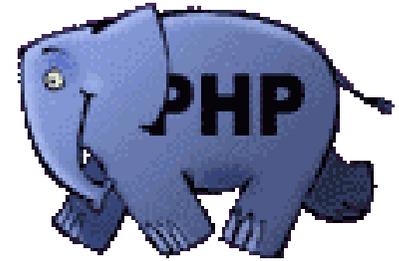




PHIP

I. Definições

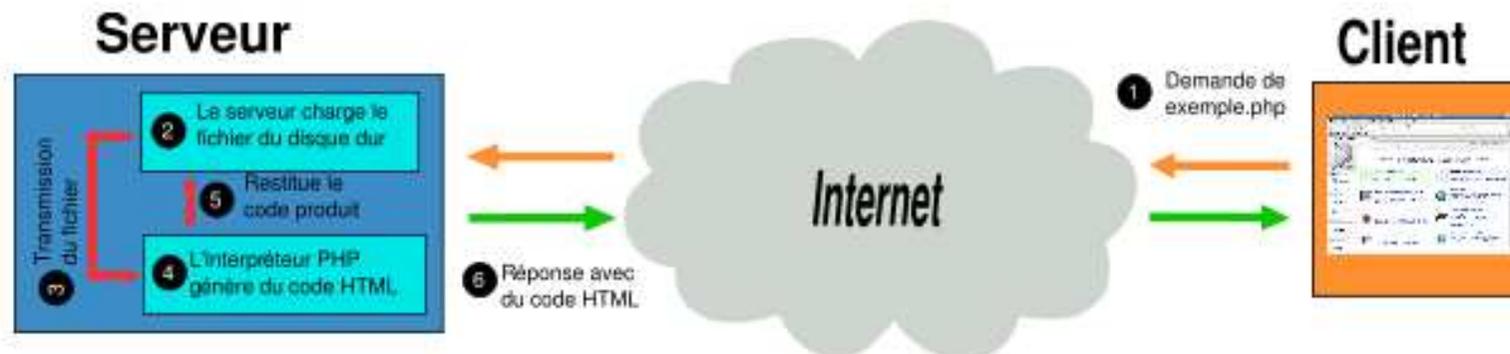


- ☞ PHP = Hypertext Preprocessor.
- ☞ = linguagem de script livre, principalmente utilizado para produzir páginas Web dinâmicas, através de um servidor HTTP.
=> o servidor interpreta o código PHP para gerar um código (HTML, CSS e às vezes JavaScript) interpretável pelo browser.
- ☞ Associado a um servidor Web (e.g., Apache), PHP permite recuperar infos que provêm de uma BD.

II. Funcionamento com o Web



- 1) Cliente → http request → servidor http.
- 2) Se página com código PHP → interpretador PHP do servidor → geração do código final da página (HTML + CSS + JS).
- 3) Restituição da página interpretada ao servidor http → envio da página ao cliente.



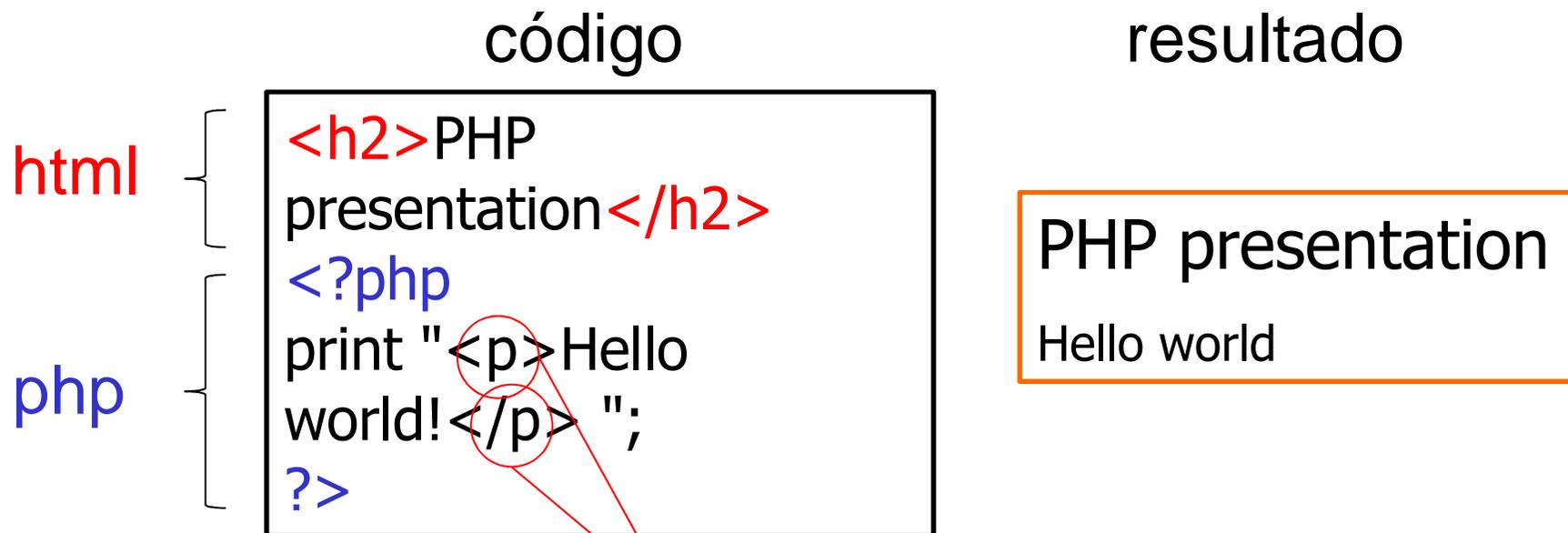
No caso em que haja uma BD, existe uma etapa suplementar na qual PHP abre uma conexão ao servidor DBMS.

III. PHP Code Tag

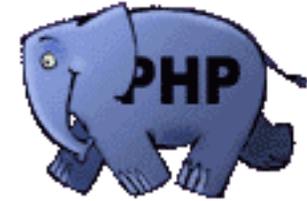


PHP pode ser integrado dentro de um código HTML:

=> nome do ficheiro = nome_ficheiro.php



Also, HTML can be inside of the code PHP, but must be inside the quotation marks.



👉 Limites do código:

```
<?php // início do código PHP
// código PHP linha 2
// código PHP linha 3
// código PHP linha 4
?> // fim do código PHP
```

👉 Outputting dados:

```
<?php
echo "PHP Presentation";
print "Hello World";
?>
```



PHP Presentation
Hello World

👉 Comentários:

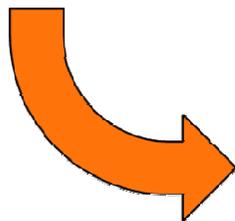
```
<?php
echo "Vivo em Portugal.<br />"; // isto é um comentário
echo "Tenho 33 anos."; // isto é um 2º comentário
/* comentário
em várias linhas
*/
?>
```

As instruções PHP acabam sempre com um “;”



If the string has a set of double quotation marks that must remain visible, use the `\` [backslash] before the quotation marks to ignore and display them.

```
<?php  
    echo "\"Multimedia Information Systems\"";  
?>;
```



"Multimedia Information Systems"

IV. As variáveis



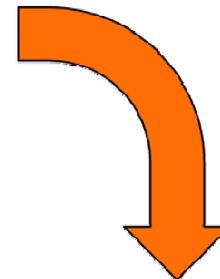
- ➔ A variable always begins with a \$ then the variable name.
- ➔ Variables are case sensitive.
- ➔ Do not start a variable with a number.
- ➔ No data types have to be defined.

```
$book  
$author  
$amount  
$Amount  
$AMOUNT
```



Exemplos de utilização das variáveis:

```
<html>
<body>
  <h2>PHP Presentation</h2><br>
  <?php
    print "<p>Hello World!</p>";
  ?>
  <br>
  <?php
    $date="Mar. 7, 2007";
  ?>
  <p>Today's date is
  <?php
    print $date;
  ?>
  </p>
</body>
</html>
```



```
PHP Presentation
Hello World!
Today's date is Mar. 7, 2007
```



👉 Operadores aritméticos:

```
<?php  
$a=15;  
$b=30;  
$total=$a+$b;  
print $total;  
print "<p><h1> $total </h1></p>";  
?>
```



45

45

`$a - $b` //subtraction

`$a * $b` //multiplication

`$a / $b` //division

`$a += 5` //`$a = $a + 5`

Also works for `*=` and `/=`



👉 Concatenação:

Use a **period** to join strings into one

```
<?php
$string1="Hello";
$string2="World";
$string3=$string1." ".$string2;
print $string3;
?>
```



Hello World

V. Transmitir variáveis



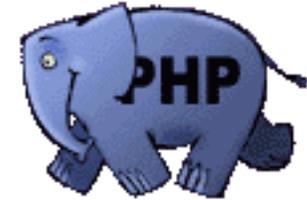
As variáveis são destruídas uma vez que a página PHP está criada.

=> Como recuperar esses valores numa outra página?

=> Solução: PHP vai permitir transmitir variáveis de uma página para outra.

Exemplo de aplicação:

Transmitir o nome de um visitante.



Transmitir variáveis através da modificação de endereço:

Ficheiro chamada.php

```
<!-- Código : HTML -->
<p>
  Nota que esta página tem apenas HTML.<br />
  Aqui são 3 links para a página alvo.php, com variáveis com
  valores diferentes:
</p>

<p>
  <a href="alvo.php?apelido=Silva&nome=João">Link para
  alvo.php?apelido=Silva&nome=João</a><br />
  <a href="alvo.php?apelido=Cardoso&nome=Clara">Link
  para alvo.php?apelido=Cardoso&nome=Clara</a><br />
  <a href="alvo.php?apelido=Fonseca&nome=Tiago">Link para
  alvo.php?apelido=Fonseca&nome=Tiago</a>
</p>
```

<http://www.meusite.com/chamada.php>

Nota que esta página tem apenas HTML.
Aqui são 3 links para a página alvo.php, com variáveis com valores diferentes:

[Link para alvo.php?apelido=Silva&nome=João](http://www.meusite.com/alvo.php?apelido=Silva&nome=João)
[Link para alvo.php?apelido=Cardoso&nome=Clara](http://www.meusite.com/alvo.php?apelido=Cardoso&nome=Clara)
[Link para alvo.php?apelido=Fonseca&nome=Tiago](http://www.meusite.com/alvo.php?apelido=Fonseca&nome=Tiago)

Ficheiro alvo.php

```
// Código : PHP
<p>Bom dia!</p>

<p>
  O seu apelido é <?php echo $_GET['apelido']; ?> , e o seu
  nome é <?php echo $_GET['nome']; ?>.
</p>

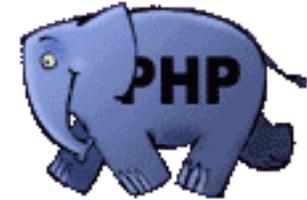
<p>
  Fazer um outro ensaio, <a href="chamada.php">clique aqui</a>
  para voltar ao chamada.php
</p>
```

<http://www.meusite.com/alvo.php?apelido=Silva&nome=João>

Bom dia!

O seu apelido é Silva, e o seu nome é João.

Fazer um outro ensaio, [clique aqui](#) para voltar ao chamada.php



Transmitir variáveis através da utilização de formulários:

Ficheiro chamada.php

```
<!-- Código : HTML -->
<p>
  Nota, também, que esta página tem apenas HTML.<br />
  Entre o seu nome:
</p>

<form action= "alvo.php" method="post">
  <p>
    <input type="text" name="nome" />
    <input type="submit" value="Validar" />
  </p>
</form>
```

Ficheiro alvo.php

```
// Código : PHP
<p>Bom dia!</p>

<p>
  Eu sei como te chamas. Chamas-te
  <?php echo $_POST['nome']; ?> !
</p>

<p>
  Se quiseres mudar de nome, <a href="appel.php">clique
  aqui</a> para voltar ao chamada.php
</p>
```

http://www.meusite.com/chamada.php

http://www.meusite.com/alvo.php

Nota, também, que esta página tem apenas HTML.
Entre o seu nome:

Paolo

Validar

Bom dia!

Eu sei como te chamas. Chamas-te Paolo!

Se quiseres mudar de nome, [clique aqui](#) para voltar ao chamada.php

VI. As funções



👉 Exemplos de funções próprias:

```
// A instrução contida nesta função é de apresentar texto (echo)
function dizerOla($nome)
{
    echo "Olá $nom !<br />";
}
// chama a função "dizerOla"
dizerOla("Paulo");
dizerOla("Maria");
dizerOla("João");
dizerOla("Ana");
dizerOla("Tiago");
```



```
Olá Paulo!
Olá Maria!
Olá João!
Olá Ana!
Olá Tiago!
```

```
// A função calcula o volume de um cone
function volumeCone($raio, $altura)
{
    $volume = $raio*$raio*3.14*$altura*(1/3);
    return $volume; // indica o valor a reenviar (aqui o volume)
}
// parâmetros (separados por uma virgula) a enviar à função
// « volumeCone » para calcular o raio e a altura
$volume = volumeCone(3, 2);
echo "O volume do cone de raio 3 e de altura 2 é:<br />
    $volume cm<sup>3</sup>";
```



```
O volume do cone de raio 3 e de altura 2 é:
18.84 cm3
```



👉 Exemplos de funções predefinidas:

```
// função predefinida « date » = recuperar a data e hora actuais  
// respeitar as maiúsculas/minúsculas!!!!!!!  
$dia= date("d"); // "d" para o dia  
$mes = date("m"); // "m" para o mês  
$ano = date("Y"); // "Y" para o ano  
$hora = date("H"); // "H" para a hora  
$minuto = date("i"); // "i" para os minutos  
  
echo "Em lisboa, são $hora hora(s) $minuto minuto(s) e<br />  
hoje é dia $dia/$mês/$ano";
```



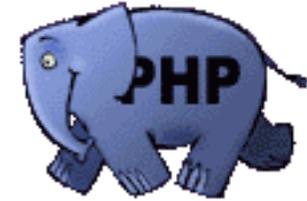
Em lisboa, são 16 hora(s) 40 minuto(s) e
hoje é dia 03/03/2011

```
// A função predefinida « Pi( ) »  
$pi = Pi();  
echo "pi = $pi"; // dá uma melhor aproximação do pi
```



pi = 3.1415926535898

VII. As condições



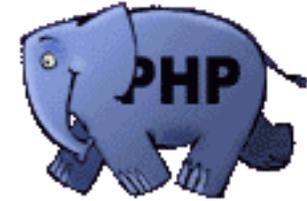
☞ A estrutura “if... else...”:

```
if (condição)
{
    Instruções A...
}
else
{
    Instruções B...
}
```

```
<?php
if ($sexo == "mulher" OR $sexo == "homem")
{
    echo "Olá ser humano!";
}
else
{
    echo "Se não for mulher nem homem, o que
    poderá ser?????????";
}
?>
```

As condições múltiplas:

- OR ou ||
- AND ou &&



Pequeno truque quando se deve imprimir ao ecrã uma grande quantidade de texto:

```
<?php
if ($variavel == 23)
{
?>
<strong>Bravo!</strong> Encontrou o número
mistério!
<?php
}
?>
```

==

```
<?php
if ($variavel == 23)
{
echo "<strong>Bravo !</strong> Encontrou
o número mistério!";
}
?>
```

1) fecho do tag PHP para inserir o código HTML

2) reabertura do tag PHP para acabar o código PHP

- Os 2 códigos dão o mesmo resultado.
- O 1º é mais vantajoso, porque permite entrar o texto apenas com a sintaxe mais simplificada do HTML.



👉 A estrutura “switch...”:

```
<?php
$nota = 10;

switch ($nota)
{
  case 0:
    echo "Sem comentário...";
    break;
  case 5:
    echo "És muito má";
    break;
  case 10:
    echo "Tens que fazer esforços";
    break;
  case 15:
    echo "Bom trabalho";
    break;
  case 20:
    echo "Excelente aluno!!!";
    break;
  default:
    echo "Desculpa, não tenho mensagem para esta nota";
}
?>
```



Tens que fazer esforços

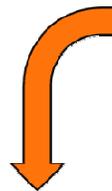
VIII. Os ciclos



☞ O ciclo “while”:

```
while (condições)
{
    Instruções
}
```

```
$count = 0;
while($count < 5)
{
    print "Hi there!!! ";
    $count++;
}
```

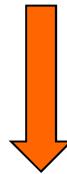


Hi there!!! Hi there!!! Hi there!!! Hi there!!! Hi there!!!



👉 O ciclo “for”:

```
for ($count = 0; $count < 5; $count++;)  
{  
    print "Hi there!!! ";  
}
```



Hi there!!! Hi there!!! Hi there!!! Hi there!!! Hi there!!!

IX. Os arrays



👉 Array numerado:

```
<?php
// aqui o array é $nome
$nome = array("João", "Miguel", "Tiago", "Rosa");

echo "$nome[2]"; // imprima ao ecrã o valor da célula 2
?>
```



Tiago

Número	Valor
0	João
1	Miguel
2	Tiago
3	Rosa
...	...



👉 Array associativo (uso de etiqueta):

```
<?php
// associação de uma etiqueta (através do símbolo « => »)
// a cada elemento do array, afim de facilitar a leitura do
// código
$contactos = array("nome" => "Marta",
                  "apelido" => "Fonseca",
                  "morada" => "Rua do Paraíso, 3",
                  "cidade" => "Lisboa");

echo "$contactos[morada]";
?>
```

Etiqueta	Valor
nome	Marta
apelido	Fonseca
morada	Rua do Paraíso, 3
cidade	Lisboa
...	...



Rua do Paraíso, 3



☞ Pequeno teste...

```
<?php
$preco = array ("Ovo" => 1.5,
               "Laranja" => 0.5,
               "Leite" => 2);

$total = (2 * $preco['Ovo']) + (1 * $preco['Leite']) + (4 * $preco['Laranja']);
echo "Total = $total euros";
?>
```

Qual é o preço total das compras?



Total = 7 euros

X. TP: página protegida por uma palavra-chave

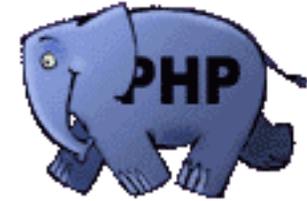


👉 Algoritmo:

- Imprimir ao ecrã uma zona de texto para entrar a palavra-chave.
- Se a palavra-chave for correcta => imprimir as infos confidenciais.
- Senão, propor entrar de novo a palavra-chave.
- Imprimir uma mensagem de erro quando a palavra é errada.

👉 **Ter o cuidado** de verificar se a palavra-chave já existe ou não:

```
if (isset($_POST['palavra-chave'])) // se a palavra-chave for introduzida...
{
    $chave = $_POST['palavra-chave']; //...criamos uma variável $chave de valor = palavra-chave
}
else // se nenhuma palavra-chave for ainda introduzida...
{
    $chave = " "; //...criamos uma variável $chave de valor nulo
}
```



👉 Exemplo do resultado a obter:

- A página de acesso:

A sua palavra-chave:

Esta página é reservada ao pessoal da NASA.

- A página protegida:

Aqui são os códigos de acesso:

CRD5-GTFT-CK65-JOPM-V29N-24G1-HH28-LLFV

Não se esqueça de visitar periodicamente esta página para ver as alterações aos códigos...



☞ Correção do código:

C:\xampp\htdocs\Tests\pageProtegee.php