

JavaScript

condicionais - Parte I



JS
JavaScript

parte 6

aprendendo Javascript

agenda

uma das instruções mais tradicionais em quaisquer soluções de software é basicamente comparar algo e decidir o que fazer. Na lição de hoje vamos conhecer mais sobre como fazer isto em JS.



pra que lado eu vou decidindo seu caminho no código

qualquer decisão que tenhamos que tomar é levada imediatamente a uma necessidade natural de se comparar e decidir. Nas linguagens de programação nossos algoritmos são guiados em quase todo o tempo por decisões. Faça isto se aquilo for igual a A, ou caso contrário faça outra coisa.

Em situações mais complexas, mais de uma condição será aplicada, ou uma quantidade finita delas.

No Javascript, o comando de decisão mais comum é o `if` (literalmente **se** em nosso idioma). Nele nós podemos certificar condições para que uma decisão verdadeira aconteça ou não, ou, se ainda haverá um fluxo de exceção.

a primeira declaração `if` mais simplista possível é a declaração com uma condição entre parêntesis que deve resultar sempre em uma resposta verdadeira.



```
// Condicional if
if (condição) {
    // Código se condição for verdadeira.
    ...
}
```



```
// Testando se a variável nome contém Alan
if (nome === 'Alan') {
    console.log(`Olá, ${nome}`);
}
```

operadores de comparação

como podemos comparar ?

no JavaScript é possível efetuar comparações utilizando um conjunto de operadores muito comum em ciência da computação. São eles:

==	compara se a variável ou valor da esquerda é igual ao valor ou variável da direita.
===	compara se o valor ou variável da esquerda é igual e do mesmo tipo do valor ou variável da direita.
!=	compara se o valor ou variável da esquerda é diferente do valor ou variável da direita.
!==	compara se o valor ou variável da esquerda é diferente e de tipo diferente do valor ou variável da direita.

<	compara se a variável ou valor da esquerda é menor que o valor ou variável da direita.
>	compara se o valor ou variável da esquerda é maior que o valor ou variável da direita.
<=	compara se o valor ou variável da esquerda é menor ou igual ao valor ou variável da direita.
>=	compara se o valor ou variável da esquerda é maior ou igual ao valor ou variável da direita.

estes operadores matemáticos são muitíssimo usados no desenvolvimento de sistemas. Atenção especial para os operadores que chamamos de *Stricts*, `===` e `!==` e dê preferência para seu uso.

comparando

testando opções

crie códigos JavaScript usando o comando `if` para decidir se executa ou não determinada(s) instruções.

```
// Condicionais com JS, operador if
if (6 == '6') {
  console.log("A comparação de 6 == '6' resultou em verdadeiro");
}

if ('a' === 'a') {
  console.log("A comparação de 'a' === 'a' resultou em verdadeiro");
}

if (5 > 6) { // Não vai exibir na console.
  console.log('A comparação de 5 > 6 retornou verdadeiro');
}

if (6 !== '6') {
  console.log("e 6 é igual a '6', mas não é do mesmo tipo, e retorna verdadeiro");
}

if (61 >= 61) {
  console.log('61 não é maior que 61 mas é igual, e retorna verdadeiro');
}
```


condições compostas

comparar “E” e “OU”

no JavaScript é possível efetuar comparações compostas no mesmo `if` se necessário. Temos 2 operadores lógicos muito utilizados. São eles.

||

operador **OU**. Significa que se um ou outro for verdadeiro já pode executar o código.

&&

operador **E**. Significa que todas as comparações devem ser verdadeiras.



```
// Condicionais compostas com JS.  
if (4 == '6' || 5 > 2) {  
    console.log('4 não é igual a 6, mas 5 é maior que 2');  
}  
  
if (1 < 6 && 'a' !== 'b') {  
    console.log('As duas condições são verdadeiras.');}  
  
let clima = 'Ensolarado';  
let tempCelsius = 27;  
  
if (clima === 'Ensolarado' && tempCelsius > 25) {  
    console.log(`☀️ dia quente com ${tempCelsius} graus celsius`);  
}
```

e se não for verdadeiro ?

senão faça aquilo

quando for necessário usar a negação da condição há uma forma bem fácil de deixar o seu código legível. Use a construção **senão** do `if`. Veja como ficaria em JavaScript.

```
// Condicionais com então em JS
if (clima === 'Ensolarado') {
  console.log('Dia ensolarado');
} else {
  console.log('Dia chuvoso');
}
```

se a condição não for verdadeira, ele vai ser desviado para o bloco de código do `else` (senão). É uma maneira fácil e estruturada de orientar sua lógica no código.

Também é possível ter uma estrutura de vários testes de forma que em uma deles seja verdadeiro, ainda que nenhuma verdadeira, o famoso `else` (senão)

```
// Condicionais complexas.
if (clima === 'Ensolarado') {
  console.log('Dia de ☀️'); // ensolarado
} else if (clima === 'Chuvoso') {
  console.log('Está ☁️'); // chuvoso
} else {
  console.log('Dia ☁️'); // nublado
}
```

condições aninhadas não é uma boa prática

uma boa prática do mundo da programação é evitar criar blocos de códigos com `if` aninhados. `if`, seguido de `if`, seguido de `if` e assim sucessivamente.

O motivo é que o código não fica legível e pode causar muita confusão especialmente em momentos de manutenção do programa. Tente condições compostas quando aplicável.

A construção abaixo é possível mas deve ser evitada.

```
// Condicionais aninhadas.
if (salario > 1280.52) {
    if (cargo === 'Tester') { // tester
        if (mes === 11) { // novembro
            if (nivel === 'Jr') {
                salario = salario * 1.08; // 8%
            } else if (nivel === 'Pleno') {
                salario = salario * 1.10; // 10%
            } else {
                salario = salario * 1.11; // 11%
            }
        }
    }
}
```




VAMOS PRATICAR

programming

revendo a prática

princípio básico

1. vimos que JavaScript é possível usar o comando **if** para decidir qual caminho nosso código vai tomar.
2. aprendemos também sobre os operadores de comparação e os relacionais.

