



JAVASCRIPT.



MATERIAL DE APOIO

Disciplina: JavaScript

Nome da aula: If, switch, for, while.

Professor(a): Marcelo Estruc

Introdução

A linguagem Javascript possui uma série de instruções que possibilitam os programas implementarem tomada de decisão, bem como repetição de um trecho de código ou até mesmo todo. Estas são conhecidas respectivamente por desvios condicionais e laços, chamados comumente por Looping.

Objetivos da aula

- Definir os conceitos de desvio condicional e de Laços.
- Demonstrar as principais instruções de desvio condicional e de Laços da linguagem Javascript.
- Explicar como trabalhar desvios condicionais de Loopings em uma rotina Javascript.

Resumo

Ao escrever um programa, pode haver uma situação em que você precise adotar um de um determinado conjunto de caminhos. Nesses casos, você precisa usar instruções condicionais que permitam que seu programa tome decisões corretas e execute as ações corretas.

JavaScript oferece suporte a declarações condicionais que são usadas para executar ações diferentes com base em condições diferentes

JavaScript suporta as seguintes formas de instrução if..else:

- declaração if
- declaração if ... else
- declaração if ... else if

Declaração if

A instrução if é a instrução de controle fundamental que permite ao JavaScript tomar decisões e executar instruções condicionalmente.

Sintaxe

A sintaxe para uma instrução if básica é a seguinte:

```
1  if (expressão) {  
2    comando(s) executados se expressão verdadeira.  
3  }  
4
```

Fonte: Autor

Aqui, **uma** expressão JavaScript é avaliada. Se o valor resultante for **verdadeiro**, a (s) instrução(ões) fornecida(s) são executadas. Se a expressão for **falsa**, **nenhuma** instrução não será executada. Na maioria das vezes, você usará operadores de comparação ao tomar decisões.

```
1  var idade = 20;  
2    if (idade > 18) {  
3    |    document.write("<b> permitido dirigir </b>");  
4
```

Fonte: Autor

Declaração if ... else

A instrução 'if ... else' é a próxima forma de instrução de controle que permite ao JavaScript executar instruções de uma forma mais controlada.

```
1  if (expressão) {  
2    Instrução(ões) a serem executadas se a expressão for verdadeira  
3  } else {  
4    Instrução(ões) a serem executadas se a expressão for falsa  
5  }  
6
```

Fonte: Autor

Aqui a expressão JavaScript é avaliada. Se o valor resultante for verdadeiro, a(s) declaração(ões) fornecida(s) no bloco 'if' são executadas. Se a expressão for falsa, então a (s) instrução(ões) fornecida (s) no bloco else são executadas.

```
1  var idade = 15;
2
3      if (idade > 18) {
4          document.write("<b> Qualifica para dirigir </b>");
5      } else {
6          document.write("<b> Não se qualifica para dirigir </b>");
7      }
8
```

Fonte: Autor

Declaração if ... else if ...

A instrução if ... else if ... é uma forma avançada de if ... else que permite ao JavaScript tomar uma decisão correta a partir de várias condições. Também denominada de “**if aninhado**”.

```
1  if (expressão 1) {
2      Instrução(ões) a serem executadas se a expressão 1 for verdadeira
3  } else if (expressão 2) {
4      Instrução(ões) a serem executadas se a expressão 2 for verdadeira
5  } else if (expressão 3) {
6      Declaração(ões) a ser executada se a expressão 3 for verdadeira
7  } else {
8      Instrução(ões) a serem executadas se nenhuma expressão for verdadeira
9  }
10
```

Fonte: Autor

Trata-se de uma série de instruções if, onde cada if é parte da cláusula else da instrução anterior. As instruções são executadas com base na condição verdadeira, se nenhuma das condições for verdadeira, o bloco **else** é executado.

```
1  var livro = "matemática";
2      if (livro == "história") {
3          document.write("<b> Livro de história </b>");
4      } else if (livro == "maths") {
5          document.write("<b> Livro de matemática </b>");
6      } else if (livro == "economia") {
7          document.write("<b> Livro de Economia </b>");
8      } else {
9          document.write("<b> Livro desconhecido </b>");
10     }
11
```

Fonte: Autor

Switch Case

Você pode usar várias instruções if ... else ... if, como visto anteriormente, para realizar uma ramificação multiway. No entanto, essa nem sempre é a melhor solução, especialmente quando todos os ramos dependem do valor de uma única variável.

Com o JavaScript 1.2, você pode usar uma instrução **switch** que lida exatamente com essa situação e o faz de maneira mais eficiente do que as instruções if ... else if repetidas.

Sintaxe

O objetivo de uma instrução switch é fornecer uma expressão para avaliar e várias instruções diferentes para executar com base no valor da expressão. O interpretador verifica cada caso em relação ao valor da expressão até que uma correspondência seja encontrada. Se nada corresponder, uma condição default será usada.

```
1  switch (expressão) {
2      condição case 1: declaração(ões)
3      break;
4
5      condição case 2: declaração(ões)
6      break;
7      ...
8
9      condição case n: declaração(ões)
10     break;
11
12     default: declaração(ões)
13 }
14
```

Fonte: Autor

As instruções **break** indicam o final de um caso particular. Se eles fossem omitidos, o interpretador continuaria executando cada instrução em cada um dos seguintes casos.

```
1  var nota = 'A';
2      document.write ("Entrando no bloco de comutação <br />");
3      switch (nota) {
4          case 'A': document.write ("Bom trabalho <br />");
5          break;
6
7          case 'B': document.write ("Muito bom <br />");
8          break;
9
10         case 'C': document.write ("Aprovado <br />");
11         break;
12
13         case 'D': document.write ("Não é tão bom <br />");
14         break;
15
16         case 'F': document.write ("Falha <br />");
17         break;
18
19         default: document.write ("grau desconhecido <br />")
20     }
21     document.write ("Saindo do bloco de comutação");
22
```

Fonte: Autor

O laço while

O loop mais básico em JavaScript é o loop while, cujo objetivo é executar uma instrução ou bloco de código repetidamente, desde que uma expressão seja verdadeira. Quando a expressão se torna falsa, o loop termina.

Sintaxe

A sintaxe do laço while em JavaScript é a seguinte:

```
1 while (expressão) {  
2     Instrução(ões) a serem executadas se a expressão for verdadeira  
3 }  
4
```

Fonte: Autor

Exemplo:

```
1 var contagem = 0;  
2     document.write ("Iniciando Loop");  
3  
4     enquanto (contagem <10) {  
5         document.write ("Contagem atual:" + contagem + "<br />");  
6         contagem ++;  
7     }  
8  
9     document.write ("Loop interrompido!");  
10
```

Fonte: Autor

O Loop do ... while

O loop do ... while é semelhante ao loop while, exceto que a verificação da condição ocorre no **final** do loop. Isso significa que o loop sempre será executado **pelo menos uma vez**, mesmo se a condição for falsa.

Sintaxe

A sintaxe para o loop do-while em JavaScript é a seguinte:

```
1  do {  
2  |    Declaração(ões) a ser executada;  
3  } while (expressão);  
4
```

Fonte: Autor

Exemplo:

```
1  var conta = 0;  
2  
3  document.write ("Iniciando Loop" + "<br />");  
4  do {  
5  |    document.write ("Contagem atual:" + conta + "<br />");  
6  |    conta ++;  
7  |    }  
8  
9  enquanto (conta <5);  
10 document.write ("Loop interrompido!");  
11
```

Fonte: Autor

Referência Bibliográfica

FLANAGAN, David. **JavaScript: O Guia Definitivo**. 6ª Ed. Porto Alegre: Bookman, 2013.

FREEMAN, Eric. **Use a cabeça!:** programação JavaScript. 1ª Ed. São Paulo: Alta Books, 2016.

Exercícios

- 1) Sobre o comando de seleção if, assinale a alternativa correta:
- a) A instrução pode avaliar mais de um teste lógico por vez.
 - b) Se a expressão avaliada for falsa a instrução logo após o if será executada.
 - c) As expressões utilizadas podem ser criadas apenas com os operadores aritméticos.
 - d) Apenas instruções de saída de dados podem ser executadas em um comando if.
 - e) Idade = 18; if (idade <= 18) { document.write(" Pode dirigir "); Nesse comando, nada será executado.

- 2) Observe a porção de código abaixo e assinale o item correto:

```
1  var conta = 0;
2
3      document.write ("Iniciando Loop" + "<br />");
4      do {
5          document.write ("Contagem atual:" + conta + "<br />");
6          conta ++;
7      }
8
9      enquanto (conta <5);
10     document.write ("Loop interrompido!");
11
```

Fonte: Autor

- a) Serão impressos os números de zero a quatro.
- b) Nenhum número será impresso.
- c) Apenas o zero será impresso.
- d) Serão impressos os números de zero a cinco.
- e) Apenas os números ímpares entre zero e cinco serão impressos.

- 3) Sobre os comandos while e do/while, assinale o item correto:
- a) Em ambos, o teste lógico é avaliado no início do laço.
 - b) Apenas no do/while o teste lógico é avaliado no início do laço.
 - c) Apenas no while o teste lógico é avaliado no início do laço.
 - d) Em ambos o teste lógico é avaliado no final do laço, sendo executados pelo menos uma vez.
 - e) Caso o teste lógico no while seja falso já na primeira execução, os comandos internos ao laço serão executados uma única vez.
- 4) Sobre a estrutura de seleção if...else if, assinale o item correto:
- a) Permite a avaliação de até cinco expressões lógicas no máximo.
 - b) Não pode ser substituída por nenhuma outra estrutura em sua função.
 - c) Utiliza-se do comando break obrigatoriamente.
 - d) Muito útil para a construção de estruturas onde vários testes precisam ser avaliados, um menu de opções, por exemplo.
 - e) Cada bloco if...else if só pode conter um comando para ser executado.

Gabarito

1. A **Letra B** é a alternativa correta, pois ao ser avaliado como falso, o próximo comando após o bloco if será avaliado.
2. A **Letra A** é a alternativa correta, pois o laço contador será executado cinco vezes, imprimindo os números de 0 a 4.
3. A **Letra C** é a alternativa correta, pois no laço while o teste lógico é feito no início do laço.
4. A **Letra D** é a alternativa correta, pois o if...else if tem a capacidade de avaliar vários testes em sua construção.