

Guilherme Cavalcanti  
[contato@guilhermecavalcanti.com](mailto:contato@guilhermecavalcanti.com)  
[@guiocavalcanti](https://twitter.com/guiocavalcanti)

## **Pacote Web**

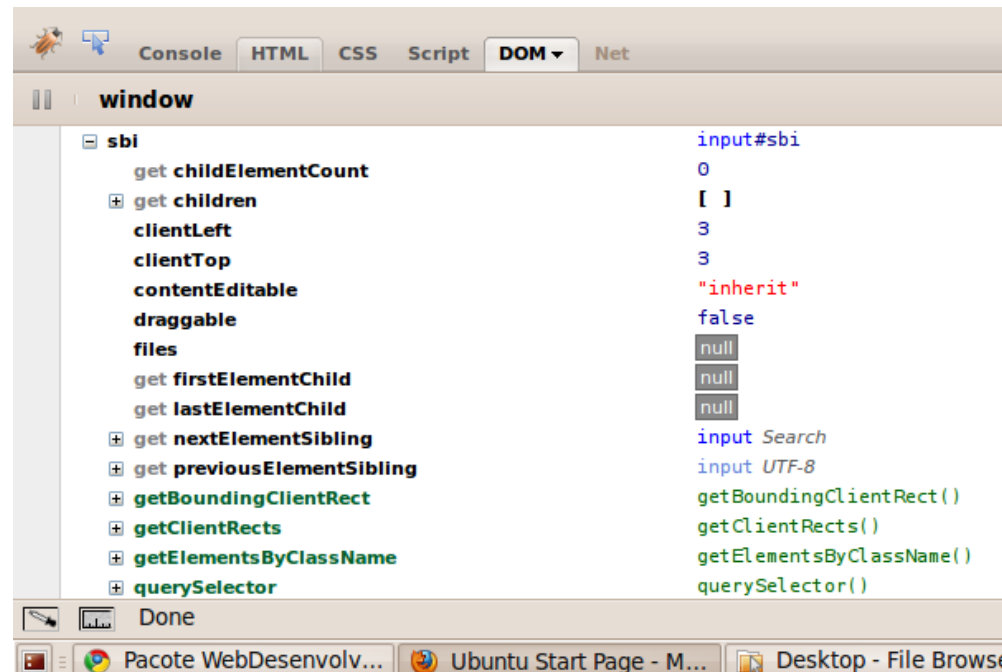
### Desenvolvendo com Padrões Web

#### Módulo 3 - Intro. JavaScript

# O que é?

- DOM

- *Document Object Model*
- É um padrão de interface para que outras linguagens possam acessar e interagir com o HTML.



# O que é?

- Uma linguagem de script usada para interagir com o DOM
- É interpretada
  - Sem compilação
- Não tem nada a ver com Java
- Fracamente tipada

# Inclusão

- Interno

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2.     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
3. <html xmlns="http://www.w3.org/1999/xhtml">
4. <head>
5.     <title>Hello JavaScript</title>
6.     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
7.
8.     <script type="text/javascript">
9.         alert("Hello World");
10.    </script>
11. </head>
12. <body>
13.
14. </body>
15. </html>
```

<http://pastebin.com/GxJY9X7f>

# Inclusão

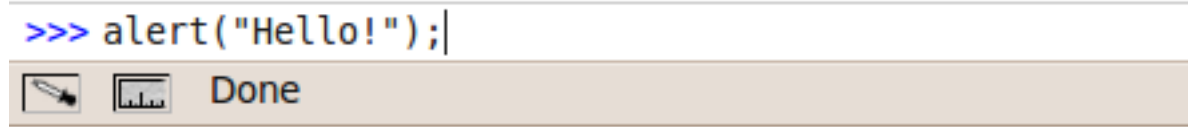
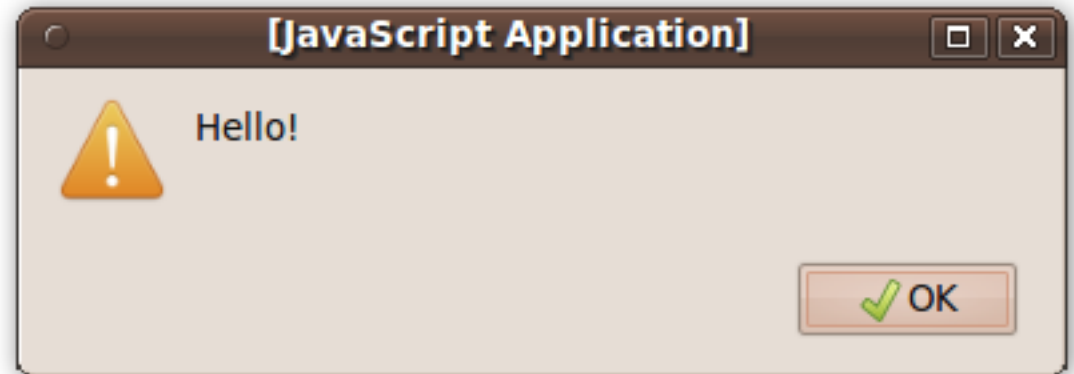
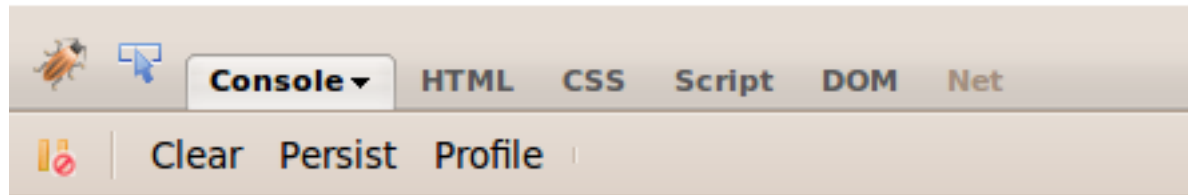
- Externo

```
8 <script type="text/javascript" src="hello-externo.js">  
9 </script>
```

<http://pastie.org/847219>

```
1 alert("Hello World");
```

# Shell interativo do firebug



# Variáveis

- Usadas para armazenar valores
- São *case-sensitive*
  - Assim como o resto da linguagem
- Precisam começar com letra ou \_
- Declaradas utilizando a keyword *var*



```
Limpar  Pesistente  Perfil  Tudo  Erro
>>> var a;
undefined
>>> a === undefined
true
>>> a = "Nome"
"Nome"
>>> a === undefined
false
```

# Operadores

- Operações básicas

- +
- -
- \*
- /

```
>>> 1 + 1
```

```
2
```

```
>>> 2 - 2
```

```
0
```

```
>>> 2 * 3
```

```
6
```

```
>>> 6 / 2
```

```
3
```

- Operadores lógicos

- e &&
- ou ||

```
>>> true && true
```

```
true
```

```
>>> true && false
```

```
false
```

```
>>> true || true
```

```
true
```

```
>>> false || false
```

```
false
```

```
>>> true || false
```

```
true
```

```
>>> true or true
```



# Operadores

- Concatenação

```
>>> x = "Pacote"  
"Pacote"  
>>> y = "Web"  
"Web"  
>>> x + y  
"PacoteWeb"
```

- Incremento/decremento

```
>>> x = 1  
1  
>>> x++  
1  
>>> x++  
2  
>>> x++  
3  
>>> x  
4
```

```
>>> x = 10  
10  
>>> x--  
10  
>>> x--  
9  
>>> x--  
8
```

# Operadores de Comparação

- ==
- !=
- >
- >=
- <=
- <

```
>>> 2 > 2  
false  
>>> 2 >= 2  
true  
>>> 2 != 2  
false  
>>> "Web" == "Web"  
true
```

```
>>> "Pacote" == "Web"  
false  
>>> "Pacote" > "Web"  
false
```

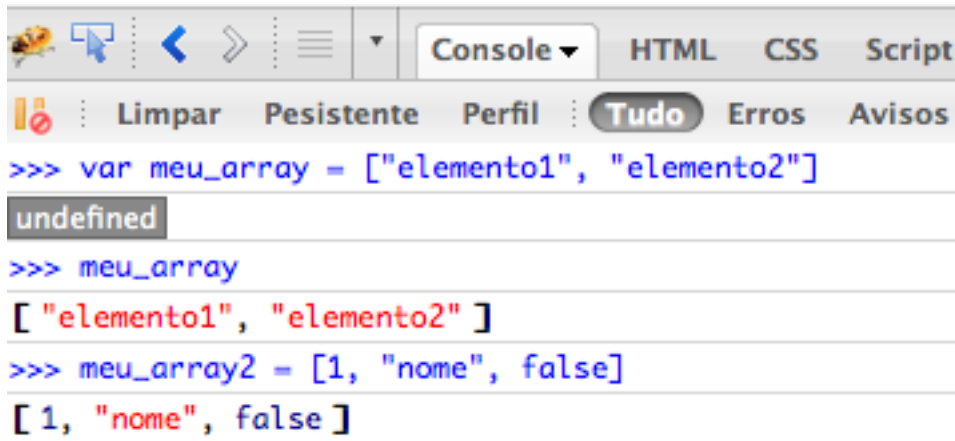
# Tipos

- Number
- String
- Boolean
- undefined

```
>>> typeof 1
"number"
>>> typeof true
"boolean"
>>> typeof "Pacote Web"
"string"
>>> typeof x
"undefined"
```

# Literais

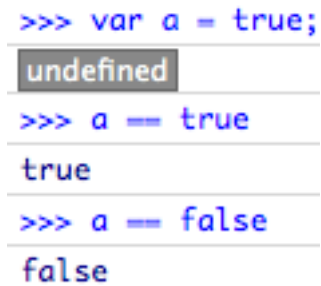
- Arrays



A screenshot of a web browser's developer console. The console has tabs for 'Console', 'HTML', 'CSS', and 'Script'. The 'Console' tab is active, showing a series of commands and their outputs. The commands are: `>>> var meu_array = ["elemento1", "elemento2"]`, `>>> meu_array`, `>>> meu_array2 = [1, "nome", false]`, and `>>> meu_array2`. The outputs are: `undefined`, `["elemento1", "elemento2"]`, `[1, "nome", false]`, and `[1, "nome", false]`.

```
>>> var meu_array = ["elemento1", "elemento2"]
undefined
>>> meu_array
[ "elemento1", "elemento2" ]
>>> meu_array2 = [1, "nome", false]
[ 1, "nome", false ]
```

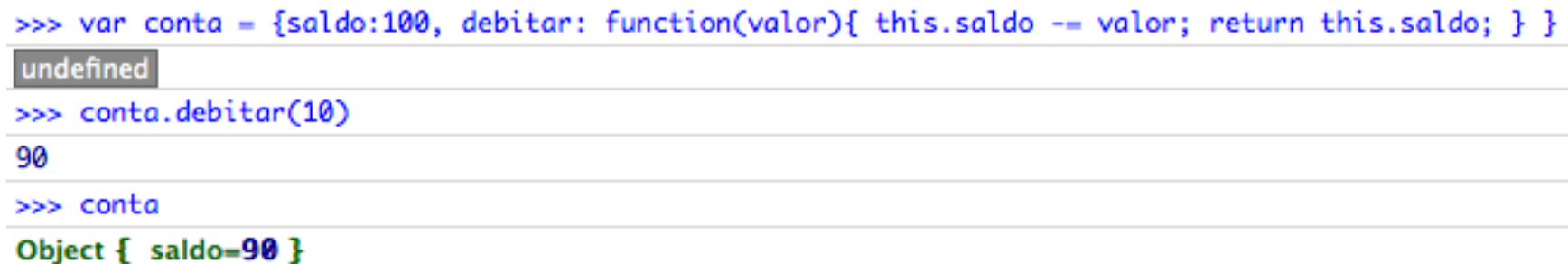
- Booleanos



A screenshot of a web browser's developer console showing boolean operations. The commands are: `>>> var a = true;`, `>>> a == true`, and `>>> a == false`. The outputs are: `undefined`, `true`, and `false`.

```
>>> var a = true;
undefined
>>> a == true
true
>>> a == false
false
```

- Objetos



A screenshot of a web browser's developer console showing object operations. The commands are: `>>> var conta = {saldo:100, debitar: function(valor){ this.saldo -= valor; return this.saldo; } }`, `>>> conta.debitar(10)`, and `>>> conta`. The outputs are: `undefined`, `90`, and `Object { saldo=90 }`.

```
>>> var conta = {saldo:100, debitar: function(valor){ this.saldo -= valor; return this.saldo; } }
undefined
>>> conta.debitar(10)
90
>>> conta
Object { saldo=90 }
```

# if

- Sequência de instruções

```
1 // Leitura de cima-baix e esquerda-direita
2 x = 10;
3 y = 20;
4 z = x + y;
5
6 document.write("O resultado é " + z);
```

<http://pastie.org/847386>

- Podemos alterar essa característica

```
1 x = 10;
2 y = 20;
3 z = x + y;
4
5 // Quebra do fluxo
6 if(z > 10){
7     document.write(x + " + " + y + " é maior do que 10")
8 }else{
9     document.write(x + " + " + y + " é menor do que 10")
10 }
11
```

<http://pastie.org/847388>

# while

```
1 document.write("<table border=\"1\">")
2
3 x = 20;
4 while(x > 0){
5     document.write("<tr><td>" + x + "</td></tr>")
6     x--
7 }
8
9 document.write("</table>")
```

<http://pastie.org/847403>

# for

```
1 for (i=0;i<=5;i++){  
2   document.write("The number is " + i);  
3   document.write("<br />");  
4 }
```

# Funções

- Modularizar melhor o código

```
6 function displaymessage(){  
7   alert("Hello World!");  
8 }
```

- Retorno

```
10 function product(a,b){  
11   return a*b;  
12 }
```



# Escopo das variáveis

- Se declaradas dentro dentro da função são locais
- Se declaradas fora das funções são acessíveis pelo documento inteiro



# Interagindo com o DOM

```
/* Maneiras de acessar os nós do DOM */  
document.getElementsByTagName("p")  
document.getElementById("ingredientes")
```

<http://pastie.textmate.org/1061917>