

Javascript

O básico

Alisson Chiquitto²

²Curso de Informática

Cianorte, 2014

Outline

- 1 Introdução
- 2 Importar o Javascript no HTML
- 3 Variáveis e Tipos de variáveis
- 4 Operadores
- 5 Estruturas de controle
- 6 Funções
- 7 HTML DOM
 - Manipulando o DOM
 - Eventos

Outline

- 1 Introdução
- 2 Importar o Javascript no HTML
- 3 Variáveis e Tipos de variáveis
- 4 Operadores
- 5 Estruturas de controle
- 6 Funções
- 7 HTML DOM
 - Manipulando o DOM
 - Eventos

- JavaScript é uma linguagem de programação interpretada;
- Javascript é uma linguagem que adiciona interatividade à páginas *HTML*
- Foi originalmente implementada como parte dos navegadores web para que scripts pudessem ser executados do lado do cliente e interagissem com o usuário sem a necessidade deste script passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido.

Comentários no CSS

Exemplo

Comentário de uma linha

```
// Meucomentario
```

Comentário de várias linhas

```
/* Meucomentario */
```

Hello World em Javascript

```
<script>  
// mostra uma mensagem  
window.alert('Olá mundo');  
</script>
```

Figura: Nosso primeiro código Javascript

- JavaScript é uma linguagem de programação interpretada;
- Javascript é uma linguagem que adiciona interatividade à páginas *HTML*
- Foi originalmente implementada como parte dos navegadores web para que scripts pudessem ser executados do lado do cliente e interagissem com o usuário sem a necessidade deste script passar pelo servidor, controlando o navegador, realizando comunicação assíncrona e alterando o conteúdo do documento exibido.

Outline

- 1 Introdução
- 2 Importar o Javascript no HTML
- 3 Variáveis e Tipos de variáveis
- 4 Operadores
- 5 Estruturas de controle
- 6 Funções
- 7 HTML DOM
 - Manipulando o DOM
 - Eventos

Importar o arquivo CSS no HTML I

```
<head>  
  <title>Importando JS</title>  
  <meta charset="utf-8">  
  
  <script src="arquivo.js"></script>  
  
</head>
```

Figura: Incluindo um arquivo Javascript ao HTML

Outline

- 1 Introdução
- 2 Importar o Javascript no HTML
- 3 Variáveis e Tipos de variáveis**
- 4 Operadores
- 5 Estruturas de controle
- 6 Funções
- 7 HTML DOM
 - Manipulando o DOM
 - Eventos

- Identificam uma variável ou função;
- Começam com uma letra, underscore (_) ou \$;
- Seguido por zero ou mais letras, dígitos, underscores (_) ou \$;

Tipagem do Javascript

O *Javascript* possui tipagem dinâmica e fraca.

Tipagem dinâmica

“A tipagem dinâmica ocorre quando a linguagem não obriga a prévia declaração de tipo de uma variável. O tipo é assumido na atribuição de valor à variável, que pode ser por presunção ou forçado com casting. Além disso, é possível modificar o tipo da variável atribuindo-lhe outro valor.”

Tipagem fraca

“A tipagem fraca ocorre quando a linguagem permite que uma variável tenha seu valor automaticamente alterado para outro tipo para possibilitar uma operação.”

Tipagem do Javascript

Exemplos

```
<script>

  var x;
  x = 'Teste';
  x = 5;

</script>
```

Figura: Exemplo de Tipagem Dinâmica

```
<script>

  var x,y;
  x = 'Nome';
  y = x + 5;

  alert(x);

  // Nome5
  alert(y);

</script>
```

Figura: Exemplo de Tipagem Fraca

Tipos de variáveis

- Number;
- String;
- Boolean;
- Object;
- **function**
- **null**
- **undefined**

Tipo: Number

- O tipo *Number* é utilizado para números inteiros e números de ponto flutuante;
- **NaN** (*not a number*): é um número especial. Ele é resultante de operações indevidas com números;

```
<script>  
  x = 5;  
  x = 3.1415;  
</script>
```

Figura: Variáveis do tipo Number

Tipo: Number

A função Number

- Converte diferentes tipos de variáveis para *Number*
- Produz o valor *NaN* em caso de falhas;
- The Number Function

Tipo: String

- *Strings* são cadeias de caracteres;

```
<script>  
  x = 'Teste';  
  x = "Teste";  
</script>
```

Figura: Variáveis do tipo String

Tipo: String

A função String

- Converte diferentes tipos de variáveis para *String*;
- The String Function

Tipo: Boolean

- **true;**
- **false;**

```
<script>  
  x = true;  
  x = false;  
</script>
```

Figura: Variáveis do tipo Boolean

Tipo: Boolean

A função Boolean

- Converte diferentes tipos de variáveis para *Boolean*;

```
<script>  
    // true  
    x = Boolean(1);  
  
    // false  
    x = Boolean(0);  
</script>
```

Figura: Conversão para Boolean

- Variáveis podem ser esvaziadas com o uso do valor **null**;

```
<script>  
    x = null;  
</script>
```

Figura: Variável do tipo *null*

- **undefined** é o valor das variáveis sem valor, ou de variáveis **inexistentes**;

Outline

- 1 Introdução
- 2 Importar o Javascript no HTML
- 3 Variáveis e Tipos de variáveis
- 4 Operadores**
- 5 Estruturas de controle
- 6 Funções
- 7 HTML DOM
 - Manipulando o DOM
 - Eventos

Operadores aritméticos

+

Soma ou Concatenação (Se ambos os operandos forem *Numbers*, então faz a soma);

—

Subtração;

*

Multiplicação;

/

Divisão;

%

Módulo (resto de uma divisão);

```
<script>
```

```
x = 5 + 6;  
x = 5 - 6;  
x = 5 * 6;  
x = 5 / 6;  
x = 5 % 6;
```

```
</script>
```

Figura: Operadores Aritméticos

Operadores aritméticos

Operadores de incremento e decremento

++

Incremento

--

Decremento

```
<script>
```

```
x = 5;
```

```
++x;
```

```
x++;
```

```
--x;
```

```
x--;
```

```
</script>
```

Figura: Pré/Pós Incremento/Decremento

Operadores de comparação

==

Igualdade

!=

Desigualdade

>

Maior

<

Menor

>=

Maior ou igual

<=

Menor ou igual

```
<script>
```

```
y = (1 == 2);
```

```
y = (1 != 2);
```

```
y = (1 > 2);
```

```
y = (1 < 2);
```

```
y = (1 >= 2);
```

```
y = (1 <= 2);
```

```
</script>
```

Figura: Operadores de comparação

Operadores lógicos

`&&`

E (AND)

`||`

Ou (OR)

`!`

Negação (NOT)

```
<script>
```

```
y = (true && false);  
y = (true || false);  
y = !true;
```

```
</script>
```

Figura: Operadores lógicos

Operador ternário

O operador ternário é uma alternativa para a instrução *if/else*.

```
<script>  
    y = (condicao) ? true : false;  
</script>
```

Figura: Operador ternário

Outline

- 1 Introdução
- 2 Importar o Javascript no HTML
- 3 Variáveis e Tipos de variáveis
- 4 Operadores
- 5 Estruturas de controle**
- 6 Funções
- 7 HTML DOM
 - Manipulando o DOM
 - Eventos

Estruturas de controle

- *if...else;*
- *break;*
- *switch;*
- *while;*
- *do;*
- *for;*
- *continue;*
- *return;*

Estrutura if...else

- Use a estrutura *if* para executar algum código somente se a condição for avaliada como *true*;
- Use a estrutura *if...else* para executar algum código se a condição é *true*, ou o outro código quando a condição é *false*;

Estrutura break

- A estrutura *break* “pula para fora” da estrutura que ele esta localizado;

```
<script>

    while(condicao) {
        break;
    }

    switch(expressao) {
        case 1:
            break;
        case 2:
            break;
    }

</script>
```

Figura: Estrutura break

Estrutura switch

- A estrutura *switch* é usada para realizar diferentes ações baseadas em diferentes condições;
- Use a palavra-chave *default* para especificar uma ação padrão;

```
<script>

    switch(expressao) {
        case 1:
            break;
        case 2:
            break;
        default:
            // padrao
    }

</script>
```

Figura: O switch

Estrutura while

- Executa um bloco de código, desde que a condição especificada seja verdadeira;

```
<script>  
    while(condicao) {  
    }  
</script>
```

Figura: Laço While

Estrutura do...while

- Variante do laço *while*;
- Esta estrutura executa o bloco uma vez, depois verifica se a condição é verdadeira. Enquanto a condição for *true*, o bloco continuará a ser executado;

```
<script>  
    do {  
        } while (condicao);  
</script>
```

Figura: Laço Do...While

Estrutura for

- Um laço *for* é composto por 3 expressões;
- **Expressão 1** será executada apenas uma vez ao inicio do laço;
- **Expressão 2** é a condição para a execução do laço. Enquanto ela continuar *true*, a próxima iteração é executada;
- **Expressão 3** é executado sempre ao final de cada iteração;

```
<script>  
    for(exp1; exp2; exp3) {  
    }  
</script>
```

Figura: Laço For

- A estrutura *continue* interrompe uma iteração, e inicia a próxima iteração do laço;

Estrutura return

- A estrutura *return* interrompe a execução de uma função, e retorna o valor especificado;

```
<script>  
  
    function soma(x,y){  
        return x + y;  
    }  
  
</script>
```

Figura: Estrutura return

Outline

- 1 Introdução
- 2 Importar o Javascript no HTML
- 3 Variáveis e Tipos de variáveis
- 4 Operadores
- 5 Estruturas de controle
- 6 Funções**
- 7 HTML DOM
 - Manipulando o DOM
 - Eventos

- Uma função é um bloco de código que será executado quando alguém chamar ele;
- No *Javascript*, uma função é um bloco de código que fica armazenado em uma variável (Variável do tipo função);
- As funções são compostas, por 4 elementos não obrigatórios: nome, parâmetros, instruções e valor de retorno;

```
<script>  
    function soma(x,y) {}  
    var soma = function(x,y) {}  
</script>
```

Figura: Funções

Outline

- 1 Introdução
- 2 Importar o Javascript no HTML
- 3 Variáveis e Tipos de variáveis
- 4 Operadores
- 5 Estruturas de controle
- 6 Funções
- 7 HTML DOM**
 - Manipulando o DOM
 - Eventos

- Quando uma *web page*, o navegador cria uma **Document Object Model** da página;
- O modelo *HTML DOM* é construído como uma árvore de objetos;
- O *DOM HTML* é um padrão para como selecionar, alterar, adicionar ou deletar elementos *HTML*;

HTML DOM

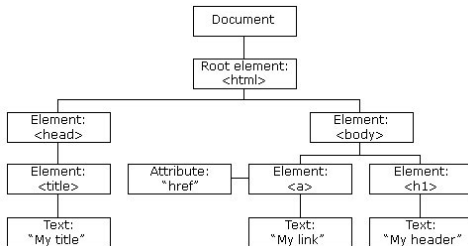
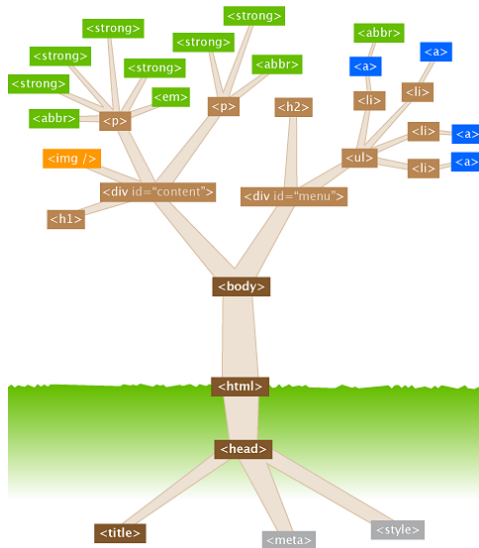


Figura: HTML DOM

HTML DOM



Com o *HTML DOM*, o *Javascript* pode:

- Alterar todos os elementos do *HTML*;
- Alterar todos os atributos dos elementos do *HTML*;
- Alterar todos os estilos *CSS* da página;
- O *JavaScript* pode remover elementos e atributos *HTML*;
- O *JavaScript* pode adicionar elementos e atributos *HTML*;
- O *JavaScript* pode reagir a todos os eventos *HTML* da página;
- O *JavaScript* pode criar novos eventos *HTML* na página;

Outline

- 1 Introdução
- 2 Importar o Javascript no HTML
- 3 Variáveis e Tipos de variáveis
- 4 Operadores
- 5 Estruturas de controle
- 6 Funções
- 7 HTML DOM**
 - Manipulando o DOM
 - Eventos

Selecionando elementos

- A maneira mais fácil para acessar elementos *HTML* é através do *id* do elemento;
- O método `document.getElementById(id)` seleciona elementos no *DOM* através do atributo *id*;

```
var i = document.getElementById('input');
```

Figura: Selecionando elementos via atributo id

Manipulando atributos HTML

- Após selecionar o elemento é possível manipular os seus atributos;

```
window.onload = function() {  
    var i = document.getElementById('input');  
    i.value = 'Teste Javascript';  
}
```

Figura: Manipulando atributos dos elementos HTML

Alterando o conteúdo de um elemento

- A maneira mais fácil de capturar ou alterar o conteúdo de um elemento é através do seu atributo *innerHTML*;
- Usando o atributo *innerHTML* é possível alterar o conteúdo de um elemento;

```
window.onload = function() {  
    var i = document.getElementById('input');  
    i.value = 'Teste Javascript';  
}
```

Figura: Manipulando atributos dos elementos HTML

Outline

- 1 Introdução
- 2 Importar o Javascript no HTML
- 3 Variáveis e Tipos de variáveis
- 4 Operadores
- 5 Estruturas de controle
- 6 Funções
- 7 HTML DOM**
 - Manipulando o DOM
 - **Eventos**

- O *HTML DOM* permite o *JavaScript* reagir a eventos;
- O *JavaScript* pode ser executado quando um evento ocorre, como por exemplo um *click* do *mouse* do usuário;

Atribuição de eventos pelo HTML

```
<script>
function validarForm() {}
function validarClick() {}
</script>

<form id="f" onsubmit="validarForm()">
<button id="b" onclick="validarClick()"></button>
</form>
```

Figura: Atribuindo função para manipular evento através do HTML

Atribuição de eventos pelo DOM

```
<script>
function validarForm() {}
function validarClick() {}

window.onload = function() {
    var f = document.getElementById('f');
    f.onsubmit = validarForm;

    var b = document.getElementById('b');
    b.onclick = validarClick;
}
</script>

<form id="f">
<button id="b"></button>
</form>
```

Figura: Atribuindo função para manipular evento através do DOM



Fifth Edition of ECMA-262.

ecmascript the language of the web.

<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-262.pdf>.



JavaScript Tutorial.

W3Schools.

<http://www.w3schools.com/js/default.asp>.



What is the Document Object Model?.

Jonathan Robie, Texcel Research - W3C.

<http://www.w3.org/TR/1998/WD-DOM-19980720/introduction.html>.



JavaScript HTML DOM.

W3Schools.

http://www.w3schools.com/js/js_htmldom.asp.



JavaScript HTML DOM Events.

W3Schools.

http://www.w3schools.com/js/js_htmldom_events.asp.



HTML DOM Events.

W3Schools.

http://www.w3schools.com/jsref/dom_obj_event.asp.