de uma página HTML e executado no navegador, o que será exibido na tela?

- a) undefined undefined 20
- b) undefined 18 20
- c) Ze 18 18
- d) Ze 18 20
- e) undefined 18 18

20.(MPE-RS / MPE-RS – 2015) Quais são os valores de saída do código Javascript abaixo?



```
<script type="text/javascript">  
<!--  
var x = 1;  
while (x < 20)  
{  
  if (x == 5){  
    break;  
  }  
  x = x + 1;  
  document.write( x + "<br />");  
}  
//-->  
</script>
```

- a) 2-3-4-5
- b) 2-3-4
- c) 1-2-3-4-5
- d) 1-2-3-4
- e) 0-1-2-3-4

21. (MPE-RS / MPE-RS – 2015) Dado o seguinte código JSON com Javascript:

```
var text = '{ "employees" : [ ' +  
'{ "firstName":"John" , "lastName":"Doe" },' +  
'{ "firstName":"Anna" , "lastName":"Smith" },' +  
'{ "firstName":"Peter" , "lastName":"Jones" } ]}';
```

É correto afirmar que o código apresentado acima:

- a) concatena três string formando uma classe e atribui dados a ela.
- b) concatena três objetos formando uma classe e atribui dados a ela.
- c) divide um objeto em três strings e atribui dados a elas.
- d) concatena três arrays em um objeto e atribui dados a ele.
- e) cria um array com três objetos e atribui dados a eles

22. (UFG / AL-GO – 2015) Leia o trecho em Javascript a seguir:



```
var a={},  
b={key:'b'},  
c={key:'c'};  
  
a[b]=123;  
a[c]=456;  
  
console.log(a[b]);
```

Qual é a saída da execução desse trecho de código Javascript?

- a) 123
- b) 456
- c) 156
- d) 423

23. (FGV / BANESTES – 2021) Considere o código javascript a seguir:

```
let txt = "";  
function funcao(value, index, array) {  
  
if (index % 2 == 0) {txt += value};  
  
}  
  
function xpto (x) {  
  
x.forEach(funcao);  
  
return txt;  
  
}  
  
alert (xpto([0, 1, 1, 2, 3, 5]));
```

A execução desse código provoca a exibição de:

- a) 011235
- b) 013
- c) 125
- d) 532110
- e) null

24. (FGV / BANESTES – 2021) Considere o código javascript a seguir:





```
document.getElementById('demo').innerHTML = Date()
```

Numa página web na qual esse código seja aplicado, o elemento que é compatível com a estrutura do comando para receber a data corrente é:

- a) <p "demo">H</p>
- b) <p class="demo">H</p>
- c) <p id="demo">H</p>
- d) <p js="demo">H</p>
- e) <p onclick="demo">H</p>

25. (CESPE / PGDF - 2021) Julgue o item a seguir, referente a linguagem de scripts.

```
<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

26. (CESPE / SEED-PR - 2021) resultado = (x for x in [5..2])

Assinale a opção que apresenta o resultado da execução do código CoffeeScript precedente.

- a) 5,5
- b) 5,4,3,2
- c) 2,2,2,2,2
- d) 5,3,1
- e) 4,3

27. (CESPE / SEED-PR - 2021) JavaScript trabalha com números usando os operadores aritméticos fornecidos pela própria linguagem. No entanto, a linguagem aceita operações matemáticas



mais complexas por meio de um conjunto de funções e constantes definidas como propriedades do objeto Math.

- 28. (CESPE / SEED-PR - 2021)** A opção `Math.ceil()` apresenta a função que permite realizar, em JS, a operação matemática de arredondar para cima o número 1,17, obtendo-se o valor 2.
- 29. (CESPE / ME - 2020)** Nos padrões web, as camadas deverão ser separadas de acordo com o objetivo para o qual elas foram desenvolvidas. Criar páginas em camadas lógicas é uma boa prática: xHTML, folhas de estilo CSS e JavaScript são voltadas, respectivamente, para as camadas de conteúdo, de apresentação visual do conteúdo e de comportamento dos elementos.
- 30. (CESPE / TJ-AM - 2019)** JavaScript é uma linguagem de programação orientada ao desenvolvimento da interface para aplicações web, cujo código-fonte é compilado pelo servidor antes de sua execução.
- 31. (CESPE / SLU-DF - 2019)** Uma função JavaScript é um bloco de código utilizado para executar tarefas repetidas e é definida pela palavra-chave `public` seguida por um nome seguido por parênteses `()`.
- 32. (CESPE / SLU-DF - 2019)** O JSX (JavaScript Syntax Extension) é de uso obrigatório no React e permite inserir a interface do usuário no código JavaScript.
- 33. (FCC / TRT 11 - 2005)** Em JavaScript, para declarar nomes de identificadores válidos utiliza-se:
- a) somente número no primeiro caractere.
  - b) somente letra ou underscore no primeiro caractere.
  - c) qualquer tipo de caractere a partir do segundo caractere.
  - d) espaço entre caracteres a partir do segundo caractere.
  - e) uma palavra reservada do JavaScript
- 34. (FCC / TRE-SE - 2007)** O cliente JavaScript, quando se soltar um botão do mouse pressionado, invocará o evento:
- a) `MouseOver`.
  - b) `MouseDown`.
  - c) `MouseUp`.
  - d) `MouseMove`.
  - e) `MouseOut`.
- 35. (FCC / MPE-SE - 2008)** Uma função embutida na linguagem JavaScript que calcula o conteúdo de uma string denomina-se:
- a) `string`.
  - b) `eval`.



- c) number.
- d) parseInt.
- e) parseFloat

**36.(FCC / TRT-1 - 2014)** O trecho de programa a seguir foi elaborado na linguagem HTML com JavaScript e será aberto por um navegador que as suporte.

```
<html>  
<body>  
<h1>Página Principal</h1>  
<form>  
comando y  
</form>  
</body>  
</html>
```



O comando que deve ser colocado no lugar do comando y, de modo que, ao se pressionar o botão Nova Página, seja aberta a página P1.html, é:

- a) `<input type = "button" text="Nova Página" click="window.location='P1.html'">`
- b) `<input type="button" text="Nova Página" onclick = "window.location='P1.html'">`
- c) `<input type="onbutton" text="Nova Página" onclick = "window.location='P1.html'">`
- d) `<input type="button" value="Nova Página" onclick = "window.location='P1.html'">`
- e) `<input type="onbutton" value="Nova Página" click = "window.location='P1.html'">`

**37.(FCC / TRT-18 - 2008)** Na linguagem JavaScript, os métodos de interface com o usuário que fazem parte do objeto window são apenas os denominados:

- a) select, submit e confirm.
- b) open, alert e write.
- c) click, select e write.
- d) open, submit e close.
- e) alert, confirm e prompt.

**38.(FCC / DPE-SP - 2010)** As tecnologias Web utilizam linguagem de scripting JavaScript, linguagem de scripting ASP e applets Java para incluir aplicações, respectivamente, no lado:



- a) servidor, cliente e servidor.
- b) cliente, servidor e cliente.
- c) servidor, servidor e cliente.
- d) cliente, cliente e servidor.
- e) cliente, cliente e cliente.

**39.(FCC / TRT-20 - 2010)** No JavaScript, a caixa de mensagem "Confirm":

- a) permite que o usuário insira um nome e/ou um número dentro da caixa de texto.
- b) Exibe apenas uma informação para o usuário.
- c) Permite que o usuário insira um nome dentro da caixa de texto.
- d) Permite que o usuário insira um número dentro da caixa de texto.
- e) Solicita uma confirmação do usuário, positiva ou negativa.

**40.(FCC – TRT-20 - 2010)** É um método do objeto History:

- a) Case.
- b) Left.
- c) Open.
- d) Forward
- e) Hostname



## GABARITO – JS

- |     |         |     |         |     |         |
|-----|---------|-----|---------|-----|---------|
| 1.  | LETRA A | 15. | LETRA C | 29. | ERRADO  |
| 2.  | LETRA A | 16. | LETRA C | 30. | ERRADO  |
| 3.  | LETRA A | 17. | LETRA D | 31. | ERRADO  |
| 4.  | LETRA A | 18. | LETRA A | 32. | ERRADO  |
| 5.  | LETRA B | 19. | LETRA B | 33. | LETRA B |
| 6.  | LETRA C | 20. | LETRA A | 34. | LETRA C |
| 7.  | LETRA D | 21. | LETRA E | 35. | LETRA B |
| 8.  | LETRA C | 22. | LETRA B | 36. | LETRA D |
| 9.  | LETRA D | 23. | LETRA B | 37. | LETRA E |
| 10. | LETRA A | 24. | LETRA C | 38. | LETRA B |
| 11. | LETRA D | 25. | CORRETO | 39. | LETRA E |
| 12. | LETRA B | 26. | LETRA B | 40. | LETRA D |
| 13. | LETRA A | 27. | CORRETO |     |         |
| 14. | LETRA A | 28. | CORRETO |     |         |



## ECMA 2021

### Conceitos Básicos

INCIDÊNCIA EM PROVA: BAIXA

ECMA é o padrão de definição da linguagem ECMAScript 2021, sendo a décima segunda especificação da linguagem ECMAScript, que teve início em 1996. Atualmente é uma das linguagens de programação de propósito geral mais utilizada, tendo em vista que já foi incorporada a maioria dos navegadores web, servidores e aplicativos. *E o que tem haver ECMA com Javascript, Professor?*



ECMAScript é baseado em várias tecnologias originárias, **sendo as mais conhecidas JavaScript (Netscape) e JScript (Microsoft)**. A linguagem foi inventada por Brendan Eich na Netscape e apareceu pela primeira vez no Navegador navegador 2.0. Ele apareceu em todos os navegadores subsequentes da Netscape e em todos os navegadores da Microsoft começando com o Internet Explorer 3.0.

O desenvolvimento da especificação de linguagem ECMAScript começou em novembro de 1996 e a primeira edição da Ecma foi adotada pela Assembleia Geral da Ecma em junho de 1997. ECMAScript é uma linguagem de programação orientada a objetos para realizar cálculos e manipular objetos computacionais dentro de um ambiente host.

Através de uma interface de usuário, a linguagem de ECMAScript consegue ser um mecanismo simples para expor funcionalidades para o ambiente de programação. Lembre-se que qualquer



linguagem de script se destina ao uso por programadores profissionais e não profissionais, para automatizar rotinas.

O ECMAScript foi originalmente **projetado para ser uma linguagem de script para automatização na Web, fornecendo um mecanismo para animar as páginas da Web em navegadores e para executar scripts do lado servidor como parte de uma arquitetura cliente-servidor baseada na Web.**

ECMAScript atualmente é usado para fornecer recursos de script para uma variedade de ambientes de host. O uso do ECMAScript foi evoluindo com o tempo, e hoje vai além do script simples no ambiente web, sendo empregado para todo o espectro de tarefas de programação em muitos ambientes e escalas diferentes. À medida que o uso do ECMAScript se expandiu, os recursos e facilidades que oferece também se expandiram.

*Mas professor por que precisa existir Javascript e ECMAScript ?*



Só você pensar o seguinte: Imagina o caos que seria cada navegador com sua própria linguagem, meio o que acontece hoje com a web, mas isso é outro assunto. A ideia da web é funcionar para todos e para isso precisamos de uma linguagem que permeie comportamento comuns a todos os navegadores. Para manter um padrão, a Netscape enviou o JavaScript à ECMA, que é a associação que cuida do ECMAScript.

Já que o nome JavaScript pertencia à Sun, registrou-se então um novo nome um padrão universal era o ECMAScript. Foi assim então que surgiu o nome ECMAScript. Porém, como o nome JavaScript sempre foi mais famoso, continuamos chamando a linguagem assim até hoje.

Galera, agora vamos começar a falar de termos mais técnicos! O documento do ECMAScript continuou evoluindo (ECMA – 262). Em 2015 foi lançado o ECMAScript 6, ECMAScript 7, 8, 9, etc. Todo ano então temos uma nova versão sendo lançada.

A nova versão 2021 traz muitas novas características, o que não considero razoável apresentar todas aqui, pelo simples fato que não existe uma cobrança pesada ao ponto de referenciar o esforço de vocês de aprender todo o ECMA 262. Então vamos as principais características, porém que quiser dar uma olhada na especificação completa, só acessar: <https://tc39.es/ecma262/#sec-intro>. Vamos então verificar o principal aspecto do ECMAScript 2021:

### ES: SYMBOL

O tipo Symbol é o conjunto de todos os valores não String que podem ser usados como chave de uma propriedade Object. Cada valor de símbolo possível é único e imutável.

ESPECIFICAÇÃO	DESCRIÇÃO
<b>@@ASYNCITERATOR</b>	Um método que retorna o AsyncIterator padrão para um objeto. Chamado pela semântica da declaração for-await-of.
<b>@@HASINSTANCE</b>	Um método que determina se um objeto construtor reconhece um objeto como uma das instâncias do construtor. Chamado pela semântica do operador instanceof.
<b>@@ISCONCATSPREADABLE</b>	Uma propriedade com valor booleano que, se true, indica que um objeto deve ser nivelado para seus elementos de matriz por Array.prototype.concat.
<b>@@ITERATOR</b>	Um método que retorna o Iterator padrão para um objeto. Chamado pela semântica da instrução for-of.
<b>@@MATCH</b>	Um método de expressão regular que corresponde à expressão regular em relação a uma string. Chamado pelo método String.prototype.match.
<b>@@MATCHALL</b>	Um método de expressão regular que retorna um iterador, que gera correspondências da expressão regular com uma string. Chamado pelo método String.prototype.matchAll.
<b>@@REPLACE</b>	Um método de expressão regular que substitui as substrings correspondentes de uma string. Chamado pelo método String.prototype.replace.
<b>@@SEARCH</b>	Um método de expressão regular que retorna o índice em uma string que corresponde à expressão regular. Chamado pelo método String.prototype.search.
<b>@@SPECIES</b>	Uma propriedade com valor de função que é a função construtora usada para criar objetos derivados.
<b>@@SPLIT</b>	Um método de expressão regular que divide uma string nos índices que correspondem à expressão regular. Chamado pelo método String.prototype.split.





<b>@@TOPRIMITIVE</b>	Um método que converte um objeto em um valor primitivo correspondente. Chamado pela operação abstrata ToPrimitive.
<b>@@TOSTRINGTAG</b>	Uma propriedade com valor de String que é usada na criação da descrição de string padrão de um objeto. Acessado pelo método interno Object.prototype.toString.
<b>@@UNSCOPABLES</b>	Uma propriedade com valor de objeto cujos nomes de propriedade próprios e herdados são nomes de propriedade que são excluídos das associações de ambiente do objeto associado.

*Professor, se eu entendo e lógica e indexação do Javascript, eu vou entender a do ECMAScript? **Sim, existe a mesma finalidade e a mesma estrutura, por isso as questões como veremos, estão 99% orientadas ao Javascript.*** O que você precisa guardar é que o escopo da finalidade de ambos...

**(UNIPAMPA – 2013)** ECMAScript é considerada uma versão anterior do JavaScript.

**Comentários:** conforme vimos, o ECMAScript é uma especificação formal de uma linguagem de script. Javascript é a implementação mais popular do ECMAScript (Errado).

**(COREN – 2018)** Em relação ao JavaScript (versão ECMAScript 2015 ou superior), é correto afirmar que para adicionar o array (arr2) ao final do array existente (arr1), era utilizada a instrução "Array.prototype.push.apply(arr1, arr2);" e atualmente, após a especificação ES2015, passa a ser "arr1.push(...arr2);".

**Comentários:** a simplificação apresentada diz respeito ao ECMAScript, e faz com que a instrução de adição de array, utilize o arr1.pus (...arr2), sem o prototype, logo a questão está perfeita (Correto).

**(COREN – 2018)** No ECMAScript, os tipos de dados primitivos são: bool, number, String, Symbol e closures.

**Comentários:** os tipos primitivos são no ECMAScript: string, number, null, undefined, boolean e symbol (Errado).



## QUESTÕES COMENTADAS – ECMAScript

1. (UFC / CCV - 2018) O que acontece quando executamos o código abaixo em um navegador com suporte à ECMAScript versão ES6 ou superior?

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

<script>
let teste = 200;
if ( teste > 100){
  let teste = 1;
}
console.log(teste);
</script>

</body>
</html>
```

- a) O valor da variável teste será sempre 200.
- b) Será exibido no console do navegador o valor 1.
- c) Será exibido dentro da janela navegador o valor 1.
- d) A instrução let introduz um escopo de bloco a variável teste.
- e) A declaração da variável teste utilizando let funcionaria da mesma forma se utilizássemos var.

### Comentários:

O erro da alternativa A) está em dizer que "teste" sempre será 200, o que é incorreto, visto que se invocarmos um `console.log(teste)` dentro do `if` teremos que teste é = 1. Isso acontece porque existe uma nova declaração da variável teste dentro do `if`, e quando chamamos uma variável dentro de um bloco de código o Javascript sempre "pegará" a variável que estiver mais perto do bloco. A B e C estão erradas porque não será exibido nem dentro do console e nem dentro da janela o valor 1. A D é nossa resposta correta, porque a instrução `let` é que introduz um escopo de bloco a variável teste no código e a E está errada, porque se utilizássemos `var` não seria o mesmo de utilizar `let`, são instruções distintas, `let` cria uma variável com escopo de bloco.

**Gabarito:** Letra D

2. (UFC / CCV - 2018) O que acontece quando executamos o código abaixo em um navegador com suporte à ECMAScript versão ES6 ou superior?



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

<script>
var ponto = [1,3],
    segmento = [ponto, [5,5]],
    triangulo = [...segmento,[1,8]];
console.log(triangulo.length)
</script>

</body>
</html>
```

- a) A variável triangulo terá o comprimento 1.
- b) A variável triangulo terá o comprimento 2.
- c) A variável triangulo terá o comprimento 3.
- d) A variável triangulo terá o comprimento 4.
- e) A variável triangulo terá o comprimento 6.

#### Comentários:

Questão de raciocínio lógico! Vejamos um triângulo tem três lados, e temos as variáveis: ponto (1,3), segmento [ponto, [5,5]] e segmento [1,8]], agora vamos de frente para trás, se length é o comprimento para o triângulo (triangulo.length), então: triangulo, concatena segmento, que concatena ponto, logo recebemos 3 de length.

**Gabarito:** Letra C

**3. (UFC / CCV - 2018)** O que acontece quando executamos o código abaixo em um navegador com suporte à ECMAScript versão ES6 ou superior?



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

<p id="demo"></p>

<script>
var obj = { name: "Joao", age: 20, city: "Fortaleza" };
var myJSON = JSON.stringify(obj);
document.getElementById("demo").innerHTML = myJSON;
</script>

</body>
</html>
```

- a) Será exibido na janela do navegador o seguinte texto: [object Object].
- b) Será exibido no console do navegador o seguinte texto: [object Object].
- c) Será exibido na janela do navegador o seguinte texto: Joao, 20, Fortaleza.
- d) Será exibido na janela do navegador o seguinte texto: {"name":"Joao","age":20,"city":"Fortaleza"}.
- e) Será exibido no console do navegador o seguinte texto: {"name":"Joao","age":20,"city":"Fortaleza"}.

#### Comentários:

Ao compilar o código, teremos os elementos: {"name":"Joao","age":20,"city":"Fortaleza"} na janela do navegador e não no console.

**Gabarito:** Letra D

4. (COVED / UFPE - 2019) Considere a execução do código JavaScript abaixo, compatível com o ECMAScript 6, e a respectiva numeração das linhas de código na coluna à esquerda, e assinale a alternativa correta.



1	var a = 5;
2	var b = 10;
3	if (a === 5) {
4	let a = 4;
5	var b = 1;
6	console.log(a);
7	console.log(b);
8	}
9	console.log(a);
10	console.log(b);

Após a execução do código, desconsiderando os caracteres de quebra de linha da função `console.log()`, o console apresentará:

- a) as saídas: 4,1,5 e 1
- b) as saídas: 4,1,4 e 1
- c) as saídas: 4,1,4 e 10
- d) uma mensagem de erro referente a linha 3
- e) uma mensagem de erro referente a linha 4

### Comentários:

Pessoal, vamos linha a linha:

Primeiro, iniciamos as variáveis e atribuímos os valores 5 e 10 (linha 1 e 2). Depois usamos a condição (IF e colocamos `==` que é um operador de igualdade, ou seja, verifica se a realmente é igual 5, com objetivo de inicializar o laço condicional. Em seguida temos o `let`, `let` transforma a variável como escopo de bloco. Ex:

```
let escopoBloco = 'Pedro';  
console.log (escopoBloco); // Pedro
```

```
Logo: console.log(a); // 4 console.log(b); // 1 console.log(a); // 5 console.log(b); // 1
```

**Gabarito:** Letra A

**5. (UFPR / COREN - 2018)** Em relação ao JavaScript (versão ECMAScript 2015 ou superior), é correto afirmar:

- a) A expressão `"var func = x => x * x;"` define a variável `x` apontando para a função `func` que calcula a raiz quadrada de `x`.



- b) Template literals ou template strings são literais string que permitem expressões embutidas, nas quais os textos são delimitados por aspas simples e as expressões indicadas e delimitadas por `#{expressão}`.
- c) Para adicionar o array (arr2) ao final do array existente (arr1), era utilizada a instrução `Array.prototype.push.apply(arr1, arr2);` e atualmente, após a especificação ES2015, passa a ser `arr1.push(...arr2);`.
- d) No ECMAScript 6, os tipos de dados primitivos são: bool, number, String, Symbol e closures.
- e) A função focus do objeto Window, quando o script é executado nos navegadores da web, remove o foco da janela atual.

### Comentários:

(a) Errado. Para definir a variável x precisaria ser `var func = function (x) {return x*x}` e não `var func = x => x*x;` (b) Errado. Template Literal é delimitado por acento grave (``) e a interpolação de expressão é feita com ``${}``; (c) Correto. Pessoal algumas questões você vão ter que descobrir as erradas e ir excluindo, é humamente improvável que alguém soubesse no detalhe a especificação ES2015 aplicada a instrução `Array.prototype.push.apply(arr1, arr2);` neste caso vale a regra: se a questão não tem erro, provavelmente ela é a correta; (d) Errado. No ECMAScript 6, os tipos de dados primitivos são: string, number, undefined, null, boolean, symbol; (e) Errado. Ao contrário, a função focus adiciona foco e não remove o foco da janela atual.

**Gabarito:** Letra C



## LISTA DE QUESTÕES – ECMASCRIPT

1. (UFC / CCV - 2018) O que acontece quando executamos o código abaixo em um navegador com suporte à ECMAScript versão ES6 ou superior?

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

<script>
let teste = 200;
if ( teste > 100){
  let teste = 1;
}
console.log(teste);
</script>

</body>
</html>
```

- a) O valor da variável teste será sempre 200.
  - b) Será exibido no console do navegador o valor 1.
  - c) Será exibido dentro da janela navegador o valor 1.
  - d) A instrução let introduz um escopo de bloco a variável teste.
  - e) A declaração da variável teste utilizando let funcionaria da mesma forma se utilizássemos var.
2. (UFC / CCV - 2018) O que acontece quando executamos o código abaixo em um navegador com suporte à ECMAScript versão ES6 ou superior?



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

<script>
var ponto = [1,3],
    segmento = [ponto, [5,5]],
    triangulo = [...segmento,[1,8]];
console.log(triangulo.length)
</script>

</body>
</html>
```

- a) A variável triangulo terá o comprimento 1.
- b) A variável triangulo terá o comprimento 2.
- c) A variável triangulo terá o comprimento 3.
- d) A variável triangulo terá o comprimento 4.
- e) A variável triangulo terá o comprimento 6.

3. (UFC / CCV - 2018) O que acontece quando executamos o código abaixo em um navegador com suporte à ECMAScript versão ES6 ou superior?

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

<p id="demo"></p>

<script>
var obj = { name: "Joao", age: 20, city: "Fortaleza" };
var myJSON = JSON.stringify(obj);
document.getElementById("demo").innerHTML = myJSON;
</script>

</body>
</html>
```

- a) Será exibido na janela do navegador o seguinte texto: [object Object].
- b) Será exibido no console do navegador o seguinte texto: [object Object].
- c) Será exibido na janela do navegador o seguinte texto: Joao, 20, Fortaleza.
- d) Será exibido na janela do navegador o seguinte texto: {"name":"Joao","age":20,"city":"Fortaleza"}.





e) Será exibido no console do navegador o seguinte texto:  
{"name":"Joao","age":20,"city":"Fortaleza"}.

4. (COVED / UFPE - 2019) Considere a execução do código JavaScript abaixo, compatível com o ECMAScript 6, e a respectiva numeração das linhas de código na coluna à esquerda, e assinale a alternativa correta.

1	var a = 5;
2	var b = 10;
3	if (a === 5) {
4	let a = 4;
5	var b = 1;
6	console.log(a);
7	console.log(b);
8	}
9	console.log(a);
10	console.log(b);

Após a execução do código, desconsiderando os caracteres de quebra de linha da função console.log(), o console apresentará:

- a) as saídas: 4,1,5 e 1
- b) as saídas: 4,1,4 e 1
- c) as saídas: 4,1,4 e 10
- d) uma mensagem de erro referente a linha 3
- e) uma mensagem de erro referente a linha 4

5. (UFPR / COREN - 2018) Em relação ao JavaScript (versão ECMAScript 2015 ou superior), é correto afirmar:

- a) A expressão "var func = x => x \* x;" define a variável x apontando para a função func que calcula a raiz quadrada de x.
- b) Template literals ou template strings são literais string que permitem expressões embutidas, nas quais os textos são delimitados por aspas simples e as expressões indicadas e delimitadas por #{expressão}.
- c) Para adicionar o array (arr2) ao final do array existente (arr1), era utilizada a instrução "Array.prototype.push.apply(arr1, arr2);" e atualmente, após a especificação ES2015, passa a ser "arr1.push(...arr2);".
- d) No ECMAScript 6, os tipos de dados primitivos são: bool, number, String, Symbol e closures.



e) A função focus do objeto Window, quando o script é executado nos navegadores da web, remove o foco da janela atual.



## GABARITO – ECMAScript

- |    |         |    |         |    |         |
|----|---------|----|---------|----|---------|
| 1. | LETRA D | 3. | LETRA D | 5. | LETRA C |
| 2. | LETRA C | 4. | LETRA A |    |         |



# JQUERY

## Conceitos Básicos

INCIDÊNCIA EM PROVA: MÉDIA

**Pessoal, é muito comum construir uma página web utilizando HTML e estilizá-la utilizando CSS!** As páginas ficam realmente muito bonitas, mas não são muito interativas, i.e., nós não conseguimos arrastar elementos pela página, abrir ou fechar painéis deslizantes, animar elementos HTML ou adicionar novos elementos a nossa página simplesmente clicando em um botão.

E é para isso que surgiu o jQuery! **Ele se trata de uma biblioteca JavaScript cross-browser, de código aberto, pequena, leve, rápida, fácil e muito rica.** Ela foi desenvolvida para simplificar scripts client-side que interagem com HTML, criar animações, manipular eventos e desenvolver aplicações AJAX. Galera, agora fica muito fácil utilizar JavaScript em seu website.

*Por que, professor?* **Porque ele pega um bocado de tarefas muito comuns que requerem várias linhas de código em JavaScript e envolve tudo em métodos que podem ser chamados com uma única linha de código.** Ele também simplifica um monte de coisas complicadas do JavaScript, como manipulação de AJAX, DOM, CSS, AJAX, entre outras utilidades, eventos e chamadas (e ainda suporta plug-ins).

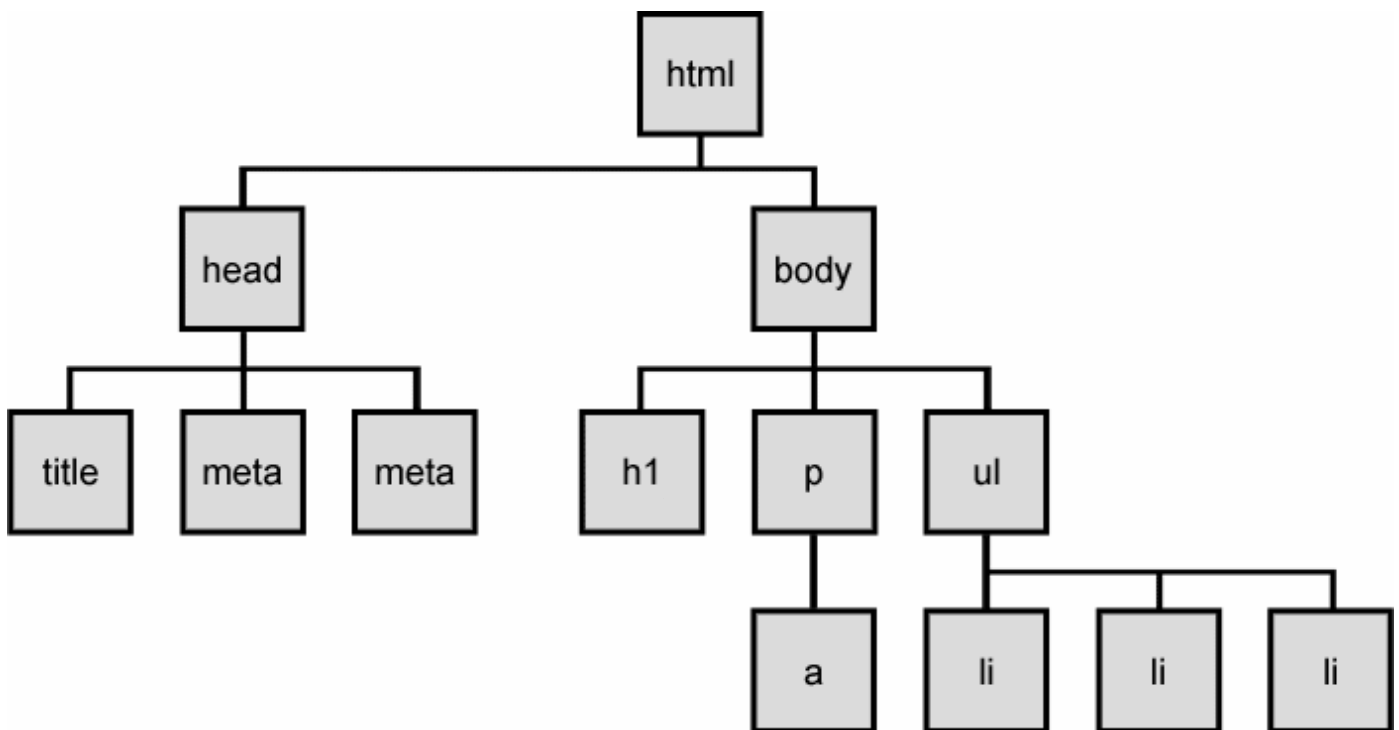
**Pode parecer contra intuitivo aprender a usar uma biblioteca antes de aprender a linguagem de fato, mas há algumas excelentes razões para fazer isso.** Em primeiro lugar, demora um pouco até que se fique familiarizado e confortável com JavaScript, e é mais difícil manipular elementos HTML diretamente por JavaScript do que por jQuery.

Em segundo lugar, jQuery oferece uma interface simples para o básico de JavaScript. Para muitos usuários é mais fácil aprender jQuery primeiro e então mergulhar nos detalhes profundos de JavaScript mais tarde. **Em terceiro lugar, podemos dizer que jQuery é muito melhor no que diz respeito a dar resultados visuais imediatos do que JavaScript.**

*Professor, o que é DOM?* **É uma convenção multiplataforma que representa como as marcações em HTML, XHTML e XML são organizadas e lidas pelo navegador.** Uma vez indexadas, estas marcações se transformam em elementos de uma árvore que você pode manipular via API – que é o que fazemos quando usamos programas ou scripts para conteúdo, estrutura ou folha de estilo de uma página.

O DOM possui uma estrutura de árvore (assim como HTML, XHTML e XML). **É assim que o navegador monta! Professor, não é isso que eu vejo quando acesso uma página HTML!** É verdade, o DOM é somente a base para uma outra árvore que é o que realmente um browser monta em tela – é a Árvore de Renderização (ou Render Tree). Abaixo, podemos ver uma Árvore DOM (de um HTML).





Isso tudo foi para dizer que é por meio da interação com o DOM que o jQuery consegue acessar e modificar HTML. Pessoal, quem já viu jQuery funcionando sabe! **É muito legal, ele simplesmente consegue mudar a estrutura de uma página dinamicamente.** Infelizmente, por PDF é impossível demonstrar animações funcionando, eu recomendo que vocês busquem compiladores online.

Vamos falar de sintaxe! Para utilizar o jQuery, é preciso fazer um link da nossa página HTML para o nosso script em jQuery (similar ao que fazemos com HTML/CSS). Para tal, podemos utilizar o elemento `<script></script>`, como é apresentado abaixo. **No entanto, para fazer a linguagem efetivamente funcionar, é necessário utilizar uma sintaxe básica também apresentada a seguir:**

```
<script type="text/javascript" src="script.js"><script>
```

```
$(seletor).ação()
```

Isso significa que o jQuery realizará uma ação sobre um determinado seletor. No entanto, a primeira sintaxe de um arquivo .js é sempre a mesma:

```
$(document).ready(ação);
```

O que isso quer dizer? **Isso significa: quando o Documento HTML (*document*) estiver pronto (*ready*), faça algo (*ação*).** Isso serve para prevenir que qualquer código jQuery rode antes de o documento estar completamente carregado. É uma boa prática esperar para, então, trabalhar com ele. Isso também permite que você tenha seu código JavaScript antes do corpo do seu documento (na seção head).



Bacana, agora vamos colocar alguma coisa na função `ready()`. Quando eu digo função, pensem em ação. Funções são a unidade básica para trabalhar com jQuery. **Por essa razão, ele inclui uma palavra reservada `function`.** Segue a sintaxe:

```
function(){  
    Faça algo;  
}
```

Se adicionarmos essa função dentro da função `ready()`, o jQuery executará o código da nossa função assim que o documento HTML for carregado. **Vamos destrinchar a sintaxe a seguir:** `$( )` é uma função disfarçada que transforma `document` em um objeto jQuery; `.ready()` é um tipo de função que ajuda a executar o código dentro dos parênteses assim que o documento estiver pronto.

```
$(document).ready(function(){  
    Faça algo;  
});
```

**Ainda não acabou: `function( ) { }` é uma ação que `.ready( )` desempenhará assim que o Documento HTML estiver carregado.** Em nosso exemplo, a função substituirá o `Faça algo`. Beleza! Então se eu quiser fazer um cabeçalho `div` do HTML ser ocultado imediatamente, basta que eu escreva um jQuery seguindo a sintaxe apresentada a seguir (tentem entender!).

```
$(document).ready(function(){  
    $('div').hide();  
});
```

*Vocês estão acompanhando? Imaginem que eu tenho um documento HTML que cria um **div vermelho em uma tela em branco**. O código acima está dizendo que assim que o documento estiver pronto, será executada a função `hide()` sobre o elemento `div` do documento HTML. *E o que essa função faz?* Ela faz o elemento sumir (ocultar) imediatamente após sua execução. *Simples, não?**

**Agora chegamos a uma parte importante da aula:** Seletores! Nós utilizamos o `div` como exemplo, mas poderíamos usar qualquer seletor CSS entre aspas e passá-lo para `$( )`, inclusive classes. *Vocês se lembram que nós podemos selecionar classes em CSS utilizando um ponto?* Pois é, se temos `class="xpto"` em HTML, podemos selecioná-la em CSS com o operador ponto: `.xpto`.

SINTAXE	DESCRIÇÃO
<code>\$("#*")</code>	Seleciona todos os elementos.
<code>\$(THIS)</code>	Seleciona o elemento HTML atual.
<code>\$("#P.INTRO")</code>	Seleciona todos os elementos <code>&lt;p&gt;</code> com classe <code>intro</code> .
<code>\$("#P:FIRST")</code>	Seleciona o primeiro elemento <code>&lt;p&gt;</code> .



<code>\$("UL LI:FIRST")</code>	Seleciona o primeiro elemento <li> do primeiro <ul>.
<code>\$("UL LI:FIRST-CHILD")</code>	Seleciona o primeiro elemento <li> de todos <ul>.
<code>\$("[HREF]")</code>	Seleciona todos os elementos com um atributo href.
<code>\$("A[TARGET='_BLANK']")</code>	Seleciona todos os elementos <a> com atributo target com valor igual a "_blank".
<code>\$("A[TARGET!='_BLANK']")</code>	Seleciona todos os elementos <a> com um atributo target com valor diferente de "_blank".
<code>\$(":BUTTON")</code>	Seleciona todos os elementos <button> e elementos <input> do tipo button.

Em jQuery, tudo o que precisamos fazer é colocar o seletor entre aspas e, então, podemos passá-lo para `$()` para criar um objeto jQuery. Há acima uma lista com todos os possíveis seletores. Lembrando que existe também diversos efeitos, mas que são impossíveis de demonstrar em texto (Hide/Show, Fade, Slide, Animate, Stop, Callback, Chaining), **recomendo que vocês testem em um compilador online.**

Existe uma biblioteca chamada JQuery UI **que contém um conjunto de widgets de interface gráfica, efeitos visuais animados e diversos temas implementados com JQuery.** Trata-se de uma biblioteca que facilita a criação de elementos de interface de usuário com um mínimo de código e sintaxe idêntica à da biblioteca jQuery, compatível com padrões web. É usada pelo Paypal, IMDB e Netflix. Uma importante função do JQuery é `.each` que realiza uma função callback

Por fim, vamos falar rapidamente sobre um importante plug-in: JQGrid. Ele é um **controle JQuery ativado por Ajax que fornece soluções para representar e manipular dados tabulares (tabelas) na web.** Uma vez que a grid é uma solução *client-side* carregando dados dinamicamente por chamadas AJAX, ela pode ser integrada com qualquer tecnologia *server-side* (PHP, ASP, JSP, Servlet, etc).

Galera, há um concorrente do JQuery chamado Dojo! **Ele também é uma biblioteca, tem código-aberto, é modular, possui sintaxe simplificada, apresenta funções para chamadas AJAX, é capaz de manipular o DOM, contém diversos componentes gráficos e efeitos visuais e pode realizar diversas validações** – proporcionando um código mais legível e menor.

**O Dojo é menos extensível que o JQuery, mas seu escopo é bem maior, isto é, ele faz praticamente tudo que o JQuery faz e mais um pouco.** O JQuery é mais leve para o navegador, mas não esqueçamos que ele é extremamente extensível, logo eu posso encher ele de plug-ins – ficando, assim, mais pesado. Se você quer fazer aplicações web ricas, você estará bem servido com qualquer um desses.

(TRT6 – 2014) Considere a página abaixo, construída utilizando jQuery:

```
<!DOCTYPE html> <html> <head> <meta http-
```



```
equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Home</title>
    <script
type="text/javascript"
src="http://ajax.microsoft.com/ajax/jquery/jquery-1.4.2.min.js"></script>
    <script
type="text/javascript">
        $(document).ready(function
        ()
        {
            ..I...
        });
    </script>
</head>
<body>
    <button
type="button">Esconder</button>
    <p class="par">Texto do
parágrafo.</p>
</body>
</html>
```

Para que, ao se clicar no botão Esconder, o texto do parágrafo se torne oculto, deve-se preencher a lacuna I com:

- a) \$('button').click(function(){\$('#p#par').hide();});
- b) \$('button:eq(0)').click(function().defineProperty('p.par').hide());
- c) \$('button').click(function(){\$('#p.par').hidden();});
- d) \$('button:eq(1)').click(function(){\$('#p#par').hide();});
- e) \$('button').click(function(){\$('#p.par').hide();});

**Comentários:** parece bem complicada essa questão pela quantidade de código exposto, mas não se engane, a questão é simples, veja: a questão pede que o parágrafo se torne oculto, para isso utilizamos o método hide. Nisso já eliminamos a letra C. Precisamos então com o método hide selecionar os elementos que precisam ficar ocultos, esse elementos estão na classe .par. Logo, basta chamar: \$('button').click(function(){\$('#p.par').hide();}); (Letra E).

**(Prefeitura de São Paulo – 2016)** A função .each () do jQuery faz com que:

- a) um atributo seja definido em cada um dos elementos selecionados.
- b) uma função seja aplicada em apenas um dos elementos selecionados.
- c) todos os subelementos de um elemento especificado sejam selecionados.
- d) uma função de callback seja invocada no contexto de cada um dos elementos selecionados.
- e) uma seleção de elementos seja dividida em um vetor com elementos individuais.

**Comentários:** a função .each () do Jquery é uma das mais importante e faz com que uma função de call-back seja invocada no contexto de cada um dos elementos selecionados (Letra D).





## QUESTÕES COMENTADAS – JQUERY

1. (FCC/ MPE-AP - 2012) O seletor jQuery:

- a) \$("[href!='#']") seleciona todos os elementos cujo atributo href não contenha '#'.
- b) \$("[href\$.jpg]") seleciona todos os elementos cuja propriedade href contenha '.jpg'.
- c) \$("div#intro .head") seleciona todos os elementos com id="head" dentro de um elemento com class="intro".
- d) \$("p#intro:first") seleciona o primeiro elemento com class="intro".
- e) \$("ul li:first-child") seleciona todos os elementos do primeiro elemento

Comentários:

SINTAXE	DESCRIÇÃO
<code>\$(".href")</code>	Seleciona todos os elementos com um atributo href.
<code> \$("ul li:first-child")</code>	Seleciona o primeiro elemento <li> de todos <ul>.

Vamos por partes: \$("[href!='#']"). Trata-se de um seletor, que vai selecionar todos os elementos que contenham o atributo href diferente (por conta do operador '!=') de '#'. Já a última opção seleciona o primeiro elemento <li> de todos os elementos <ul>. Todas as outras opções não fazem nenhum sentido.

**Gabarito:** Letra A

2. (VUNESP/ TCE-SP - 2015) O método do jQuery que permite ocultar um elemento selecionado é:

- a) .show(false)
- b) .hide()
- c) .mask()
- d) .visible(false)
- e) .display(false)

Comentários:

Trata-se do hide(): ele faz o elemento sumir (ocultar) imediatamente após sua execução.

**Gabarito:** Letra B

3. (FCC/ TRT-13 – 2014) Considere a página abaixo, construída utilizando jQuery:



```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">
    <title>Home</title>
    <script type="text/javascript"
src="http://ajax.microsoft.com/ajax/jquery/jquery-1.4.2.min.js"></script>
    <script type="text/javascript">
      $(document).ready(function() {
        ..I...
      });
    </script>
  </head>
  <body>
    <button type="button">Esconder</button>
    <p class="par">Texto do parágrafo.</p>
  </body>
</html>
```

Para que, ao se clicar no botão Esconder, o texto do parágrafo se torne oculto, deve-se preencher a lacuna I com:

- a) \$('button').click(function(){\$('p#par').hide();});
- b) \$('button:eq(0)').click(function().defineProperty('p.par').hide());
- c) \$('button').click(function(){\$('p.par').hidden();});
- d) \$('button:eq(1)').click(function(){\$('p#par').hide();});
- e) \$('button').click(function(){\$('p.par').hide();});

### Comentários:

- Para selecionar o elemento button: \$('button');
- Para tratar de um clique: .click( );
- Para tratar da ação em si: function( );
- Para selecionar o parágrafo: \$('p.par');
- Para ocultar o parágrafo: .hide( );

*O que aconteceu, galera? Vamos ler de trás para frente: o código está dizendo que é para ocultar .hide( ) o parágrafo \$('p.par') quando clicarem function( ).click( ) no botão \$('button').*

**Gabarito:** Letra E



4. (CESPE/ ANAC – 2012) Em jQuery, o método `hide()` permite ocultar elementos HTML. Assim, a sintaxe correta para ocultar o elemento corrente é `$(this).hide()` e, para ocultar todos os elementos `<p>` da página, é `$("#<p>").hide()`.

#### Comentários:

Há um erro nessa questão: não é necessário colocar `<p>`, mas apenas `p`, isto é, jQuery interpreta apenas o elemento - `$("#p").hide()`.

**Gabarito:** Letra E

---

5. (ESAF / CGU – 2012) São plug-ins da biblioteca jQuery destinados a criar funcionalidade para a interface do usuário:

- a) Plug-ins nativos para interação. Widgets. Effects.
- b) Plug-ins interativos tipo slider. Wide-rangers. Effects.
- c) Plug-ins nativos da família datepicker. MainWidgets. Efforts.
- d) Widgets nativos para marcação. Widgets de monitoramento. Plug-ins de localização.
- e) Plug-ins nativos para interação. Upload widgets. Effective plug-ins.

#### Comentários:

Existem plug-ins nativos de interação, widgets e effects.

**Gabarito:** Letra A

---

6. (FUNDEP / UFVMJ – 2017) O JavaScript é uma linguagem leve, interpretada e baseada em objetos, mais conhecida como a linguagem de script para páginas Web. Assinale a alternativa que apresenta corretamente uma biblioteca de JavaScript.

- a) JQuery
- b) HTML
- c) AJAX
- d) CSS

#### Comentários:

*Qual a biblioteca de Javascript mais famosa? JQuery! :)*

**Gabarito:** Letra A

---

7. (AOCP – TCE-PA - 2012) O nome dado à biblioteca JavaScript cross-browser desenvolvida para simplificar os scripts executados no lado do cliente e que interagem com o HTML é:



- a) JSP.
- b) J2ME.
- c) JSON.
- d) J2EE.
- e) JQuery

#### Comentários:

*Biblioteca JavaScript cross-browser mais famosa? JQuery! O que é cross-browser, professor? É a habilidade de um site, aplicação web ou script side-client suportar múltiplos navegadores.*

**Gabarito:** Letra E

8. (FCC / MPE-PB – 2015) Considere o bloco de código a seguir:

```
<script>
$(document).ready(function(){
    $("div").click(function(){
        $(this).hide();
    });
});
</script>
```

Este bloco de código utiliza uma biblioteca JavaScript criada para simplificar a programação. Trata-se da biblioteca conhecida como:

- a) jUnit.
- b) AngularJS.
- c) jSON.
- d) AJAX.
- e) jQuery.

#### Comentários:

Questão trata de... JQuery! *Como você sabe, professor?* Pela sintaxe, galera! O comando `$(document).ready(function()` verifica se o documento está pronto para ser manipulado. O comando `$("div").click(function()` seleciona uma determinada div que, ao ser clicada, esconde um objeto.

**Gabarito:** Letra E

9. (ESAF / CGU – 2012) Assinale a opção correta relativa a jQuery:



- a) Não admite programação encadeada.
- b) Utiliza indicadores SSP para localizar elementos componentes da estrutura de marcação jHTML da página.
- c) Possui arquitetura compatível com instalação de plug-ins e extensões em geral.
- d) Possui estrutura para armazenamento de inputs de especificação multicore.
- e) Utiliza pop-ups C2S para retirar elementos inconsistentes da estrutura de marcação XHTML da referência.

### Comentários:

O Dojo é menos extensível que o JQuery, mas seu escopo é bem maior, isto é, ele faz praticamente tudo que o JQuery faz e mais um pouco. O JQuery é mais leve para o navegador, mas não esqueçamos que ele é extremamente extensível, logo eu posso encher ele de plug-ins – ficando, assim, mais pesado. Se você quer fazer aplicações web ricas, você estará bem servido com qualquer um desses.

(a) Errado, permite programação encadeada, i.e., o famoso Method Chaining. O que é isso, professor? É a capacidade da sintaxe de uma linguagem de invocar múltiplos métodos com cada método retornando um objeto (Ex: `$('#elemento').empty().append('<div id="nova_div">').find('#nova_div').css("background-color", "#foo");`); (b) Errado, esse item não faz o menor sentido; (c) Correto, é uma das grandes vantagens; (d) Errado, JQuery não suporta programação em threads (multicore); (e) Errado, *pop-ups C2S*? Nada disso, questão viajante.

**Gabarito:** Letra C

**10. (IF/PE – IF/PE - 2016)** O comando abaixo escrito em jQuery faz com que,

```
$( ':text' ).blur( function() {  
    $( "p" ).css( "border", "3px dashed red" );  
});
```

- a) quando o elemento text for selecionado na página, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.
- b) quando o elemento de ID text perder o foco sobre ele, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.
- c) quando um input text ganhar o foco sobre ele, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.



d) quando um input text perder o foco sobre ele, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.

e) quando qualquer input text for modificado na página, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.

### Comentários:

`$( ':text' ).` = esse comando utiliza um pseudo-seletor para selecionar um elemento input text; `.blur(function() {` = quando esse elemento perder foco; `$( "p" ).css( "border", "3px dashed red" )` = adicione uma borda de 3px vermelho e tracejada em todas as tags <p> da página.

**Gabarito:** Letra D

**11. (AOC/ INES – 2013)** Sobre as características da Biblioteca jQuery, assinale a alternativa INCORRETA.

a) A principal limitação desta biblioteca é a necessidade de instalação de um plug-ins, em cada navegador que fará acesso a páginas com suas funcionalidades.

b) Utiliza seletores CSS para localizar elementos componentes da estrutura de marcação HTML da página.

c) Possui arquitetura compatível com instalação de plugins e extensões em geral.

d) É indiferente às inconsistências de renderização entre navegadores.

e) Admite programação encadeada, ou seja, cada método retorna um objeto.

### Comentários:

(a) Errado, não é obrigatória a instalação de plug-ins e não precisa instalar em cada navegador. Você baixa a biblioteca, coloca em sua página e referencia no Documento HTML; (b) Correto, é possível utilizar seletores CSS para localizar elementos; (c) Correto, ele é extensível por plug-ins; (d) Correto, ele é de fato indiferente às inconsistências entre navegadores; (e) Correto, programação encadeada é o method chaining, ou seja, métodos encadeados que retornam objetos.

**Gabarito:** Letra A

**12. (CESPE / TCE-PA – 2016)** Após a incorporação do jQuery ao HTML5, o desenvolvimento de funcionalidades por meio dessa biblioteca JavaScript ficou limitado a aplicações para dispositivos móveis.



## Comentários:

*Einh? Tá maluco?* Nada de limitado a aplicações para dispositivos móveis! Inclusive para aplicações para dispositivos móveis, nós temos o JQuery Mobile. Questão errada.

**Gabarito:** Errado

**13. (FGV / IBGE – 2017)** Para definir a cor de fundo de todos os elementos <p> para azul em uma página HTML, o código jQuery correto é:

- a) `$(this).css("background-color","blue");`
- b) `$("#p").css("background-color"," blue");`
- c) `$("#p").css ("background-color"," blue");`
- d) `$("#p").css("background-color"," blue");`
- e) `(".p").css("background-color"," blue");`

## Comentários:

Para definir a cor de fundo de todos os elementos <p>, nós temos que primeiro selecioná-lo e, para selecionar, nós usamos `$("#p")`. Após isso, nós utilizamos o método `.css("background-color","blue")` para colorir a cor de fundo de azul.

**Gabarito:** Letra C

**14. (QUADRIX / CRO-PR – 2016)** Um código em Javascript puro, para atribuir o valor "3" a um elemento qualquer, ficaria assim:

```
document.getElementById( 'Concurso' ).value = 3;
```

Como ficaria esse mesmo código em jQuery?

- a) `#Concurso:=3;`
- b) `$("#Concurso").val( 3 );`
- c) `$(Concurso).val( 3 );`
- d) `$('#Concurso').val( 3 );`
- e) `$('#Concurso')=3`

## Comentários:

Esse código atribui o valor 3 para o elemento cujo Id seja "Concurso". Logo, em JQuery, será `$("#Concurso").val(3)`.

**Gabarito:** Letra D



**15. (FCC / TRE-SP – 2017)** Considere uma página HTML contendo a instrução `<div id="d1" ></div>`. Para mostrar no interior deste contêiner, utilizando jQuery, o tamanho do contêiner, utiliza-se a instrução:

- a) `$("#d1").innerHTML($("#d1").width());`
- b) `$("#d1").html($("#d1").size());`
- c) `$("#d1").show($("#d1").width());`
- d) `$("#d1").getElementByld("d1").innerHTML= $("#d1").width();`
- e) `$("#d1").html($("#d1").width());`

### Comentários:

(a) Errado, essa propriedade retorna o conteúdo de um HTML, porém é usada no JavaScript puro; (b) Errado, esse elemento – sim – retorna o conteúdo de um HTML no JQuery (equivalente ao innerHTML do JavaScript) e esse item estaria correto se fosse width() em vez de size(); (c) Errado, o método show() mostra um elemento - estaria correto se fosse html(); (d) Errado, getElementByld é utilizado no JavaScript e, não, no JQuery; (e) Correto, é exatamente assim :)

**Gabarito:** Letra E

**16. (AOCP / INES – 2013)** Sobre a Biblioteca jQuery analise as assertivas e assinale a alternativa que aponta a(s) correta(s).

I. jQuery é uma biblioteca JavaScript Criada por John Resig e disponibilizada como software livre e aberto, ou seja, de emprego e uso regido segundo licença conforme as regras estabelecidas pelo MIT e pelo GPL.

II. jQuery destina-se a adicionar interatividade e dinamismo às páginas web, proporcionando ao desenvolvedor funcionalidades necessárias à criação de scripts que visem a incrementar, de forma progressiva e não obstrutiva, a usabilidade e a acessibilidade e o design, enriquecendo a experiência do usuário.

III. Diferentemente do JavaScript sozinho, a biblioteca jQuery pode ser utilizada do lado do servidor para fazer acesso à base de dados através de um ODBC.

IV. Com jQuery em uma página Web podemos adicionar efeitos visuais e animações, acessar e manipular o DOM, alterar conteúdos entre outras funcionalidades.

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





e) I, II, III e IV.

### Comentários:

(I) Correto, é livre e aberto; (II) Correto, essa é uma bela descrição das vantagens de se utilizar JQuery/JS; (III) Errado, JQuery ou JavaScript não podem ser utilizados para fazer acesso a base de dados através de um ODBC; (IV) Correto, ele permite vários efeitos visuais, animações, você pode manipular o DOM da página, alterar conteúdos HTML, etc.

**Gabarito:** Letra D

**17. (FGV / IBGE – 2016)** O comando jQuery para selecionar e esconder todos os elementos com a classe "escondido", de uma página é:

- a) \$(class=".escondido").hide();
- b) \$(".escondido").hide();
- c) \$(escondido).hide();
- d) \$("#escondido").hide();
- e) \$(hide).escondido();

### Comentários:

Galera, é possível matar essa questão de um jeito muito fácil. Qual é o Seletor CSS de classe? É o ponto, logo só pode ser a Letra B. Aqui eu seleciono \$(".escondido") e com o método hide() eu escondo.

**Gabarito:** Letra B

**18. (FCC / TRT-13 – 2014)** Considere a página abaixo, construída utilizando jQuery:

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Home</title>
    <script type="text/javascript"
src="http://ajax.microsoft.com/ajax/jquery/jquery-1.4.2.min.js"></script>
    <script type="text/javascript">
      $(document).ready(function() {
        ..I...
      });
    </script>
  </head>
  <body>
    <button type="button">Esconder</button>
```



```
<p class="par">Texto do parágrafo.</p>
</body>
</html>
```

Para que, ao se clicar no botão Esconder, o texto do parágrafo se torne oculto, deve-se preencher a lacuna I com

- a) \$('button').click(function(){\$('#p#par').hide();});
- b) \$('button:eq(0)').click(function().defineProperty('p.par').hide());
- c) \$('button').click(function(){\$('#p.par').hidden();});
- d) \$('button:eq(1)').click(function(){\$('#p#par').hide();});
- e) \$('button').click(function(){\$('#p.par').hide();});

### Comentários:

Vamos lá! Primeiro, vamos selecionar o botão de esconder. Como só há um botão, podemos utilizar \$('button'); em seguida, ao clicar, deve tornar o texto do parágrafo oculto, logo utilizamos o método click(), no entanto esse método deve receber uma função. Então, fica click(function(){}); por fim, para esconder o texto do parágrafo, nós utilizamos o seletor \$('#p.par'), ele busca o parágrafo p que tenha a classe "par" e, então, utilizamos o método hide().

**Gabarito:** Letra E

19.(CESPE / TRE-BA – 2017) Considere que, no desenvolvimento de um sistema para a Web, um analista pretenda criar páginas dinâmicas que serão executadas na máquina do cliente. Considere, ainda, que, nesse trabalho, o analista opte por uma biblioteca que ofereça capacidade de seleção e manipulação de HTML (Hyper Text Markup Language), manipulação de CSS (Cascade Style Sheets) e requisições AJAX (Asynchronous JavaScript e XML). Nessa situação, o analista deverá utilizar a ferramenta:

- a) JQuery.
- b) JSP (Java Server Pages).
- c) JSF (Java Server Faces).
- d) APEX (Oracle Application Express).
- e) JPA (Java Persistence API).

### Comentários:

O enunciado da questão afirma que o objetivo do analista era criar páginas dinâmicas. Logo, já podemos descartar o APEX (Ambiente para desenvolvimento rápido de software que roda sobre o Oracle) e o JPA (Especificação de mapeamento objeto-relacional). Em seguida, ele afirma que essas páginas dinâmicas seriam executadas na máquina do cliente, i.e., client-side. Logo, já podemos descartar o JSP (Especificação para criar páginas web dinâmicas em um servidor web, portanto server-side) e o JSF (Especificação para criar interfaces de usuário por meio de componentes web).



Em seguida, o enunciado relata que o analista deseja optar por uma biblioteca que ofereça capacidade de seleção e manipulação de HTML (Hyper Text Markup Language), manipulação de CSS (Cascade Style Sheets) e requisições AJAX (Asynchronous JavaScript e XML). Ora, só pode ser o JQuery: ele é uma biblioteca do JavaScript; ele fornece várias formas de seleção e manipulação de HTML e CSS – é capaz, por exemplo, de manipular o DOM de uma Página Web ou suas folhas de estilo; e ele é capaz de realizar requisições AJAX.

Importante: não sejam tão rigorosos quando o enunciado diz que se trata de uma ferramenta. JQuery é primariamente uma biblioteca, mas não há nada errado em dizer que se trata de uma ferramenta de programação, por exemplo, para capturar seletores de uma Página Web.

**Gabarito:** Letra A

20. (IESE / IFC-SC – 2015) Sobre JavaScript e jQuery:

- a) `var $('#id');` (jQuery) e `document.getElementById('id');` (JavaScript) são similares e ambos retornam objetos HTML DOM.
- b) `$('#id');` (jQuery) e `document.getElementById('id');` (JavaScript) são similares, sendo que a única diferença é que o primeiro retorna um objeto jQuery e o segundo retorna um objeto HTML DOM.
- c) `var document.$('#id');` (jQuery) e `document.getElementById('id');` (JavaScript) são similares e ambos retornam objetos HTML DOM.
- d) `var document.$('id');` (jQuery) e `document.getElementById('id');` (JavaScript) são similares, sendo que a única diferença é que o primeiro retorna um objeto jQuery e o segundo retorna um objeto HTML DOM.

**Comentários:**

A resposta correta é o item B: `$('#id');` e `document.getElementById('id');` são equivalentes para jQuery e Javascript, respectivamente. Observando os outros itens, temos que o item A difere do item B apenas pelo uso de aspas na abordagem usando jQuery. Por fim, os itens C e D utilizam uma sintaxe que não existe nem no jQuery nem no Javascript ao utilizar `document.$( ... )`.

**Gabarito:** Letra B

21. (IESES / IFC-SC – 2015) Em JavaScript,

```
document.getElementById('id');
```



retorna um objeto HTML DOM através de seu id. Utilizando jQuery, pode-se realizar uma tarefa semelhante, porém retornando um objeto jQuery ao invés de um objeto HTML DOM através de:

- a) \$('#id');
- b) var document.\$(#id);
- c) var \$(#id);
- d) var document.\$('id');

#### Comentários:

Para obtermos um objeto a partir do seu id usando jQuery utilizamos a sintaxe \$('#id'), pois a mesma recebe como parâmetro o seletor CSS do objeto.

**Gabarito:** Letra A

---

**22. (CESPE / STM - 2018)** Em JQuery, o método \$.get() é usado para executar uma solicitação HTTP.

#### Comentários:

Perfeito! O método \$.get() é usado para executar uma solicitação HTTP.

**Gabarito:** Correto

---

**23. (CESPE / STM - 2018)** O seletor JQuery \$(":disabled") seleciona todos os elementos que estão desativados (disabled).

#### Comentários:

Exato! O comando seletor \$(":disabled") seleciona todos os elementos que estão desativados (disabled).

**Gabarito:** Correto

---

**24. (IBFC / EMDEC - 2019)** O jQuery é uma biblioteca javascript/CSS/HTML e utiliza variáveis na forma de seletores CSS e trabalha em qualquer projeto de frontend que envolva arquivos Javascript.

#### Comentários:

Perfeito! O jQuery é uma biblioteca javascript/CSS/HTML e utiliza variáveis na forma de seletores CSS e trabalha em qualquer projeto de frontend que envolva arquivos Javascript.



**Gabarito:** Correto

**25. (COMPERVE / TJ-RN - 2020)** O jQuery pode selecionar elementos HTML a partir de seletores, otimizando o desenvolvimento. Sobre a utilização de seletores no jQuery é correto afirmar:

- a) \$('td, th'); retornará todos os elementos th que são filhos de td.
- b) \$('#formulario'); retornará todos os elementos da classe "formulario".
- c) \$('p:first'); retornará o primeiro dos elementos p.
- d) \$('.paragrafo'); retornará o elemento com id igual a "paragrafo".

**Comentários:**

(a) Errado. Para selecionar os elementos *th* que são filhos de *td*: \$("td > th"); (b) Errado. Para selecionar os elementos da classe *formulario*: \$(".formulario"); (c) Correto. \$('p:first'); retornará o primeiro dos elementos p; (d) Errado. Para selecionar os elementos com *id* igual a *paragrafo*: \$("#paragrafo").

**Gabarito:** Letra C

**26. (IFSUL / IFSUL - 2019)** Sobre a biblioteca JQuery, é um framework client-side desenvolvido na linguagem Java para a manipulação de elementos HTML.

**Comentários:**

JQUERY é um framework Javascript e não um framework Java.

**Gabarito:** Errado

**27. (FGV / TJRO - 2021)** No contexto da jQuery, o código `$(document).ready(){ ..... }` previne que as funções jQuery sejam executadas antes da carga total da página. A primeira linha desse script pode ser substituída por um método mais abreviado:

- a) \$(begin{
- b) \$(function(){
- c) \$(go(){
- d) \$(main{
- e) \$(ready){

**Comentários:**

No JQuery, podemos inicializar com `$(document).ready(function() { ... });` ou com `$(function() { ... });`;

**Gabarito:** Letra B



**28.(FGV / MPE-AL - 2018)** Em JQuery, o comando que oculta todos os elementos da classe "xxx" é `$(".xxx").hide()`.

#### Comentários:

Perfeito! O comando `$(".xxx").hide()` oculta todos os elementos de uma classe.

**Gabarito:** Correto

---

**29.(AOCP / INES - 2013)** jQuery é uma biblioteca JavaScript Criada por John Resig e disponibilizada como software livre e aberto, ou seja, de emprego e uso regido segundo licença conforme as regras estabelecidas pelo MIT e pelo GPL.

#### Comentários:

Perfeito! jQuery é uma biblioteca JavaScript Criada por John Resig e disponibilizada como software livre e aberto, ou seja, de emprego e uso regido segundo licença conforme as regras estabelecidas pelo MIT e pelo GPL.

**Gabarito:** Correto

---

**30.(AOCP / INES - 2013)** jQuery destina-se a adicionar interatividade e dinamismo às páginas web, proporcionando ao desenvolvedor funcionalidades necessárias à criação de scripts que visem a incrementar, de forma progressiva e não obstrutiva, a usabilidade e a acessibilidade e o design, enriquecendo a experiência do usuário.

#### Comentários:

Exato! jQuery destina-se a adicionar interatividade e dinamismo às páginas web, proporcionando ao desenvolvedor funcionalidades necessárias à criação de scripts que visem a incrementar, de forma progressiva e não obstrutiva, a usabilidade e a acessibilidade e o design, enriquecendo a experiência do usuário.

**Gabarito:** Correto

---

**31.(AOCP / INES - 2013)** Diferentemente do JavaScript sozinho, a biblioteca jQuery pode ser utilizada do lado do servidor para fazer acesso a base de dados através de um ODBC.

#### Comentários:

O correto seria dizer que trata-se de uma biblioteca de funções JavaScript que interage com o HTML, desenvolvida para simplificar os scripts interpretados no navegador do cliente (client-side).



**Gabarito:** Errado

---

**32. (AOCP / INES - 2013)** Com jQuery em uma página Web podemos adicionar efeitos visuais e animações, acessar e manipular o DOM, alterar conteúdos entre outras funcionalidades.

**Comentários:**

Perfeito! Com jQuery em uma página Web podemos adicionar efeitos visuais e animações, acessar e manipular o DOM, alterar conteúdos entre outras funcionalidades.

**Gabarito:** Correto

---

**33. (FGV / AL-RO - 2018)** Analise o código jQuery a seguir.

```
$(document).ready(function(){  
    $("button").click(function(){  
        $("p").hide();  
    });  
});
```

Assinale a opção que indica o efeito observado, quando o botão referido no código é clicado

- a) Nenhum efeito, pois o método hide requer dois parâmetros.
- b) O botão referido é ocultado.
- c) Toda ocorrência do carácter "p" no texto do botão é substituída por uma string vazia.
- d) Todo elemento da página cujo atributo name é igual a "p" é ocultado.
- e) Todo elemento <p> da página é ocultado.

**Comentários:**

Nesse caso, temos que todo elemento <p> da página será ocultado, ao rodar a function () { \$("p").hide();

**Gabarito:** Letra E

---

**34. (VUNESP / TJM-SP - 2017)** A biblioteca do jQuery é distribuída em duas versões: desenvolvimento (.js) e produção (.min.js).

**Comentários:**

Perfeito! A biblioteca do jQuery é distribuída em duas versões: desenvolvimento (.js) e produção (.min.js).



**Gabarito:** Correto

---

**35. (CESPE / SEDF - 2017)** JQUERY é uma tecnologia para desenvolvimento web front-end.

**Comentários:**

Perfeito! JQUERY é uma tecnologia para desenvolvimento web front-end.

**Gabarito:** Correto

---

**36. (CESPE / TCE-PA - 2016)** Após a incorporação do jQuery ao HTML5, o desenvolvimento de funcionalidades por meio dessa biblioteca JavaScript ficou limitado a aplicações para dispositivos móveis.

**Comentários:**

Para inclusão do jQuery em aplicações móveis, foi criada a adaptação do mesmo que é denominado jQuery Mobile. Não existe essa limitação citada.

**Gabarito:** Errado

---

**37. (CESPE / TRE-BA - 2017)** No desenvolvimento de um sistema para a Web, um analista pretenda criar páginas dinâmicas que serão executadas na máquina do cliente. Considere, ainda, que, nesse trabalho, o analista opte por uma biblioteca que ofereça capacidade de seleção e manipulação de HTML (Hyper Text Markup Language), manipulação de CSS (Cascade Style Sheets) e requisições AJAX (Asynchronous JavaScript e XML). Nessa situação, o analista deverá utilizar a ferramenta JQUERY.

**Comentários:**

Perfeito! No desenvolvimento de um sistema para a Web, um analista pretenda criar páginas dinâmicas que serão executadas na máquina do cliente. Considere, ainda, que, nesse trabalho, o analista opte por uma biblioteca que ofereça capacidade de seleção e manipulação de HTML (Hyper Text Markup Language), manipulação de CSS (Cascade Style Sheets) e requisições AJAX (Asynchronous JavaScript e XML). Nessa situação, o analista deverá utilizar a ferramenta JQUERY.

**Gabarito:** Correto

---

**38. (AOCF / FRUNPRESF - 2021)** JavaScript é uma linguagem que sofre muito com compatibilidade entre navegadores. A jQuery sofre com o mesmo problema. Animações, manipulação de DOM e outras tarefas corriqueiras são mais complexas e menos produtivas ao usar o jQuery.





### Comentários:

jQuery é uma biblioteca de Javascript rápida, compacta e com muitas funcionalidades. É uma API que funciona em vários navegadores e facilita tarefas como percorrer e manipular documentos HTML (DOM), lidar com eventos, criar animações e trabalhar com Ajax

**Gabarito:** Errado

---

**39.(ESAF / CGU - 2012)** JQUERY possui arquitetura compatível com instalação de plug-ins e extensões em geral.

### Comentários:

Perfeito! JQUERY possui arquitetura compatível com instalação de plug-ins e extensões em geral.

**Gabarito:** Correto

---

**40.(VUNESP / Prefeitura SP - 2016)** A função .each() do jQuery faz com que uma função de callback seja invocada no contexto de cada um dos elementos selecionados.

### Comentários:

Perfeito! A função .each() do jQuery faz com que uma função de callback seja invocada no contexto de cada um dos elementos selecionados.

**Gabarito:** Correto

---

**41.(IF-PE / IF-PE - 2016)** O comando abaixo escrito em jQuery faz com que,

```
$('#:text').blur(function( ) {  
    $("p").css("border", "3px dashed red");  
});
```

- quando o elemento text for selecionado na página, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.
- quando o elemento de ID text perder o foco sobre ele, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.
- quando um input text ganhar o foco sobre ele, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.



d) quando um input text perder o foco sobre ele, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags da <P> página.

e) quando qualquer input text for modificado na página, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.

### Comentários:

Pessoal, o código:

```
$('.text').blur(function( ) {  
    $('p').css("border", "3px dashed red");  
});
```

ao ser executado adiciona uma borda de 3px dashed red (pixels vermelha e tracejada) em todas as tags da <P> página.

**Gabarito:** Letra D

**42.(VUNESP / FUNDUNESP - 2016)** No jQuery, uma Promise é um objeto que fornece métodos que gerenciam a execução de callbacks, permitindo, inclusive, que seja alterado o estado de sucesso ou falha da função que a produziu.

### Comentários:

No JQUERY, uma promise é um objeto que fornece acesso a um subconjunto de métodos do objeto Deferred, prevenindo que usuários modifiquem o estado do objeto que a produziu.

**Gabarito:** Errado

**43.(FGV / MPE-AL - 2018)** A biblioteca JQuery pode ser obtida por *download* de arquivos, em *sites* como "http://jquery.com/download/", ou a partir de um CDN (*Content Delivery Network*) como, por exemplo, Google Microsoft, "https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js".

Assinale a opção que descreve uma vantagem da segunda alternativa sobre a primeira.

- a) Inclui, adicionalmente, todas as bibliotecas do JavaScript.
- b) Não requer licença, pois é de domínio público.
- c) Os dados poderão ser obtidos rapidamente, do cache do servidor.
- d) Permite acesso ao código fonte.
- e) Utiliza alto grau de compactação.



### Comentários:

A vantagem é bem simples: na segunda alternativa, através do CDN (*Content Delivery Network*), os dados poderão ser obtidos rapidamente, do cache do servidor.

**Gabarito:** Letra C

---

**44.(AOCP / INES - 2016)** A principal limitação da biblioteca JQUERY é a necessidade de instalação de um plug-ins, em cada navegador que fará acesso a páginas com suas funcionalidades.

### Comentários:

Não existe isso de necessidade de instalação de plug-in em cada navegador que fará acesso a páginas com suas funcionalidades. O JQUERY é uma biblioteca leve, rápida e cheia de recursos para Javascript.

**Gabarito:** Errado

---

**45.(AOCP / INES - 2016)** O JQUERY utiliza seletores CSS para localizar elementos componentes da estrutura de marcação HTML da página.

### Comentários:

Perfeito! O JQUERY utiliza seletores CSS para localizar elementos componentes da estrutura de marcação HTML da página.

**Gabarito:** Correto

---

**46.(AOCP / INES - 2016)** O JQUERY possui arquitetura compatível com instalação de plugins e extensões em geral e é indiferente às inconsistências de renderização entre navegadores.

### Comentários:

Perfeito! O JQuery funciona em todos os navegadores com arquitetura compatível com instalação de plugins.

**Gabarito:** Correto

---

**47.(AOCP / INES - 2016)** O JQUERY admite programação encadeada, ou seja, cada método retorna um objeto.

### Comentários:



Perfeito! O JQuery admite programação encadeada, ou seja, cada método retorna um objeto.

**Gabarito:** Correto

---

**48.(VUNESP / TCE-SP - 2016)** O método do jQuery que permite ocultar um elemento selecionado é o `.show(false)`.

**Comentários:**

O método do jQuery que permite ocultar um elemento selecionado é o `.hide()`.

**Gabarito:** Errado

---

**49.(IADES / CRF-DF - 2017)** O jQuery oferece muitos seletores embasados em atributo, o que permite que sejam selecionados elementos embasados no conteúdo de atributos arbitrários, com o uso de expressões regulares simplificadas.

**Comentários:**

Perfeito! O jQuery oferece muitos seletores embasados em atributo, o que permite que sejam selecionados elementos embasados no conteúdo de atributos arbitrários, com o uso de expressões regulares simplificadas

**Gabarito:** Correto

---

**50.(IADES / CRF-DF - 2017)** Considere a página HTML a seguir para responder à questão.

```
<body>
  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
    <li>Item 4</li>
  </ul>
  <button type="button">Clique</button>
</body>
```

O seletor jQuery para selecionar os dois últimos itens da lista é o `$( 'ul > li:eq(3)' )`



### Comentários:

Para selecionar os dois últimos itens de uma lista no JQUERY utilizamos o `$('#ul > li:gt(1)')`.

**Gabarito:** Errado

---

**51. (IADES / CRF-DF - 2017)** O método jQuery para converter uma string JSON em um objeto JavaScript é o `$.convertJSON()`

### Comentários:

O método jQuery para converter uma string JSON em um objeto JavaScript é o `$.parseJSON()`

**Gabarito:** Errado

---

**52. (VUNESP / Pref.Prudente SP - 2016)** O seletor jQuery capaz de selecionar todos os elementos de uma página web é o `$('#all')`.

### Comentários:

O seletor jQuery capaz de selecionar todos os elementos de uma página web é o `$('#*')`.

**Gabarito:** Errado

---

**53. (IBFC / EMBASA - 2017)** Assinale, das alternativas abaixo, a única que identifica corretamente uma das principais características do jQuery.

- a) Tem problemas com a implementação dos recursos do CSS, em todas as suas versões
- b) O jQuery apresenta incompatibilidade técnica para trabalhar com AJAX e DOM
- c) É uma biblioteca de funções, puramente em linguagem Java, que interage com HTML
- d) A resolução da incompatibilidade entre os navegadores (browsers)

### Comentários:

O JQUERY resolve o problema de incompatibilidade entre navegadores. Vejamos as demais alternativas:

(a) Errado. Ao contrário, implementa recursos do CSS; (b) Errado. Não existe essa incompatibilidade; Errado. Não é uma biblioteca em Java, é uma biblioteca Javascript.

**Gabarito:** Letra D

---



54. (FGV / IBGE - 2016) O comando jQuery para selecionar e esconder todos os elementos com a classe "escondido", de uma página é:

- a) `$(class=".escondido").hide();`
- b) `$(".escondido").hide();`
- c) `$(escondido).hide();`
- d) `$("#escondido").hide();`
- e) `$(hide).escondido();`

#### Comentários:

Para esconder um elemento temos que usar o `hide()`, definida então a classe ".escondido", basta escrever: `$(".escondido").hide();`

**Gabarito:** Letra B

---



## LISTA DE QUESTÕES – JQUERY

- (FCC/ MPE-AP - 2012)** O seletor jQuery:
  - `$("[href!='#']")` seleciona todos os elementos cujo atributo href não contenha '#'.
  - `$("[href$.jpg']")` seleciona todos os elementos cuja propriedade href contenha '.jpg'.
  - `$("div#intro .head")` seleciona todos os elementos com id="head" dentro de um elemento com class="intro".
  - `$("p#intro:first")` seleciona o primeiro elemento com class="intro".
  - `$("ul li:first-child")` seleciona todos os elementos do primeiro elemento
- (VUNESP/ TCE-SP - 2015)** O método do jQuery que permite ocultar um elemento selecionado é:
  - `.show(false)`
  - `.hide()`
  - `.mask()`
  - `.visible(false)`
  - `.display(false)`
- (FCC/ TRT-13 – 2014)** Considere a página abaixo, construída utilizando jQuery:

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Home</title>
    <script type="text/javascript"
src="http://ajax.microsoft.com/ajax/jquery/jquery-1.4.2.min.js"></script>
    <script type="text/javascript">
      $(document).ready(function() {
        ..I...
      });
    </script>
  </head>
  <body>
    <button type="button">Esconder</button>
    <p class="par">Texto do parágrafo.</p>
  </body>
</html>
```

Para que, ao se clicar no botão Esconder, o texto do parágrafo se torne oculto, deve-se preencher a lacuna I com:



- a) `$('#button').click(function(){$('#p#par').hide();});`  
b) `$('#button:eq(0)').click(function().defineProperty('p.par').hide());`  
c) `$('#button').click(function(){$('#p.par').hidden();});`  
d) `$('#button:eq(1)').click(function(){$('#p#par').hide();});`  
e) `$('#button').click(function(){$('#p.par').hide();});`
4. **(CESPE/ ANAC – 2012)** Em jQuery, o método `hide()` permite ocultar elementos HTML. Assim, a sintaxe correta para ocultar o elemento corrente é `$(this).hide()` e, para ocultar todos os elementos `<p>` da página, é `$("#<p>").hide()`.
5. **(ESAF / CGU – 2012)** São plug-ins da biblioteca jQuery destinados a criar funcionalidade para a interface do usuário:
- a) Plug-ins nativos para interação. Widgets. Effects.  
b) Plug-ins interativos tipo slider. Wide-rangers. Effects.  
c) Plug-ins nativos da família datepicker. MainWidgets. Efforts.  
d) Widgets nativos para marcação. Widgets de monitoramento. Plug-ins de localização.  
e) Plug-ins nativos para interação. Upload widgets. Effective plug-ins.
6. **(FUNDEP / UFVMJ – 2017)** O JavaScript é uma linguagem leve, interpretada e baseada em objetos, mais conhecida como a linguagem de script para páginas Web. Assinale a alternativa que apresenta corretamente uma biblioteca de JavaScript.
- a) JQuery  
b) HTML  
c) AJAX  
d) CSS
7. **(AOCP – TCE-PA - 2012)** O nome dado à biblioteca JavaScript cross-browser desenvolvida para simplificar os scripts executados no lado do cliente e que interagem com o HTML é:
- a) JSP.  
b) J2ME.  
c) JSON.  
d) J2EE.  
e) JQuery
8. **(FCC / MPE-PB – 2015)** Considere o bloco de código a seguir:

```
<script>
$(document).ready(function(){
    $("div").click(function(){
        $(this).hide();
    });
});
```





```
});  
});  
</script>
```

Este bloco de código utiliza uma biblioteca JavaScript criada para simplificar a programação. Trata-se da biblioteca conhecida como:

- a) jQuery.
- b) AngularJS.
- c) JSON.
- d) AJAX.
- e) jQuery.

9. (ESAF / CGU – 2012) Assinale a opção correta relativa a jQuery:

- a) Não admite programação encadeada.
- b) Utiliza indicadores SSP para localizar elementos componentes da estrutura de marcação jHTML da página.
- c) Possui arquitetura compatível com instalação de plug-ins e extensões em geral.
- d) Possui estrutura para armazenamento de inputs de especificação multicore.
- e) Utiliza pop-ups C2S para retirar elementos inconsistentes da estrutura de marcação XHTML da referência.

10. (IF/PE – IF/PE - 2016) O comando abaixo escrito em jQuery faz com que,

```
$( ':text' ).blur(function() {  
    $( "p" ).css( "border", "3px dashed red" );  
});
```

- a) quando o elemento text for selecionado na página, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.
- b) quando o elemento de ID text perder o foco sobre ele, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.
- c) quando um input text ganhar o foco sobre ele, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.
- d) quando um input text perder o foco sobre ele, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.



e) quando qualquer input text for modificado na página, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.

**11. (AOCPI/ INES – 2013)** Sobre as características da Biblioteca jQuery, assinale a alternativa INCORRETA.

a) A principal limitação desta biblioteca é a necessidade de instalação de um plug-ins, em cada navegador que fará acesso a páginas com suas funcionalidades.

b) Utiliza seletores CSS para localizar elementos componentes da estrutura de marcação HTML da página.

c) Possui arquitetura compatível com instalação de plugins e extensões em geral.

d) É indiferente às inconsistências de renderização entre navegadores.

e) Admite programação encadeada, ou seja, cada método retorna um objeto.

**12. (CESPE / TCE-PA – 2016)** Após a incorporação do jQuery ao HTML5, o desenvolvimento de funcionalidades por meio dessa biblioteca JavaScript ficou limitado a aplicações para dispositivos móveis.

**13. (FGV / IBGE – 2017)** Para definir a cor de fundo de todos os elementos <p> para azul em uma página HTML, o código jQuery correto é:

- a) `$(this).css("background-color","blue");`
- b) `$("#?p").css("background-color"," blue");`
- c) `$("#p").css ("background-color"," blue");`
- d) `$("##p").css("background-color"," blue");`
- e) `$(".p").css("background-color"," blue");`

**14. (QUADRIX / CRO-PR – 2016)** Um código em Javascript puro, para atribuir o valor "3" a um elemento qualquer, ficaria assim:

```
document.getElementById( 'Concurso' ).value = 3;
```

Como ficaria esse mesmo código em jQuery?

- a) `#Concurso:=3;`
- b) `$( "Concurso" ).val( 3 );`
- c) `$(Concurso).val( 3 );`
- d) `$( '#Concurso' ).val( 3 );`
- e) `$( '#Concurso' )=3`



**15. (FCC / TRE-SP – 2017)** Considere uma página HTML contendo a instrução `<div id="d1" ></div>`. Para mostrar no interior deste contêiner, utilizando jQuery, o tamanho do contêiner, utiliza-se a instrução:

- a) `$("#d1").innerHTML($("#d1").width());`
- b) `$("#d1").html($("#d1").size());`
- c) `$("#d1").show($("#d1").width());`
- d) `$("#d1").getElementById("d1").innerHTML= $("#d1").width();`
- e) `$("#d1").html($("#d1").width());`

**16. (AOCP / INES – 2013)** Sobre a Biblioteca jQuery analise as assertivas e assinale a alternativa que aponta a(s) correta(s).

I. jQuery é uma biblioteca JavaScript Criada por John Resig e disponibilizada como software livre e aberto, ou seja, de emprego e uso regido segundo licença conforme as regras estabelecidas pelo MIT e pelo GPL.

II. jQuery destina-se a adicionar interatividade e dinamismo às páginas web, proporcionando ao desenvolvedor funcionalidades necessárias à criação de scripts que visem a incrementar, de forma progressiva e não obstrutiva, a usabilidade e a acessibilidade e o design, enriquecendo a experiência do usuário.

III. Diferentemente do JavaScript sozinho, a biblioteca jQuery pode ser utilizada do lado do servidor para fazer acesso a base de dados através de um ODBC.

IV. Com jQuery em uma página Web podemos adicionar efeitos visuais e animações, acessar e manipular o DOM, alterar conteúdos entre outras funcionalidades.

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

**17. (FGV / IBGE – 2016)** O comando jQuery para selecionar e esconder todos os elementos com a classe "escondido", de uma página é:

- a) `$(class=".escondido").hide();`
- b) `$(".escondido").hide();`
- c) `$(escondido).hide();`
- d) `$("#escondido").hide();`
- e) `$(hide).escondido();`



**18. (FCC / TRT-13 – 2014)** Considere a página abaixo, construída utilizando jQuery:

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Home</title>
    <script type="text/javascript"
src="http://ajax.microsoft.com/ajax/jquery/jquery-1.4.2.min.js"></script>
    <script type="text/javascript">
      $(document).ready(function() {
        ..I...
      });
    </script>
  </head>
  <body>
    <button type="button">Esconder</button>
    <p class="par">Texto do parágrafo.</p>
  </body>
</html>
```

Para que, ao se clicar no botão Esconder, o texto do parágrafo se torne oculto, deve-se preencher a lacuna I com

- a) \$('button').click(function(){\$('#p#par').hide();});
- b) \$('button:eq(0)').click(function().defineProperty('p.par').hide());
- c) \$('button').click(function(){\$('#p.par').hidden();});
- d) \$('button:eq(1)').click(function(){\$('#p#par').hide();});
- e) \$('button').click(function(){\$('#p.par').hide();});

**19. (CESPE / TRE-BA – 2017)** Considere que, no desenvolvimento de um sistema para a Web, um analista pretenda criar páginas dinâmicas que serão executadas na máquina do cliente. Considere, ainda, que, nesse trabalho, o analista opte por uma biblioteca que ofereça capacidade de seleção e manipulação de HTML (Hyper Text Markup Language), manipulação de CSS (Cascade Style Sheets) e requisições AJAX (Asynchronous JavaScript e XML). Nessa situação, o analista deverá utilizar a ferramenta:

- a) JQuery.
- b) JSP (Java Server Pages).
- c) JSF (Java Server Faces).
- d) APEX (Oracle Application Express).
- e) JPA (Java Persistence API).

**20. (IESE / IFC-SC – 2015)** Sobre JavaScript e jQuery:



- a) **var `$('#id)`**; (jQuery) e **`document.getElementById('id')`**; (JavaScript) são similares e ambos retornam objetos HTML DOM.
- b) **`$('#id)`**; (jQuery) e **`document.getElementById('id')`**; (JavaScript) são similares, sendo que a única diferença é que o primeiro retorna um objeto jQuery e o segundo retorna um objeto HTML DOM.
- c) **var `document. $('#id)`**; (jQuery) e **`document.getElementById('id')`**; (JavaScript) são similares e ambos retornam objetos HTML DOM.
- d) **var `document. $('id)`**; (jQuery) e **`document.getElementById('id')`**; (JavaScript) são similares, sendo que a única diferença é que o primeiro retorna um objeto jQuery e o segundo retorna um objeto HTML DOM.

21. (IESES / IFC-SC – 2015) Em JavaScript,

```
document.getElementById('id');
```

retorna um objeto HTML DOM através de seu id. Utilizando jQuery, pode-se realizar uma tarefa semelhante, porém retornando um objeto jQuery ao invés de um objeto HTML DOM através de:

- a) `$('#id)`;
- b) `var document. $('#id)`;
- c) `var $('#id)`;
- d) `var document. $('id)`;

22. (CESPE / STM - 2018) Em JQuery, o método `$.get()` é usado para executar uma solicitação HTTP.

23. (CESPE / STM - 2018) O seletor JQuery `$(".disabled")` seleciona todos os elementos que estão desativados (disabled).

24. (IBFC / EMDEC - 2019) O jQuery é uma biblioteca javascript/CSS/HTML e utiliza variáveis na forma de seletores CSS e trabalha em qualquer projeto de frontend que envolva arquivos Javascript.

25. (COMPERVE / TJ-RN - 2020) O jQuery pode selecionar elementos HTML a partir de seletores, otimizando o desenvolvimento. Sobre a utilização de seletores no jQuery é correto afirmar:

- a) `$('td, th)`; retornará todos os elementos th que são filhos de td.
- b) `$('#formulario)`; retornará todos os elementos da classe "formulario".
- c) `$('p:first)`; retornará o primeiro dos elementos p.
- d) `$('.paragrafo)`; retornará o elemento com id igual a "paragrafo".



26. (IFSUL / IFSUL - 2019) Sobre a biblioteca JQuery, é um framework client-side desenvolvido na linguagem Java para a manipulação de elementos HTML.
27. (FGV / TJRO - 2021) No contexto da jQuery, o código `$(document).ready(){ ..... }` previne que as funções jQuery sejam executadas antes da carga total da página. A primeira linha desse script pode ser substituída por um método mais abreviado:
- a) `$(begin{`
  - b) `$(function(){`
  - c) `$(go(){`
  - d) `$(main{`
  - e) `$(ready){`
28. (FGV / MPE-AL - 2018) Em JQuery, o comando que oculta todos os elementos da classe "xxx" é `$(".xxx").hide()`.
29. (AOCP / INES - 2013) jQuery é uma biblioteca JavaScript Criada por John Resig e disponibilizada como software livre e aberto, ou seja, de emprego e uso regido segundo licença conforme as regras estabelecidas pelo MIT e pelo GPL.
30. (AOCP / INES - 2013) jQuery destina-se a adicionar interatividade e dinamismo às páginas web, proporcionando ao desenvolvedor funcionalidades necessárias à criação de scripts que visem a incrementar, de forma progressiva e não obstrutiva, a usabilidade e a acessibilidade e o design, enriquecendo a experiência do usuário.
31. (AOCP / INES - 2013) Diferentemente do JavaScript sozinho, a biblioteca jQuery pode ser utilizada do lado do servidor para fazer acesso a base de dados através de um ODBC.
32. (AOCP / INES - 2013) Com jQuery em uma página Web podemos adicionar efeitos visuais e animações, acessar e manipular o DOM, alterar conteúdos entre outras funcionalidades.
33. (FGV / AL-RO - 2018) Analise o código jQuery a seguir.

```
$(document).ready(function(){  
    $("button").click(function(){  
        $("p").hide();  
    });  
});
```

Assinale a opção que indica o efeito observado, quando o botão referido no código é clicado

- a) Nenhum efeito, pois o método hide requer dois parâmetros.
- b) O botão referido é ocultado.
- c) Toda ocorrência do carácter "p" no texto do botão é substituída por uma string vazia.



- d) Todo elemento da página cujo atributo name é igual a "p" é ocultado.
- e) Todo elemento <p> da página é ocultado.

- 34.(VUNESP / TJM-SP - 2017)** A biblioteca do jQuery é distribuída em duas versões: desenvolvimento (.js) e produção (.min.js).
- 35.(CESPE / SEDF - 2017)** JQUERY é uma tecnologia para desenvolvimento web front-end.
- 36.(CESPE / TCE-PA - 2016)** Após a incorporação do jQuery ao HTML5, o desenvolvimento de funcionalidades por meio dessa biblioteca JavaScript ficou limitado a aplicações para dispositivos móveis.
- 37.(CESPE / TRE-BA - 2017)** No desenvolvimento de um sistema para a Web, um analista pretenda criar páginas dinâmicas que serão executadas na máquina do cliente. Considere, ainda, que, nesse trabalho, o analista opte por uma biblioteca que ofereça capacidade de seleção e manipulação de HTML (Hyper Text Markup Language), manipulação de CSS (Cascade Style Sheets) e requisições AJAX (Asynchronous JavaScript e XML). Nessa situação, o analista deverá utilizar a ferramenta JQUERY.
- 38.(AOCP / FRUNPRESP - 2021)** JavaScript é uma linguagem que sofre muito com compatibilidade entre navegadores. A jQuery sofre com o mesmo problema. Animações, manipulação de DOM e outras tarefas corriqueiras são mais complexas e menos produtivas ao usar o jQuery.
- 39.(ESAF / CGU - 2012)** JQUERY possui arquitetura compatível com instalação de plug-ins e extensões em geral.
- 40.(VUNESP / Prefeitura SP - 2016)** A função .each() do jQuery faz com que uma função de callback seja invocada no contexto de cada um dos elementos selecionados.
- 41.(IF-PE / IF-PE - 2016)** O comando abaixo escrito em jQuery faz com que,

```
$('.text').blur(function( ) {  
    $("p").css("border", "3px dashed red");  
});
```

- a) quando o elemento text for selecionado na página, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.
- b) quando o elemento de ID text perder o foco sobre ele, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.



c) quando um input text ganhar o foco sobre ele, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.

d) quando um input text perder o foco sobre ele, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags da <P> página.

e) quando qualquer input text for modificado na página, seja adicionada uma borda de 3 pixels vermelha e tracejada em todas as tags <P> da página.

**42.(VUNESP / FUNDUNESP - 2016)** No jQuery, uma Promise é um objeto que fornece métodos que gerenciam a execução de callbacks, permitindo, inclusive, que seja alterado o estado de sucesso ou falha da função que a produziu.

**43.(FGV / MPE-AL - 2018)** A biblioteca JQuery pode ser obtida por *download* de arquivos, em *sites* como "http://jquery.com/download/", ou a partir de um CDN (*Content Delivery Network*) como, por exemplo, Google Microsoft, "https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js".

Assinale a opção que descreve uma vantagem da segunda alternativa sobre a primeira.

a) Inclui, adicionalmente, todas as bibliotecas do JavaScript.

b) Não requer licença, pois é de domínio público.

▪ c) Os dados poderão ser obtidos rapidamente, do cache do servidor.

d) Permite acesso ao código fonte.

e) Utiliza alto grau de compactação.

**44.(AOCP / INES - 2016)** A principal limitação da biblioteca JQUERY é a necessidade de instalação de um plug-ins, em cada navegador que fará acesso a páginas com suas funcionalidades.

**45.(AOCP / INES - 2016)** O JQUERY utiliza seletores CSS para localizar elementos componentes da estrutura de marcação HTML da página.

**46.(AOCP / INES - 2016)** O JQUERY possui arquitetura compatível com instalação de plugins e extensões em geral e é indiferente às inconsistências de renderização entre navegadores.

**47.(AOCP / INES - 2016)** O JQUERY admite programação encadeada, ou seja, cada método retorna um objeto.

**48.(VUNESP / TCE-SP - 2016)** O método do jQuery que permite ocultar um elemento selecionado é o .show(false).

**49.(IADES / CRF-DF - 2017)** O jQuery oferece muitos seletores embasados em atributo, o que permite que sejam selecionados elementos embasados no conteúdo de atributos arbitrários, com o uso de expressões regulares simplificadas.





50. (IADES / CRF-DF - 2017) Considere a página HTML a seguir para responder à questão.

```
<body>

  <ul>

    <li>Item 1</li>

    <li>Item 2</li>

    <li>Item 3</li>

    <li>Item 4</li>

  </ul>

  <button type="button">Clique</button>

</body>
```

O seletor jQuery para selecionar os dois últimos itens da lista é o `$('ul > li:eq(3)')`

51. (IADES / CRF-DF - 2017) O método jQuery para converter uma string JSON em um objeto JavaScript é o `$.convertJSON()`

52. (VUNESP / Pref.Prudente SP - 2016) O seletor jQuery capaz de selecionar todos os elementos de uma página web é o `$('all')`.

53. (IBFC / EMBASA - 2017) Assinale, das alternativas abaixo, a única que identifica corretamente uma das principais características do jQuery.

- a) Tem problemas com a implementação dos recursos do CSS, em todas as suas versões
- b) O jQuery apresenta incompatibilidade técnica para trabalhar com AJAX e DOM
- c) É uma biblioteca de funções, puramente em linguagem Java, que interage com HTML
- d) A resolução da incompatibilidade entre os navegadores (browsers)

54. (FGV / IBGE - 2016) O comando jQuery para selecionar e esconder todos os elementos com a classe "escondido", de uma página é:

- a) `$(class=".escondido").hide();`
- b) `$(".escondido").hide();`
- c) `$(escondido).hide();`
- d) `$("#escondido").hide();`
- e) `$(hide).escondido();`



## GABARITO – JQUERY

- |     |         |     |         |     |         |
|-----|---------|-----|---------|-----|---------|
| 1.  | LETRA A | 19. | LETRA A | 37. | CORRETO |
| 2.  | LETRA B | 20. | LETRA B | 38. | ERRADO  |
| 3.  | LETRA E | 21. | LETRA A | 39. | CORRETO |
| 4.  | LETRA E | 22. | CORRETO | 40. | CORRETO |
| 5.  | LETRA A | 23. | CORRETO | 41. | LETRA D |
| 6.  | LETRA A | 24. | CORRETO | 42. | ERRADO  |
| 7.  | LETRA E | 25. | LETRA C | 43. | LETRA C |
| 8.  | LETRA E | 26. | ERRADO  | 44. | ERRADO  |
| 9.  | LETRA C | 27. | LETRA B | 45. | CORRETO |
| 10. | LETRA D | 28. | CORRETO | 46. | CORRETO |
| 11. | LETRA A | 29. | CORRETO | 47. | CORRETO |
| 12. | ERRADO  | 30. | CORRETO | 48. | ERRADO  |
| 13. | LETRA C | 31. | ERRADO  | 49. | CORRETO |
| 14. | LETRA D | 32. | CORRETO | 50. | ERRADO  |
| 15. | LETRA E | 33. | LETRA E | 51. | ERRADO  |
| 16. | LETRA D | 34. | CORRETO | 52. | ERRADO  |
| 17. | LETRA B | 35. | CORRETO | 53. | LETRA D |
| 18. | LETRA E | 36. | ERRADO  | 54. | LETRA B |



# AJAX

## Conceitos Básicos

INCIDÊNCIA EM PROVA: MÉDIA

O desenvolvimento de aplicações desktop apresenta uma riqueza que, por muito tempo, pareceu fora do alcance das aplicações web, gerando um *gap* no nível de experiência do usuário com a aplicação. A riqueza de interação e o grau de responsividade das aplicações desktop davam um banho nas aplicações web, mas esse *gap* diminuiu com a criação do AJAX!

*Professor, AJAX é uma tecnologia?* Não, é um conjunto de tecnologias, cada uma com seu propósito, que se juntam harmonicamente para melhorar a experiência do usuário, tornando páginas web mais ricas e interativas. *Professor, que tecnologias são essas?* **Jessé Garret, criador do AJAX, afirmou em um artigo que ela incorpora as seguintes tecnologias:**

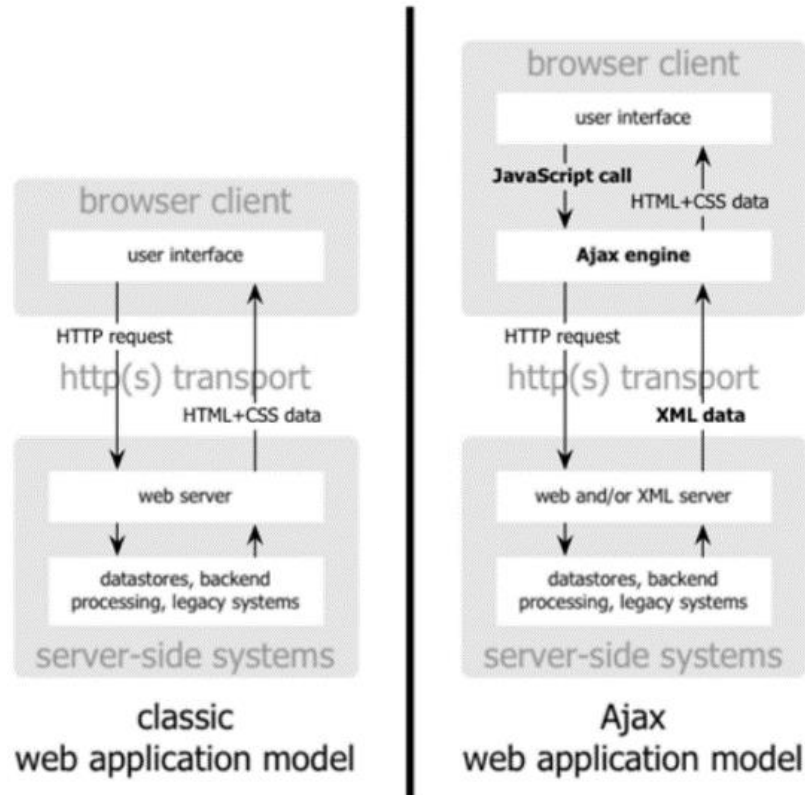
- **Apresentação:** utiliza XHTML e CSS;
- **Interação e Exibição dinâmicas:** utiliza DOM;
- **Manipulação e Intercâmbio de Dados:** utiliza XML e XSLT;
- **Recuperação Assíncrona de Dados:** utiliza XMLHttpRequest;
- e JavaScript para reunir todas essas tecnologias.

Vamos ver rapidamente cada uma dessas tecnologias:

TECNOLOGIAS	DESCRIÇÃO
XHTML	Trata-se de uma linguagem de marcação que estende o HTML para criação de páginas web de forma mais acessível e interoperável.
CSS	Trata-se de uma linguagem de folhas de estilo utilizada para definir a apresentação de documentos escritos em uma linguagem de marcação.
DOM	Trata-se de um modelo de objetos de documentos para criação de interfaces dinâmicas e editáveis independentes de linguagem e plataforma.
XML	Trata-se de uma linguagem de marcação para propósitos especiais de intercâmbio e compartilhamento de dados.
XSLT	Trata-se de uma linguagem de marcação usada para transformar documentos XML em outros documentos ou objetos.
XMLHTTPREQUEST	Trata-se de um objeto utilizado para enviar e receber requisições HTTP assincronamente.
JAVASCRIPT	Trata-se de uma linguagem de programação de scripts interpretada e client-side.

**Galera, um modelo de aplicação web clássica funciona mais ou menos assim: o usuário envia uma Requisição HTTP ao Servidor Web.** Esse realiza algum tipo de processamento (recupera dados, manipula números, etc) e retorna uma Página HTML ao cliente. Essa abordagem faz muito sentido, mas não oferece uma boa experiência ao usuário.





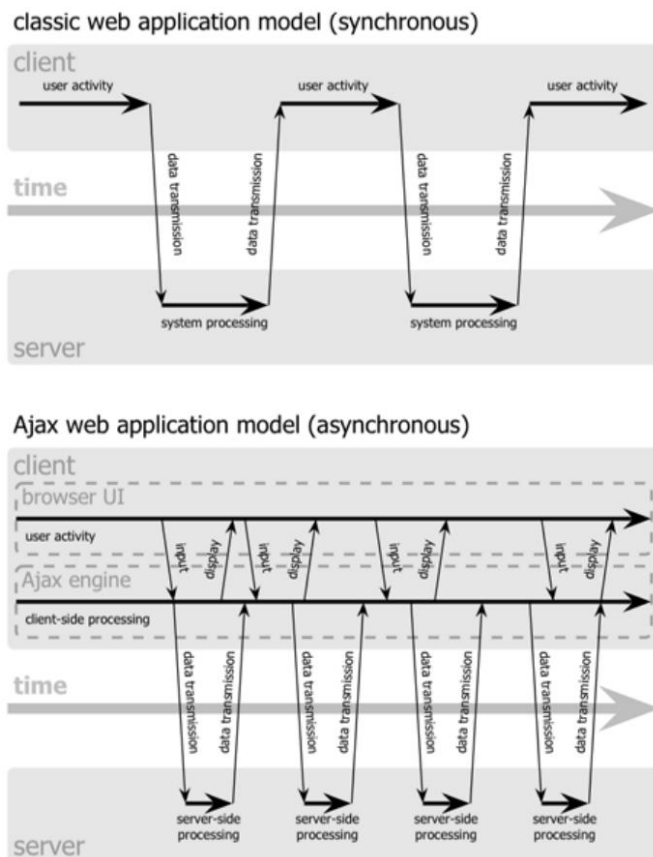
O que faz o usuário enquanto o servidor está realizando seus processamentos? Nada, só espera! E, a cada nova tarefa, espera novamente. Quem aí gosta de esperar? Quem curte um belo chá-de-cadeira? Ninguém! Por que toda vez que o usuário requisitar algo é obrigatório que haja uma interrupção na interface? **Ora, uma vez que a interface foi carregada, o usuário está pouco se importando com o resto.**

As Aplicações AJAX eliminam esse inicia-interrompe-inicia-interrompe ao introduzir uma camada chamada *Engine AJAX* entre o usuário e o servidor! Em vez de carregar uma página web, ao iniciar a sessão, o navegador carrega a *Engine AJAX* (escrita em JavaScript). **Ela permite que a interação do usuário com a aplicação ocorra de forma assíncrona – independente de comunicação com o servidor.**

Portanto, o usuário jamais fica olhando para uma tela branca do navegador, sem fazer nada e esperando que o servidor faça seu trabalho. Toda ação do usuário que normalmente gera uma Requisição HTTP se transforma em uma chamada JavaScript à *Engine AJAX*. **Qualquer resposta que não requeira uma viagem de volta ao servidor (Ex: validação de dados, etc) é manipulada pela própria *Engine AJAX*.**

**Se ela necessitar de alguma coisa do servidor, a *Engine AJAX* realiza as requisições assincronamente, geralmente utilizando XML – sem enrolar a interação do usuário com a aplicação.** A imagem abaixo apresenta a diferença entre uma aplicação web clássica que utiliza uma comunicação síncrona com o servidor e uma aplicação web AJAX que utiliza uma comunicação assíncrona com o servidor.





**O Objeto XMLHttpRequest é utilizado para trocar dados com o servidor de forma assíncrona.**

Alguns dizem que ele é o sonho de um desenvolvedor, porque pode-se atualizar uma parte de uma página web sem precisar recarregar a página inteira; é possível requerer e receber dados do servidor após a página ser carregada; e é possível enviar dados ao servidor em background.

Todos os navegadores modernos (IE7+, Firefox, Chrome, Safari e Opera) possuem um Objeto XMLHttpRequest embutido. A sintaxe para criá-lo é bem simples:

```
xmlhttp = new XMLHttpRequest();
```

**Todos os navegadores modernos possuem um Parser XML! Ele converte um documento XML em um DOM, que pode ser manipulado com JavaScript.** Ele define uma forma padrão de acessar e manipular documentos XML de maneira hierárquica. Dessa forma, todos os elementos podem ser acessados por meio da Árvore DOM!

Para capturar a resposta do servidor, existem duas possibilidades: utilizar as propriedades *responseXML* ou *responseText*. O segundo recupera dados como uma *string*, em forma de texto; e o primeiro recupera dados como um Objeto DOM, em formato XML! **É possível, ainda, capturar dados em outros formatos, tais como: HTML e JSON!**



**No AJAX, o objeto XMLHttpRequest possui diversas propriedades importantes para realizar a comunicação com o servidor.** Quando uma solicitação para um servidor é enviada, normalmente deseja-se executar algumas ações com base na resposta. Existe um evento chamado *onreadystatechange*, responsável por especificar o que vai acontecer quando o servidor processar a resposta. Há três propriedades:

- **onreadystatechange:** armazena uma função a ser chamada de forma automática cada vez que a propriedade `readyState` mudar;
- **readyState:** armazena o status do XMLHttpRequest.
  - 0: Requisição não iniciada;
  - 1: Conexão com servidor estabelecida;
  - 2: Requisição recebida;
  - 3: Requisição processada;
  - 4: Requisição finalizada e resposta pronta.
- **Status:**
  - 200: "Ok".
  - 404: Página não-encontrada.

**AJAX apresenta uma grande desvantagem: se o usuário desativar o JavaScript de seu navegador, a aplicação pode parar de funcionar.** No entanto, há diversas vantagens: executa processos em paralelo às requisições/respostas; manipula XML nas aplicações para desktop de forma simples; troca mensagens de forma assíncrona; entre outras.

Por fim, alguns desenvolvedores utilizam iFrames para embutir aplicações interativas em páginas web, incluindo aquelas que empregam AJAX (Ex: Google Maps). **O iFrame é utilizado para embutir um Documento HTML em outro Documento HTML.** Ele é frequentemente utilizado também para inserir conteúdo de outra fonte (como uma propaganda) em uma página web.

**(UFRJ – 2012)** Considere o desenvolvimento de aplicações web com Ajax (Asynchronous Javascript and XML).

I - O objeto XMLHttpRequest desempenha importante papel, estando relacionado à comunicação assíncrona com o servidor web.

II - Para utilizar Ajax é preciso antes instalar e configurar um servidor Ajax, o qual interage com o servidor web.

III - Uma das características de Ajax é a exibição dinâmica de dados e uso do modelo de objetos DOM (Document Object Model).



Podemos afirmar que:

- a) somente as alternativas I e II estão corretas.
- b) somente a alternativa I está correta.
- c) somente a alternativa II está correta.
- d) somente as alternativas I e III estão corretas.
- e) somente as alternativas II e III estão corretas.

---

**Comentários:** para solicitar dados de um servidor web, utiliza o objeto XMLHttpRequest, logo a I está correta. A II está errada, porque não temos esse requisito em momento algum. A III está correta, o AJAX usa exibição dinâmica de dados e emprega o DOM (Document Object Model) (Letra D).

**(CCV – 2019)** Uma das formas de desenvolver uma página dinâmica é através do uso de AJAX (Asynchronous JavaScript and XML), onde ele é utilizado para atualizar apenas parte da página Web. Sobre o AJAX, é correto afirmar:

- a) o AJAX baseia-se em tecnologias já disponíveis no lado cliente da navegação Web, entre elas a linguagem JavaScript.
- b) para que o cliente usufrua dos benefícios do AJAX, é necessária a instalação de um programa AJAX específico para cada tipo de plataforma.
- c) quando o AJAX é utilizado, todas as requisições enviadas ao servidor são assíncronas devido a limitação imposta pelo protocolo da camada de aplicação.
- d) o AJAX utiliza um protocolo proprietário para a troca de informações do cliente com o servidor, não sendo possível saber como esse procedimento é realizado.
- e) para manter o conteúdo da página atualizado, o servidor mantém uma sessão síncrona com o cliente utilizando o protocolo SMTP (Simple Mail Transfer Protocol), permitindo a troca mais rápida de informações.

---

**Comentários:** vamos por partes: (a) Correto. AJAX baseia-se em tecnologias do lado cliente, entre elas o Javascript, conforme vimos; (b) Errado. Não existe isso de instalar um programa AJAX, os navegadores que utilizam o AJAX; (c) Errado. AJAX suporta requisição síncrona, logo não faz sentido essa alternativa; (d) Errado. AJAX utiliza tecnologias não proprietárias, como XML, Javascript. etc foram que falar em usar protocolo proprietário não faz o menor sentido; (e) Errado. AJAX não usa SMTP, não faz sentido também falar do protocolo SMTP e do AJAX, são coisa distintas (Letra A).



## QUESTÕES COMENTADAS – AJAX

1. (FCC/ TRE-MS 2007) Asynchronous Javascript and XML (Ajax) é uma técnica de desenvolvimento de aplicações web cujo objetivo é a troca de pequenas porções de dados entre um browser e um servidor web de modo a evitar a recarga de toda a página cada vez que um cliente solicita uma mudança. Sobre Ajax considere as afirmativas abaixo.

I. É uma técnica indicada para melhorar a experiência do usuário, reduzir a utilização de banda e claramente separar dados, formatação, estilo e funcionalidade.

II. O modelo de objetos html/xml (DOM) é acessado e transformado por linguagens de script como JavaScript.

III. O objeto XMLHttpRequest e/ou o objeto iFrame são utilizados para troca de dados assíncrona com o servidor.

É correto o que se afirma em:

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

### Comentários:

(I) Perfeito, perfeito, perfeito, nada a acrescentar; (II) Sim, item perfeito também; (III) Também está correto! Pode-se utilizar o iFrame (Inline Frame) para embarcar aplicações interativas em páginas web.

**Gabarito:** Letra E

2. (FCC / MPE-RS – 2008) Para capturar as respostas de uma aplicação Web dinâmica em AJAX básico utiliza-se somente:

- a) responseText e responseXML.
- b) responseText e responseHTML.
- c) responseText.
- d) responseXML.
- e) responseHTML

### Comentários:





Trata-se da primeira opção: `responseText` e `responseXML`. Respectivamente para capturar as respostas de uma aplicação Web dinâmica em AJAX básico utiliza-se somente `responseText` e `responseXML`.

**Gabarito:** Letra A

3. (FCC / MPE-RS – 2008) Dentre as tecnologias que compõem o AJAX, aquelas que têm como principal função fazer o intercâmbio e a manipulação de dados são:

- a) JavaScript e XMLHttpRequest.
- b) XML e XSLT.
- c) HTML e XHTML.
- d) JavaScript e CSS.
- e) DOM e CSS.

#### Comentários:

- **Apresentação:** HTML/XHTML e CSS;
- **Exposição e Interação Dinâmica:** DOM;
- **Intercâmbio de Dados:** XML ou JSON;
- **Manipulação de Dados:** XSLT;
- **Recuperação Assíncrona de Dados:** XMLHttpRequest;

\* JavaScript é o responsável por reunir todas essas tecnologias.

**Gabarito:** Letra B

4. (FCC / MPE-RS – 2008) Para fazer o intercâmbio e a manipulação de dados em uma aplicação Web, o modelo AJAX normalmente utiliza:

- a) XML e XSLT.
- b) HTML, XHTML e CSS.
- c) XMLHttpRequest.
- d) JavaScript.
- e) DOM.

#### Comentários:

- **Apresentação:** HTML/XHTML e CSS;
- **Exposição e Interação Dinâmica:** DOM;
- **Intercâmbio de Dados:** XML ou JSON;
- **Manipulação de Dados:** XSLT;
- **Recuperação Assíncrona de Dados:** XMLHttpRequest;



\* JavaScript é o responsável por reunir todas essas tecnologias.

**Gabarito:** Letra A

---

5. (FCC / TJ-PE – 2012) Esse objeto é o ponto chave do AJAX. Pode ser considerado um objeto Javascript que torna possível a comunicação assíncrona com o servidor. O objeto citado é do tipo:

- a) XMLHttpRequest.
- b) XMLHttpRequest.
- c) HttpServletResponse.
- d) HttpServletRequest.
- e) XMLHttpRequest

**Comentários:**

O objeto responsável pela comunicação assíncrona com o servidor é chamado: *XMLHttpRequest*.

**Gabarito:** Letra E

---

6. (FCC / TCE-PR – 2011) No AJAX, o objeto XMLHttpRequest possui diversas propriedades importantes para realizar a comunicação com o servidor. Quando uma solicitação para um servidor é enviada, normalmente deseja-se executar algumas ações com base na resposta. O evento utilizado para especificar o que vai acontecer quando a resposta do servidor está pronta para ser processada é o:

- a) oncompletedrequest.
- b) onserverreturn.
- c) onendstatus.
- d) onreadystatechange.
- e) onreadystate.

**Comentários:**

Trata-se do *onreadystatechange*. O evento utilizado para especificar o que vai acontecer quando a resposta do servidor está pronta para ser processada é *onreadystatechange*.

**Gabarito:** Letra D

---

7. (FCC / INFRAERO – 2011) Representa uma desvantagem do uso de AJAX:



- a) Troca mensagens entre o cliente e o servidor de forma assíncrona, ou seja, envia requisições e continua o processamento sem precisar aguardar a resposta.
- b) Trata-se principalmente de JavaScript que executa no navegador do usuário. Se for desativado o processamento do JavaScript no navegador, a aplicação pode falhar.
- c) Recebe respostas às requisições na mesma página sem a necessidade de refresh.
- d) Executa os processos em paralelo às requisições/respostas.
- e) Manipula o conteúdo XML nas aplicações para desktop de forma simples.

### Comentários:

Todas apresentam vantagens, exceto a segunda opção! De fato, se o usuário desativar o JavaScript em seu navegador, a aplicação pode falhar e essa é uma grande desvantagem do AJAX.

**Gabarito:** Letra B

**8. (FCC / TRE-RN – 2011)** Agrega um conjunto de tecnologias conhecidas trabalhando juntas para tornar páginas Web mais interativas com o usuário, utilizando-se de solicitações assíncronas de informações:

- a) XML.
- b) JavaScript.
- c) XHTML.
- d) AJAX.
- e) CSS.

### Comentários:

Trata-se obviamente do AJAX! Conforme estudamos, logo o AJAX agrega um conjunto de tecnologias conhecidas trabalhando juntas para tornar páginas Web mais interativas com o usuário, utilizando-se de solicitações assíncronas de informações.

**Gabarito:** Letra D

**9. (FGV / FIOCRUZ – 2010)** Asynchronous JavaScript and XML (AJAX) é um termo criado recentemente para duas características poderosas dos browsers que existem há anos mas tem sido ignoradas por muitos criadores de páginas web até recentemente, quando aplicações como Gmail, Google suggest e Google Maps foram lançadas. AJAX não é uma tecnologia, na verdade, envolve várias, cada uma atuando da sua própria maneira, tornando-se juntas uma poderosa ferramenta. Nesse contexto, além de suporte à Javascript, analise as afirmativas abaixo, associadas às tecnologias incorporadas pelo AJAX.

- I. Troca e manipulação de dados usando XML e XSLT.
- II. Retorno de dados assincronamente usando XMLHttpRequest.



III. Apresentação baseada nas Web Standards usando XHTML e CSS.

Assinale:

- a) se somente a afirmativa I estiver correta.
- b) se somente a afirmativa II estiver correta.
- c) se somente a afirmativa III estiver correta.
- d) se somente as afirmativas II e III estiverem corretas.
- e) se todas as afirmativas estiverem corretas.

#### Comentários:

Todos os itens estão corretos. Assim: O AJAX realiza a troca e manipulação de dados usando XML e XSLT. O retorno de dados assincronamente usando XMLHttpRequest e a apresentação baseada nas Web Standards usando XHTML e CSS.

**Gabarito:** Letra E

**10. (CESGRANRIO / DNMP – 2006)** AJAX é um termo que se refere a um conjunto de tecnologias para desenvolvimento de aplicações WEB. Assinale a opção que contempla uma dessas tecnologias.

- a) Ant
- b) XML
- c) WiMAX
- d) VoIP
- e) Jakarta

#### Comentários:

AJAX contempla JavaScript e XML, conforme estudamos ao longo da aula, logo a alternativa B está correta.

**Gabarito:** Letra B

**11. (CESPE / TJ-RO – 2012)** Que conjunto de tecnologias é utilizado em aplicações WEB na Internet para propiciar forte interatividade e dinamismo?

- a) CSS
- b) SMTP
- c) AJAX
- d) FTP
- e) UTF-8



### Comentários:

Trata-se do AJAX! Ele é um conjunto de tecnologias que propiciam forte interatividade e dinamismo.

**Gabarito:** Letra C

**12. (CESPE / MEC – 2011)** AJAX (do inglês Asynchronous Javascript And XML) tem sido largamente utilizado no desenvolvimento de aplicações WEB. Um dos conceitos centrais do AJAX é a possibilidade de serem feitas requisições ao servidor através de código Javascript, rodando no navegador do usuário. Esse recurso é empregado principalmente para permitir que:

- a) o processamento das regras de negócio da aplicação seja distribuído aos clientes, minimizando a carga do servidor.
- b) o usuário envie uma requisição assíncrona e verifique o resultado da mesma mais tarde, sem precisar aguardar pela resposta imediata do servidor.
- c) as páginas carreguem mais rapidamente, pois as requisições são paralelizadas e compactadas durante o trânsito.
- d) as páginas compostas diretamente em XML, ao invés de XHTML ou HTML, sejam apresentadas no navegador do usuário.
- e) partes de uma página web sejam atualizadas, sem que o browser recarregue a página inteira do servidor, proporcionando uma interface com melhor usabilidade

### Comentários:

(a) Não, o processamento das regras de negócio continua sendo no Servidor; (b) Não há nada de errado com esse item, no entanto o recurso não é empregado principalmente para isso! *Eu posso verificar o resultado mais tarde?* Sim, mas o Ajax é utilizado principalmente para que eu possa verificar o resultado imediatamente sem a necessidade de recarregar a página inteira; (c) Não, é para que partes das páginas recarreguem de forma dinâmica e interativa; (d) Não, são apresentadas como XHTML ou HTML; (e) Perfeito, é exatamente isso!

**Gabarito:** Letra E

**13. (CESGRANRIO / IBGE – 2010)** O código HTML, em construção, abaixo demonstra a utilização de AJAX.

```
<html>
```



```
<head>
<script language="JavaScript">
function submitForm()
{
var xhr=null;
try
{ xhr = new object(); }
catch(e) {}
xhr.onreadystatechange = function()
{
document.ajax.dyn.value="Wait server...";
if(xhr.readyState == 4)
{
if(xhr.status == 200)
{ document.ajax.dyn.value="Recebido:" + xhr.responseText; }
else
{ document.ajax.dyn.value="Erro: " + xhr.status + " " + xhr.statusText; }
}
};
xhr.open("GET", "data.xml", true);
xhr.send(null);
}
</script>
</head>
<body>
<FORM method="POST" name="ajax" action="">
<INPUT type="submit" value="Submit" ONCLICK="submitForm()">
<INPUT type="text" name="dyn" value="">
</FORM>
</body>
</html>
```

Para que esse código possa utilizar a tecnologia AJAX, na linha "xhr = new object();", "xhr" deve receber um objeto Javascript que torna possível a comunicação assíncrona com o servidor, sem a necessidade de recarregar a página por completo. Para tanto, no código acima, "object" deve ser substituído por:

- a) responseXML.
- b) MIME.
- c) XMLHttpRequest.
- d) DOMParser.
- e) setRequestReader.

### Comentários:

O objeto JS para comunicação assíncrona em questão, trata-se do Objeto XMLHttpRequest. XMLHttpRequest é um objeto que fornece funcionalidade ao cliente para transferir dados entre um cliente e um servidor.



**Gabarito:** Letra C

---

**14. (CESPE / PREVIC – 2011)** Ajax não é meramente uma tecnologia. É uma abordagem moderna para desenvolvimento de sites iterativos. A abordagem de desenvolvimento tradicional tem semelhanças e diferenças em relação ao Ajax. Uma característica exclusiva de Ajax em relação à abordagem tradicional é que:

- a) executa as requisições através do protocolo HTTP.
- b) usa javascript como linguagem para desenvolver código no lado do cliente.
- c) usa (x)html / css para definir o aspecto visual da página.
- d) permite recuperação assíncrona de dados usando XMLHttpRequest.
- e) representa os objetos no lado cliente com DOM.

**Comentários:**

A única opção que apresenta uma característica de fato exclusiva em relação à abordagem tradicional é quarta opção: permite recuperação assíncrona de dados usando *XMLHttpRequest*.

**Gabarito:** Letra D

---

**15. (CESGRANRIO/ LIQUIGÁS - 2011)** Duas das tecnologias que compõem o Ajax são:

- a) DOM e CSS
- b) ASP e XML
- c) Java e XML
- d) Java e CSS
- e) JavaScript e ASP

**Comentários:**

*DOM?* Sim! *CSS?* Sim! *ASP?* Não! *XML?* Sim! *Java?* Não! *JavaScript?* Sim! Portanto, é a primeira opção: DOM e CSS!

**Gabarito:** Letra A

---

**16. (CESGRANRIO / LIQUIGÁS – 2013)** Considere as afirmativas a seguir sobre a tecnologia AJAX.

I – Uma das aplicações mais frequentes da tecnologia AJAX é na atualização assíncrona do conteúdo de páginas HTML.

II – Boa parte da funcionalidade da tecnologia AJAX é viabilizada pelo objeto XMLHttpRequest, que é capaz de transmitir requisições HTTP de modo assíncrono.



III – A tecnologia AJAX utiliza código Java para a manipulação do conteúdo de páginas HTML.

Está correto o que se afirma em:

- a) I, apenas
- b) II, apenas
- c) I e II, apenas
- d) II e III, apenas
- e) I, II e III

#### Comentários:

(I) Sim, perfeito! (II) Sim, perfeito! (III) Não, utiliza DOM! Logo, a alternativa C com I e II, apenas está aderente ao que estudamos na aula.

**Gabarito:** Letra C

---

**17. (CESGRANRIO / IBGE – 2014)** Com o objetivo de criar páginas dinâmicas para o desenvolvimento de aplicações web, AJAX é um termo que descreve um(a):

- a) protocolo
- b) banco de dados
- c) coleção de tecnologias
- d) linguagem de programação
- e) linguagem de marcação

#### Comentários:

AJAX é um conjunto de tecnologias, tais como: XML, JSON, XSTL, HTML, XHTML, DOM, JavaScript, CSS e XMLHttpRequest

**Gabarito:** Letra C

---

**18. (FGV / TCM-SP – 2015)** Considere a requisição AJAX construída, no cliente, através da chamada à seguinte função em Javascript:





```
function send (msg)
{
var message = "GET=' " + msg + "'";

var ajax = new XMLHttpRequest ();

ajax.onreadystatechange = function ()
{
    if (this.readyState == 4 &&
        this.status == 200)
        alert (this.responseText);
}

ajax.open ("POST", "PUT.PHP", true);

ajax.setRequestHeader ("Content-type",
    "application/x-www-form-urlencoded");
ajax.setRequestHeader ("Content-length",
    message.length);
ajax.setRequestHeader ("Connection",
    "close");

ajax.send (message);
}
```

O método HTTP utilizado na requisição ao servidor será, neste caso:

- a) PUT;
- b) POST;
- c) GET;
- d) HEAD;
- e) SEND.

### Comentários:

Neste caso, a requisição ao servidor está representada na variável `ajax = new XMLHttpRequest()` e sua chamada em `ajax.open("POST", "PUT.PHP", true);`

O protótipo do método `open`, neste caso, é `XMLHttpRequest.open(method, url, async)`, logo o método HTTP a ser utilizado é o POST.

Vale notar que as strings GET e PUT foram usadas apenas para confundir o aluno, sendo elas, apenas, parte da mensagem e url de envio, respectivamente.

**Gabarito:** Letra B



**19. (UFRJ / UFRJ - 2012)** Para utilizar Ajax é preciso antes instalar e configurar um servidor Ajax, o qual interage com o servidor web.

**Comentários:**

Não existe essa dependência de que para utilizar Ajax é preciso antes instalar e configurar um servidor Ajax, o qual interage com o servidor web.

**Gabarito:** Errado

---

**20. (CESGRANRIO / LIQUIGÁS - 2013)** Uma das aplicações mais frequentes da tecnologia AJAX é na atualização assíncrona do conteúdo de páginas HTML.

**Comentários:**

Perfeito! Uma das aplicações mais frequentes da tecnologia AJAX é na atualização assíncrona do conteúdo de páginas HTML.

**Gabarito:** Correto

---

**21. (CESGRANRIO / LIQUIGÁS - 2013)** A tecnologia AJAX utiliza código Java para a manipulação do conteúdo de páginas HTML.

**Comentários:**

No AJAX, o JAVA é utilizado para manipulação do conteúdo de páginas HTML.

**Gabarito:** Correto

---

**22. (FUNCAB / PRODAM - 2014)** A utilização de Ajax em uma página permite a recuperação de dados de uma URL sem a necessidade de recarregar a página inteira. O objeto JavaScript que foi adotado pelos principais navegadores para realizar operações Ajax é chamado XMLHttpRequest.

**Comentários:**

Perfeito! Conforme vimos, o objeto JavaScript que foi adotado pelos principais navegadores para realizar operações Ajax é chamado de XMLHttpRequest.

**Gabarito:** Correto

---



**23. (CESPE / MPE-RS - 2012)** AJAX é uma técnica de construção de páginas web, e sua principal vantagem é permitir a atualização de partes da página web.

**Comentários:**

Perfeito! De fato, AJAX é uma técnica de construção de páginas web, e sua principal vantagem é permitir a atualização de partes da página web.

**Gabarito:** Correto

---

**24. (CESGRANRIO / PETROBRÁS - 2010)** Ajax não é meramente uma tecnologia. É uma abordagem moderna para desenvolvimento de sites iterativos. A abordagem de desenvolvimento tradicional tem semelhanças e diferenças em relação ao Ajax. Uma característica exclusiva de Ajax em relação à abordagem tradicional é que permite recuperação assíncrona de dados usando XMLHttpRequest.

**Comentários:**

De fato, o Objeto XMLHttpRequest é utilizado para trocar dados com o servidor de forma assíncrona.

**Gabarito:** Correto

---

**25. (AOC / FUNPRESP-JUD - 2021)** JavaScript é uma linguagem que sofre muito com compatibilidade entre navegadores. A jQuery sofre com o mesmo problema. Animações, manipulação de DOM e outras tarefas corriqueiras são mais complexas e menos produtivas ao usar o jQuery.

**Comentários:**

É exatamente tudo ao contrário do que está escrito. JS não sofre de compatibilidade com navegadores e JQuery tbm não, logo manipulação de DOM são menos complexas e mais produtivas ao utilizar JQuery.

**Gabarito:** Errado

---

**26. (IFS / IFS - 2021)** O JavaScript possui características de uma linguagem funcional, portanto pode-se passar funções como parâmetros para outras funções, algo comumente encontrado nos códigos em JavaScript e Node.

Sabendo disso, analise as afirmativas abaixo:



```
I.    function soma(a,b){  
      |   return a + b;  
      |  
      }  
  
      function executar(funcao,a,b){  
      |   return funcao(a,b)  
      |  
      }  
  
      let resultado = executar(soma,1,2)  
  
      console.log(resultado)
```

```
II.   function executar(a,b,funcao){  
      |   return funcao(a,b)  
      |  
      }  
  
      let resultado = executar(1,2,function(a,b){  
      |   return a+b  
      |  
      })  
  
      console.log(resultado)
```

```
III.  var funcao = function(a,b){  
      |   return a+b  
      |  
      }  
  
      function executar(a,b,funcao){  
      |   return funcao(a,b)  
      |  
      }  
  
      let resultado = executar(1,2,funcao)  
  
      console.log(resultado)
```

Está(ão) correta(s) a(s) afirmativa(s)

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

Comentários:



No item I temos a declaração da função, que soma e passa os argumentos para execução. No item II, a função executar recebe uma função anônima e no item III, temos a atribuição da variável `funcao` à uma função anônima, para por fim realizar a execução.

**Gabarito:** Letra D

**27. (IFS / IFS - 2021)** A linguagem *JavaScript* provê uma série de métodos que facilitam a manipulação de *arrays*. Sobre o método de manipulação de *array of*, é correto afirmar que:

- a) cria um novo *array* a partir de um *array* existente.
- b) preenche o *array* com um valor estático.
- c) devolve `@@iterator`, contendo os valores do *array*.
- d) cria um novo *array* a partir dos argumentos passados para o método.

### Comentários:

O método **Array.of()** cria uma nova instância de `Array` a partir de um número variável de argumentos.

**Gabarito:** Letra D

**28. (IFS / IFS - 2021)** Considere o seguinte código *JavaScript*:

```
let o = {one:1,two:2,three:3};  
for(let p in o) console.log(p);
```

Ao final da execução, quais valores serão impressos?

- a) 1, 2, 3
- b) one:1, two: 2, three: 3
- c) p, p, p
- d) 'one', 'two', 'three'

### Comentários:

Para imprimir os valores em vez das chaves, é necessário utilizar o `console.log(o[p])`, como temos: `console.log(p)`, nossa saída será: 'one', 'two', 'three'.

**Gabarito:** Letra D

**29. (IFS / IFS - 2021)** Considere o seguinte código *JavaScript*:

```
var a = [1,2,3,4,5];  
a.slice(0,3);
```



```
a.splice(1,1);  
a.pop();
```

Qual o valor da variável a ao término da execução do código?

- a) [1,3,4]
- b) [1,3,4,5]
- c) [1,3]
- d) [3,4,5]

### Comentários:

Array inicial = "[1,2,3,4,5]" e o índice começa em zero, teremos:

#### **a.slice(0,3);**

- > Retorna uma lista com os elementos de 0 a 3 (não incluso)
- > Essa função não altera a lista original, apenas retorna uma nova lista, então não irá influenciar no código.

#### **a.splice(1,1);**

- > Essa função altera a lista original
- > A partir da posição 1, remove 1 elemento
- > Sobra: [1,3,4,5]

#### **a.pop();**

- > Remove o último elemento do array, sobra: [1,3,4]

**Gabarito:** Letra A

**30.(IFS / IFS - 2021)** Em JavaScript, o operador new cria e inicializa um novo objeto. Qual operador **NÃO** representa a criação de um objeto de tipo nativo JavaScript?

- a) var o = new Object();
- b) var l = new ArrayList();
- c) var a = new Array();
- d) var d = new Date();

### Comentários:

Para se criar um array no JS, a única forma das citadas que não serve é var l = new ArrayList (); ArrayList é utilizado na linguagem JAVA e não no JS.

**Gabarito:** Letra B



31. (IUDS / IF-RJ – 2021) Em javascript, o que faz a função Math.round()?

- a) Retorna o valor decimal, mais próximo.
- b) Retorna o valor de um número arredondado para o inteiro, mais próximo.
- c) Retorna o valor de um número arredondado para o inteiro, logo abaixo.
- d) Retorna o valor de um número arredondado para o inteiro, logo acima.

**Comentários:**

A função Math.round() retorna o valor arredondado para o inteiro mais próximo.

**Gabarito:** Letra B

---

32. (QUADRIX / CFT – 2021) Na linguagem de programação JavaScript, por meio do operador + (adição), é permitido que uma nova variável de texto (*string*) possua valor igual à justaposição dos valores de outras variáveis.

**Comentários:**

Perfeito! No JS, por meio do operador + (adição), é permitido que uma nova variável de texto (*string*) possua valor igual à justaposição dos valores de outras variáveis.

**Gabarito:** Correto

---

33. (QUADRIX / CFT – 2021) O comprimento máximo de uma linha de código no JavaScript é de 128 caracteres.

**Comentários:**

Em tese, não existe uma limite de extensão (comprimento) de uma linha de código no JS.

**Gabarito:** Errado

---

34. (QUADRIX / CFT – 2021) As variáveis, em JavaScript, possuem um tipo definido, por isso um *array* não pode conter valores de tipos diferentes.

**Comentários:**

JS não tem tipo definido. A atribuição de tipos é dinâmica, logo o array pode sim conter valores de tipos diferentes.

**Gabarito:** Errado

---



**35. (QUADRIX / CFT – 2021)** Como toda linguagem de programação, a JavaScript também possui limitações. Sua maior limitação é não permitir que determinados objetos gráficos apresentados na página (imagem, botão etc.) respondam dinamicamente às ações do usuário.

**Comentários:**

No JS, o DOM representa todos os elementos da página web. Assim, os scripts em JS podem manipular os elementos da página, logo é possível que os mesmos respondam dinamicamente às ações do usuário.

**Gabarito:** Errado

---





## LISTA DE QUESTÕES – AJAX

1. (FCC/ TRE-MS 2007) Asynchronous Javascript and XML (Ajax) é uma técnica de desenvolvimento de aplicações web cujo objetivo é a troca de pequenas porções de dados entre um browser e um servidor web de modo a evitar a recarga de toda a página cada vez que um cliente solicita uma mudança. Sobre Ajax considere as afirmativas abaixo.

I. É uma técnica indicada para melhorar a experiência do usuário, reduzir a utilização de banda e claramente separar dados, formatação, estilo e funcionalidade.

II. O modelo de objetos html/xml (DOM) é acessado e transformado por linguagens de script como JavaScript.

III. O objeto XMLHttpRequest e/ou o objeto iFrame são utilizados para troca de dados assíncrona com o servidor.

É correto o que se afirma em:

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

2. (FCC / MPE-RS – 2008) Para capturar as respostas de uma aplicação Web dinâmica em AJAX básico utiliza-se somente:

- a) responseText e responseXML.
- b) responseText e responseHTML.
- c) responseText.
- d) responseXML.
- e) responseHTML

3. (FCC / MPE-RS – 2008) Dentre as tecnologias que compõem o AJAX, aquelas que têm como principal função fazer o intercâmbio e a manipulação de dados são:

- a) JavaScript e XMLHttpRequest.
- b) XML e XSLT.
- c) HTML e XHTML.
- d) JavaScript e CSS.
- e) DOM e CSS.

4. (FCC / MPE-RS – 2008) Para fazer o intercâmbio e a manipulação de dados em uma aplicação Web, o modelo AJAX normalmente utiliza:



- a) XML e XSLT.
- b) HTML, XHTML e CSS.
- c) XMLHttpRequest.
- d) JavaScript.
- e) DOM.

5. **(FCC / TJ-PE – 2012)** Esse objeto é o ponto chave do AJAX. Pode ser considerado um objeto Javascript que torna possível a comunicação assíncrona com o servidor. O objeto citado é do tipo:

- a) XMLHttpRequest.
- b) XMLHttpRequestObject.
- c) HttpServletResponse.
- d) HttpServletRequest.
- e) XMLHttpRequest

6. **(FCC / TCE-PR – 2011)** No AJAX, o objeto XMLHttpRequest possui diversas propriedades importantes para realizar a comunicação com o servidor. Quando uma solicitação para um servidor é enviada, normalmente deseja-se executar algumas ações com base na resposta. O evento utilizado para especificar o que vai acontecer quando a resposta do servidor está pronta para ser processada é o:

- a) oncompletedrequest.
- b) onserverreturn.
- c) onendstatus.
- d) onreadystatechange.
- e) onreadystate.

7. **(FCC / INFRAERO – 2011)** Representa uma desvantagem do uso de AJAX:

- a) Troca mensagens entre o cliente e o servidor de forma assíncrona, ou seja, envia requisições e continua o processamento sem precisar aguardar a resposta.
- b) Trata-se principalmente de JavaScript que executa no navegador do usuário. Se for desativado o processamento do JavaScript no navegador, a aplicação pode falhar.
- c) Recebe respostas às requisições na mesma página sem a necessidade de refresh.
- d) Executa os processos em paralelo às requisições/respostas.
- e) Manipula o conteúdo XML nas aplicações para desktop de forma simples.

8. **(FCC / TRE-RN – 2011)** Agrega um conjunto de tecnologias conhecidas trabalhando juntas para tornar páginas Web mais interativas com o usuário, utilizando-se de solicitações assíncronas de informações:

- a) XML.



- b) JavaScript.
- c) XHTML.
- d) AJAX.
- e) CSS.

9. (FGV / FIOCRUZ – 2010) Asynchronous JavaScript and XML (AJAX) é um termo criado recentemente para duas características poderosas dos browsers que existem há anos mas tem sido ignoradas por muitos criadores de páginas web até recentemente, quando aplicações como Gmail, Google suggest e Google Maps foram lançadas. AJAX não é uma tecnologia, na verdade, envolve várias, cada uma atuando da sua própria maneira, tornando-se juntas uma poderosa ferramenta. Nesse contexto, além de suporte à Javascript, analise as afirmativas abaixo, associadas às tecnologias incorporadas pelo AJAX.

- I. Troca e manipulação de dados usando XML e XSLT.
- II. Retorno de dados assincronamente usando XMLHttpRequest.
- III. Apresentação baseada nas Web Standards usando XHTML e CSS.

Assinale:

- a) se somente a afirmativa I estiver correta.
- b) se somente a afirmativa II estiver correta.
- c) se somente a afirmativa III estiver correta.
- d) se somente as afirmativas II e III estiverem corretas.
- e) se todas as afirmativas estiverem corretas.

10. (CESGRANRIO / DNMP – 2006) AJAX é um termo que se refere a um conjunto de tecnologias para desenvolvimento de aplicações WEB. Assinale a opção que contempla uma dessas tecnologias.

- a) Ant
- b) XML
- c) WiMAX
- d) VoIP
- e) Jakarta

11. (CESPE / TJ-RO – 2012) Que conjunto de tecnologias é utilizado em aplicações WEB na Internet para propiciar forte interatividade e dinamismo?

- a) CSS
- b) SMTP
- c) AJAX
- d) FTP
- e) UTF-8



**12. (CESPE / MEC – 2011)** AJAX (do inglês Asynchronous Javascript And XML) tem sido largamente utilizado no desenvolvimento de aplicações WEB. Um dos conceitos centrais do AJAX é a possibilidade de serem feitas requisições ao servidor através de código Javascript, rodando no navegador do usuário. Esse recurso é empregado principalmente para permitir que:

a) o processamento das regras de negócio da aplicação seja distribuído aos clientes, minimizando a carga do servidor.

b) o usuário envie uma requisição assíncrona e verifique o resultado da mesma mais tarde, sem precisar aguardar pela resposta imediata do servidor.

c) as páginas carreguem mais rapidamente, pois as requisições são paralelizadas e compactadas durante o trânsito.

d) as páginas compostas diretamente em XML, ao invés de XHTML ou HTML, sejam apresentadas no navegador do usuário.

e) partes de uma página web sejam atualizadas, sem que o browser recarregue a página inteira do servidor, proporcionando uma interface com melhor usabilidade

**13. (CESGRANRIO / IBGE – 2010)** O código HTML, em construção, abaixo demonstra a utilização de AJAX.

```
<html>
<head>
<script language="JavaScript">
function submitForm()
{
var xhr=null;
try
{ xhr = new object(); }
catch(e) {}
xhr.onreadystatechange = function()
{
document.ajax.dyn.value="Wait server...";
if(xhr.readyState == 4)
{
if(xhr.status == 200)
{ document.ajax.dyn.value="Recebido:" + xhr.responseText; }
else
{ document.ajax.dyn.value="Erro: " + xhr.status + " " + xhr.statusText; }
}
};
xhr.open("GET", "data.xml", true);
xhr.send(null);
}
```



```
</script>  
</head>  
<body>  
<FORM method="POST" name="ajax" action="">  
<INPUT type="submit" value="Submit" ONCLICK="submitForm()">  
<INPUT type="text" name="dyn" value="">  
</FORM>  
</body>  
</html>
```

Para que esse código possa utilizar a tecnologia AJAX, na linha "xhr = new object();", "xhr" deve receber um objeto Javascript que torna possível a comunicação assíncrona com o servidor, sem a necessidade de recarregar a página por completo. Para tanto, no código acima, "object" deve ser substituído por:

- a) responseXML.
- b) MIME.
- c) XMLHttpRequest.
- d) DOMParser.
- e) setRequestReader.

**14. (CESPE / PREVIC – 2011)** Ajax não é meramente uma tecnologia. É uma abordagem moderna para desenvolvimento de sites iterativos. A abordagem de desenvolvimento tradicional tem semelhanças e diferenças em relação ao Ajax. Uma característica exclusiva de Ajax em relação à abordagem tradicional é que:

- a) executa as requisições através do protocolo HTTP.
- b) usa javascript como linguagem para desenvolver código no lado do cliente.
- c) usa (x)html / css para definir o aspecto visual da página.
- d) permite recuperação assíncrona de dados usando XMLHttpRequest.
- e) representa os objetos no lado cliente com DOM.

**15. (CESGRANRIO/ LIQUIGÁS - 2011)** Duas das tecnologias que compõem o Ajax são:

- a) DOM e CSS
- b) ASP e XML
- c) Java e XML
- d) Java e CSS
- e) JavaScript e ASP

**16. (CESGRANRIO / LIQUIGÁS – 2013)** Considere as afirmativas a seguir sobre a tecnologia AJAX.

I – Uma das aplicações mais frequentes da tecnologia AJAX é na atualização assíncrona do conteúdo de páginas HTML.



II – Boa parte da funcionalidade da tecnologia AJAX é viabilizada pelo objeto XMLHttpRequest, que é capaz de transmitir requisições HTTP de modo assíncrono.

III – A tecnologia AJAX utiliza código Java para a manipulação do conteúdo de páginas HTML.

Está correto o que se afirma em:

- a) I, apenas
- b) II, apenas
- c) I e II, apenas
- d) II e III, apenas
- e) I, II e III

**17. (CESGRANRIO / IBGE – 2014)** Com o objetivo de criar páginas dinâmicas para o desenvolvimento de aplicações web, AJAX é um termo que descreve um(a):

- a) protocolo
- b) banco de dados
- c) coleção de tecnologias
- d) linguagem de programação
- e) linguagem de marcação

**18. (FGV / TCM-SP – 2015)** Considere a requisição AJAX construída, no cliente, através da chamada à seguinte função em Javascript:



```
function send (msg)
{
var message = "GET=' " + msg + "' ";

var ajax = new XMLHttpRequest ();

ajax.onreadystatechange = function ()
{
    if (this.readyState == 4 &&
        this.status == 200)
        alert (this.responseText);
}

ajax.open ("POST", "PUT.PHP", true);

ajax.setRequestHeader ("Content-type",
    "application/x-www-form-urlencoded");
ajax.setRequestHeader ("Content-length",
    message.length);
ajax.setRequestHeader ("Connection",
    "close");

ajax.send (message);
}
```

O método HTTP utilizado na requisição ao servidor será, neste caso:

- a) PUT;
- b) POST;
- c) GET;
- d) HEAD;
- e) SEND.

- 19. (UFRJ / UFRJ - 2012)** Para utilizar Ajax é preciso antes instalar e configurar um servidor Ajax, o qual interage com o servidor web.
- 20. (CESGRANRIO / LIQUIGÁS - 2013)** Uma das aplicações mais frequentes da tecnologia AJAX é na atualização assíncrona do conteúdo de páginas HTML.
- 21. (CESGRANRIO / LIQUIGÁS - 2013)** A tecnologia AJAX utiliza código Java para a manipulação do conteúdo de páginas HTML.
- 22. (FUNCAB / PRODAM - 2014)** A utilização de Ajax em uma página permite a recuperação de dados de uma URL sem a necessidade de recarregar a página inteira. O objeto JavaScript que



foi adotado pelos principais navegadores para realizar operações Ajax é chamado XMLHttpRequest.

- 23. (CESPE / MPE-RS - 2012)** AJAX é uma técnica de construção de páginas web, e sua principal vantagem é permitir a atualização de partes da página web.
- 24. (CESGRANRIO / PETROBRÁS - 2010)** Ajax não é meramente uma tecnologia. É uma abordagem moderna para desenvolvimento de sites iterativos. A abordagem de desenvolvimento tradicional tem semelhanças e diferenças em relação ao Ajax. Uma característica exclusiva de Ajax em relação à abordagem tradicional é que permite recuperação assíncrona de dados usando XMLHttpRequest.
- 25. (AOC / FUNPRESP-JUD - 2021)** JavaScript é uma linguagem que sofre muito com compatibilidade entre navegadores. A jQuery sofre com o mesmo problema. Animações, manipulação de DOM e outras tarefas corriqueiras são mais complexas e menos produtivas ao usar o jQuery.
- 26. (IFS / IFS - 2021)** O JavaScript possui características de uma linguagem funcional, portanto pode-se passar funções como parâmetros para outras funções, algo comumente encontrado nos códigos em JavaScript e Node.

Sabendo disso, analise as afirmativas abaixo:





```
I.    function soma(a,b){  
      |   return a + b;  
      |  
      }  
  
      function executar(funcao,a,b){  
      |   return funcao(a,b)  
      |  
      }  
  
      let resultado = executar(soma,1,2)  
  
      console.log(resultado)
```

```
II.   function executar(a,b,funcao){  
      |   return funcao(a,b)  
      |  
      }  
  
      let resultado = executar(1,2,function(a,b){  
      |   return a+b  
      |  
      })  
  
      console.log(resultado)
```

```
III.  var funcao = function(a,b){  
      |   return a+b  
      |  
      }  
  
      function executar(a,b,funcao){  
      |   return funcao(a,b)  
      |  
      }  
  
      let resultado = executar(1,2,funcao)  
  
      console.log(resultado)
```

Está(ão) correta(s) a(s) afirmativa(s)

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

27. (IFS / IFS - 2021) A linguagem *JavaScript* provê uma série de métodos que facilitam a manipulação de *arrays*. Sobre o método de manipulação de *array of*, é correto afirmar que:



- a) cria um novo *array* a partir de um *array* existente.
- b) preenche o *array* com um valor estático.
- c) devolve `@@iterator`, contendo os valores do *array*.
- d) cria um novo *array* a partir dos argumentos passados para o método.

28.(IFS / IFS - 2021) Considere o seguinte código JavaScript:

```
let o = {one:1,two:2,three:3};  
for(let p in o) console.log(p);
```

Ao final da execução, quais valores serão impressos?

- a) 1, 2, 3
- b) one:1, two: 2, three: 3
- c) p, p, p
- d) 'one', 'two', 'three'

29.(IFS / IFS - 2021) Considere o seguinte código JavaScript:

```
var a = [1,2,3,4,5];  
a.slice(0,3);  
a.splice(1,1);  
a.pop();
```

Qual o valor da variável a ao término da execução do código?

- a) [1,3,4]
- b) [1,3,4,5]
- c) [1,3]
- d) [3,4,5]

30.(IFS / IFS - 2021) Em JavaScript, o operador `new` cria e inicializa um novo objeto. Qual operador **NÃO** representa a criação de um objeto de tipo nativo JavaScript?

- a) `var o = new Object();`
- b) `var l = new ArrayList();`
- c) `var a = new Array();`
- d) `var d = new Date();`

31.(IUDS / IF-RJ – 2021) Em javascript, o que faz a função `Math.round()`?

- a) Retorna o valor decimal, mais próximo.
- b) Retorna o valor de um número arredondado para o inteiro, mais próximo.
- c) Retorna o valor de um número arredondado para o inteiro, logo abaixo.



d) Retorna o valor de um número arredondado para o inteiro, logo acima.

- 32. (QUADRIX / CFT – 2021)** Na linguagem de programação JavaScript, por meio do operador + (adição), é permitido que uma nova variável de texto (*string*) possua valor igual à justaposição dos valores de outras variáveis.
- 33. (QUADRIX / CFT – 2021)** O comprimento máximo de uma linha de código no JavaScript é de 128 caracteres.
- 34. (QUADRIX / CFT – 2021)** As variáveis, em JavaScript, possuem um tipo definido, por isso um *array* não pode conter valores de tipos diferentes.
- 35. (QUADRIX / CFT – 2021)** Como toda linguagem de programação, a JavaScript também possui limitações. Sua maior limitação é não permitir que determinados objetos gráficos apresentados na página (imagem, botão etc.) respondam dinamicamente às ações do usuário.



## GABARITO – AJAX

- |     |         |     |         |     |         |
|-----|---------|-----|---------|-----|---------|
| 1.  | CORRETO | 13. | LETRA E | 25. | ERRADO  |
| 2.  | LETRA E | 14. | LETRA C | 26. | ERRADO  |
| 3.  | LETRA A | 15. | LETRA D | 27. | LETRA D |
| 4.  | LETRA B | 16. | LETRA A | 28. | LETRA D |
| 5.  | LETRA A | 17. | LETRA C | 29. | LETRA D |
| 6.  | LETRA E | 18. | LETRA C | 30. | LETRA A |
| 7.  | LETRA D | 19. | ERRADO  | 31. | LETRA B |
| 8.  | LETRA B | 20. | CORRETO | 32. | LETRA B |
| 9.  | LETRA D | 21. | CORRETO | 33. | CORRETO |
| 10. | LETRA E | 22. | ERRADO  | 34. | ERRADO  |
| 11. | LETRA B | 23. | CORRETO | 35. | ERRADO  |
| 12. | LETRA C | 24. | CORRETO |     |         |



## Single Page Application (SPA)

Em um paradigma de requisição/resposta, quando utilizamos um navegador web, nós enviamos uma requisição ao servidor, ele processa essa requisição e retorna uma página HTML. Pois bem... vocês já se perguntaram porque é necessário que o servidor processe as informações e me retorne à página pronta? Ora, porque ele não me retorna apenas as informações e eu mesmo monto a página?

Eu economizo dados, diminuo o tráfego, desonero o servidor e posso fazer uma aplicação muito mais rápida. É uma maneira de equilibrar a carga entre o cliente e o servidor, além de melhorar a experiência proporcionada ao usuário da aplicação por meio de maior ênfase nas técnicas de desenvolvimento de front-end. O que é isso, professor? Grosso modo, é a apresentação visual do site.

Galera, não é só isso! O JavaScript evoluiu muito desde sua integração nos browsers. Inicialmente ele era visto apenas como uma ferramenta para fazer botões funcionarem ou para a criação de menus a pedido de web designers. Com a evolução dos browsers e da linguagem HTML, o JavaScript ganhou força e muito poder de processamento.

A Web 2.0 veio trazer o JavaScript para uma posição de destaque. E adivinhe de que lado roda JavaScript? Cliente, oras! Em outras palavras, tudo parece nos levar a um aprimoramento da experiência do usuário, com transferência de dados em background (AJAX, WebSocket, etc) e aplicações cada vez mais interativas. Vocês conseguem perceber isso?

Com mais e mais profissionais estudando e utilizando JavaScript, grandes empresas como Google, Yahoo!, Apple e Microsoft passaram a desenvolver padrões e técnicas para incrementar a performance de seus scripts, separar e isolar porções de código, e fazer uso de orientação à objeto mais avançada na linguagem. O que aconteceu a partir daí?

Foram criadas aplicações inteiramente contidas no navegador e que não precisam fazer requisições de novas páginas ao servidor. Foram compiladas as melhores práticas e criado, então, o conceito de **Single Page Application (SPA)**. São **aplicações completas**, desenvolvidas em JavaScript, que funcionam quase como se estivessem sendo executadas nativamente no desktop.

Vocês devem ter percebido que a tradução do conceito para português é Aplicação de Única Página. O nome leva a crer que estamos desenvolvendo um site inteiro em uma única página com todas as funcionalidades – uma abaixo da outra. Não é isso! Tem esse nome porque geralmente todo código (HTML, CSS e JavaScript) é recuperado em uma única carga inicial – o navegador utiliza apenas uma URL.



Após essa página inicial ser carregada, toda a lógica de apresentação está no cliente. Eu sei, vocês devem estar achando um pouco complexo. Querem ver um exemplo? Abram o GMail, loguem-se em suas contas e clique em uma mensagem na Caixa de Entrada. Observem que o navegador continua na mesma página, mas o código JavaScript esconde a caixa de entrada e traz a mensagem para a tela.

Modo de exibição do Gmail: [padrão](#) | [HTML básico](#) [Saiba mais](#)

Não sei se vocês se lembram de como era antigamente, então – para ver claramente a diferença – mudem agora o Modo de Exibição de Padrão para Básico! Repitam a mesma operação, i.e., entrem na Caixa de Entrada e cliquem em um e-mail qualquer. Observem, dessa vez, que uma nova página é carregada com o corpo da mensagem que você clicou.

Vocês percebem a diferença de velocidade? No primeiro caso, é quase instantâneo – parece até uma aplicação desktop. O SPA oferece dois benefícios principais aos usuários: remover aquela mudança brusca de página quando você clica em um link – em geral, os principais controles de navegação e interface permanecem na página e apenas o pedaço de conteúdo que você deseja alterar que realmente muda.

Além disso, há um aumento significativo de velocidade da carga de dados de resposta vinda de um servidor mais leve, visto que se trata de uma parte da página e, não, da página inteira. Uma carga de dados mais leve pode ser transmitida pela rede mais rapidamente e o navegador pode incorporar o novo conteúdo de forma mais rápida do que redesenhar uma página inteiramente nova.



Vamos resumir nosso papo? Single Page Application é apenas um website em que uma página não é, de fato, carregada. Em uma metodologia tradicional, um cliente faz uma requisição a um servidor, que retorna uma Página HTML (+CSS, JavaScript, etc) que é carregada no cliente. Se você clicar em outro link ou fizer qualquer outra requisição, esse processo se repetirá.

Nessa abordagem alternativa, acontece um pouco diferente! Uma requisição é feita ao servidor e a página retornada é renderizada no cliente. A partir daí, o cliente faz requisições apenas de dados ao servidor, que rapidamente os busca e retorna de volta ao cliente, que recebe, faz o que quiser com os dados e modifica apenas as partes da página que precisam ser modificadas. É isso, vamos às questões...



# 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.