INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Desenvolvimento Web I

Professor: Euclides Paim

euclides.paim@ifc.edu.br

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Introdução *JavaScript*

Professor: Euclides Paim

euclides.paim@ifc.edu.br

Introdução ao *JavaScript*

Intro *JavaScript*

JavaScript é a linguagem de programação da Web. A esmagadora maioria dos sites modernos utilizam *JavaScript* ou um de seus *frameworks* e todos os navegadores modernos - seja em desktops, jogos consoles, tablets e smartphones - incluem interpretadores *JavaScript*, tornando essa linguagem de programação a mais onipresente da história.

JavaScript faz parte da tríade de tecnologias que todos os desenvolvedores da Web devem aprender: HTML para especificar o conteúdo de páginas da web, CSS para especificar a apresentação de páginas da web e *JavaScript* para especificar o comportamento das páginas da web. Durante as próximas aulas iremos conhecer esta linguagem.

Introdução ao *JavaScript*

JavaScript não é Java

JavaScript frequentemente é confundida com a linguagem Java, provavelmente devido à semelhança do nome. Há também algumas semelhanças na sintaxe no entanto tudo mais é diferente. O nome "*script*", que significa roteiro, já indica que se trata de uma linguagem interpretada. Podemos apontar ainda outras diferenças:

- Interpretada. Programas em *Java* são compilados, programas em *JavaScript* são interpretados linha-por-linha.
- Simples. Programas em Java são bem mais poderosos que programas *JavaScript.*
- Baseada em objetos enquanto *Java* é uma linguagem orientada a objetos.
- Extensão do HTML. Não se usa *Java* em uma página Web. Não existe *JavaScript* (*client-side*) sem HTML.

Introdução ao *JavaScript*

O que se pode fazer com *JavaScript*?

Com *JavaScript* pode-se fazer diversas coisas que não são possíveis apenas com a linguagem HTML, como:

- Realizar operações matemáticas e computação.
- Gerar documentos com aparência definida na hora da visualização, com base em informações do cliente como versões do browser, cookies e outras propriedades.
- Abrir janelas do browser, trocar informações entre janelas, manipular com propriedades do browser como o histórico, barra de estado, *plug-ins* e *applets*.
- Interagir com o conteúdo do documento, alterando propriedades da página, dos elementos HTML e tratando toda a página como uma estrutura de objetos (DOM).
- Interagir com o usuário através do tratamento de eventos.

Introdução ao *JavaScript*

Formas de usar *JavaScript*

Há pelo menos três maneiras de incluir *JavaScript* em uma página Web:

- Dentro de blocos HTML <SCRIPT> ... </SCRIPT> em várias partes da página: para definir funções, gerar HTML ou alterar o procedimento normal de interpretação do HTML da página pelo browser.
- Em um arquivo externo, importado pela página: para definir funções que serão usadas por várias páginas de um site.
- Dentro de descritores HTML sensíveis a eventos: para tratar eventos do usuário em links, botões e componentes de entrada de dados, durante a exibição da página.

Obs.: As três formas podem ser usadas em uma mesma página.

Introdução ao *JavaScript*

A tag <script>

Em HTML, o código JavaScript é inserido entre tags <script> </script>.

```
<script>
  document.getElementById("demo").innerHTML = "Meu primeiro JavaScript";
</script>
```

Obs.: Exemplos antigos de *JavaScript* podem conter um atributo do tipo *type,* este atributo não é mais necessário uma vez que *JavaScript* é a linguagem de *script* padrão no HTML atualmente.

~~<script type = "text / javascript">~~.

Introdução ao *JavaScript*

JavaScript no <head> ou <body>

Você pode colocar qualquer número de *scripts* em um documento HTML. Os *scripts* podem ser colocados no <body>, ou na seção <head> de uma página HTML, ou em ambos.

Obs.: Colocar *scripts* na parte inferior do elemento <body> melhora a velocidade de exibição, porque a interpretação *scripts* retarda o *display*.

Introdução ao *JavaScript*

JavaScript externo

Scripts também podem ser colocados em arquivos externos. *Scripts* externos são práticos quando o mesmo código é utilizado em diferentes páginas da web . Arquivos *JavaScript* tem a extensão de arquivo **".js"**. Para usar um script externo, coloque o nome do arquivo de script no atributo *src* (fonte) de uma tag <script>:

```
<script src="myScript.js"></script>
```

Você pode colocar uma referência de *script* externo nas seções <head> ou <body> como preferir. O *script* irá se comportar como se ele estivesse localizado exatamente onde a tag <script> está localizada.

Introdução ao *JavaScript*

Vantagens *JavaScript* externo

Colocar *scripts* em arquivos externos tem algumas vantagens:

- Ele separa o código HTML
- Deixa o HTML e *JavaScript* mais fáceis de ler e manter
- Arquivos *JavaScript* em cache podem acelerar o carregamento da página

Para adicionar vários arquivos de script em uma página - use várias marcas de script:

```
<script src="myScript1.js"></script>
<script src="myScript2.js"></script>
```

Introdução ao *JavaScript*

Os *scripts* externos podem ser referenciados com uma URL completa ou com um caminho relativo à página web atual. Este exemplo vincula um *script* localizado na mesma pasta da página atual:

```
<script src="myScript1.js"></script>
```

Obs.: Exemplo nos arquivos da aula. (Exemplo1)

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Introdução ao *JavaScript*

Output

O *JavaScript* pode "exibir" dados de diferentes maneiras:

- Escrevendo em um elemento HTML, **usando *innerHTML***
- Escrevendo na saída HTML usando ***document.write()***
- Escrevendo em uma caixa de alerta, usando ***window.alert()***
- Escrevendo no console do navegador, usando ***console.log()***

Introdução ao *JavaScript*

Output

Usando ***innerHTML:***

Para acessar um elemento HTML, o JavaScript pode usar o método:

document.getElementById(id)

O atributo *id* define o elemento HTML. A propriedade *innerHTML* define o conteúdo HTML. Alterar a propriedade *innerHTML* de um elemento é uma maneira comum de exibir dados em HTML.

Obs.: Exemplo nos arquivos da aula. (Exemplo2)

Introdução ao *JavaScript*

Output

Usando ***document.write()***

Para fins de teste, é conveniente usar *document.write()*:

```
<script>document.write(5 + 6);</script>
```

O uso de *document.write()* após o carregamento de um documento HTML **excluirá todo o HTML existente.**

Obs.: Exemplos de uso nos arquivos da aula. (Exemplo3)

Introdução ao *JavaScript*

Output

Usando ***window.alert()***

Você pode usar uma caixa de alerta para exibir dados:

```
<script>
    window.alert(5 + 6);
</script>
```

Obs.: Exemplos de uso nos arquivos da aula. (Exemplo4)

Introdução ao *JavaScript*

Output

Usando ***console.log ()***

Para fins de depuração, você pode chamar o método *console.log()* no navegador para exibir dados. Aprenderemos mais sobre depuração em um capítulo posterior.

```
<script>
    console.log(5 + 6);
</script>
```

Obs.: Exemplo de uso nos arquivos da aula. (Exemplo5)

Introdução ao *JavaScript*

Output

Usando ***console.log ()***

Você também pode testar alguns dos métodos estudados neste capítulo digitando eles diretamente no console.

1. Pressione F12 no seu teclado para ativar o depurador.
2. Selecione "Console" no menu.
3. Digite um dos exemplos a seguir:

```
> document.write(5+6);
> document.write("Usando o cosole para testes");
> window.alert(5 + 6);
```

Introdução ao *JavaScript*

Output

JavaScript Print

O *JavaScript* não possui nenhum objeto ou método de impressão. Você não pode acessar dispositivos de saída a partir do *JavaScript*. A única exceção é que você pode chamar o método *window.print()* no navegador para imprimir o conteúdo da janela atual. Exemplo:

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

<button onclick="window.print()">Imprima esta página!</button>

</body>
</html>
```

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Introdução ao *JavaScript*
Resumo

- Histórico do JavaScript
- JavaScript não é Java
  - Interpretada
  - Simples
  - Extensão do HTML...

- O que podemos fazer com JavaScript:
  - Matemática e Computação
  - Interagir com o HTML
  - Interagir com o usuário...

- Onde usar JavaScript:
  - Na *head*, no *body*, arquivo externo...

- Formas de saída dos resultados em JavaScript:
  - usando ***innerHTML;*** escrevendo na saída HTML
  usando ***document.write();*** usando ***window.alert();*** usando o ***console.log().***

INSTITUTO FEDERAL
Catarinense
Campus Camboriú

Referências

Referências Básicas

Livro NIEDERAUER, Juliano. Desenvolvendo websites com PHP: aprenda a criar websites dinâmicos e interativos com PHP e bancos de dados. 2. ed. rev. e atual. São Paulo: Novatec, 2011.

Livro SILVA, Maurício Samy. HTML 5: a linguagem de marcação que revolucionou a web. São Paulo: Novatec, 2011.

Livro SILVA, Maurício Samy. CSS3: desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3. São Paulo: Novatec, 2012.

Referências Complementares

Livro DEITEL, Paul J. Ajax,. Rich Internet applications e desenvolvimento Web para programadores. . Pearson Prentice Hall. 2009

Livro DALL'OGLIO, Pablo. PHP: programando com orientação a objetos. . Novatec. 2009

Livro SOARES, Walace. PHP 5: conceitos, programação e integração com banco de dados. . Érica. 2010

Livro SILVA, Maurício Samy. Criando sites com HTML: sites de alta qualidade com HTML e CSS. . Novatec. 2010

Livro FLANAGAN, David. o guia definitivo. . O Really. 2012

Referências na Internet

https://www.w3schools.com

https://developer.mozilla.org/pt-BR/docs/Web