UNINOVE

Rotinas de Programação (Estruturas de repetição)

APRESENTAR AS ESTRUTURAS DE REPETIÇÃO

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

Estruturas de repetição

Estruturas de repetição ou de laço permite facilmente criar repetições sobre determinadas condições, controladas ou indeterminadas, conhecido também como loop. O recurso de repetição é importantíssimo em qualquer linguagem de programação e o JavaScript não poderia estar de fora.
Estas estruturas permitem, sobre determinadas condições, realizar comandos repetidamente até que certas condições estejam atendidas.
Vamos analisar as três estruturas de repetição, cada uma com sua particularidade e vantagens.

Repetição while( )

A estrutura while() realiza repetições em seu bloco de comandos enquanto o seu teste lógico for verdadeiro. Ainda, o while() primeiramente testa a condição lógica, e enquanto a condição for verdadeira, os comandos delimitados ficam em repetição.
Vamos verificar a estrutura a seguir:

```
1. <script type="text/javascript">
2.         var x = 1;   // variável de controle inicializada
3.         while( <condição> ){
4.                 <comandos>
5.                 <comandos>
6.                 x++;    // incremento da variável de controle
7.         }
8. </script>
```

De maneira prática, vamos avaliar ma contagem, de 1 a 10:

```
1. <script type="text/javascript">
2.         var x = 1;
3.         while(x <= 10){
4.                 document.write("<br> Contagem:  " + x);
5.                 x++;
6.         }
7. </script>
```

Neste exemplo, o retorno no arquivo HTML foi:

Programação de Interfaces (aula 6)

Contagem: 1

Contagem: 2

Contagem: 3

Contagem: 4

Contagem: 5

Contagem: 6

Contagem: 7

Contagem: 8

Contagem: 9

Contagem: 10

Se caso o primeiro teste lógico der como resultado falso, nenhum comando delimitado na repetição será realizado, veja o exemplo.

```
1. <script type="text/javascript">
2.         var x = 100;
3.         while(x <= 10){
4.                 document.write("<br> Contagem:  " + x);
5.                 x++;
6.         }
7. </script>
```

Podemos observar que necessitamos de uma variável de controle, um teste lógico e um incremento. Sobre a variável de controle, é importante que ela possua um valor inicial. O teste lógico irá testar a variável de controle até uma condição desejada, e o incremento irá alterar o valor da variável de controle, pois caso não seja feito, a estrutura irá se repetir indefinidamente, como no caso a seguir:

```
1. <script type="text/javascript">
2.             while(true){
3.                 document.write("<br> Bom dia!");
4.             }
5. </script>
```

Esta é uma estrutura básica e bem comum, no entanto, há outras formas de se realizar uma repetição do tipo while(), utilizando-se de expressões logicas, por exemplo:

```
1. <script type="text/javascript">
2.         var x = 10;
3.         while(x > 0){
4.                 document.write("<br> Contagem regressiva:  " + x);
5.                 x--;
6.         }
7. </script>
```

Neste caso, a variável de controle iniciou com o valor 10, o teste lógico determina que a repetição se dá enquanto x for maior do que 0, e ainda há o decremento, onde o valor da variável diminui a cada iteração, criando assim uma contagem regressiva.

Assim como os testes lógicos na estrutura de decisão, podemos também criar múltiplas condições de avaliação lógica. Tenha muita atenção na aplicação destes testes lógicos.

Uma maneira de estudar programação é práticar em uma linguagem mais simples, natural, neste caso o pseudocódigo. Apesar de não ser tema e foco deste tópico e disciplina, segue abaixo uma imagem apresentando a estrutura do pseudocódigo e um vídeo explicativo sobre a estrutura de repetição while( ), passo a passo, em pseudocódigo.

Para saber mais sobre pseudocódigo veja em:

http://www.apoioinformatica.inf.br/produtos/visualg/linguagem

(http://www.apoioinformatica.inf.br/produtos/visualg/linguagem)

Legenda: ESTRUTURA DE UM PSEUDOCóDIGO

A seguir você verá um vídeo explicativo sobre a estrutura de repetição while( ).

Em pseudocódigo a estrutura é chamada de enquanto( ).

Legenda: VíDEO DEMONSTRAçãO DE UM LAçO TIPO WHILE (ENQUANTO)

Caso seja necessário, reveja o vídeo e leia o conteúdo indicado no link acima.

Repetição do{ } while( )

A estrutura do{ } while( ) é semelhante a estrutura de repetição while, com a diferença que o teste lógico é feito no final do bloco, ou seja, é possível que os comandos do bloco sejam executados ao menos uma vez, mesmo que o teste lógico seja falso. Veja o modelo:

```
<script type="text/javascript">
        var x = 1;      // variável de controle inicializada
        do{
                <comandos>
                <comandos>
                x++;    // incremento da variável de controle
        }while( <condição> );
</script>
```

Esta estrutura de repetição possui basicamente os mesmos elementos do while, são eles:

- Variável de controle.
- Teste lógico;
- Incremento.

O resultado final deste código é o mesmo do primeiro exemplo de contador.

Veja o seguinte exemplo, tambem com um contador, de 1 a 10, porem neste novo formato:

```
<script type="text/javascript">
        var x = 1;
        do{
                document.write("<br> Contagem:  " + x);
                x++;
        }while(x <= 10);

</script>
```

Apesar de serem parecidos, há uma diferença na maneira como são iniciados, veja este código:

```
<script type="text/javascript">
        var x = 100;
        do{
                document.write("<br> Contagem:  " + x);
                x++;
        }while(x <= 10);

</script>
```

Neste exemplo, a mensagem “Contagem: 100” é apresentada na tela, pois o teste lógico está ao final do bloco de repetição, o que permite a execução deste bloco ao menos uma vez, mesmo que o teste lógico resulte como falso (x <= 10).
Você deve transformar esta característica em vantagem, pois esta estrutura de repetição permite que os comandos sejam executados ao menos uma vez, mesmo com a condicional resultando em falso.

Repetição for( )

A estrutura de repetição for( ) entrega na prática os mesmos resultados das estruturas anteriores. A diferença é que os elementos como variável de controle, teste lógico e incremento estão na mesmo sentença do comando, separados por um ponto e virgula(;).
Vamos verificar o exemplo a seguir (contagem):

```
1. <script type="text/javascript">
2.         for ( <variável de controle> ; <condição> ; <incremento>){
3.         <comandos>
4.                 .....
5.         }
6. </script>
```

Vamos verificar o exemplo a seguir (contagem de 1 a 10):

```
1. <script type="text/javascript">
2.         for (var x = 1; x < 10; x++){
3.                 document.write("<br> Contagem:  " + x);
4.         }
5. </script>
```

Da mesma maneira a contagem regressiva é bem simples, basta utilizar o decremento x-- e inverter a lógica de finalização da repetição, veja o exemplo:

```
1. <script type="text/javascript">
2.         for (var x = 10; x > 0; x--){
3.                 document.write("<br> Contagem regressiva:  " + x);
4.         }
5.  </script>
```

Uma situação muito comum é utilizar as estruturas de repetição em conjunto com outros comandos, como uma estrutura de decisão. No exemplo a seguir, vamos exibir uma contagem de números pares, com base em uma repetição variando de 1 a 50, veja:

```
<script type="text/javascript">
        for (var x = 1; x <= 50; x++){
                if ( x % 2 == 0)
                        document.write("<br> Contagem:  " + x);
        }
</script>
```

Note que são impressos na tela apenas os valores pares, conforme a expressão lógica:

```
if ( x % 2 == 0)
    document.write("Contagem: " + x);
```

Este recurso pode ser útil em várias situações, por exemplo, destacar somente os pares, divisíveis por 10 (duas condições), em uma sequência de 1 a 50 acompanhe o exemplo.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Tópico 6</title>
        <meta charset="UTF-8">
    </head>
    <body>
        <div>Programação de Interfaces (aula 6)</div>
        <script type="text/javascript">
            for (var x = 1; x <= 50; x++){
                if ( (x % 2 == 0) && (x % 5 == 0))
                    document.write("<br><b> Contagem:  " + x + "</b>");
               else
                    document.write("<br> Contagem:  " + x);
            }
        </script>
    </body>
</html>
```

Outra característica do for é a disponibilidade do comando break; que permite sair da repetição antes de sua finalização, veja no exemplo a seguir, onde uma repetição entre números de 1 a 50 terá sua parada ocasionada após a variável de controle ser maior do que 9.

```
1. <script type="text/javascript">
2.         for (var x = 1; x <= 50; x++){
3.                 document.write("<br> Contagem:  " + x);
4.                 if (x > 9)
5.                         break;
6.         }
7.  </script>
```

DICA:

A interrupção de uma repetição por meio do comando break é possível de se utilizar, nos comandos while( ) e do{ } while( ), além no comando for( ). Tenha cautela ao usar este tipo de recurso, pois se aplicado de maneira errada pode ocasionar paradas e falhas abruptas em seu código.

Outra prática muito útil é de se utilizar mais de uma estrutura de repetição, alinhados e sequenciais. Veja no exemplo a seguir:
Tente criar novas repetições com estruturas alinhadas, isso será útil em muitas situações, como aplicação de matrizes.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Tópico 6</title>
        <meta charset="UTF-8">
    </head>
    <body>
        <div>Programação de Interfaces (aula 6)</div>
        <script type="text/javascript">
            for (var l = 1; l <= 5; l++){
                for (var c = 1; c <= 5; c++){
                    document.write("+ ");
                }
                document.write("<br>");
            }
        </script>
   </body>
</html>
```

DICA:

Para saber mais, veja em:

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Lacos_e_iteracoes (https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Lacos_e_iteracoes)

https://www.w3schools.com/js/js_loop_for.asp (https://www.w3schools.com/js/js_loop_for.asp)

REVISãO:

Nesta aula apresentamos novas estruturas:

- Conceito de estrutura de repetição.
- Comandos de repetição while( )
- Comandos de repetição do{ } while( )
- Comandos de repetição for( )

Conclusão

As estruturas de repetição são essenciais em diversos momentos da programação em JavaScript. Com este recurso é possível repetir comandos ou organizar estruturas de dados de maneira muito mais simples. Os tipos diferentes de repetições devem atender a casos específicos, a fim de facilitar e se adequar a cada situação. Treine intensamente este recurso.

ATIVIDADE

Para uma estrutura de repetição for( ), indique a alternativa correta:

A. for (var l = 1, l <= 5, l++){?
   // comandos
   }

B. for (var l = 1; l++, I <= 5){?
   // comandos
   }

C. for (var l = 1 && l <= 5 && l++){?
   // comandos
   }

D. for (var l = 1; l <= 5; l++){?
   // comandos
   }

ATIVIDADE

Quanto a estrutura de repetição while( ), escolha a alternativa correta:

A. var
   x = 1;
   while(x <= 10){
   document.write("
 Contagem: " + x);
   x++;
   }

B. var x = 1;
   while(x && > 10){
   document.write("
 Contagem: " + x);
   x++;

```
}
```

C.
```
var x = 1;
while(x <= 10)
document.write("<br> Contagem: " + x);
x++;
```

D.
```
var x = 1;
while(x <= 10)(
document.write("<br> Contagem: " + x);
x++;
]
```

ATIVIDADE

Quanto a estrutura de repetição do{ } while( ) , escolha a alternativa que apresente um código correto.

A.
```
var x = 100;
do{
document.write("<br> Contagem: " + x);
}while(x <= 10)
```

B.
```
var x = 100;
do{
document.write("<br> Contagem: " + x);
x++;
}while(x <= 10);
```

C.
```
var x = 100;
do{
document.write("<br> Contagem: " + x);
x++;
}while(x <= 10)
```

D.
```
var x = 100;
do[
document.write("<br> Contagem: " + x);
x++;
]while(x <= 10);
```

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.