

Desenvolvimento Web
client-side

3 Intro JavaScript
Guilherme Cavalcanti
@guiocavalcanti

CONTEXTO: SEMÂNTICA

Comportamento

JavaScript

Formatação

CSS

Informação

HTML

AS SEEN ON...



[http://shop.oreilly.com/product/9780596517748.do?
sortby=bestSellers](http://shop.oreilly.com/product/9780596517748.do?sortby=bestSellers)

.appendTo **DEVELOPER
LEARNING CENTER**

<http://learn.appendto.com>

DOM

- Document Object Model
 - Padrão de interface para que outras linguagens possam interagir com o HTML



JAVASCRIPT

- Linguagem de script usada para interagir com o DOM
- Interpretada
- Não tem nada a ver com Java
- Fracamente tipificada

INCLUINDO JAVASCRIPT

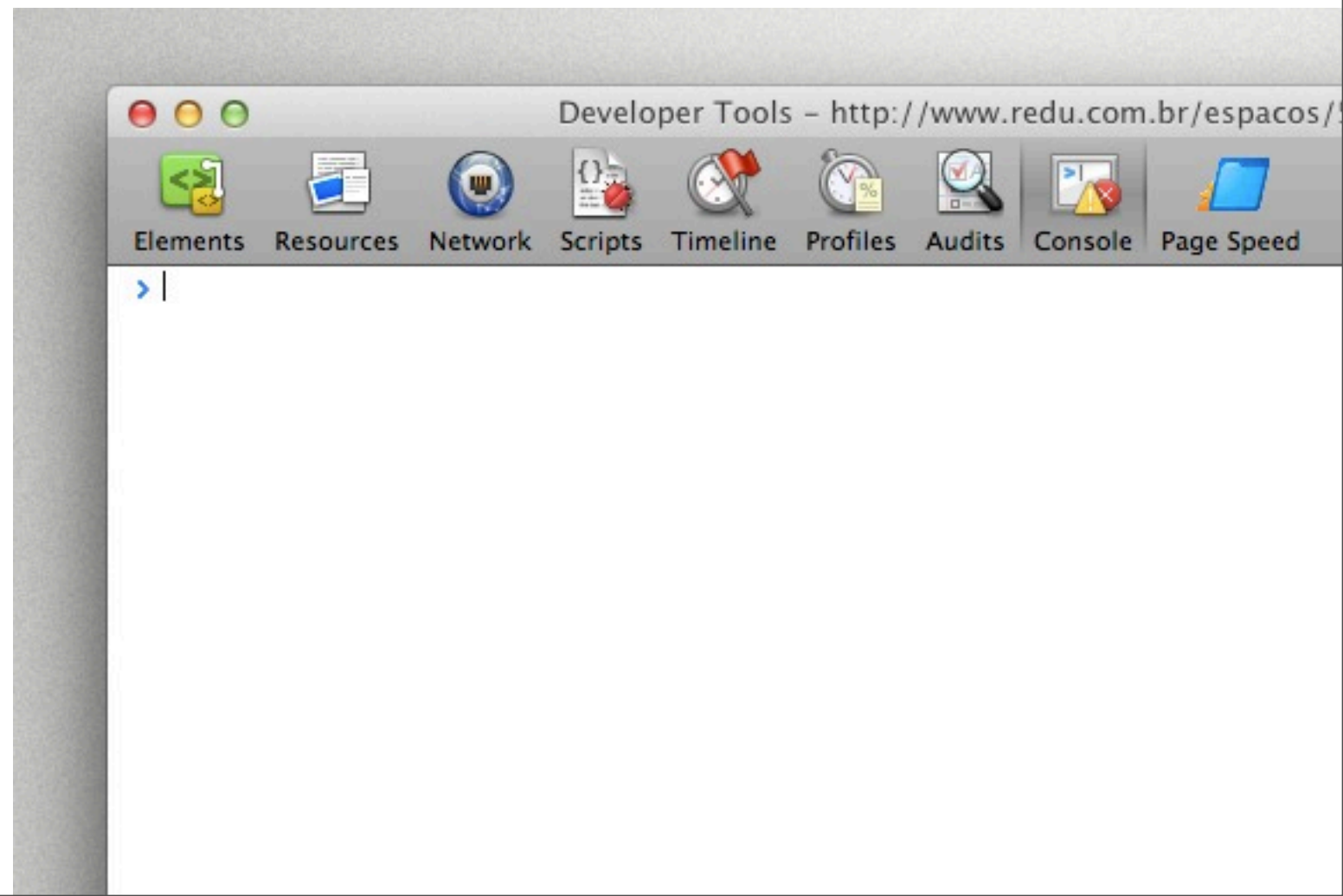
```
<script type="text/javascript" src="arquivo.js">  
</script>
```

CONSOLE INTERATIVO

- Para quem conhece python e ruby é um conceito comum
- Como se fosse uma sessão de shell no contexto do interpretador

CONSOLE INTERATIVO

- Builtin no Chrome
- No firefox através do FireBug



VARIÁVEIS

- Fracamente tipificada
- Declaradas com a keyword **var**
- Casesensitive
- Segue o padrão *camelCase*

```
var myName = "guilherme";  
console.log(myName);
```

```
myName = 12;  
console.log(myName);
```

[http://jsfiddle.net/guiocavalcanti/
aYgnB/](http://jsfiddle.net/guiocavalcanti/aYgnB/)

NUMBER

```
var someNumber = 1;  
var otherNumber = 2;
```

```
console.log(someNumber + otherNumber);  
console.log(someNumber * otherNumber);  
console.log(someNumber / otherNumber)
```

<http://jsfiddle.net/guiocavalcanti/8eVWwq/>

NUMBER

- Não há float ou double, apenas Number

```
var myNum = 1.2;  
var myOtherNum = 1;
```

NUMBER

- Transformando Strings em Numbers

```
var myNumberAsString = "1.232";  
console.log(parseFloat(myNumberAsString));  
  
console.log(parseInt(myNumberAsString));
```

NOT A NUMBER

- Literal chamado NaN
- Tem-se como resultado quando não é possível realizar alguma operação com Number
- Dá para testar através da função `isNaN()`;

```
var myName = "guiocavalcanti";  
var result = 12 / myName;  
  
console.log(result);  
console.log(isNaN(result));
```

STRINGS

- Coerção automática em alguns casos
- Strings duplas ou simples
- Aspas podem ser escapadas

```
var coercao = "11" * 2;  
console.log(coercao);
```

```
console.log("abc" == 'abc')
```

```
console.log("abc d\"a")
```

COMPARAÇÕES

- Cuidado! Há coerção de tipos

```
console.log("1" == 1);  
// true  
console.log("9" == 9.0);  
// true  
console.log(0 == false);  
// true
```

COMPARAÇÕES ESTRITAS

- Não há coerção de valores

```
console.log("1" === 1);  
// false  
console.log("9" === 9.0);  
// false  
console.log(0 === false);  
// false
```


IF, ELSE IF

- Com uma linha chaves podem ser omitidos

```
var condition = true;
```

```
if(condition) {  
    // ...  
} else if(condition) {  
    // ...  
}
```

O QUE É AVALIADO COMO FALSO?

- 0, -0
- String vazia
- NaN
- undefined
- null

```
var nan = NaN, zero = 0,  
    undef = undefined;
```

```
if (nan) {  
    console.log("Hay");  
} else {  
    console.log("Ho");  
}
```

EXERCÍCIO

// q1 : Dê console.log() nos valores que são avaliados para false

```
var q1Var1 = "hello",  
    q1Var2 = 0,  
    q1Var3 = true,  
    q1Var4 = "false",  
    q1Var5 = -1,  
    q1Var6 = undefined,  
    q1Var7 = null,  
    q1Var8;
```

// BEGIN Question 1 Answer

// END Question 1 Answer

// q2: Concatene as strings abaixo e imprima no console:

```
var q3Var1 = "hello, ",  
    q3Var2 = "is it me you're looking for?";
```

WHILE

```
var index = 0;  
  
while(index < 100) {  
    console.log(index);  
    index = index + 1;  
}
```

FOR

```
var arr = [1,2,3,4,5];  
  
for(var i = 0; i < arr.length; i++){  
    console.log(arr[i]);  
}
```

FUNCTIONS

```
var myFunction = function(){  
    console.log("hey");  
};
```

```
myFunction();
```

```
var sayHey = function(str){  
    console.log("hey " + str);  
};
```

```
sayHey("ho!");
```

CRIANDO OBJETOS

```
var MyObject = function(arg1, arg2) {  
    this.arg1 = arg1;  
    this.arg2 = arg2;  
};
```

```
MyObject.prototype.args = function(){  
    return this.arg1 + this.arg2;  
}
```

```
var o1 = new MyObject("a", "b");  
var o2 = new MyObject("a2", "b2");
```

```
console.log(o1.args());  
console.log(o2.args());
```

ARRAY

```
var arr = ["abc", "def", "gh"]  
  
arr.pop(); // "gh"  
arr.push("ij"); // 4  
arr.push("ij", "zw"); // 6  
arr.splice(1,2) // ["abc", "def"]
```

[https://developer.mozilla.org/en/JavaScript/Reference/
Global_Objects/Array#Methods](https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/Array#Methods)

PROJETO: CONJUNTO

- Crie um objeto que representa um conjunto de string. Este objeto deve ter as seguintes operações:
 - Adicionar uma string ex: `set.add("minha string"); => set`
 - Checar a existência de uma string ex: `set.exist("string"); => false`
 - Remover uma string: ex: `set.remove("string"); => set`
 - União de dois conjuntos: ex: `set.union(set2); => set`
 - Limpar conjunto ex: `set.clear(); => set`
 - Transformar conjunto em string: ex: `set.add("1"); set.add("2"); set.toString(); => "{1,2}"`

PROJETO: FIZZBUZZ

- Se o número for divisível por 3, imprima "Fizz".
- Se o número for divisível por 5, imprima "Buzz".
- Se o número for divisível por 5 e 3 imprima "FizzBuzz".
- Nos demais casos imprima apenas o número

<https://github.com/guiocavalcanti/pacote-web-2013-1/blob/master/js/projetos/fizzbuzz.zip?raw=true>