

Trabalhando com condições

Transcrição

Nosso grande rival futebolístico, o Livros Velhos Futebol Clube, é o favorito e está indo muito bem! Ele já somou 28 pontos no campeonato! Será que ele está melhor ou pior que o CDC Futebol Clube?

Para verificar seu desempenho, vamos fazer uma pergunta para o JavaScript. Algo como "se pontos é maior que 28, mostre que estamos bem". Para fazer isso, teremos o seguinte código:

```
if(pontos > 28) {  
    mostra("Nosso time está indo MELHOR que o Livros Velhos!");  
}
```

O `if` é a palavra chave do JavaScript. Ele faz a comparação que está dentro dos parênteses e, no caso de ser verdadeira, executa o que está dentro das chaves.

Faça o teste! Rode seu código preenchendo ele com 7 vitórias e 4 empates, o que totalizará 25 pontos. Nesse caso, o que acontece com o `if`? E se você colocar que houve 10 vitórias e 1 empate?

Além do operador `>` é preciso utilizar o `<`. Assim, podemos verificar se nossa equipe está pior ou melhor que o time Livros Velhos:

```
if(pontos < 28) {  
    mostra("Nosso time está indo PIOR que o Livros Velhos!");  
}
```

Falta apenas uma comparação: a de igualdade. Como saber se estamos empatados com o Livros Velhos? Tudo indica que o `=` deveria resolver o problema. Infelizmente, o operador `=` não funcionará como esperado, pois conforme vimos ele serve para **atribuir** valores. Isto é, ele pega o valor que está à direita do sinal de igual e copia para a variável da esquerda.

A solução é utilizar o operador que faz a comparação de igualdade, que é o `==`. São dois sinais de igual, um seguido do outro! Por exemplo:

```
if(pontos == 28) {  
    mostra("Nosso time está EMPATADO com o Livros Velhos!");  
}
```

Pode parecer estranho, mas esse `==` é utilizado em muitas linguagens, inclusive, na **sintaxe** do JavaScript. A forma como cada linguagem trabalha é diferente e cada uma tem suas peculiaridades e detalhes. Um detalhe da sintaxe que chama a atenção do aluno é um comando como o `if`, no final do fechamento das chaves (`}`), não utilizar ponto e vírgula!

Até agora criamos um arquivo chamado `pontos_futebol.html` que descobre a situação do nosso time no campeonato de futebol! Vamos rever o código. Ele começa com a declaração das duas funções que temos utilizado mais, a `pulaLinha()` e a

```
mostra() :
```

```
<script>
function pulaLinha() {
    document.write("<br>");
}

function mostra(frase) {
    document.write(frase);
    pulaLinha();
}
```

No código perguntamos ao usuário qual é a quantidade de vitórias e empates do time. Fizemos isso utilizando a função `prompt` :

```
var vitorias = parseInt(prompt("Quantos jogos o CDC Futebol Clube venceu?"));
var empates = parseInt(prompt("Quantos jogos o CDC Futebol Clube empatou?"));
```

Em seguida, calculamos a quantidade de pontos totais do time. Considerando que cada vitória vale 3 pontos e cada empate vale 1, temos o seguinte código:

```
var pontos = (vitorias * 3) + empates;
```

Já mencionamos que nosso arqui-rival, o time Livros Velhos, está com 28 pontos. Nosso objetivo, agora, é mostrar por meio do programa se estamos na frente do outro time, atrás ou empatados. Para realizar essas condições, usamos novamente o `if` :

```
var vitorias = parseInt(prompt("Quantos jogos o CDC Futebol Clube venceu?"));
var empates = parseInt(prompt("Quantos jogos o CDC Futebol Clube empatou?"));

var pontos = (vitorias * 3) + empates;

if(pontos > 28) {
    mostra("Nosso time está indo MELHOR que o Livros Velhos!");
}

if(pontos < 28) {
    mostra("Nosso time está indo PIOR que o Livros Velhos!");
}

if(pontos == 28) {
    mostra("Nosso time está EMPATADO com o Livros Velhos!");
}
</script>
```

