Javascript

SOFTEX PERNAMBUCO

Softex

MINISTÉRIO DA EDUCAÇÃO
E INOVAÇÃO

GOVERNO FEDERAL
BRASIL

Ainda hoje...

- **Continuação Javascript:**
  - Operadores:
    i. aritméticos unários: **++**
  - Strings:
    i. operadores
    ii. **interpolação** vs **concatenação**
  - Introdução a Funções
  - Exercícios

OPERADORES ARITMÉTICOS UNÁRIOS

```
var incremento = 5;
console.log(incremento++); // 5
console.log(incremento); // 6

var incremento2 = 5;
console.log(++incremento2); // 6
console.log(incremento2); // 6
```

> *Mesma coisa para o decremento* `--x`

OPERADORES ARITMÉTICOS (STRINGS)

```
var soma = '100' + 50; // 10050
var subtracao = '100' - 50; // 50
var multiplicacao = '100' * '2'; // 200
var divisao = 'Comprei 10' / 2; // NaN (Not a Number)
```

> *É possível verificar se uma variável é NaN ou não com a função* `isNaN()`

Interpolação de Strings (Template String)

A inclusão das template literals na ECMAScript 6 (ES6) nos permite interpolar strings em JavaScript

```
const idade = 4.5;
const idadeTerrestre = `Estima-se que a Terra tenha ${idade} bilhões de anos.`;

console.log(idadeTerrestre);
```

Como fazíamos isso antes?

Antes das *template literals*, teríamos algo assim:

```
const idadeTerrestre = "Estima-se que a Terra tenha " + idade + " bilhões de anos."
```

FUNÇÕES

Bloco de código que pode ser executado e reutilizado. Valores podem ser passados por uma função e a mesma retorna outro valor.

```
function pi() {
  return 3.14;
}

var total = 5 * pi(); // 15.7
```

> *Parênteses* `()` *executam uma função*

FUNÇÕES

Bloco de código que pode ser executado e reutilizado. Valores podem ser passados por uma função e a mesma retorna outro valor.

```
function areaQuadrado(lado) {
  return lado * lado;
}

areaQuadrado(4) // 16
areaQuadrado(5) // 25
areaQuadrado(2) // 4
```

> *Chamada de function declaration*

PARÂMETROS E ARGUMENTOS

Ao `criar` uma função, você pode definir `parâmetros`.

Ao `executar` uma função, você pode passar `argumentos`.

```
// peso e altura são os parâmetros
function imc(peso, altura) {
  const imc = peso / (altura ** 2);
  return imc;
}

imc(80, 1.80) // 80 e 1.80 são os argumentos
imc(60, 1.70) // 60 e 1.70 são os argumentos
```

> *Separar por vírgula cada parâmetro. Você pode definir mais de um parâmetro ou nenhum também*

PARÊNTESES EXECUTA A FUNÇÃO

```
function corFavorita(cor) {
  if(cor === 'azul') {
    return 'Você gosta do céu';
  } else if(cor === 'verde') {
    return 'Você gosta de mato';
  } else {
    return 'Você não gosta de nada';
  }
}
corFavorita(); // retorna 'Você não gosta de nada'
```

> *Se apenas definirmos a função com o function e não executarmos a mesma, nada que estiver dentro dela irá acontecer*

PODE OU NÃO RETORNAR UM VALOR

Quando não definimos o return, ela irá retornar `undefined`. O código interno da função é executado normalmente, independente de existir valor de return ou não.

```
function imc(peso, altura) {
  const imc = peso / (altura ** 2);
  console.log(imc);
}

imc(80, 1.80); // retorna o imc
console.log(imc(80, 1.80)); // retorna o imc e undefined
```

VALORES RETORNADOS

Uma função pode retornar qualquer tipo de dado e até outras funções.

```
function terceiraIdade(idade) {
  if(typeof idade !== 'number') {
    return 'Informe a sua idade!';
  } else if(idade >= 60) {
    return true;
  } else {
    return false;
  }
}
```

> *Cuidado, retornar diferentes tipos de dados na mesma função não é uma boa ideia.*

Exercícios

Javascript - Lista 1

SOFTEX
PERNAMBUCO

Softex

MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO

GOVERNO FEDERAL
BRASIL
UNIÃO E RECONSTRUÇÃO