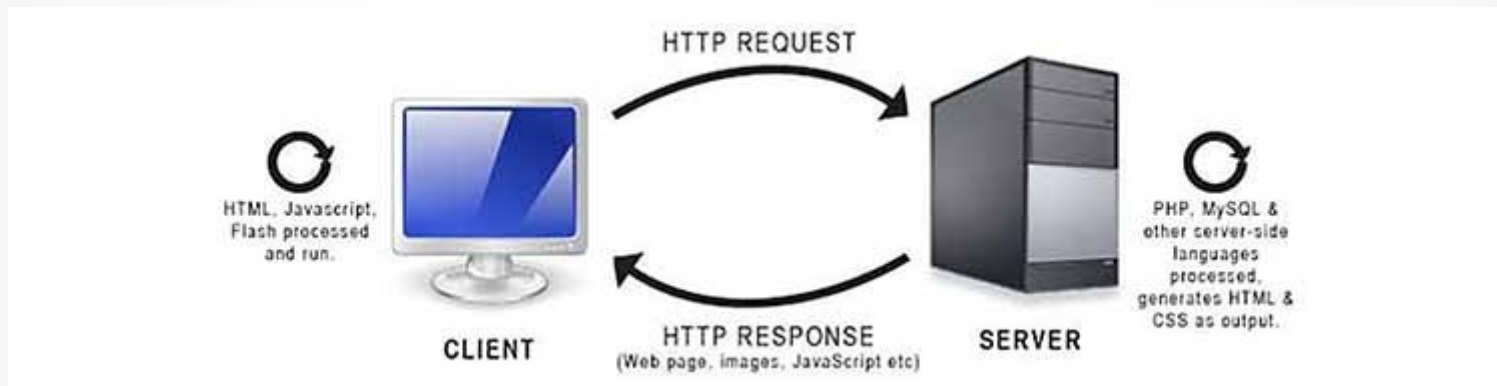


Introdução ao JavaScript

- A estrutura da linguagem
- Suas constantes e variáveis
- A concatenação de strings
- Depurando o código JavaScript

JavaScript

- Linguagem de programação *client-side*



- Linguagem de *script* leve
- Adiciona interatividade e personaliza comportamentos nos sites

Estrutura de uma página *web*

- 3 camadas:
 - Estrutural: marcação HTML
 - Apresentação: CSS
 - Comportamental: JavaScript

O que o JavaScript pode fazer?

- Todos os elementos, atributos e textos de uma página *web* podem ser acessados pelo JavaScript
- *Scripts* que reagem à entrada do usuário
 - Exemplo: sugerir o termo completo a um usuário que digita um texto em uma caixa de pesquisa (ex. Google)

O que o JavaScript pode fazer?

- Solicitar conteúdo e as informações do servidor e injetá-los no documento atual, conforme necessário, sem recarregar a página inteira ("Ajax")
- Mostrar e ocultar conteúdo quando um usuário clicar em um elemento
- Alterar comportamentos do navegador
- Etc...

JavaScript: Exemplos

Whoops! Some errors occurred.

- That username is already in use.
- Email confirmation doesn't match

Username

Must be at least 4 characters

Email

Confirm Email

Password

Confirm Password

JavaScript: Exemplos

what can javascript do



what can javascript do

what can javascript be used for

what can javascript do for a website

what can javascript programs do

JavaScript: Exemplos



Execução de programas JavaScript

- Os programas são executados na ordem em que aparecem na página
- 3 formas de incluir JavaScript em uma página:
 - através da tag **<SCRIPT> ... </SCRIPT>** no documento HTML ou através de um arquivo de extensão **.js** separado
 - através da seleção de um **link**
 - através da associação a um **evento**

Exemplos

- Agora vamos praticar essas 3 formas de incluir o código JavaScript em um arquivo HTML
- Abra um editor de sua preferência (Bloco de Notas, Notepad++, Sublime, ...) e crie uma página HTML básica.

Exemplos – Código HTML Básico

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Incluindo JavaScript em um documento HTML</title>
  </head>
  <body>
  </body>
</html>
```

Salve como basico.html

1º - Inclusão através da *tag script*

- Pode aparecer em qualquer lugar da página, inclusive na área de cabeçalho (dentro da tag <HEAD> ... </HEAD>).
- Quando incluída na área de cabeçalho, o programa é executado antes que a página seja totalmente carregada.

<script type="text/javascript">

<!--

...//codigo Javascript ...

-->

</script>

- A inclusão do código entre comentários é recomendável para que navegadores que não conheçam JavaScript não tentem exibir o código.

1º - Inclusão através da tag script

- Acrescente o seguinte código à sua página HTML:

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Script no documento HTML</title>
    <script type="text/javascript">
      document.write("Meu primeiro script" + "<br>");
    </script>
  </head>
  <body>
    <script type="text/javascript">
      document.write("Hello World!");
    </script>
  </body>
</html>
```

- Salve como “script_documento.html”
- Teste o seu código clicando duas vezes no arquivo

JavaScript em arquivo separado

- Crie um arquivo “script.js”
- Inclua o seguinte código:

```
document.write("Meu segundo script"  
+ "<br>" + "Hello World!");
```

- Salve o arquivo “script.js”

JavaScript em arquivo separado

- Abra o arquivo “básico.html”
- Inclua o seguinte código:

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Script em arquivo externo</title>
    <script src="script.js"></script>
  </head>
  <body>
  </body>
</html>
```

- Salve como “script_externo.html” no mesmo diretório de “script.js”. Faça o teste.

2º - Inclusão de JavaScript através da seleção de um link

- Para executar um código JavaScript através da seleção de um link, você deve incluí-lo em uma tag HTML:

` `

2º - Inclusão de JavaScript através da seleção de um link

- Abra o arquivo basico.html
- Inclua o seguinte código:

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Script através de links HTML</title>
  </head>
  <body>
    <a href="javascript: document.write('Hello World!')">Clica aqui</a>
  </body>
</html>
```

- Salve o arquivo como “script_link.html”. Faça o teste.

3º Inclusão de JavaScript através da associação a um evento

- Para executar um código JavaScript através da associação ao um evento, você deve incluir o código em um atributo evento da *tag* HTML desejada:

<tag-html onEvento=" ... ">

3º Inclusão de JavaScript através da associação a um evento

- Abra o arquivo “basico.html”
- Inclua o seguinte código:

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Script através de eventos</title>
  </head>
  <body>
    <h1 onclick="document.write('Hello World!')">Clica aqui</h1>
  </body>
</html>
```

- Salve o arquivo como “script_evento.html”. Faça o teste.

Estrutura léxica da linguagem

- Linguagem *case sensitivity* (como na linguagem C e diferente de HTML)
- As palavras-chave da linguagem, as variáveis, os nomes de funções ou qualquer outro identificador deve sempre ser escrito da mesma forma (por exemplo, a palavra-chave *break*)

Estrutura léxica da linguagem

- Espaços em branco, tabulações (*tabs*) e as mudanças de linha (*newlines*) entre os comandos são ignorados
- O caracter **ponto e vírgula (;)** deve ser usado como limitadores, após cada comando. (boa prática)

Inclusão de comentários

- 2 formas:
 - Comentário de linha: `//`
 - Comentário de bloco: `/* ... */`

Tipos de dados

- 3 tipos primitivos de dados: números, strings e booleanos

Tipo	Constante
string	"Lanchonete Bob's", "Pi = 3.14", "A"
	'vi o filme "Central do Brasil"', 'Pi=3.14', 'A'
inteiro na base 10	8, -12, -9007199254740992, 9007199254740992
inteiro octal	056, -0123, 0377,
inteiro hexadecimal	0x7A, -0x10, 0xCAFE911, 0xff, 0377
real	3.14, -10.28, .33333333, 6.02e+23, 1.4738E-32
booleano	true , false

Variáveis

- Os nomes de variáveis podem conter letras, dígitos (exceto no início do nome), o caracter “_” e o caracter “\$”. Exemplos:
 - **minhaVariavel**, **minhavariavel2**, **minha_variavel**, **minhavariavel\$**
- Palavras reservadas não podem ser usadas como nome de variáveis:
 - **true**, **var**, **if**, **while** etc...

Declaração de variáveis

- A criação de uma variável em JavaScript é chamada de "declaração" de uma variável.
- Não é obrigatório declarar previamente uma variável nem definir explicitamente o seu tipo, mas...

É uma boa prática de programação declarar todas as variáveis no início de um *script*.

- Para declarar variável, usa-se a palavra-chave: **var**

Aprenda mais sobre declaração de variáveis em JavaScript aqui:

https://www.w3schools.com/js/js_variables.asp

Declaração de variáveis

- São exemplos de declarações e usos de variáveis:

```
var i;           //Cria a variavel i
i = 10;          //Guarda o valor 10 em i
var i = 2;       //Cria e armazena 2
i = 2;           /*idem a linha anterior: a palavra
                  "var" é opcional, a principio */
i = "onze";      /* agora o tipo da variável
                  passou a ser string. */
```

Números

- Inteiros na base 10.
- Inteiros octais.
- Inteiros hexadecimais.
- Pontos flutuantes.

Inteiros na base 10

- **Não iniciam com zero**, à exceção do próprio zero.
- Todos os inteiros entre -2^{53} ($= -9007199254740992$) e 2^{53} ($= 9007199254740992$), inclusive, podem ser representados em JavaScript.

Octal (base 8)

- **Começa com o dígito zero**, seguido por uma seqüência de números entre 0 e 7.
 - Exemplos:
 - **01234**
 - **-077**
 - **0312**

Hexadecimais

- **Começa com 0x ou 0X**, seguido de dígitos de 0 a 9 ou letras **a** (ou **A**) a **f**(ou **F**), as quais representam os números 10 a 15.
 - Exemplos:
 - **0xFF**
 - **-0xCCFF**
 - **0abcdef**

Valores numéricos especiais

- Um ponto flutuante se torna maior que o valor máximo representável = **Infinity**
- Um valor negativo se torna mais negativo que o maior negativo possível de ser representado = **-Infinity**
- Operação matemática que resulta um valor indefinido (exemplo zero dividido por zero) = “not-a-number” (**NaN**)

Ponto Flutuante

- Podem ter um ponto decimal, uma parte inteira do número e uma parte fracionária (depois do ponto decimal).
- Também podem ser representados na notação exponencial: onde a letra e (ou E) seguida de um sinal + ou - introduz expoentes inteiros que representam potências de 10.
 - Exemplos:
 - $100.10 = 1.001e+2$
 - $0.01 = 1E-2$

Constantes numéricas especiais

Constante	Significado
Number.MAX_VALUE	O maior valor possível
Number.MIN_VALUE	O menor (mais próximo a zero) possível
Number.POSITIVE_INFINITY	Representação especial de infinito
Number.NEGATIVE_INFINITY	Representação especial de infinito negativo
Number.NaN	Valor especial not-a-number

https://www.w3schools.com/jsref/jsref_obj_number.asp

Strings

- Sequência de letras, dígitos, caracteres de pontuação e outros, que são representados pela linguagem como texto.
- Em JavaScript, um caractere isolado é representado como uma *string* de tamanho 1.
- **Definição de *strings*:** valor contido entre '...' ou “...”

Concatenação

- Se você usar o símbolo `+` (mais) com números, terá como resultado uma operação de adição.
- Utilizando esse mesmo símbolo com **strings**, será feita a concatenação das mesmas.
 - Exemplo: `"2 dividido por " + "0" + " = " + "matematicamente indefinido!"`

Booleanos

- 2 valores: **true** (verdadeiro) ou **false** (falso)
- São geralmente o resultado de comparações feitas no programa
 - Exemplo: **a==4**
- São tipicamente usados nas **estruturas de controle** do fluxo dos programas

Depuração de código

- Console JavaScript
 - Google Chrome:
 - Ferramentas do Desenvolvedor → Console
 - Atalho: Tecla F12 ou Ctrl+Shift+J no Windows
 - Firefox:
 - Menu de configurações → Desenvolvimento Web → Ferramentas de Desenvolvimento
 - Atalho: Tecla F12

Erros comuns de JavaScript: alguns exemplos de mensagens no Chrome

- **Falta de pontuação:**

- `alert('hello';`

- Mensagem: “Unexpected token ;”
 - Neste caso, o navegador encontrou o ponto e vírgula em vez do parêntese de fechamento.

- **Falta ou erro em aspas:**

- `alert ('hello ") ;`

- Mensagem: “Uncaught SyntaxError: Unexpected token ILLEGAL”

Erros comuns de JavaScript: alguns exemplos de mensagens no Chrome

- **Comandos incorretos:**

- `aler ('hello');`
 - Mensagem: “Uncaught ReferenceError: aler is not defined”

- **Erro de sintaxe:**

- Ocasionalmente, o Chrome não tem ideia do que você estava tentando fazer e fornece esta mensagem de erro genérico: “Syntax error”.
- Pode não ser um erro de digitação, mas você pode ter colocado em conjunto um ou mais comandos em uma maneira que não é permitido.
- Neste caso, você precisa olhar atentamente para a linha onde o erro foi encontrado e tentar descobrir o que está errado.

Material para Estudo

- **Site w3schools:**

- <https://www.w3schools.com/js/default.asp>

- **Para reforçar o conteúdo visto nessa aula:**

- https://www.w3schools.com/js/js_intro.asp

- https://www.w3schools.com/js/js_where.asp

- https://www.w3schools.com/js/js_variables.asp

- https://www.w3schools.com/js/js_datatypes.asp

- **Livros:**

- FLANAGAN, David. JavaScript: O Guia Definitivo. 6ª ed., Porto Alegre: Bookman, 2013. 1062 p.

- SILVA, Maurício Samy. JavaScript: guia do programador. São Paulo: Novatec, 2010. 604 p.