



Universidade Federal do Rio Grande do Norte - UFRN

Instituto Metr pole Digital – IMD
JavaScript: Operadores L gicos,
Estrutura Condicional, Estruturas de
Repeti  o

Nelson Ion de Oliveira



Agenda

- Operadores L gicos
- Estrutura condicional
- Estrutura de repeti  o
- Tipos de vari veis

Operadores lógicos

Igualdade	==	a == b
	===	a === b
Diferença	!=	a != b
	!==	a !== b
Maior do que	>	a > b
Maior ou igual	>=	a >= b
Menor do que	<	x < y
Menor ou igual	<=	x <= y
E (AND)	&&	x && y
OU (OR)		x y

Operadores Lógicos

- Os operadores **===** e **!==** avaliam os operandos e então os compara.
- Essa operação é realizada sem conversão de tipo.

```

< 1 === 1
> true
< 1 === '1'
> false
< 1 !== 1
> false
< 1 !== '1'
> true

```

Operadores Lógicos

- Na operação com os operadores `==` e `!=` a operação é menos estrita.
- Se os valores dos operandos não forem do mesmo tipo, algumas conversões de tipo serão tentadas e realiza-se uma nova comparação.

```
◀ 1 == 1  
▶ true  
◀ 1 == '1'  
▶ true  
◀ 1 != '1'  
▶ false  
◀ 1 != 1  
▶ false
```

Estrutura Condicional IF

```
if (condição) {  
    ação para condição verdadeira  
}else {  
    ação para condição falsa  
}
```

Estrutura de repetição While

```
while (condição) {  
    ação que será executada enquanto condição  
    for verdadeira  
}
```

Estrutura de repetição For

```
for (i = 1; i <= 10; i++) {  
    //ação que será repetida  
    window.alert("Oi " + i);  
}
```

Tipos de dados e variáveis

- Como dito anteriormente, uma variável em JavaScript é criada pela simples associação de algum valor a ela.
 - Seu tipo é definido **dinamicamente**.
- A palavra-chave **var** também pode ser usada para criar uma variável.
 - **var** nome;
 - nome = “Qualquer nome”;

Tipos de dados e variáveis

- Se uma variável é inicializada (atribui-se um valor) sem ser declarada com a palavra chave var, ela é adicionada ao contexto global da página automaticamente, ou seja, torna-se uma variável global.
- Dica: procure sempre declarar suas variáveis usando o **var**, pois facilitará sua leitura do código.

Tipos primitivos

- Em JavaScript existem os seguintes tipos primitivos:
 - Number
 - String
 - Boolean
 - Undefined
 - Null

Tipos primitivos

- Number – inclui números inteiros bem como de ponto flutuante

```
◀ x=1
▶ 1
◀ typeof x
▶ "number"
◀ x=9.9
▶ 9.9
◀ typeof x
▶ "number"
```

Bônus: typeof
Retorna o tipo do
dato

Tipos primitivos

- String – sequência de caracteres entre aspas (simples ' ' ou dupla " ")

```
◀ x = "oi"
▶ "oi"
◀ typeof x
▶ "string"
◀ x = 'IMD'
▶ "IMD"
◀ typeof x
▶ "string"
```

Tipos primitivos

- Boolean – assume apenas dois valores: *true* ou *false*

```
◀ x = true
▶ true
◀ typeof x
▶ "boolean"
◀ x = false
▶ false
◀ typeof x
▶ "boolean"
```

Tipos primitivos

- Undefined – quando tentamos acessar uma variável que não existe, recebemos esse valor 'especial':

```

< typeof undefined
> "undefined"
< typeof q
> "undefined"
< x
> false
< x.a
> undefined
< typeof x.a
> "undefined"

```

Tipos primitivos

- Null – outro tipo especial. Que significa vazio ou ausência de valor. Variáveis com esse valor foram definidas explicitamente.

```

> v = null
< null
> typeof v
< "object"
> typeof null
< "object"
> 1 + v
< 1
> 1*v
< 0

```

Chromium

```

< v = null
> null
< typeof v
> "object"
< typeof null
> "object"
< 1 + v
> 1
< 1*v
> 0

```

Firefox

Referências

- Textos e Exemplos retirados dos sites:
 - www.w3schools.com/js/
 - www.codecademy.com

Dúvidas?



Exercício

No SIGAA há uma lista de exercícios.