Operadores lógicos

Em javascript há quatro operadores lógicos:

|| → ou

&& → e

! → não

→ se algum de seus argumentos for true, ele retorna true, caso contrário ele retorna false:

true || true → V

false || true → V

true || false → V

false || false → F

número 0 ou false,
qualquer outro número: true

→ retorna true se ambos os operandos forem verdadeiros e false caso contrário:

true && true → true

false && true → false

true && false → false

false && false → false

! (NOT)

↳ converte o operando para o tipo boolean (true/false)

↳ retorna o valor inverso

!true → false

!false → true

operador de coalescência nulo '??'

coalescência nulish é escrito com dois pontos de interrogação ??

a ?? b

- se a está definido, então a
- se a não estiver definido, então b

?? retorna o primeiro argumento se não for null / undefined. Caso o contrário, o segundo.

loops: While e for

- uma maneira de repetir o mesmo código várias vezes.

while "enquanto" a condição for verdadeira, o loop será executado.

```
let i = 0
while (i < 3) {
    alert(i)  → saída: 0, 1, 2
    i++
}
```

do while "faça... enquanto", primeiro o loop é executado e depois a condição é verificada

```
let i = 0

do {
    alert(i)
    i++
} while (i < 3)
```

for (início , condição, incremento)

```
for ( let i = 0; i < 3; i++ ) {
    alert(i)
}
```

a declaração "trocar"

uma instrução switch pode substituir vários if

Ele fornece uma maneira mais descritiva de comparar um valor com vários variantes.

```
let a = 18
let b = 3

switch (a > b) {
  case true:
    console.log('é maior')

  case false:
    console.log('é menor')
}
```

funções

São blocos de código. Permite que o código seja chamado várias vezes sem repetição.

```
function showmessage() {
    alert('Hello World!')
}
```

ao clicar no botão a função será chamada e a função executada

```
<button onclick = "showmessage()"> clique! </button>
```