

Praticando os operadores

Algo muito comum para quem trabalha com desenvolvimento de software é a comparação de valores entre variáveis ou objetos para executar ou um comportamento, ou outro. O resultado dessas comparações são dois: `true` ou `false` (verdadeiro ou falso).

Para testar nossos conhecimentos nos operadores de comparação do JavaScript, observe as variáveis abaixo:

```
a = []  
b = 20  
c = true  
d = ''
```

[COPIAR CÓDIGO](#)

Analise as comparações abaixo e marque aquelas cuja saída no console será **verdade**.



```
if (b === 20 && c !== false && d == 0) {  
  console.log('Verdade')  
} else {  
  console.log('Falso')  
}
```



Alternativa correta! Certo! A variável `b` possui o mesmo valor e o mesmo tipo e a primeira comparação é verdadeira. Como o valor armazenado na variável `c` é `true` e a comparação diz que `c` não é `false`, o resultado também é verdadeiro. Na última comparação, o

↑
VOLTAR
AO
TOPO



valor de `d`, que é uma string vazia é comparado com `0` e isso também é verdadeiro. Sendo assim, saída do console será **Verdade**.



```
if (a == 0) {  
  console.log('Verdade')  
} else {  
  console.log('Falso')  
}
```



Alternativa correta! Certo! O valor de uma lista vazia é zero, e ao comparar com um número, temos `0 == 0` resultando `true`.

C

```
if (d) {  
  console.log('Verdade')  
} else {  
  console.log('Falso')  
}
```



Quase... Uma string vazia no JavaScript é considerada `false`.



```
if (b == '20') {  
  console.log('Verdade')  
} else {  
  console.log('Falso')  
}
```



Alternativa correta! Certo! Quando comparamos `20` do tipo inteiro com `'20'` do tipo string com o operador `==` (igual, igual), apenas os valores das variáveis são comparados, resultando no valor `true`.



PRÓXIMA ATIVIDADE

