Operadores de Igualdade e Comparação

> Em vários momentos enquanto estamos programando e até atuando no dia a dia, precisamos fazer algumas comparações, se algo é igual ou diferente à outra coisa e assim por diante, no JavaScript temos alguns comparadores específicos para cada situação.

"Iguais"

No Javascript possuímos algumas possibilidades de atribuições e comparações com números em geral. Temos 3 possíveis casos utilizando o '=':

- = : Operador de Atribuição.
- == : Operador de Comparação de Igualdade;
- === : Operador de Comparação de Igualdade Estrita;

= → Operador de Atribuição

Um igual é responsável por atribuir valor à alguma variável. Seja ela string, boolean, number e etc... sempre usamos somente um '=' para definirmos seus valores, por exemplo:

```
// Number:
let idade = 30;

// String:
let nome = 'Clóvis';

// String:
let profissao = 'Engenheiro';

// Boolean:
let empregado = true;
```

== → Operador de Comparação de Igualdade

Dois iguais são responsáveis por comparar valores em si. Devemos tomar cuidado para não utilizarmos somente um '=', pois poderá acarretar em erros e mau funcionamento da lógica em geral.

Podemos comparar em diversos casos, um exemplo é na estrutura de decisão 'if':

```
let numero = 10;
let string = '20';

// Se, numero for igual a string,
if (numero == string) {
  // Retorna o conteúdo da primeira chaves ({})
  console.log('São iguais');
}  // Caso seja diferente,
  else {
  // Retorna o conteúdo da segunda chaves
  console.log('São diferentes');
}

//Retorna: São diferentes
```

Exemplo com string:

```
let nome1 = 'Clóvis';
let nome2 = 'Francisco';

console.log(nome1 == nome2);

// Retorna: false
```

```
let nome1 = 'Clóvis';
let nome2 = 'Clóvis';

console.log(nome1 == nome2);

// Retorna: true
```

Porém, com o operador de igualdade não é possível compararmos seus tipos, então será possível um retorno verdadeiro quando comparar 10 == '10'.

Exemplo com string e number:

```
let numero1 = 10;
let numero2 = '10';

if (numero1 == numero2) {
  console.log('São iguais');
} else {
  console.log('São diferentes');
}

//Retorna: São iguais
```

Com valores booleanos também:

```
let numero = 1;
let boolean = true;

if (numero == boolean) {
  console.log('São iguais');
} else {
  console.log('São diferentes');
}

//Retorna: São iguais
```

É estranho pensar que valores como 'true' e 1 são iguais, porém, diferentemente de nós humanos, as máquinas acreditam que são iguais. Em determinado momento da história, pensaram que seria preciso diferenciar esses valores para as máquinas e então, criaram o "estritamente igual" (===).

=== → Operador de Comparação de Igualdade Estrita

Para nós humanos é fácil entender que os valores são diferentes, é logicamente visível, mas isso não acontece para as máquinas, temos então que utilizar o operador de comparação de igualdade estrita. Diferentemente do operador anterior, esses três iguais são responsáveis por comparar os valores e também o tipo da variáel em questão. Por exemplo:

```
// Number:
let numero = 1;
// Boolean
let boolean = true;

if (numero === boolean) {
  console.log('São iguais');
} else {
  console.log('São diferentes');
}

//Retorna: São diferentes
```

```
// Number:
let numero = 1;
// String:
let string = '1';

if (numero === string) {
  console.log('São iguais');
} else {
  console.log('São diferentes');
}

//Retorna: São diferentes
```

Agora sim conseguimos explicar para a máquina o que queremos fazer. Vamos imaginar que a máquina entenda da seguinte forma: esses valores são iguais, mas eles não possuem a mesma estrutura(tipo), portanto, são diferentes.

!= → Operador de Comparação de Diferença

Esse operador executa da mesma maneira que o comparador de igualdade '==', porém, de forma contrária. Então, iremos comparar se uma variável é diferente da outra variável e não mais igual.

Por exemplo:

```
let number1 = 10;
let number2 = 100;

console.log(number1 != number2);

// true
```

Nesse exemplo acima vemos sua funcionalidade caso ambas variáveis tenham valores diferentes.

```
let number = 1;
let string = '1';
let boolean = true;

console.log(number != string);
console.log(number != boolean);
console.log(string != boolean);

// false
// false
// false
```

Já nesse acima vemos que só temos valores false, ou seja, todos constam que são iguais aos outros. Isso ocorre exatamente pelo mesmo motivo que ocorre ao comparmos com '==', para o Javascript, 1, '1' e true possuem o mesmo valor, mesmo tendo tipos diferentes.

É bem estranho pensar que esses valores são iguais, para nós humanos isso não faz sentido e, por isso, criaram o comparador de diferença estrita.

!== → Operador de Comparação de Diferença Estrita

Para nós humanos é fácil entender que os valores são diferentes, é logicamente visível, mas isso não acontece para as máquinas, temos então que utilizar o operador de comparação de diferença estrita.

Por exemplo:

```
let number = 1;
let string = '1';
let boolean = true;

console.log(number !== string);
console.log(number !== boolean);
console.log(string !== boolean);

// true
// true
// true
```

Agora sim conseguimos comparar os valores juntamente aos seus tipos.

< → Comparador de Menoridade

Utilizamos esse comparador quando precisamos ver se um valor é menor que outro. Nas aulas de Matemática que tivemos durante toda a vida aprendemos que o menor valor vai do lado esquerdo e o maior do lado direito: `menor < maior`.

Com o Javascript podemos comparar essa determinada expressão, pois temos valores do tipo boolean que nos retornam true ou false.

Por exemplo:

```
let numMaior = 5;
let numMenor = 3;

console.log(numMaior < numMenor);
// false, pois 5(numMaior) é maior que 3(numMenor)

console.log(numMenor < numMaior);
// true, pois 3(numMenor) é menor que 5(numMaior)

console.log(numMaior < numMaior);
// false, pois 5(numMaior) é igual e não menor que 5(numMaior)

console.log(numMenor < numMenor);
// false, pois 3(numMenor) é igual e não menor que 3(numMenor)
```

Podemos utilizar o método length para vermos outro exemplo:

```
let string1 = 'Clóvis tem 30 anos';
let string2 = 'O Clóvis mora em Marte e faz o Beginners PRO com uma internet via rádio'

console.log(string1.length);
// 18

console.log(string2.length);
// 71

console.log(string1.length < string2.length);
// true, pois 18 é menor que 71

console.log(string2.length < string1.length);
// false, pois 71 não é menor que 18
```

Já que o método em questão é usado para nos retornar o tamanho da string, podemos compará-los em nossa aplicação.

> → Comparador de Maioridade

Utilizamos esse comparador quando precisamos ver se um valor é maior que outro. Nas aulas de Matemática que tivemos durante toda a vida aprendemos que o maior valor vai do lado esquerdo e o menor do lado direito: `maior > menor`.

Por exemplo:

```
let numMaior = 5;
let numMenor = 3;

console.log(numMaior > numMenor);
// true, pois 5(numMaior) é maior que 3(numMenor)

console.log(numMenor > numMaior);
// false, pois 3(numMenor) não é maior que 5(numMaior)

console.log(numMaior > numMaior);
// false, pois 5(numMaior) é igual e não maior que 5(numMaior)

console.log(numMenor > numMenor);
// false, pois 3(numMenor) é igual e não maior que 3(numMenor)
```

Podemos também comparar valores booleanos, como:

```
let boolean1 = true;
let boolean2 = false;

console.log(boolean1 > boolean2);
// true, pois boolean1(1) é maior que boolean2(0)

console.log(boolean2 > boolean1);
// false, pois boolean2(0) não é menor que boolean1(1)
```

Ps.: true = 1 ↔ false = 0.

<= →Comparador de Menoridade ou Igualdade

Utilizamos esse comparador quando precisamos verificar que algum valor é menor ou igual ao outro.

Na matemática temos alguns exemplos como:

- 1 ≤ 2, ✔️ pois 1 é menor que 2;
- 2 ≤ 2, ✔️ pois 2 não é menor que 2, mas é igual a 2;
- 2 ≤ 1, ❌ pois 2 não é menor e nem igual a 1;

Exemplo de uso:

```
let number1 = 5;
let number2 = 6;
let number3 = 6;

console.log(number1 <= number2);
// true, pois number1(5) é menor que number2(6)

console.log(number2 <= number3);
// true, pois number2(6) não é menor que number3(6), mas é igual

console.log(number3 <= number1);
// false, pois number3(6) não é menor e nem igual a number1(5)
```

Assim como os outros comparadores, podemos utilizar outros tipos de dados como string e boolean, e não somente number.

>= → Comparador de Maioridade ou Igualdade

Utilizamos esse comparador quando precisamos verificar que algum valor é maior ou igual ao outro.

Na matemática temos alguns exemplos como:

- 1 ≥ 2, ❌ pois 1 não é menor e nem igual a 2;
- 2 ≥ 2, ✔️ pois 2 não é maior que 2, mas é igual a 2;
- 2 ≥ 1, ✔️ pois 2 é maior que 1;

Por exemplo:

```
let string1 = 'Clóvis';
let string2 = 'Márcia';
let comprimento = 7;

console.log(string1.length);
// 6

console.log(string2.length);
// 6

console.log(string1.length >= string2.length);
// true, pois string1(6) é igual a string2(6)

console.log(string2.length >= string1.length);
// true, pois string2(6) é igual a string1(6)


console.log(string2.length >= comprimento);
// false, pois string2(6) é menor que comprimento(7)
```

Bibliografias:

- https://www.w3schools.com/js/js_assignment.asp;
- https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Expressions_and_operators#operadores_de_comparação;