

## Condições

### Expressão if

Num momento ou outro da programação, teremos necessidade de testar uma condição. O que vai permitir executar ou não uma série de instruções. Na sua forma a mais simples, a expressão if apresenta-se assim:

```
if (condição verdadeira) {  
uma ou várias instruções;  
}
```

Assim, se a condição é verdadeira, as instruções executam-se. Se ela não é, as instruções não se executam e o programa passa para o comando seguinte. De maneira um pouco mais evoluída, tem-se a expressão if...else

```
if (condição verdadeira) {  
instrução1;  
}  
else {  
instrução2;  
}
```

Se a for verdadeira (true), o bloco de instruções 1 executa-se. Se não for, ou seja, falso (false), o bloco de instruções 2 se executa.

Graças aos operadores lógicos "e" e "ou", a expressão de teste pode testar uma associação de condições.

Assim **if ((condição1) && (condição2))**, testará se a condição 1 e a condição 2 são verdadeiras.

E **if ((condição1) || (condição2))**, testará se uma ao menos das condições é verdadeira.

#### If1

```
<HTML>  
<HEAD>  
<SCRIPT LANGUAGE="JAVASCRIPT" TYPE="TEXT/JAVASCRIPT">  
  <!--  
function control(form1)  
{  
  var test = document.form1.input.value;  
  if (test<0)  
  {  
    alert("negativo: " + test);  
  }  
  else  
  {  
    alert("positivo: " + test);  
  }  
}  
  -->  
</SCRIPT>  
</HEAD>  
<BODY>  
<FORM NAME="form1">  
<INPUT TYPE="text" NAME="input" VALUE=""><BR>  
<INPUT TYPE="button" NAME="botao" VALUE="veja se o número é positivo"  
onClick="control(form1)">  
</FORM>  
</BODY>  
</HTML>
```

**If2**

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="JAVASCRIPT" TYPE="TEXT/JAVASCRIPT">
  <!--
function control(form1)
{
  var test = document.form1.input.value;
  if (test>0 && test<=10)
  {
    alert("situa-se entre 1 e 10 " + test);
  }
  else
  {
    alert("outro valor fora do intervalo: " + test);
  }
}
  -->
</SCRIPT>

</HEAD>
<BODY>
<FORM NAME="form1">
<INPUT TYPE="text" NAME="input" VALUE=""><BR>
<INPUT TYPE="button" NAME="botao" VALUE="veja se o número é positivo"
onClick="control(form1)">
</FORM>
</BODY>
</HTML>

```

**If3**

```

<HTML>
<HEAD>

<SCRIPT LANGUAGE="JAVASCRIPT" TYPE="TEXT/JAVASCRIPT">
  <!--
function control(form1)
{
  var test = document.form1.input.value;

  if (test>969999999 || test<910000000)

    {
      alert("houve um engano no número volte a tentar " + test);
    }
  else
  {
    alert("número aceite: " + test);
  }
}
  -->
</SCRIPT>

</HEAD>
<BODY>
<FORM NAME="form1">
dite o número de telemovel
<INPUT TYPE="text" NAME="input" VALUE=""><BR>
<INPUT TYPE="button" NAME="botao" VALUE="inserir" onClick="control(form1)">
</FORM>
</BODY>
</HTML>

```

## Expressão for

A expressão for permite executar um bloco de instruções num certo número de vezes em função de determinado critério. A sintaxe é:

```
for (valor inicial ; condição ; progressão) {  
instruções;
```

```
}
```

**um exemplo concreto**

```
for (i=1, i<10, i++) {  
document.write(i + "<BR>")  
}
```

Na primeira passagem, a variável i, é inicializada a 1. Sendo a variável inferior a 10. Ela é então incrementada uma unidade pelo operador de incremento i++

No fim da execução das instruções, voltamos ao contador. A variável i (que vale 2) é ainda inferior a 10. Ela é aumentada 1 e as instruções prosseguem, até que i vale 10. A variável i não satisfaz mais a condição i<10. O ciclo interrompe-se e o programa continua depois da chaveta fechada

## While

A instrução while permite executar uma instrução (ou um grupo de instruções) um certo número de vezes.

```
while (condição verdadeira){  
continuar a fazer alguma coisa  
}
```

Enquanto que a condição é verdadeira, o Javascript continua executar as instruções entre as chavetas. Uma vez que a condição não mais verdadeira, o ciclo interrompe-se e continua o script.

```
While1  
<HTML>  
<HEAD>  
<SCRIPT LANGUAGE="JAVASCRIPT" TYPE="TEXT/JAVASCRIPT">  
    <!--  
        cont=1;  
while (cont<5) {  
document.write ("linha : " + cont + "<BR>");  
cont++;  
}  
        -->  
</SCRIPT>  
  
</HEAD>  
<BODY>  
</BODY>  
</HTML>
```

Vejamos como funciona este exemplo. Primeiro a variável que irá servir de contador cont é inicializada a 1. O ciclo while arranca então com o valor 1 que é inferior a 5. A condição é verdadeira. Executa as instruções das chavetas. Primeiro, "linha : 1" é inserido e depois o contador é incrementado de 1 e toma assim o valor 2. A condição ainda é verdadeira. as instruções entre as chavetas são executadas. Até inserir a linha 4. O contador depois da incrementação vale 5. A condição não sendo verdadeira, continuamos no script e é então o fim do ciclo.

## Break

A instrução break permite interromper prematuramente um ciclo for ou while. Para ilustrar isto, retomamos o nosso exemplo:

### Break1

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JAVASCRIPT" TYPE="TEXT/JAVASCRIPT">
    <!--
        cont=1;
while (cont<5) {
if (cont == 4)
break;
document.write ("linha : " + cont + "<BR>");
cont++;
}
document.write("fim do ciclo");

        -->
</SCRIPT>

</HEAD>
<BODY>
<FORM>

</FORM>
</BODY>
</HTML>
```

O funcionamento é semelhante ao exemplo precedente só que quando o contador vale 4. Nesse momento, o break faz-nos sair do ciclo e "fim do ciclo" é inserido.

O resultado no ecrã será:

```
linha : 1
linha : 2
linha : 3
fim do ciclo
```

## Continue

A instrução continue permite de saltar uma instrução num ciclo for ou while e de continuar de seguida com o ciclo (sem sair deste como faz o break).

### Continue1

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JAVASCRIPT" TYPE="TEXT/JAVASCRIPT">
    <!--
        cont=1;
while (cont<5) {
if (cont == 3){
cont++
continue;}
document.write ("linha : " + cont + "<BR>");
cont++;
}
document.write("fim do ciclo");

        -->
</SCRIPT>

</HEAD>
<BODY>
<FORM>

</FORM>
</BODY>
</HTML>
```

Aqui, o ciclo começa, quando o contador vale 3, devido a instrução continue, salta-se a instrução document.write (linha : 3 não é afixada) e continua-se com o ciclo. Nota-se que tivemos de inserir cont++ antes continue; para evitar um ciclo infinito.

O resultado no ecrã será:

```
linha : 1
linha : 2
linha : 4
fim do ciclo
```