



Operadores E e OU



Temos alguns operadores interessantes e muito utilizados que são: && e ||, achou estranho? Vamos entender mais nessa aula.

&& →

Esse operador lógico é correspondente ao 'E' do nosso português. Imaginamos duas variáveis, a numero1 e numero2, ambas são diferentes e ao colocarmos no console.log() nos é retornado true ou false, por exemplo:

```
let numero1 = 10;
let numero2 = 13;

// false + true = false;
// true + false = false;
// true + true = true;
// false + false = false;

console.log(numero1 == numero2 && 13 == numero2);
// false, pois numero1 não é igual a numero2 e 5 não é igual a 5

console.log(numero1 == 10 && numero2 == 13);
// true, pois numero1 é igual a 10 e numero2 é igual a 13
```

Para retornar false um dos lados tem que ter resultado como false, para true ambos lados precisam ser verdadeiros, como:

```
let validacao1 = true && true; // Retorna true
let validacao2 = true && false; // Retorna false
let validacao3 = false && true; // Retorna false
let validacao4 = false && (6 == 12); // Retorna false
let validacao5 = "Faculdade" && "Beginners PRO";
// Retorna Beginners PRO
let validacao6 = "Beginners PRO" && "Faculdade";
// Retorna Faculdade
let validacao7 = false && "Faculdade"; // Retorna false
let validacao8 = "Faculdade" && false; // Retorna false
```

|| →

Esse operador lógico é correspondente ao 'OU' do nosso português. Agora, ao contrário do operador anterior, esse retornará true caso uma das condições seja verdadeira e false caso ambas sejam falsas. Imaginamos duas variáveis, a numero1 e numero2,

que ao compararmos nos traz:

```
let numero1 = 10;
let numero2 = 13;

// true + true = true;
// true + false = true;
// false + true = true;
// false + false = false;

console.log(numero1 == 10 || numero2 == 13);
// Retorna true, pois ambas condições são verdadeira
console.log(numero1 == 15 || numero2 == 13);
// Retorna true, pois a segunda condição é verdadeira
console.log(numero1 == 10 || numero2 == 1);
// Retorna true, pois a primeira condição é verdadeira
console.log(numero1 == 1 || numero2 == 1);
// Retorna false, pois ambas são falsas
```

! →

Esse operador nega determinada lógica. Devemos usar no começo para inverter seu valor booleano. Por exemplo:

```
let validacao1 = true;
let validacao2 = false;
let validacao3 = "Faculdade";

console.log(!validacao1);
// Retorna: false
console.log(!validacao2);
// Retorna: true
console.log(!validacao3);
// Retorna: false
```

Bibliografias:

- https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Expressions_and_operators;