estgoh
Politécnico de Coimbra

PI II / DW II

2.3. PHP – Instruções condicionais e ciclos

Nuno Miguel Gil Fonseca
nuno.fonseca@estgoh.ipc.pt

PHP - *Hypertext Preprocessor*

- Instruções condicionais
  - If's

```
<?php
   $var = 3;
   if($var == 1) {
      print("um");
   } elseif($var == 2) {
      print ("dois");
   } else {
      print ("três");
   }
?>
```

```
<?php
if($var){
      print $var;
}
?>
```

=

```
<?php
if($var)
      print $var;
?>
```

=

```
<?php
   if($var) print $var;
?>
```

PHP - *Hypertext Preprocessor*

- Instruções condicionais
  - If's

```
<?php
   $var = 3;
   if($var == 1):
      print("um");
   elseif($var == 2):
      print ("dois");
   else:
      print ("três");
   endif;
?>
```

PHP - *Hypertext Preprocessor*

- Instruções condicionais
  - Case *switching*

```
$nome = "Ze";
switch ($nome) {
    case "Ze" : print "O nome é Ze\n"; break;
    case "Joao" : print "O nome é Joao\n"; break;
    // ... outros casos ...
    default : print "Não sei qual é o nome!\n";
}
```

```
$nome = "Ze";
if ($nome == "Ze") {
  print "O nome é Ze\n";
} elseif ($nome == "João") {
  print "O nome é Joao\n";
} else {
  print "Não sei qual é o nome!\n";
}
```

- **Instruções condicionais**
  - **Case *switching***
    - Quando mais do que um valor levam ao mesmo resultado

```
$nome = "Jo";
switch ($nome) {
   case "Jose" :
   case "Ze" : print "O nome é Ze\n"; break;
   case "John" :
   case "Jo" :
   case "Joao" : print "O nome é Joao\n"; break;

   // ... outros casos ...

   default : print "Não sei qual é o nome!\n";
}
```

- **Instruções condicionais**
  - **Match**
    - Com o PHP8 surgiu uma alternativa ao *switch/case*

```php
$nome = "Jo";

$frase = match($nome){
    "Jose", "Ze" => "O nome é Ze\n",
    "Bond" => call_007(),
    "John", "Jo", "Joao" => "O nome é Joao\n",
    default => "Não sei qual é o nome!\n"
};

function call_007(){
    return "My name is Bond, James Bond!";
}

echo $frase;
```

- Nota1: Não é necessário que o match devolva um valor.
- Nota2: Caso seja necessário executar vários *statements*, deverá recorrer-se a uma função

PHP - *Hypertext Preprocessor*

- **Ciclos**
  - **while...**
    - Executa um pedaço de código enquanto a condição for avaliada como verdadeira.
      - O código poderá nunca ser executado

```
while (condição verdadeira) {
    // código a executar
}
```

```
$i = 0;
while ($i < 10) {
    echo $i, " ";
    $i++;
}
```

```
0 1 2 3 4 5 6 7 8 9
```

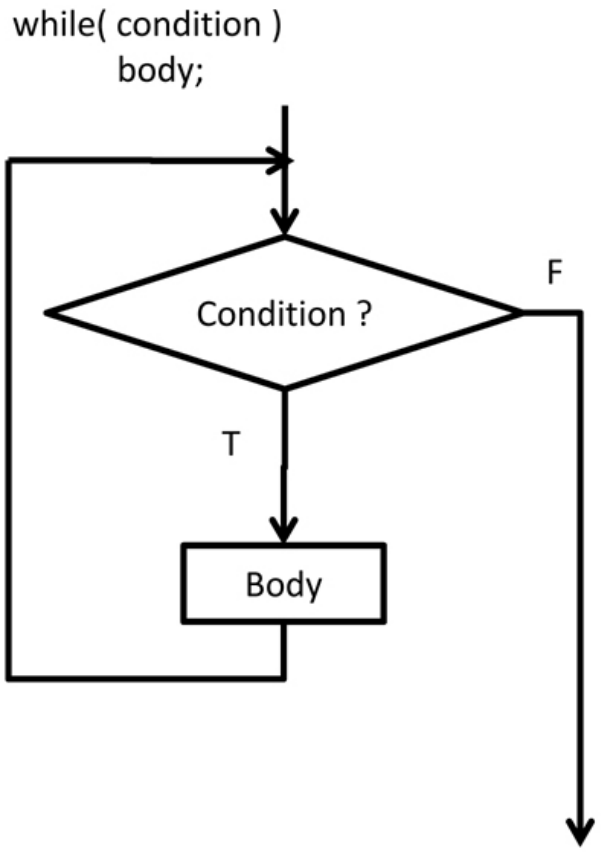
- **Ciclos**
  - **do... while...**
    - Semelhante ao anterior, no entanto, o código é sempre executado pelo menos uma vez

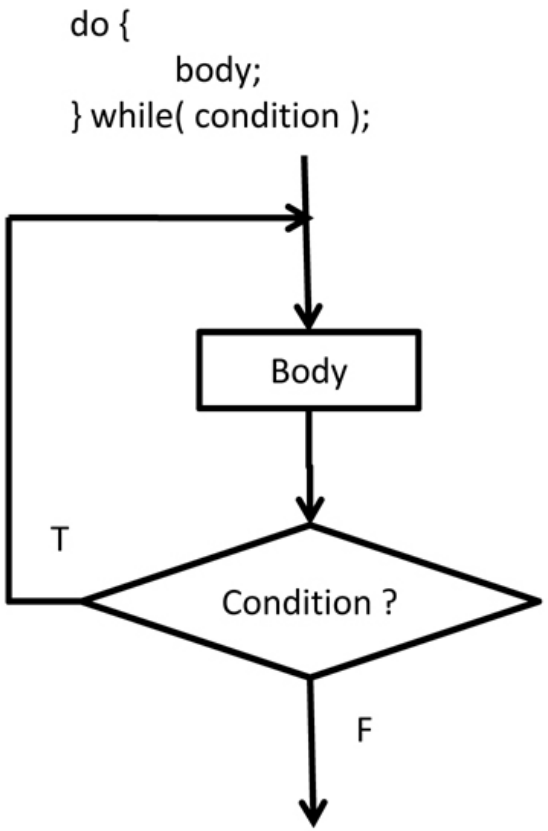
```
do {
    // código a executar
} while (condição verdadeira);
```

```
$i = 0;
do {
    echo $i, " ";
    $i++;
} while ($i < 10);
```

```
0 1 2 3 4 5 6 7 8 9
```

- Ciclos
  - while… vs. do... while...





PHP - *Hypertext Preprocessor*

- **Ciclos**
  - **For**
    - Executa um pedaço de código enquanto se verificar uma determinada condição.

```
for ($i = 1; $i < 10; $i++) {
    print("Número: $i \n");
}
```

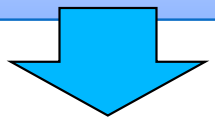
    - Em termos práticos isto é o mesmo que:

```
$i = 1;
while ($i < 10) {
     print("Número: $i \n");
     $i++;
}
```

PHP - *Hypertext Preprocessor*

- **Ciclos**
  - **For**
    - Boa prática: para poupar alguns recursos, evite fazer cálculos de tamanhos de *arrays* e *strings* dentro do `for` (**a não ser que o tamanho possa ser alterado entretanto...**)

```
$a_minha_string = "Qualquer coisa";
for ($i = 1; $i < strlen($a_minha_string); $i++) {
    print("Número: $i \n");
}
```



```
$a_minha_string = "Qualquer coisa";
$len = strlen($a_minha_string);
for ($i = 1; $i < $len; $i++) {
    print("Número: $i \n");
}
```

    - Nota: `strlen` devolve o numero de caracteres de uma *string*

- **Ciclos**
  - **Instruções especiais**
    - `continue`
      - Indica que se pode saltar para a próxima iteração do ciclo
    - `break`
      - Termina a execução do ciclo

```php
<?php
    for ($i = 1; $i < 10; $i = $i + 1) {
        if ($i == 3) continue;
        if ($i == 7) break;
        print("$i ");
    }
?>
```

```
1 2 4 5 6
```

- **Ciclos encadeados**
  - **Por vezes pode fazer sentido fazer interações dentro de outras iterações.**
    - Por exemplo, iterar numa matriz de 5 x 5

```
$matriz = [[1,2,3,4,5], [1,2,3,4,5], [1,2,3,4,5], [1,2,3,4,5], [1,2,3,4,5]];

for ($linha = 0; $linha < 5; $linha++) {
    for ($coluna = 0; $coluna < 5; $coluna++) {
            echo $matriz[$linha][$coluna], " ";
    }
    echo "\n";
}
```

```
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
```