

# Javascript

Patrícia Guimarães e Rodrigo Chaves

# O que é Javascript

JavaScript is a cross-platform, object-oriented scripting language. It is a small and lightweight language.

# DOM

O Modelo de Objeto de Documento (do inglês Document Object Model - DOM) é uma convenção multiplataforma e independente de linguagem para representação e interação com objetos em documentos HTML, XHTML e XML. Os nós de cada documento são organizados em uma estrutura de árvore, chamada de árvore DOM.

Usamos JS para manipular o DOM!

# Tipos

## Simples:

- ▶ Null
- ▶ Undefined
- ▶ Numbers
- ▶ Booleans
- ▶ String

## Complexos:

- ▶ Arrays
- ▶ Objects
- ▶ Functions

# Print

```
console.log("pudding");
```

# Variáveis

```
var pudim = 42
```

# Estrutura Condicional

if

```
if (condition) {  
    statement_1;  
} else {  
    statement_2;  
}
```

- ▶ false
- ▶ undefined
- ▶ null
- ▶ 0
- ▶ NaN
- ▶ the empty string ("")

# Estruturas de Repetição

for

```
var step;
```

```
for (step = 0; step < 5; step++) {
```

```
    // Runs 5 times, with values of step 0 through 4.
```

```
    console.log('Walking east one step');
```

```
}
```



# Estruturas de Repetição

for

```
var n = 0;  
var x = 0;  
while (n < 3) {  
    n++;  
    x += n;  
}
```

# Objetos

Objetos são hashes que podem receber variáveis (e funções) para cada chaves.

```
// bad  
const item = new Object();
```

```
// good  
const item = {};
```

# Funções

A function definition (also called a function declaration, or function statement) consists of the function keyword, followed by:

- ▶ The name of the function.
- ▶ A list of parameters to the function, enclosed in parentheses and separated by commas.
- ▶ The JavaScript statements that define the function, enclosed in curly brackets, { }.

# Funções

```
function square(number) {  
  return number * number;  
}
```

# Funções Anônimas

Sim! Funções sem nome!!

```
var square = function(number) { return number * number; };  
var x = square(4); // x gets the value 16
```

# Closures

A closure is an inner function that has access to the outer (enclosing) function's variables—scope chain. The closure has three scope chains: it has access to its own scope (variables defined between its curly brackets), it has access to the outer function's variables, and it has access to the global variables.

# Closures

A função `makeFullName` consegue chamar a variável `nameIntro`.

```
function showName( firstName, lastName ) {  
  var nameIntro = "Your name is ";  
  function makeFullName() {  
    return nameIntro + firstName + " " + lastName;  
  }  
  return makeFullName();  
}  
  
showName("Rodrigo", "Chaves");
```

# Callbacks

Callback é uma função que é usada como “callback”. Ela é tipicamente passada como argumento de outra função e/ou chamada quando um evento for acontecido, ou quando uma parte de código receber uma resposta de que estava à espera.



# Callbacks

```
function callback(e) {  
    alert('Aconteceu um evento ' + e.type);  
}  
window.addEventListener('click', callback);
```

# Selecionando Elementos

Selecionando a partir da Classe:

```
var y = document.getElementsByClassName('foo');
```

# Seleccionando Elementos

Seleccionando a partir do ID:

```
var y = document.getElementById('foo');
```

# Referências

- ▶ Mozilla Developer Guide
- ▶ Seleção