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 formulação a mais simples, a expressão **if** apresenta-se assim:

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

Assim, se a condição é verdadeira, as instruções são executadas, caso contrário (else), não são executadas 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 for verdadeira (true), o bloco de instruções 1 se executa. Se não for (é false), o bloco de instruções 2 se executa.

Graça aos operadores lógicos "e" e "ou", a expressão de teste pode testar uma associação de condições. Assim **if ((condição1) && (condição2))**, testará a condição 1 e a condição 2. E **if ((condição1) || (condição2))**, testará se uma ao menos, das condições, é verdadeira.

Para ser mais completo (e para aqueles que gostam da escrita concisa), também há:

```
(expressão) ? instrução a : instrução b
```

Se a expressão entre parêntese é verdadeira, a instrução **a** é executada. Se a expressão entre parêntese é falsa, a instrução **b** é executada.

Expressão for

A expressão **for** permite se executar um bloco de instruções um certo número de vez em função da realização de um certo critério. Sua sintaxe é:

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

Pegamos 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 de uma unidade pelo operador de incrementação **i++**

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

While

A instrução **while** permite de 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 chaves. Uma vezes que a condição não mais verdadeira, o ciclo interrompe-se e continua-se o script.

Pegamos este exemplo:

```
cont=1;
while (cont<5) {
document.write ("linha : " + cont + "<BR>");
cont++;
}
document.write("fim do ciclo");
```

Vejamos como funciona este exemplo. Primeiro a variável que irá servir de contador **cont** é inicializada em 1. o ciclo **while** arranca então com o valor 1 que é inferior a 5. A condição é verdadeira. Executa-se as instruções das chaves.
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 chaves são executadas. E isso até inserir a linha 4. Aí , o contador depois da incrementação vale 5. A condição já não sendo verdadeiro, continuamos no script e é então o fim do ciclo.

Atenção ! Com o sistema de ciclo, o risco existe (se a condição é sempre verdadeira), de fazer o ciclo indefinidamente a 'instrução. O que pode trancar o browser !

Break

A instrução **break** permite interromper prematuramente um ciclo **for** ou **while**.

Para ilustrar isto, retomamos o nosso exemplo:

```
cont=1;
while (cont<5) {
if (cont == 4)
break;
document.write ("linha : " + cont + "<BR>");
cont++;
}
document.write("fim do ciclo");
```

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

O resultado na tela 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 em seguida com o ciclo (sem sair deste como faz o **break**).

Retomamos o nosso exemplo ;

```
cont=1;
while (cont<5) {
if (cont == 3){
cont++
continue;}
```

```
        document.write ("linha : " + cont + "<BR>");
        cont++;
        }
        document.write("fim do ciclo");
```

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

O resultado na tela será:

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