

Digital College Fortaleza Ceará

Prof. Carlos Marques

Resumo Fernando Chaves

Aula 21

- Objeto

ex:

```
let pessoa = {  
  nome: "Carlos",  
  idade: 31,  
  experiencias: [  
    {empresa: "A", descrição: "-----"},  
    {-----, -----, -----}  
  ],  
}
```

Forma de acessar um elemento dentro do objeto

- 1) let n = pessoa.nome
- 2) let n = pessoa["nome"]

Acessar elementos dentro do objeto que está dentro de um array

```
pessoa.experiencias[0].empresa
```

Para imprimir os elementos:

```
for (let i = 0; pessoa.experiencias.length; i++) {  
  console.log(pessoa.experiencias[i])  
}
```

- Declaração e chamada de função

ex:

1) function soma (a, b) {

let c = 10 // variável local

return a + b

}

let x = 6

soma (x, 10) sendo x igual a 6, 6 passa a ser o parâmetro "a" retornando 16 como resultado

Também é possível colocar uma function dentro de uma variável:

let y = soma

y (2, 10) = 12 sendo a = 2 e b = 10

2) function soma (a, b) {

let resultado = a + b

return resultado

}

Em uma calculadora ao clicar no botão de igual teria a ativação de uma função OBS: op == operação (soma, subtração, multiplicação, divisão, ...)

function executarOperacao (op, a, b) {

return op (a, b)

}

executarOperacao(soma, 1, 2) // Chamada da function executarOperacao

op = função soma, a = 1 e b = 2, assim: return == 3

Se tivesse uma function multiplicação ao chamar executarOperacao teria

op = função multiplicação, a = 1, b = 2, assim: return == 2

OBS: A ordem dos parâmetros importa

3) Uma função pode receber uma função com parâmetros diretamente

executarOperacao(function(a, b) {

```
        return a - b
    }, 3, 10)
```

seria como ter feito assim:

```
function subtracao(a, b) {
    return a - b
}
```

executarOperacao(subtracao, 3, 10)

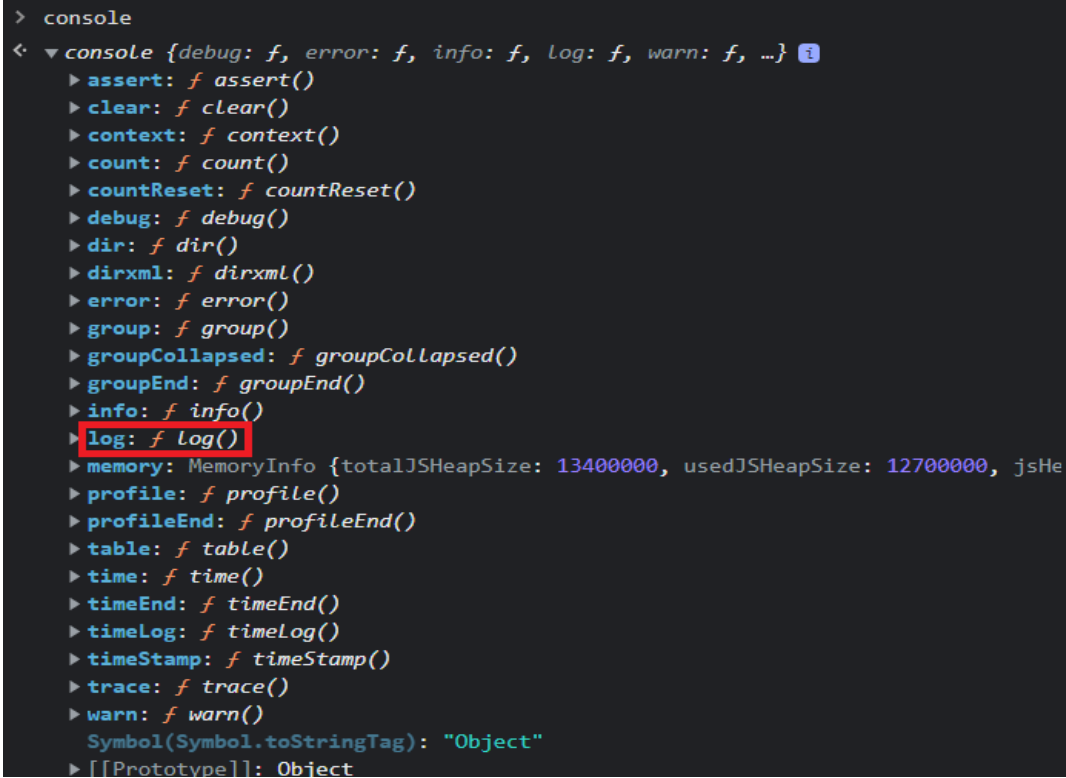
- Interação do Javascript com o navegador

console.log

console == objeto

log == chave e dentro dessa chave possui uma função

É possível ver dentro do console no DevTools



```
> console
< ▼ console {debug: f, error: f, info: f, log: f, warn: f, ...} ⓘ
  ▶ assert: f assert()
  ▶ clear: f clear()
  ▶ context: f context()
  ▶ count: f count()
  ▶ countReset: f countReset()
  ▶ debug: f debug()
  ▶ dir: f dir()
  ▶ dirxml: f dirxml()
  ▶ error: f error()
  ▶ group: f group()
  ▶ groupCollapsed: f groupCollapsed()
  ▶ groupEnd: f groupEnd()
  ▶ info: f info()
  ▶ log: f log()
  ▶ memory: MemoryInfo {totalJSHeapSize: 13400000, usedJSHeapSize: 12700000, jsHe
  ▶ profile: f profile()
  ▶ profileEnd: f profileEnd()
  ▶ table: f table()
  ▶ time: f time()
  ▶ timeEnd: f timeEnd()
  ▶ timeLog: f timeLog()
  ▶ timeStamp: f timeStamp()
  ▶ trace: f trace()
  ▶ warn: f warn()
  Symbol(Symbol.toStringTag): "Object"
  ▶ [[Prototype]]: Object
```

