

Recode Pro 2020

JavaScript



JavaScript

Aula 1



JS

COMO FAZER UM SANDUÍCHE

@Newsflare



DESAFIO DE INSTRUÇÕES EXATAS

INCRÍVEL

JS

Também chamada de JS, é a linguagem de criação de scripts para a Web.

Originalmente criada na Netscape por Brendan Eich em 1994.

Java e JavaScript são “coisas” completamente distintas e desconexas.

JavaScript não permite a criação de applets nem de aplicativos.

JavaScript reside dentro de documentos HTML e pode prover diferentes níveis de interatividades não suportados pelo HTML sozinho.

Diferenças chaves em relação ao Java:

- Java é uma linguagem de programação;
- JavaScript é uma linguagem de script;
- Aplicativos Java são executados pela máquina virtual Java;
- Scripts JavaScript são executados pelos browsers;
- Java é compilado;
- JavaScript é texto puro;
- Cada tecnologia requer um plug-in diferente.

É utilizado por bilhões de páginas para:

- Adicionar funcionalidades;
- Verificar formulários; e
- Comunicar com servidores.

JS

Formas de utilização do JS

- No elemento HTML.
- Na tag `<script>`.
- No arquivo externo.

Assim, dispomos de três opções para poder fazer um trabalho bem direcionado e atender com excelência as demandas do dia a dia.

Orientação a objetos

Diferente da Linguagem HTML, a linguagem JavaScript corresponde a programação orientada a objetos, isto significa que todos os elementos de uma página da Web são tratados como objetos.

Estes objetos são agrupados de acordo com seu tipo ou finalidade.

Dentro da linguagem JavaScript, são criados automaticamente objetos que permitem que o usuário possa criar novos objetos de acordo com sua conveniência.

JS

propriedades e valores próprios que são ajustados pelo conteúdo da própria página.

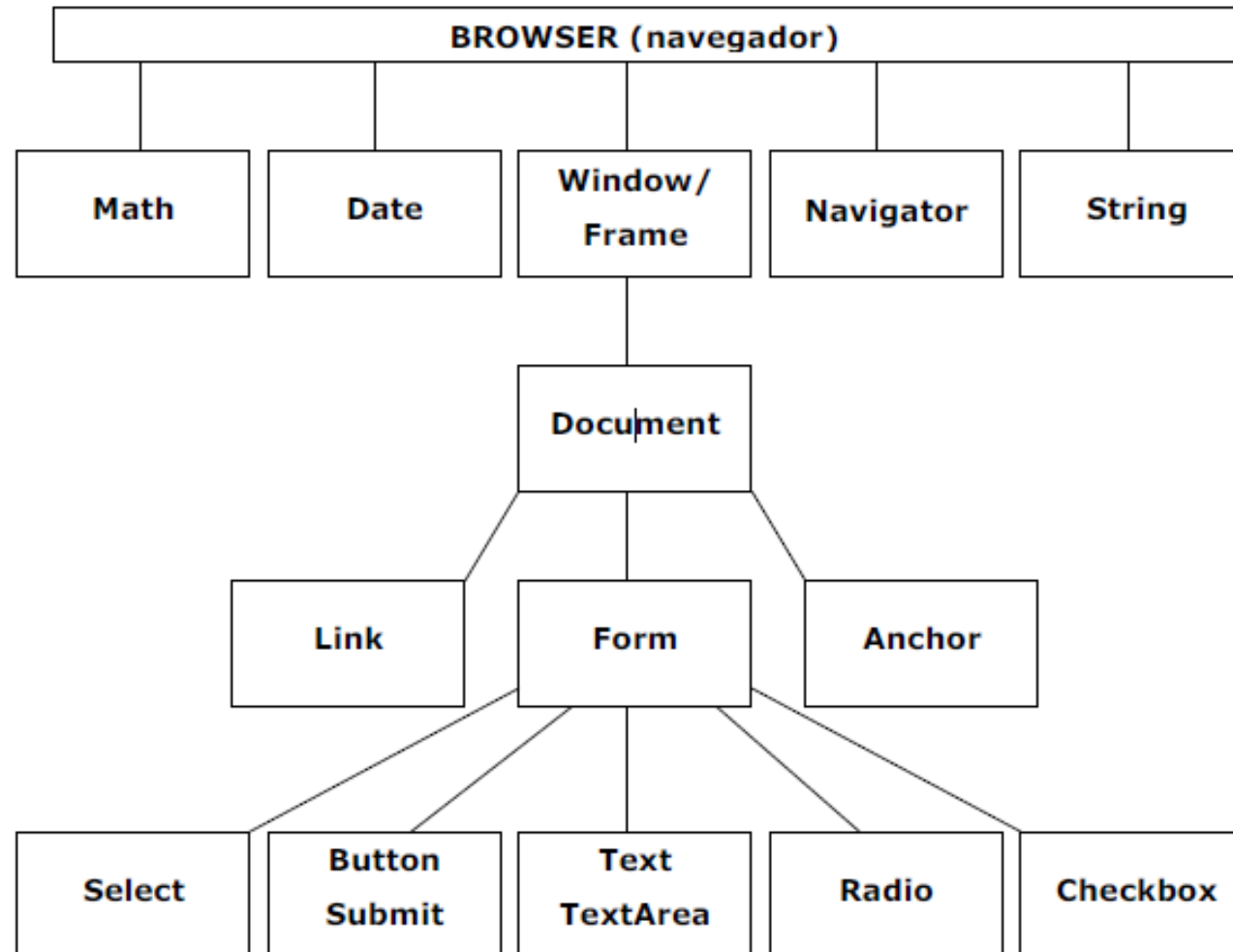
Todos eles seguem uma hierarquia que reflete toda a estrutura de uma página HTML.

A linguagem JavaScript pode ser utilizada para a criação de scripts tanto do lado cliente como do lado servidor. Seguindo a hierarquia de objetos da linguagem JavaScript, são criados os seguintes objetos ao ser carregada uma página:

- **window** → O objecto mais acima na hierarquia, contém propriedades que se aplicam a toda a janela. Há também um objecto desta classe para todas as "sub-janelas" de um documento com frames location: Contém as propriedades da URL actual.
- **history** → Contém as propriedades das URLs visitadas anteriormente.
- **document** → Contém as propriedades do documento contido na janela, tais como o seu conteúdo, título, cores, etc.

Hierarquia dos objetos do JavaScript

A linguagem JavaScript manipula vários tipos de objetos através do uso de suas propriedades e métodos. Estes objetos são representados por uma hierarquia, fazendo com que alguns objetos se tornem propriedades de outros.



Propriedade de objetos

Cada objeto existente na manipulação do JavaScript possui propriedades (características).

Exemplo: sabemos que um documento HTML possuem título e corpo, estas características do documento podemos chamar de propriedades que existem neste documento.

Estas propriedades existem de dois tipos, algumas são os objetos propriamente ditos enquanto outras não.

Exemplo: o objeto form (formulário) que é uma propriedade do objeto document (documento), conforme mostrado no organograma apresentado anteriormente.

JS

Já a propriedade de título da página (title), é pertencente ao objeto document não havendo nenhuma propriedade sobre ela. Concluindo, podemos dizer que a propriedade form do objeto document é um objeto-filho e o objeto document é o objeto-pai.

Em geral, as propriedades podem conter valores (string, números, booleanos, array entre outros tipos). A utilização de propriedades se dá acompanhada de seu objeto sendo separados por um ponto apenas.

Veja abaixo a sintaxe de utilização de propriedades:

nomeobjeto.propriedade

Métodos dos objetos

Além das propriedades, os objetos podem conter métodos que são funções pré-definidas pela linguagem JavaScript que irão executar determinada operação.

Exemplo: dentro de um documento o usuário poderá utilizar o método de escrever neste documento para exibir um texto qualquer.

Os métodos estarão sempre associados à algum objeto presente no documento e cada método faz parte de um objeto específico.

JS

Não use métodos em objetos que não o utilizam, isto causa erro na execução do script. Na maior parte os métodos são usados para alterar o valor de uma propriedade ou executar uma tarefa específica. **Sintaxe:**

nomeobjeto.metodo(argumento)

nomeobjeto → faz referência ao objeto a ser utilizado e o qual sofrerá uma ação do método.

método → é o nome de identificação do método usado e entre parênteses.

argumento → é a expressão ou valor opcional que será usada para alterar sobre o objeto.

Eventos

Em linguagens orientadas a objetos é comum a manipulação de eventos que é qualquer reação ou ação que executará determinado procedimento, normalmente ocorre por ato executado pelo usuário:

- Clicar em um botão;
- Selecionar algum objeto; e
- Pressionar alguma tecla.

Eventos são quaisquer ações iniciadas por parte do usuário.

Sua utilização se dá como atributos da linguagem HTML, ou seja dentro dos próprios Tag's HTML.

JS

Sintaxe:

`<TAG nomeevento="Instruções JavaScript">`

Onde é apresentado **TAG** é uma instrução da linguagem HTML.

Onde é evento é o nome do evento gerado da linguagem JavaScript.

Onde “**Instruções JavaScript**” serão as instruções JavaScript à serem executadas.

Elas estarão sempre entre aspas.

JS

Caso haja mais de um comando JavaScript a ser executado para o mesmo evento estes deverão estar separados por ponto e vírgula (;), conforme mostrado no exemplo a seguir:

Exemplo:

```
<TAG nomeevento="JavaScript1; JavaScript2; JavaScript3">
```

JS

Exemplo do javascript na TAG <button>:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Usando o Button com onclick</title>
```

```
  </head>
```

```
  <body>
```

```
    <button  onclick='document.write("Ação realizada com  
    sucesso")'>Clique no botão!</button>
```

```
  </body>
```

```
</html>
```

Manipulador de eventos que podem ser usados

EVENTO	MANIPULADOR	DESCRIÇÃO
blur	onBlur	Ocorre quando o usuário retira o foco de um objeto de formulário.
change	onChange	Ocorre quando o usuário muda o valor de um objeto de formulário.
click	onClick	Ocorre quando o usuário clica sobre o objeto.
focus	onFocus	Ocorre quando o usuário focaliza o objeto.
load	onLoad	Ocorre quando o usuário carrega a página.
unload	onUnload	Ocorre quando o usuário abandona a página.
mouseover	onMouseOver	Ocorre quando o ponteiro do mouse passa sobre um link ou âncora. Válidos apenas para hiperlinks.
select	onSelect	Ocorre quando o usuário seleciona um elemento de um formulário.

EVENTO	MANIPULADOR	DESCRIÇÃO
submit	onSubmit	Ocorre quando o usuário envia um formulário.
mouseDown	onMouseDown	Ocorre quando o botão do mouse é pressionado.
mousemove	onMouseMove	Ocorre quando o ponteiro do mouse se movimenta sobre o objeto.
mouseout	onMouseOut	Ocorre quando o ponteiro do mouse afasta de um objeto. Válidos apenas para hiperlinks.
mouseUp	onMouseUp	Ocorre quando o botão do mouse é solto.
keyDown	onKeyDown	Ocorre quando uma tecla é segurada.
keyPress	onKeyPress	Ocorre quando uma tecla é pressionada.
keyUp	onKeyUp	Ocorre quando uma tecla é solta.

JS

A Tag <script>

Para inserir códigos JavaScript, iremos fazê-lo em uma Tag HTML apropriada para a interpretação dos códigos JS:

- <script>...</script>

Uma das formas de trabalho na tag **<script>**, é o uso de uma **função nomeada** (function) e, no elemento, devemos inserir o nome da função que será executada.

<script>

function limpar()

{

<comando>;

}

</script>

limpar é o nome da função a ser chamada em
outro ponto

Instrução a ser executada

JS

Em documentos HTML, a utilização da linguagem JavaScript, se dá sob a forma de funções, as quais são chamadas em determinadas situações ou em resposta a determinados eventos, estas funções podem estar localizadas em qualquer parte do código HTML, a única restrição é que devem começar com a declaração `<SCRIPT>` e termina com o respectivo `</SCRIPT>`.

Por convenção costuma-se colocar todas as funções no início do documento (**entre as TAG `<HEAD>` e `</HEAD>`**), isso para garantir que o código JavaScript seja carregado antes que o usuário interaja com a Home Page, ou seja, antes da TAG `<BODY>`.

Observação:

É importante ressaltar que todas as linhas devem ser terminadas com “;” (ponto e virgula) a menos que a próxima instrução seja um “else” e se você precisar escrever mais de uma linha para executar uma condição seja ela em uma estrutura “for”, “if” ou “while”, este bloco de instruções deve estar entre “{ }” (chaves).

Inclusive a definição de funções segue este modelo, ou seja, todo o código da função deve estar limitado por { (no início) e } (no final).

Um browser que não suporta JavaScript, ele não conhece a TAG.

JS

Classe document

Propriedades

- title – Define ou Retorna o Título da Página;
- url – Retorna o URL completo da página;

Métodos

- write() – Escreve texto no documento;
- writeln() – Escreve uma linha de texto no documento;

JS

Método `document.write()`

Esta instrução na realidade segue a sintaxe de ponto da linguagem JavaScript, uma das maneiras de seguir a hierarquia dos objetos presentes na linguagem.

Nesta linha de comando temos o método `write()` que é pertencente ao objeto `document` que retrata o documento como um todo.

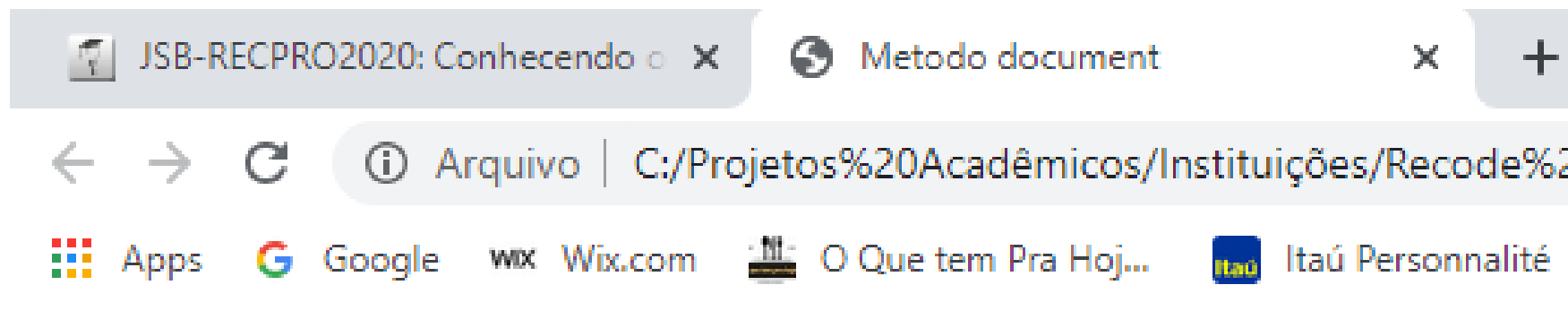
Exemplo:

```
document.write("Texto inserido com instruções JavaScript");
```

JS

Exemplo:

```
<html>
  <head>
    <title>Metodo document</title>
  </head>
  <body>
    <h1>Uso do document.write</h1>
    <script>
      document.write("Entendendo o uso deste
      metodo");
    </script>
  </body>
</html>
```



Uso do document.write

Entendendo o uso deste metodo

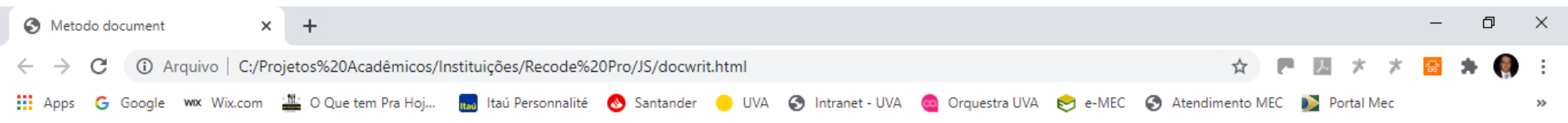
Exercício 1:

Duração de 20 minutos.

1. Criar um arquivo HTML de nome docwrite1.html.
2. Criar três tags de h1, h2 e h3; e
3. Em cada uma das tags, colocar uma mensagem com o uso do `document.write`.

Código do exercício 1:

```
<html>
  <head>
    <title>Metodo document</title>
  </head>
  <body>
    <h1>Uso do document.write no H1</h1>
    <script>
      document.write("Entendendo o uso deste metodo no H1");
    </script>
    <h2>Uso do document.write no H2</h2>
    <script>
      document.write("Entendendo o uso deste metodo no H2");
    </script>
    <h3>Uso do document.write no H3</h3>
    <script>
      document.write("Entendendo o uso deste metodo no H3");
    </script>
  </body>
</html>
```



Uso do document.write no H1

Entendendo o uso deste metodo no H1

Uso do document.write no H2

Entendendo o uso deste metodo no H2

Uso do document.write no H3

Entendendo o uso deste metodo no H3



JS

Método alert()

A finalidade deste método é emitir uma caixa de diálogo do windows passando com uma mensagem e um botão de OK.

Este método é pertencente ao objeto window do JavaScript. Observe a sintaxe de seu funcionamento.

Exemplo:

```
window.alert("Meu Primeiro Script");
```

ou

```
alert("Meu Primeiro Script");
```

JS

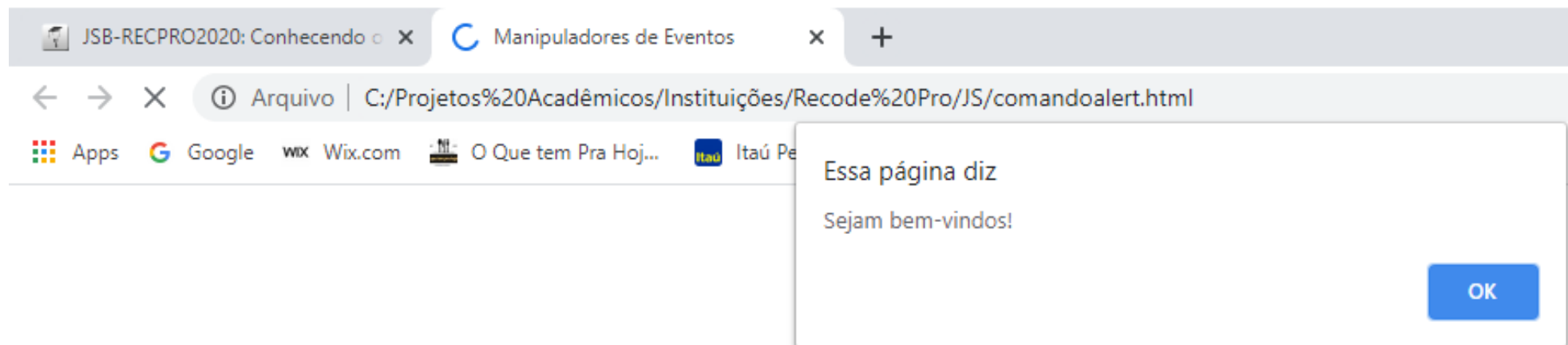
Alert (caixa de mensagem para o usuário)

Comando inicial para dar uma mensagem para o usuário.

Exemplo:

```
<html>
  <head>
    <title>Manipuladores de Eventos</title>
  </head>
  <body>
    <h1>Comando alert para o usuario</h1>
    <script>
      alert ("Sejam bem-vindos! ");
    </script>
  </body>
</html>
```


JS



Exercício 2:

Duração de 20 minutos.

1. Crie um arquivo HTML de nome exercalert.html;
2. Criar um título “Trabalhando com o Alert”;
3. Criar uma tag `<h1>` com o título “Comando alert para o usuário”; e
4. Dentro da TAG `<script>` crie três mensagens quaisquer para informações ao usuário.

Código do exercício 2:

```
<html>
  <head>
    <title>Trabalhando com o Alert</title>
  </head>
  <body>
    <h1>Comando alert para o usuario</h1>
    <script>
      alert ("Você acaba de ser sorteado");
      alert ("Agora é preciso entender as regras");
      alert ("seu prêmio já pode ser resgatado");
    </script>
  </body>
</html>
```