



Desenvolvimento Web 1 – Aulas 22 / 23 / 24

IFPE – Campus Igarassu
2016.1

Ranieri Valença
03/10/2016
04/10/2016
10/10/2016



Tópicos de JavaScript

- O que é JavaScript? O que ela faz?
- Onde colocamos o JavaScript?
 - Tag <script>
 - Arquivo externo
 - Atributo
- JavaScript Básico
 - Escrevendo no documento HTML
 - Variáveis
 - alert(), prompt() e confirm()
 - Tipos de dados
 - JSON



Tópicos de JavaScript

- JavaScript Básico
 - Operadores e Comparadores
 - Estruturas de controle
 - Math
 - Date
 - Funções
 - setTimeout() e setInterval()
 - Eventos
- DOM
 - Document Object Model
 - Manipulando elementos do DOM



O que é JavaScript? O que ela faz?

É a **linguagem de programação** da
Web.

Com JavaScript podemos alterar o **conteúdo** do
HTML, alterar **atributos de elementos**
HTML e até alterar **estilos (CSS)** de
elementos HTML

Onde colocamos o JavaScript?

```
<html>
  <head>
    <title>JavaScript Tests</title>
    <script>
      document.write("Hello world");
    </script>
  </head>
  <body>
    <p>Lorem ipsum</p>
  </body>
</html>
```

Onde colocamos o JavaScript?

```
<html>
  <head>
    <title>JavaScript Tests</title>
  </head>
  <body>
    <p>Lorem ipsum</p>
    <script>
      document.write("Hello world");
    </script>
  </body>
</html>
```

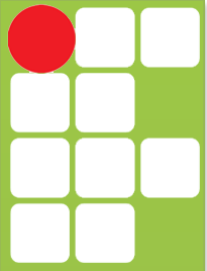
Onde colocamos o JavaScript?

```
<html>
  <head>
    <title>JavaScript Tests</title>
    <script src="external01.js"></script>
  </head>
  <body>
    <p>Lorem ipsum</p>
    <script src="external01.js"></script>
  </body>
</html>
```



Onde colocamos o JavaScript?

```
<html>
  <head>
    <title>JavaScript Tests</title>
  </head>
  <body>
    <p onclick="alert('Hello world');">Lorem ipsum</p>
  </body>
</html>
```

JavaScript básico – Escrevendo no documento

```
document.write("conteudo");
```




JavaScript básico – Variáveis

Declaramos variáveis em JavaScript usando a palavra reservada “**var**”.

JavaScript é mais uma linguagem C-like, ou seja, sua sintaxe é parecida com C. Assim, **variáveis** (e também funções, como veremos mais à frente) **devem começar com letra ou underline (_), e podem conter apenas letras, números ou underlines.**

JavaScript também é **case sensitive** (diferencia letras maiúsculas e minúsculas)



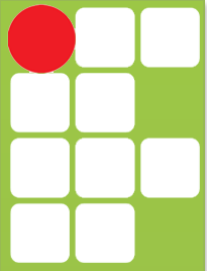
JavaScript básico – alert(), prompt() e confirm()

Comandos chatos ;)

alert() → lança um alerta na tela.

prompt() → abre um popup para que o usuário digite algo; a função retorna o que for digitado.

confirm() → abre um popup com os botões “ok” e “cancelar”.



JavaScript básico – Tipos de dados

Números (number)

```
var number = 3;
```

Strings

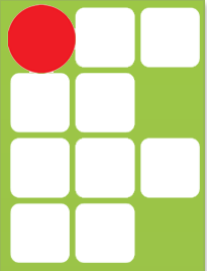
```
var str = "teste";
```

Boolean

```
var bool = false;
```

Arrays

```
var arr = [1, 2, "teste", false];
```



JavaScript básico – Tipos de datos

`typeof(x);`



JavaScript básico – JSON

JavaScript **O**bject **N**otation

Notação de Objetos de JavaScript



JavaScript básico – JSON

```
var obj = {  
    name: "Tony Stark",  
    alias: "Iron man",  
    affiliations: [  
        "SHIELD",  
        "Avengers"  
    ],  
    age: 30  
}
```



JavaScript básico – JSON

```
var data = {  
  "funcionarios": [  
    {"nome": "John", sobrenome: "Doe"},  
    {"nome": "Anna", sobrenome: "Smith"},  
    {"nome": "Peter", sobrenome: "Jones"},  
  ]  
}
```




JavaScript básico – Operadores

+

- soma
- concatenação

-

- subtração

*

- multiplicação

/

- divisão

%


- resto (módulo)

--

- decremento

++

- incremento



JavaScript básico – Comparadores

==

- igualdade

===

- igualdade (com verificação de tipo)

!=

- desigualdade

!==

- desigualdade (com verificação de tipo)

>

- maior que

<

- menor que

>=

- maior ou igual

<=

- menor ou igual



JavaScript básico – Estruturas de controle

```
var x = prompt("Qual número?");  
if (x > 10) {  
    alert("número grande");  
}
```



JavaScript básico – Estruturas de controle

```
var x = prompt("Qual número?");  
while (x > 10) {  
    document.write("número: " + x);  
    x--;  
}
```



JavaScript básico – Estruturas de controle

```
var x = prompt("Qual número?");  
for (i = 0; i < x; i++) {  
    document.write(x + "<br>");  
}
```



JavaScript básico – Estruturas de controle

```
var arr = ["elise", "leesin", "trundle"]
for (i in arr) {
    document.write("indice: " + i);
    document.write("<br>");
    document.write("player: " + arr[i]);
    document.write("<br>");
}
```



JavaScript básico – Math

Biblioteca matemática

`Math.min(x, y, ..);`

`Math.max(x, y, ..);`

`Math.sin(x); Math.cos(x); Math.tan(x);`

`Math.log(x); Math.log10(x); Math.log2(x);`

`Math.sqrt(x); Math.pow(x, y);`

`Math.ceil(x); Math.floor(x); Math.round(x);`

`Math.random()`

`Math.PI`

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



JavaScript básico – Exercício (1)

Jogo da adivinhação

1. Sorteie um número aleatório entre 0 e 100
2. Peça ao usuário que digite um número, e diga se o número digitado pelo usuário é maior ou menor que o sorteado
3. Continue pedindo um número até que o usuário acerte
4. Mostre quantas jogadas o usuário precisou para acertar



JavaScript básico – Date

```
var d = new Date();
```

```
var d = new Date(milliseconds);
```

```
var d = new Date(dateString);
```

```
var d = new Date(year, month, day,  
    hours, minutes, seconds,  
    milliseconds);
```



JavaScript básico – Date

```
Date.now();
```

```
var d = new Date();  
d.getFullYear();  
d.getMonth();  
d.getDay();  
d.getHours();  
d.getMinutes();  
d.getSeconds();  
d.getMilliseconds();
```

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



JavaScript básico – Funções

```
function fn(arg1, arg2, arg3) {  
    for (i = 0; i < arg3; i++) {  
        document.write(arg1);  
        document.write("<br>");  
        document.write(arg2);  
        document.write("<br><br>");  
    }  
}
```

```
fn("Hello", "World", 3);
```



JavaScript básico – Funções

```
var fn = function(arg1, arg2, arg3) {  
    for (i = 0; i < arg3; i++) {  
        document.write(arg1);  
        document.write("<br>");  
        document.write(arg2);  
        document.write("<br><br>");  
    }  
}
```

```
fn("Hello", "World", 3);
```



JavaScript básico – Funções

```
(fn = function(arg1, arg2, arg3) {  
    for (i = 0; i < arg3; i++) {  
        document.write(arg1);  
        document.write("<br>");  
        document.write(arg2);  
        document.write("<br><br>");  
    }
```

```
})( "Hello", "World", 3);
```

```
fn("teste", 2, 1);
```



JavaScript básico – setTimeout

```
function foo() {  
    document.write(new Date());  
}
```

```
setTimeout(foo, 1000);
```



JavaScript básico – setInterval

```
function foo() {  
    document.write(new Date());  
}
```

```
setInterval(function() {  
    foo();  
}, 1000);
```



JavaScript básico – setInterval

```
setInterval(function() {  
    document.write(new Date());  
}, 1000);
```




JavaScript básico – setInterval

```
(fn = function(arg1, arg2) {  
    for (i = 0; i < arg2; i++) {  
        document.write(arg1);  
        document.write("<br>");  
    }  
})("Hello" + "World", 3);
```

```
setInterval(function() {  
    fn("setInterval", 1)  
}, 1000);
```



JavaScript básico – Exercício (2)

Relógio

Faça um relógio que atualiza a hora em tempo real, alterando o conteúdo de um elemento ``.



JavaScript básico – Eventos

Eventos são ações que acontecem no HTML.

Clique, duplo clique, passar o mouse por cima de um elemento são exemplos de eventos.

Ações de JavaScript podem ser disparadas por esses eventos.



JavaScript básico – Eventos

Para utilizar uma função de JavaScript vinculado a um evento em um elemento HTML, utilizamos o atributo "onnomedoevento" (tudo minúsculo mesmo):

```
<body onkeydown="document.write('wat?');">  
  <p onclick="fn('hello', 'world', 3);">  
    Lorem ipsum  
  </p>  
  ...
```

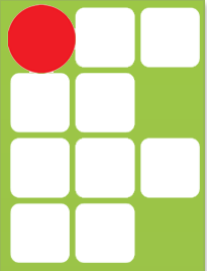
http://www.w3schools.com/jsref/dom_obj_event.asp



JavaScript básico – Exercício (3)

Cronômetro

Crie um cronômetro que pode ser iniciado, pausado e resetado utilizando JavaScript e HTML.



JavaScript básico – Erros comuns

[http://www.w3schools.com/js/js_mistakes.a
sp](http://www.w3schools.com/js/js_mistakes.asp)



DOM – Document Object Model

No modelo de objeto-documento, tudo do
HTML é um **Nó**:

- O documento HTML por si;
- Todos os elementos HTML;
- Os atributos dos elementos HTML;
- Texto dentro dos elementos HTML.

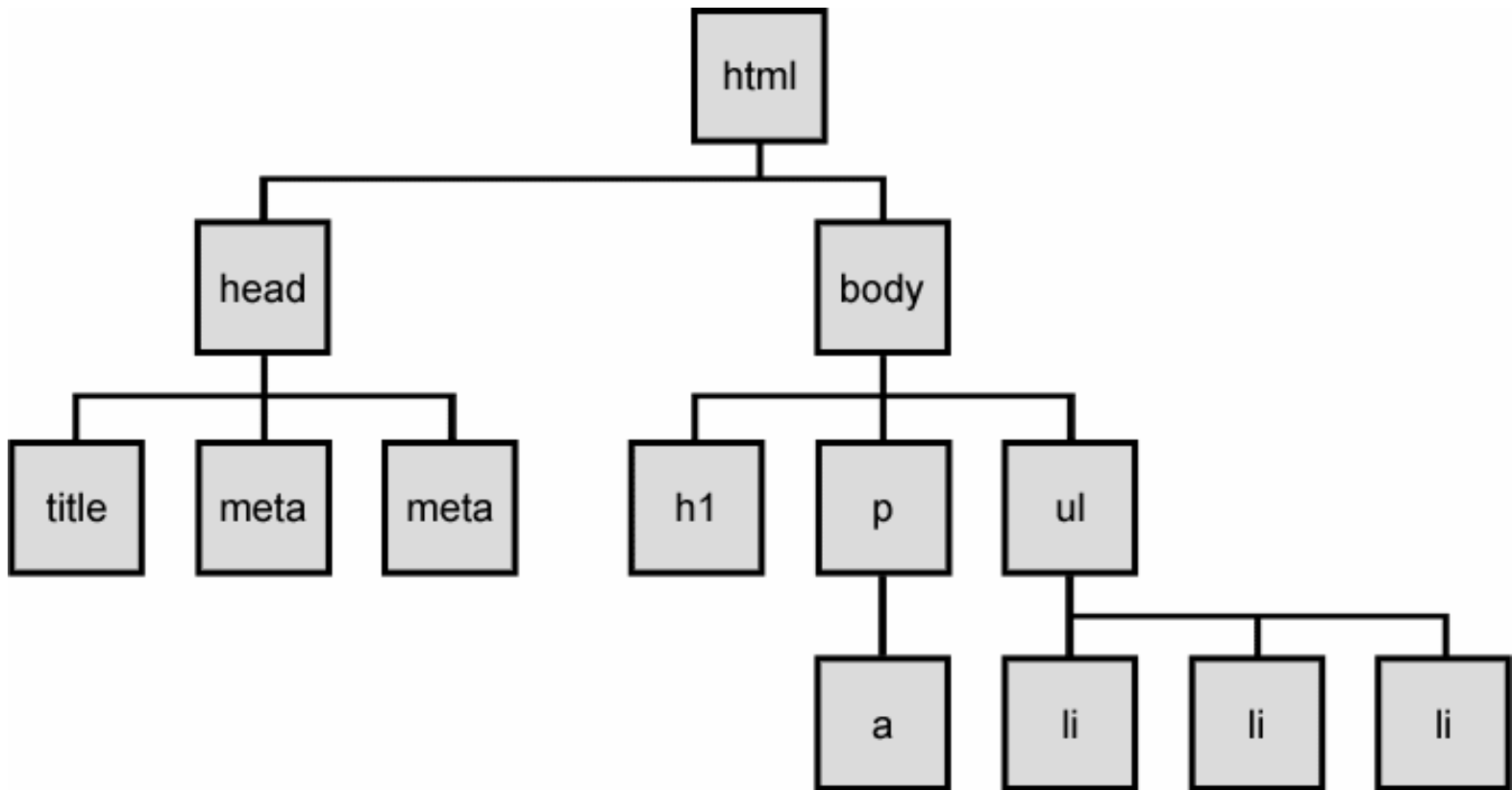


DOM – Document Object Model

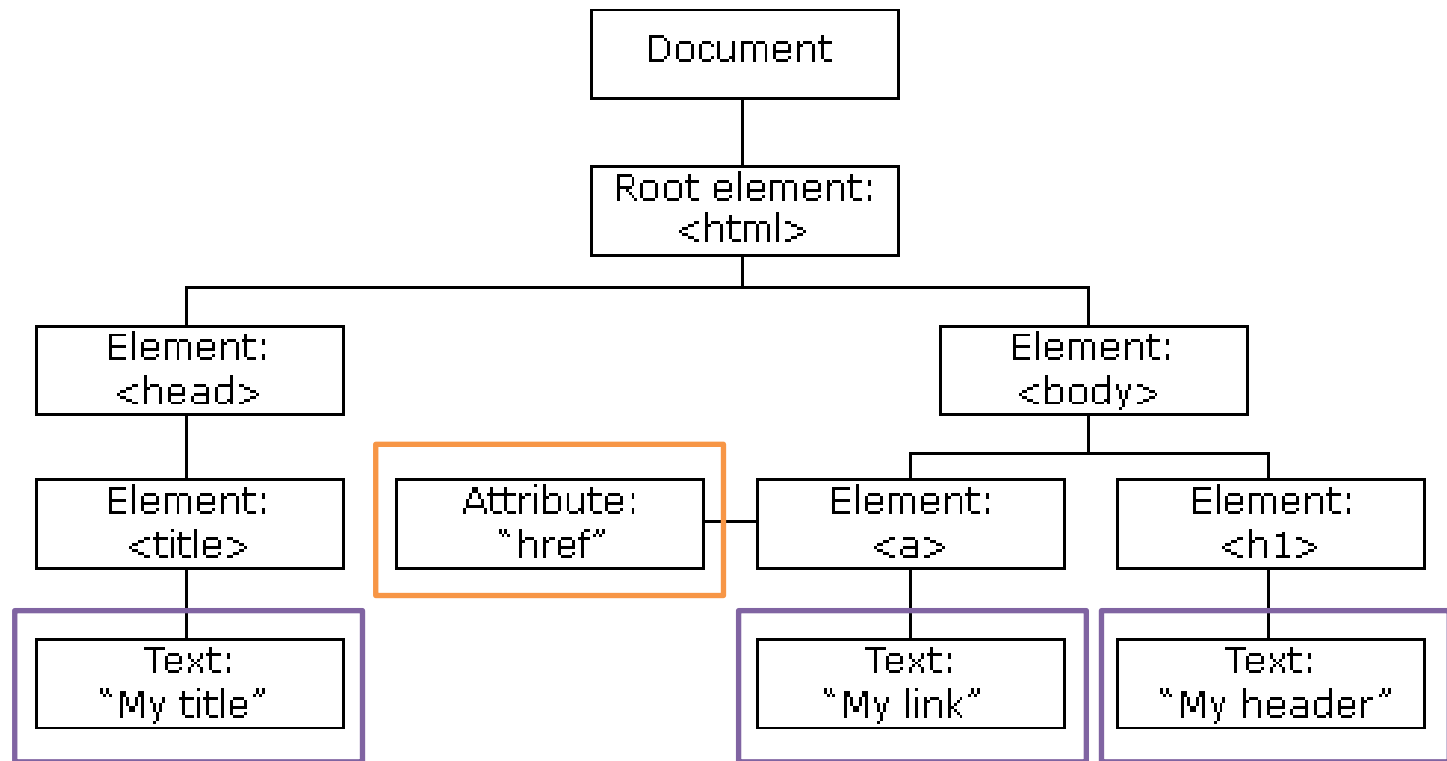
Cada um desses **nós** é tratado como um objeto de JavaScript, que tem seus próprios atributos e **métodos**.

Usando JavaScript, podemos acessar esses atributos e métodos, e assim alterar a estrutura DOM de um documento, assim como fizemos quando utilizamos `document.write()` anteriormente.

DOM – Document Object Model



DOM – Document Object Model





DOM – Manipulando o DOM

Para podermos manipular um Nó, "*precisamos*" antes tê-lo em uma variável.

Para acessar um desses Nós, utilizamos os métodos *get* do objeto *document*:

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

```
document.getElementsByClassName( ' ' );
```

```
document.getElementsByTagName( ' ' );
```

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



DOM – Manipulando o DOM

Uma vez que temos um Nó, podemos chamar métodos e alterar seus atributos:

```
var p = document.getElementById('myP');  
p.innerHTML = Date();  
p.style.background = '#97F0A3';  
var im = document.getElementById('myImg');  
im.setAttribute('src', 'path/to/img.png');
```

*http://www.w3schools.com/jsref/dom_obj_all.asp
http://www.w3schools.com/jsref/dom_obj_style.asp*

