

ntander Coders 2024.1 | Front-End | #1178

Isaias Soares

Id: 1178008



Condicionais IF/ELSE

Condicionais IF/ELSE

Usamos `if` para determinar a condição e qual será o caminho caso a condição seja satisfeita.

```
< let estaSol = true;

if(estaSol){
    console.log('fui para a praia');
}
```

Se dentro do corpo do `if` só houver um comando as chaves podem ser omitidas:

```
let estaSol = true;

if(estaSol)
    console.log('fui para a praia');
```

`else` é um comando opcional que determina um segundo caminho caso a condição não seja satisfeita:

```
let estaSol = true;

if(estaSol){
  console.log('fui para a praia');
}
else{
  console.log('fui para o campo');
}
```

As chaves do `else` também podem ser omitidas quando há apenas um comando no corpo:

```
let estaSol = true;

if(estaSol)
  console.log('fui para a praia');
else
  console.log('fui para o campo');
```

Operadores

Vários operadores produzem valores booleanos, que podem ser usados como condições em comandos `if` como:

Operador	Descrição
<code>==</code>	igual
<code>!=</code>	diferente

Operador	Descrição
<	menor
<=	menor igual
>	maior
>=	maior igual

Booleanos podem ser combinados usando operadores:

Operador	Descrição
	ou
&&	e

O operador E `&&` combina os booleanos a e b de acordo com a seguinte tabela de verdade:

a	<code>&&</code>	b	resultado
true	<code>&&</code>	true	true
true	<code>&&</code>	false	false
false	<code>&&</code>	true	false
false	<code>&&</code>	false	false

O operador OU `||` combina os booleanos a e b de acordo com a seguinte tabela de verdade:

a		b	resultado
true		true	true

a		b	resultado
true		false	true
false		true	true
false		false	false

Booleanos podem ser negados pelo operador not ! :

Operador	Descrição
!	não ou negação

Comparação de tipos

Como JavaScript é uma linguagem de tipagem dinâmica, pode ser um pouco difícil determinar os tipos de variáveis, especialmente aquelas obtidas fora de nosso código. Sendo assim, o JavaScript tem dois operadores extras de igualdade e diferença que usam um igual a mais: === e !==.

Esses operadores além de comparar os valores também comparam os tipos. Veja esse exemplo comparando 10 com texto com 10 como número:

```
"10" == 10 // true
"10" === 10 // false
```

O resultado esperado da igualdade é o de baixo, "10" e 10 são coisas diferentes, por isso usamos === como **operador padrão de igualdade no JavaScript**. Da mesma forma, usamos !== como operador "diferente".

Truthy e Falsy

Normalmente em uma linguagem apenas variáveis booleanas ou operadores/funções que produzem um booleano podem ser usados como condição de um if. No entanto, JavaScript é um pouco mais permissivo. Ele tem valores que são interpretados como

true e false, quando são usados como booleanos.

Por exemplo, o número `0`, o `undefined` e o `null` são interpretados como `false`, assim como `NaN` e uma string vazia `""`. Todos os demais são interpretados como `true`. Como esses valores não são rigorosamente `true` ou `false`, chamamos eles de `truthy` e `falsy`.

Referências e Materiais Complementares

- `if...else`

Próximo Tópico