UNINOVE

Rotinas de Programação (Estruturas de decisão)

APRESENTAR AS ESTRUTURAS DE DECISÃO.

AUTOR(A): PROF. DANIEL FERREIRA DE BARROS JUNIOR

Estruturas de decisão

Em diversas situações, o seu programa JavaScript terá que tomar decisões baseado em regras pré-estabelecidas. Esta capacidade permite ao JavaScript, assim como em várias outras linguagens de programação, criar estruturas de decisão relevantes e de apoio ao seu programa. Esta capacidade de decisão se faz por meio de instruções, ou expressões lógicas, onde espera-se uma resposta, verdadeira ou falsa.

Decisão condicional

Uma decisão condicional é delimitada por uma expressão lógica, como uma forma de pergunta, onde se relaciona duas condições. Veja o exemplo:

```
var x = 5;
```

Neste caso, uma variável x é criada, e seu valor inicial está determinado como 5.

Em algumas situações devemos testar o valor desta variável, por exemplo, se x é um número positivo, ou seja, se é maior do que o valor zero.

Se x for maior do que zero temos um número positivo!

Como testar este valor?

(x > 0) ????

Na decisão condicional, temos o comando if, que permite obter uma resposta lógica para este tipo de pergunta. Em nosso exemplo, ficaria assim:

```
if (x > 0 )
```

Mais não basta isso, devemos delimitar uma região para se caso a expressão lógica leve a condição verdadeira, veja:

```
if ( x > 0 ) {
        // comando executados se caso o valor for maior do que zero
```

}

Neste espaço, delimitado entre os colchetes { }, podemos inserir os comando que devem ser executados se caso o valor for maior do que zero. No entanto, o valor pode não ser maior do que zero, o que leva a expressão lógica e retorna o valor falso.

Neste caso, o retorno da expressão lógica direciona a outro ponto da estrutura condicional. Veja o exemplo de como delimitar.

```
1.  if ( x > 0 ) {
2.      // comandos executados se caso o valor for maior do que zero
3. } else {
4.      // comandos executados se caso o valor não for maior do que zero
5. }
```

Agora temos uma região delimitada ao retorno verdadeiro e falso! Poderíamos ler a expressão como: “Se o valor de x for maior do que zero, execute estes comandos, se não, execute estes outros comandos”.

Note que estamos sempre relacionando dois elementos, seja duas variáveis, uma variável e um valor, uma variável com uma expressão, uma variável com um retorno de função, ou seja, sempre estamos buscando respostas lógicas por meio de um relacionamento entre valores. Para tanto, vale relembrar a nossa lista de operadores relacionais:

No programa a seguir temos um exemplo que demonstra esta estrutura condicional.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Tópico 5</title>
        <meta charset="UTF-8">
    </head>
    <body>
        <div>Programação de Interfaces (aula 5)</div>
        <script type="text/javascript">
            var x = 5;

            if (x > 0){
                document.write("<br> positivo");
            }
            else{
                document.write("<br> negativo");
            }
        </script>
    </body>
</html>
```

Em algumas situações é necessário utilizar uma pergunta dentro de outra pergunta, ou seja, uma condição lógica dentro de outra condição logica.

Faremos isso logo após a instrução else (se não), o que permite após este ponto, iniciar uma nova condição lógica.

Usando o caso anterior, o nosso programa agora quer saber se um numero é positivo, negativo ou nulo. Note que testamos a condição negativa ou nula apenas se a condição “positiva” (x > 0) for falsa.

Veja o exemplo a seguir:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Tópico 5</title>
        <meta charset="UTF-8">
    </head>
    <body>
        <div>Programação de Interfaces (aula 5)</div>
        <script type="text/javascript">
            var x = 5;

            if (x > 0){
                document.write("<br> positivo");
            }
            else{
                if (x < 0){
                    document.write("<br> negativo");
                }
                else{
                    document.write("<br> NULO");
                }
            }
        </script>
    </body>
</html>
```

Em certos casos necessitam testar mais de uma expressão logica e utilizando os operadores lógicos, como descritos na aula anterior.

No exemplo a seguir, vamos testar uma expressão lógica em 3 condições:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Tópico 5</title>
        <meta charset="UTF-8">
    </head>
    <body>
        <div>Programação de Interfaces (aula 5)</div>
        <script type="text/javascript">
            var a = 1, b = 5, c = 3, d = 1;

            if ( (a < b) && (c + d > 0) || (c % 2 == 0) ){
                document.write("<br> verdadeiro");
            }
            else{
                document.write("<br> falso");
            }
        </script>
    </body>
</html>
```

Neste exemplo temos a expressão:

```
if ( (a < b) && (c + d > 0) || (c % 2 == 0) )
```

De acordo com os valores iniciais, (a < b) e verdadeiro, (c + d > 0) é verdadeiro e (c % 2 == 0) é falso, uma vez que a variável c não é par. De acordo com os operadores lógicos && e ||, a resposta final é dada como verdadeira.

Caso seja alterado para &&, teríamos o seguinte fragmento de código:

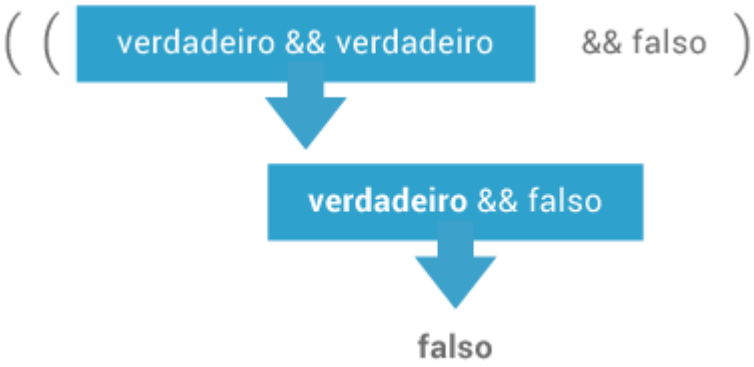
```
 if ( (a < b) && (c + d > 0) && (x % 2 == 0) )
```

Neste caso, a resposta final será dada como falso, pois a última questão lógica (c % 2 == 0) não é verdadeiro.

Teríamos a seguinte condição lógica:

verdadeiro e verdadeiro e falso

veja na imagem a seguir este mesmo resultado:



DICA:

Em certos casos, podemos ter expressões logicas bem complexas. A dica é isolar cada expressão com parênteses e realizar uma verificação manual, ou seja, com um "teste de mesa", onde escrevemos a expressão lógica em um papel e avaliamos as respostas lógicas passo a passo, de acordo com a tabela verdade.

Em algumas situações é necessário utilizar uma pergunta dentro de outra pergunta, ou seja, uma condição lógica dentro de outra condição logica.

Faremos isso logo após a instrução else (se não), o que permite após este ponto, iniciar uma nova condição lógica.

Usando o caso anterior, o nosso programa agora quer saber se um numero é positivo, negativo ou nulo. Note que testamos a condição negativa ou nula apenas se a condição "positiva" (x > 0) for falsa.

Veja o exemplo a seguir:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Tópico 5</title>
        <meta charset="UTF-8">
    </head>
    <body>
        <div>Programação de Interfaces (aula 5)</div>
        <script type="text/javascript">
            var x = 5;

            if (x > 0){
                document.write("<br> positivo");
            }
            else{
                if (x < 0){
                    document.write("<br> negativo");
                }
                else{
                    document.write("<br> NULO");
                }
            }
        </script>
    </body>
</html>
```

O uso de múltiplas expressões if / else pode tornar o seu código extenso e de difícil manutenção ou alteração, dado ao encadeamento de expressões logicas. Quando necessário, o uso das chaves deve estar alinhados aos comandos if / else, a fim de evitar erros na programação.

Existe uma alternativa para simplificar o seu código (em algumas situações), caso seja necessário utiliza múltiplas condições.

Decisão condicional de múltiplas escolhas

Podemos utilizar a estrutura condicional de múltipla escolha, SWITCH(), para avaliar e direcionar a execução de comandos de acordo as condições preestabelecidas, de valores numéricos (inteiros ou real) ou expressões de texto.

Por exemplo, analise a seguinte estrutura:

```
switch ( variável ){
        case <valor_1>:
                        <comandos>
                        <comandos>
                        break;

        case <valor_1>:
                        <comandos>
                        <comandos>
                        break;
        ...
        default:
                        <comandos>
                        break;
}
```

O comando switch recebe um valor e direciona a continuidade de execução ao bloco de comandos indicado. Um ponto importante e o comando break; Caso ele não seja determinado, os comandos continuaram a serem executados, na ordem, até que se finalize a estrutura switch ou encontre um comando break;

Vamos avaliar o seguinte código, sobre o dia da semana:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Tópico 5</title>
        <meta charset="UTF-8">
    </head>
    <body>
        <div>Programação de Interfaces (aula 5)</div>
        <script type="text/javascript">
            var x = 5;
            switch(x){
                case 1:
                    document.write("<br> DOMINGO");
                    break;
                case 2:
                    document.write("<br> SEGUNDA");
                    break;
                case 3:
                    document.write("<br> TERÇA");
                    break;
                case 4:
                    document.write("<br> QUARTA");
                    break;
                case 5:
                    document.write("<br> QUINTA");
                    break;
                case 6:
                    document.write("<br> SEXTA");
                    break;
                case 7:
                    document.write("<br> SÁBADO");
                    break;
                default:
                    document.write("<br>Dia inválido!");
                    break;
            }
        </script>
    </body>
</html>
```

Neste caso, o programa direciona o valor da variável x para o case 5, pois x = 5. Se caso o valor for alterado, entre 1 a 7, a resposta será dada de “DOMINGO” a “SÁBADO”. Para valores fora deste intervalo, a resposta padrão será dada como “Dia inválido! ”.

Tente recriar este código com apenas as instruções IF() e ELSE! Analise e avalie a diferença de complexidade.

Podemos codificar quantos comando forem necessários entre a determinação case e o break, por exemplo:

```
1. switch(x){
2.         case 2:
3.                 document.write("<br>Sejá bem vindo ao nosso site!");
4.                 document.write("<br>Hoje é segunda-feira, dia X de dezembr
5.                 document.write("<br>O Valor atual do seu saldo é de R$...
6.                 .......
7.                 break;
8.         ......
9. }
```

Podemos também avaliar condições do tipo String, veja o código de exemplo:

```
1. <script type="text/javascript">
2.         var x = "JAPÃO";
3.         switch(x){
4.                 case "Brasil":
5.                         document.write("<br> BEM VINDO AO BRASIL!");
6.                         break;
7.                 case "Portugal":
8.                         document.write("<br> BEM VINDO A PORTUGAL!");
9.                         break;
10.                case "JAPÃO":
11.                        document.write("<br> BEM VINDO AO JAPÃO");
12.                        break;
13.                default:
14.                        document.write("<br>Local inválido!");
15.                        break;
16.        }
17. </script>
```

DICA:

A instrução switch é um ótimo recurso para criação de menu, ou blocos de comandos, pois simplifica a codificação, e facilita a posterior manutenção ou aperfeiçoamento do código, caso necessário. Reescreva algumas sentenças IF() ELSE para melhor entendimento. Para todos os casos, um bom alinhamento da escrita do seu código facilita a leitura e interpretação da lógica envolvida. Avalie!

Conclusão

Nesta aula avaliamos o uso de estruturas condicionais IF() Else e do SWITCH(). Verificamos que as estrutura condicionais são importante e fundamental recurso dos nossos futuros programas. Associamos ainda o uso os operadores relacionais e lógicos nas sentenças de decisão. Pratique!

ATIVIDADE

Escolha a alternativa correta para o seguinte código:

```
<script type="text/javascript">
  var x = "JAPONES";
  switch(x){
    case "Brasil":
      document.write("<br> BEM VINDO AO BRASIL!");
      break;
    case "Portugal":
      document.write("<br> BEM VINDO A PORTUGAL!");
      break;
    case "JAPÃO":
      document.write("<br> BEM VINDO AO JAPÃO");
      break;
```

```
    default:
      document.write("<br>Local inválido!");
      break;
  }
</script>
```

A. BEM VINDO AO JAPÃO?

B. Local inválido!

C. BEM VINDO AO BRASIL!");?

D. Local inválidado!

ATIVIDADE

No caso de uma estrutura de decisão, o if significa o "SE", e a expressão else, qual é o significado dela?

A. AS VEZES

B. SE SEMPRE

C. SE

D. SE NÂO

ATIVIDADE

Qual a finalidade do comando break() utilizado na estrutura SWITCH( )?

A. Parar todo o programa.

B. Finalizar a sequência de comandos CASE.

C. Finalizar o arquivo HTML

D. Finalizar o JavaScript do servidor

REFERÊNCIA

MORRISON, M. Use a cabeça JavaScript. 5º Ed. Rio de Janeiro: Alta Books, 2012. 606 p.

OLIVIERO. C. A. J. Faça um site JavaScript orientado por projeto. 6º ed. São Paulo: Érica, 2010. 266 p.

ZAKAS, Nicholas C. JavaScript de alto desempenho. 8º Ed. São Paulo: Novatec, 2010. 245 p.