PHP – O QUE É?

PHP (Hypertext PreProcessor)

É uma linguagem de programação, principalmente para web e que é normalmente incorporada em HTML

É uma das mais usadas para desenvolvimento web

É open-source, gratuito

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – PORQUÊ?

Agnóstico de plataforma (Windows, Linux Mac, etc.)

Muito estável

Grande leque de extensões

Desenvolvimento rápido

Comunidade forte

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – BREVE HISTÓRIA

1994: Rasmus Lerdorf desenvolve a página "Personal Home Page Tools" para contar o número de visualizações do seu CV

1996: PHP 2.0

1997: PHP 3.0

1999: PHP 4.0

2004: PHP 5.0

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – FUNCIONAMENTO

Quando é feito um pedido, o servidor faz um pré-processamento

Analisa toda a página e resolve as secções PHP

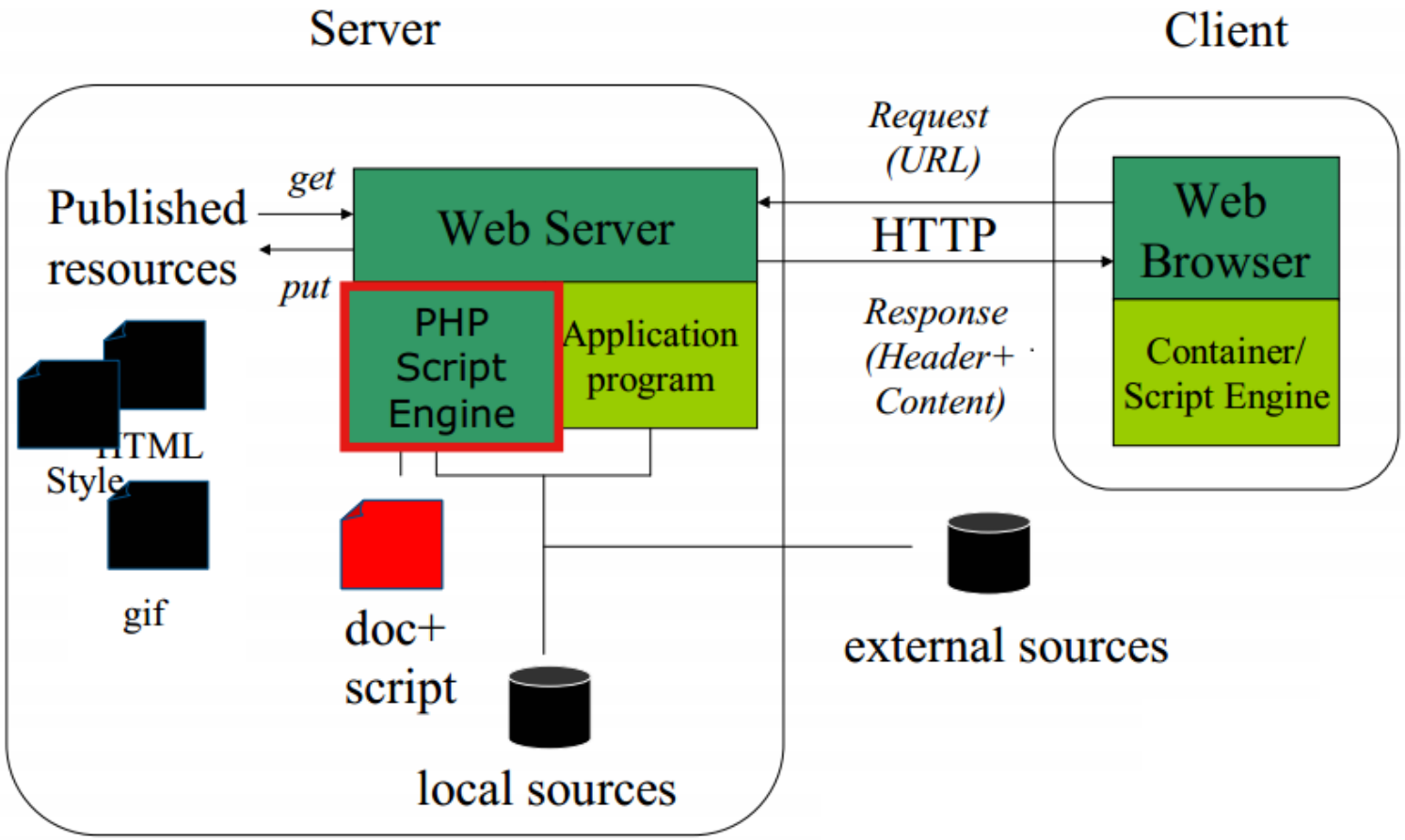
Para correr PHP é necessário:

- Navegador web
- Servidor web configurado para correr PHP
- Instalação do software PHP no servidor

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – FUNCIONAMENTO



SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – CARACTERÍSTICAS

Não é exigente em termos sintáticos

É uma linguagem orientada à conveniência e não à correção

HTML não é PHP

A sintaxe é similar à linguagem C

```
$soma = 2 + 2; // com espaços simples
$soma <tab>=<tab>2<tab>+<tab>2 ; // com espaços e identações
$soma =
2+
2; // em múltiplas linhas
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – CARACTERÍSTICAS

Parênteses perfazem blocos

- Curvos () para uma expressão numa linha
- Chavetas {} para expressões em várias linhas

```
if (3 == 2 + 1)
     print("Certo, não perdi a cabeça completamente!<BR>");
if (3 == 2 + 1)
{
     print("Certo, não perdi a cabeça ");
     print("completamente!<BR>");
}
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – ETIQUETAS

As instruções PHP encontram-se dentro de etiquetas (tags) especiais

```
<?php …?>
```

Ou em alternativa (dependendo da versão e do servidor)

```
<? … ?>
<script language="php"> … <script>
<% … %>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – COMENTÁRIOS

Comentários multilinha /*

```
/* Este é
um comentário
em PHP */
```

Comentários por linha # e //

```
# Este é um comentário
// em duas linhas de PHP
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – VARIÁVEIS

Todas as variáveis começam com o $

`$x, $email, $data`

Não são obrigatoriamente declaradas

Diferenciação de maiúsculas/minúsculas

`$Email` **é diferente de** `$email`

Tipos

Integer, Double, Boolean, NULL, Strings, Arrays, Objects, Resources

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – VARIÁVEIS

Atribuição de variáveis

```
$pi = 3 + 0.14159; // aproximadamente
```

Reatribuição de variáveis

```
$a_minha_variavel= "Isto deveria ser um número, vamos alterar";
$ a_minha_variavel = 5;
```

Como não é obrigatório declarar o tipo, não têm valor pré-definido

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – VARIÁVEIS

- Alcance (Scope)
  - Variáveis locais
  - Variáveis parâmetros de funções
  - Variáveis globais
  - Variáveis estáticas

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – VARIÁVEIS LOCAIS

```
<?
$x = 4;
function atribuir_x () {
  $x = 0;
  print "\$x dentro da função $x.";
}
atribuir_x();
print "\$x fora da função é $x.";
?>
```

Vai imprimir o resultado:

```
$x dentro da função é 0.
$x fora da função é 4.
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – VARIÁVEIS PARÂMETROS DE FUNÇÃO

```
<?
// multiplicar um valor por 10 e retornar
function multiplicar ($valor ) {
 $ valor = $valor * 10;
 return $valor;
}
$retval = multiplicar (10);
Print "O valor de retorno é $retval\n";
?>
```

Vai imprimir o resultado:

```
O valor de retorno é 100.
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – VARIÁVEIS GLOBAIS

```
<?
$variavel= 15;
function incrementar() {
   GLOBAL $variavel;
   $variavel ++;
   print "O valor da variável é $variavel ";
}
Incrementar();
?>
```

Vai imprimir o resultado:

```
O valor da variável 16.
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – VARIÁVEIS ESTÁTICAS

```
<?
function acompanhar_contagem() {
 STATIC $contar= 0;
 $contar ++;
 print $contar;
 print "";
}
acompanhar_contagem();
acompanhar_contagem();
acompanhar_contagem();
?>
```

Vai imprimir o resultado:

```
1
2
3
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – CONSTANTES

- Não têm o $ como prefixo
- São em maiúsculas por convenção
- Podem ser obtidas com a função `constant()`

```
<?php
define("MIN_TAMANHO", 50);
echo MIN_TAMANHO;
echo constant("MIN_TAMANHO"); //mesmo resultado da linha anterior
?>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – CONSTANTES

Diferenças para variáveis:

- Não podem ser definidas por simples declaração, têm que ser instanciadas com a função `define()`
- Podem ser definidas e acedidas de qualquer lado independentemente do alcance
- Uma vez definidas não podem ser redefinidas ou indefinidas

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – CONSTANTES MÁGICAS

Nome	Descrição
__LINE__	Linha atual do ficheiro
__FILE__	Caminho completo e nome do ficheiro […]
__FUNCTION__	O nome da função […]
__CLASS__	O nome da classe […]
__METHOD__	O nome do método da classe […]

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – CONVERSÕES

Não são declarados tipos

```
$primeiro_numero = 55.5;
$segundo_numero = "Não é um número de todo";
```

Conversões automáticas de tipo

```
$pi = 3 + 0.14159;
```

Tipos atribuídos por contexto

```
$sub = substr(12345, 2, 2);
print("A substring é $sub<BR>");
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – TIPOS

Integers são números inteiros sem parte decimal

Doubles são números reais com parte decimal

Booleans são valores boleanos, apenas têm dois valores possíveis: TRUE ou FALSE

NULL é um tipo especial que apenas tem um valor: NULL

Strings são sequências de caracteres ou palavras

Arrays são grupos ou coleções indexadas de valores

Objects referem-se instâncias de classes definidas pelo utilizador, que podem conter outros valores e funções específicas da classe

Resources são variáveis especiais que armazenam referências para recursos externos ao PHP, como ligações a bases de dados)

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – SAÍDAS

Echo

A maneira mais simples de imprimir um texto:

```
echo "Isto vai imprimir uma mensagem no navegador.";
echo("Isto vai imprimir uma mensagem no navegador.");
```

Print

Similar ao echo, embora apenas aceite um argumento e retorne um valor

```
print("3.14159"); // imprime um texto
print(3.14159); // imprime um número
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – ARRAYS PRÉ-DEFINIDOS

Aceder a variáveis a partir de uma página:

- Dependendo do método de submissão
  - `$_GET["<nome>"]`
  - `$_POST["<nome>"]`
- Independente do método
  - `$_REQUEST["<nome>"]`

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – EXEMPLO DE PÁGINA

```
<form action="reply1.php" method="GET">
  <p><input type="text" name="a_minha_cor"></p>
  <p><input type="submit"></p>
</form>
<h1>Exemplo de resposta PHP</h1>
<p>Escreveu a seguinte cor:
   <?php echo ($_GET["a_minha_cor"]);>
</p>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – OPERADORES DE COMPARAÇÃO

Operador	Nome	Comportamento
==	Igual	…
!=	Diferentes	…
<	Maior que	…
>	Menor que	…
<=	Menor ou igual	…
>=	Maior ou igual	…
===	Idêntico	Verdadeiro se os argumentos são iguais e do mesmo tipo, falso caso contrário

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – ESTRUTURAS DE CONTROLO

if (teste boleano)
else
elseif (teste boleano)

```
if ($dia== 5)
  print("Cinco!<BR>");
elseif ($dia == 4)
  print("Quatro!<BR>");
else
  print("Outro qualquer<BR>");
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – ESTRUTURAS DE CONTROLO

switch(expressão)

```
switch($dia)
{
   case 5:
    print("Cinco!<BR>");
    break;
   case 4:
    print("Quatro!<BR>");
    break;
   default:
    print("Outro qualquer<BR>");
}
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – ESTRUTURAS DE REPETIÇÃO

while (condição)

```
while (FALSE)
  print("Isto nunca irá ser imprimido.<BR>");
```

do (afirmação)…while (expressão)

for (expressão inicial;condição final;incremento)

```
  print("<TR>");
  print("<TH> </TH>");
  for($Contador = $numero_inicial;$count_1 <=
  $numero_final;$contador++)
    print("<TH>$contador </TH>");
  print("</TR>");
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – FUNÇÕES

A sintaxe normal para chamada de funções é:

```
nome_da_funcao(expressão1, expressão2, …, expressão n);
```

Funções nativas PHP

```
sqrt(9); //raiz quadrada de 9
rand(10, 10 + 10); //número aleatório entre 10 e 20
strlen("Isto tem 22 caracteres"); //retorna o comprimento do texto
pi(); //returna o valor aproximado de pi
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – FUNÇÕES

Para definir funções tem que se seguir a seguinte forma:

```
function nome_da_funcao($argumento1, $argumento2, …)
{
   bloco de instruções
}
```

- Palavra-chave `function`
- Nome da função
- Lista de parâmetros de entrada
- Corpo da função, delimitado por chavetas

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – FUNÇÕES

As funções apenas podem ser usadas depois de definidas

Para reutilizar funções em todo um servidor, pode-se usar:

```
include "as_minhas_funcoes.inc";
require "as_minhas_funcoes.inc";
```

Caso apenas pretendamos utilizar uma vez:

```
include_once($_SERVER['DOCUMENT_ROOT']."/caminho/para/as_minhas_funcoes.inc");
require_once($_SERVER['DOCUMENT_ROOT']."/caminho/para/as_minhas_funcoes.inc");
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – PASSAGEM DE DADOS

O protocolo HTTP é *stateless*, ou seja, cada pedido é independente, não tem memória nem sabe a identidade do cliente

Por vezes necessitamos de passar dados de uma página para outras

O PHP retém a variável descartada nas mudanças de página e fazem uso dela

O PHP é extremamente bom nesta gestão, facilitando as tarefas de passagem de dados

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – PASSAGEM DE DADOS

O método GET passa argumentos de uma página para a seguinte utilizando a query em URL

Quando usado para lidar com as páginas, o GET acrescenta os nomes das variáveis e valores ao URL

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – PASSAGEM DE DADOS

```
<HTML>
 <HEAD>
    <TITLE>Um exemplo do método GET, parte 1</TITLE>
 </HEAD>
<BODY>
<FORM ACTION=”sports.php” METHOD=”GET”>
 <P>Selecione o seu desporto favorito:<BR>
 <SELECT NAME=”Desporto”>
  <OPTION VALUE=”Basebol”> Basebol </OPTION>
  <OPTION VALUE=”Basquetebol”> Basquetebol </OPTION>
  <OPTION VALUE=”Futebol”> Futebol </OPTION>
  <OPTION VALUE=”F1”> F1 </OPTION>
  <OPTION VALUE=”Tenis”> Tenis </OPTION>
 </SELECT>
 <P><INPUT TYPE=”submit” NAME=”Submeter” VALUE=”Favorito”></P>
</FORM>
</BODY>
</HTML>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – PASSAGEM DE DADOS

Quando o botão Submeter (tipo `submit`) é clicado, o navegador junta:

- O URL dentro do atributo `action` da página
- Um ponto de interrogação `?`
- O nome de uma variável, o sinal igual que correspondem ao atributo valor
- Um `&` e o próximo par de nome/variável separador por `&`

```
http://<servidor>/sports.php?Desporto=Futebol&Submeter=Favorito
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – PASSAGEM DE DADOS

Após a submissão e já na nova página, o PHP recolhe as variáveis e coloca-as dentro da variável especial $_GET

```
<HTML>
 <HEAD>
   <TITLE>Um exemplo do método GET, parte 2</TITLE>
 </HEAD>
<BODY>
<P>Identificou o seu desporto favorite como:
<?php echo $_GET['Desporto']; ?>!</P>
</BODY>
</HTML>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – PASSAGEM DE DADOS

Cada item submetido pelo método GET é acedido pelo array $_GET

Cada item submetido pelo método POST é acedido pelo array $_POST

A sintaxe para referenciar um item no array global é simples e consistente: $_ARRAY_NAME['nome_do_indice'], onde:

- Nome do índice é a parte do nome que pertence ao par nome/valor (para o método GET) ou
- Nome do índice é um nome de um campo HTML (para o método POST)

No exemplo anterior, $_GET['Desporto'] indica o valor do formulário chamado 'Desporto' enviado pela operação GET

Neste caso o $_POST['Desporto'] está indefinido porque nenhuns dados foram submetidos anteriormente

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – PASSAGEM DE DADOS

Problemas

- O método GET não é adequado para gerir credenciais de entrada, as variáveis utilizador e palavra-chave são visíveis
- Cada submissão é guardada no registo do servidor, dados incluídos
- Armazenar dados numa variável de ambiente do servidor está limitado ao tamanho de 255 caracteres

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – PASSAGEM DE DADOS

Outros usos para GET

```
estrutura_veiculos.php
  tubos.inc
sistemas_mecanicos.php
  travagem.inc
sistemas_eletricos.php
  sensor_chuva.inc
corrida.php
  estrategias_corrida.inc
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – PASSAGEM DE DADOS

```
<BODY>
<TABLE BORDER=0 CELLPADDING=0 WIDTH=”100%“>
<TR>
<TD ALIGN=CENTER VALIGN=TOP>
<A HREF=”sistemas_mecanicos.php?Nome=travagem”>
 <B>Travagem</B></A><BR>
 <A HREF=”sistemas_mecanicos.php?Nome=direcao”>
 <B>Direcao</B></A><BR>
 <A HREF=”sistemas_mecanicos.php?Nome=suspensao”>
 <B>Suspension</B></A><BR>
 <A HREF=”sistemas_mecanicos.php?Nome=pneus”>
 <B>Pneus e rodas</B></A><BR>
</TD>
<TD ALIGN=LEFT VALIGN=TOP>
<?php include($_GET[‘Nome’] . “inc”); ?>
</TD></TR></TABLE>
</BODY>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – PASSAGEM DE DADOS

Utilizando o POST

```
<?php
if (!IsSet($_POST['Submit']) || $_POST['Submit'] != 'Calcular'){
 $_POST['IdadeAtual'] = "";
 $_POST['IdadeReforma'] = "";
 $_POST['Contrib'] = "";
 $Total = 0;
 $GanhoAnual = 7;}
else {
      $GanhoAnual = $_POST['GanhoAnual'];
      $Years = $_POST['IdadeReforma'] - $_POST['IdadeAtual'];
      $ContagemAnos = 0;
      $Total = $_POST['Contrib'];

      while ($ContagemAnos <= $Anos) {
      $Total = round($Total * (1.0 + $GanhoAnual) + $_POST['Contrib']);
      $ContagemAnos = $ContagemAnos + 1;}
}
?>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – PASSAGEM DE DADOS

Utilizando o POST

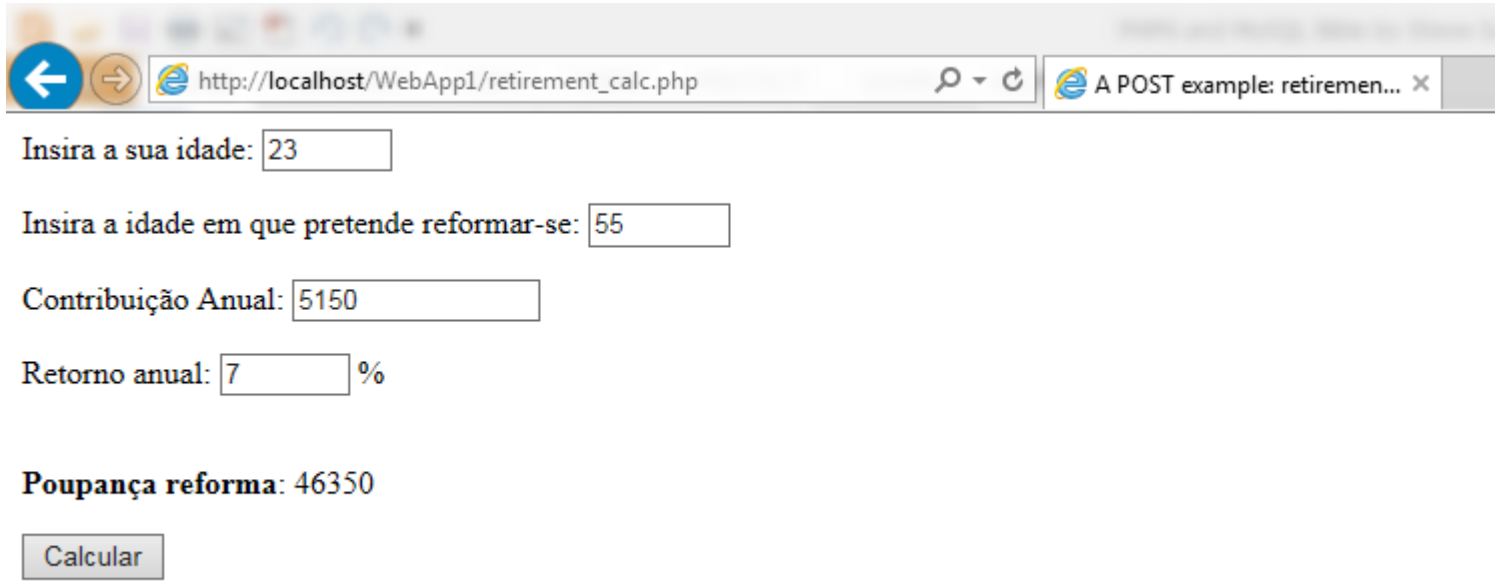
```
<FORM ACTION="<?php echo $_SERVER['PHP_SELF']; ?>"
METHOD="POST">
<P>Insira a sua idade:
<INPUT TYPE="text" SIZE=5 NAME="IdadeAtual" VALUE="<?php echo $_POST['IdadeAtual']; ?>">
<P>Insira a idade em que pretende reformar-se:
<INPUT TYPE="text" SIZE=6 NAME="IdadeReforma" VALUE="<?php echo $_POST['IdadeReforma']; ?>">
<P>Contribuição Anual:
<INPUT TYPE="text" SIZE=15 NAME="Contrib" VALUE="<?php echo $_POST['Contrib']; ?>">
<P>Retorno anual:
<INPUT TYPE="text" SIZE=5 NAME="GanhoAnual" VALUE="<?php echo $GanhoAnual; ?>"> % <BR><BR>
<P><B>Poupança reforma</B>: <?php echo $Total; ?>
<P><INPUT TYPE="submit" NAME="Submit" VALUE="Calcular">
</FORM>
</BODY>
</HTML>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – PASSAGEM DE DADOS

Utilizando o POST



SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – PASSAGEM DE DADOS

Variáveis Super Globais (PHP 6)

- Automaticamente disponíveis em todo o lado
- $_GET
- $_POST
- $_COOKIE
- $_ENV
- $_SERVER

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – STRINGS

Sequências de caracteres tratadas como um todo

A maneira mais fácil de usar é com plicas ou aspas:

```
$a_minha_string = 'Uma string' // usada normalmente
$outra_string = "Outra string" // usada para interpolar valores ou variáveis no texto
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – STRINGS

Interpolação

```
$desporto = 'vóleibol';
$plan = "Vou jogar $desporto nas férias"; // certo
$desporto3 = 'basquete';
$plano1 = "Vou jogar $desportobol nas férias"; // errado
$plano1 = "Vou jogar {$desporto1}bol nas férias"; // certo
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – STRINGS

Índices

Pode-se aceder aos caracteres dos índices das strings com chavetas

```
$a_minha_string = "Duplos";
for ($indice = 0; $indice < 6; $indice++) {
    $string_a_imprimir = $a_minha_string{$indice};
    print("$string_a_imprimir$string_a_imprimir");
}
```

Resulta na impressão de `DDuuppllooss`

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – STRINGS

Operadores

Concatenação .

```
$uma_afirmacao = "Quero descrever uma coisa ";
$outra_afirmacao = " E ainda outra coisa";
print($uma_afirmacao . "..." . $outra_afirmacao);
```

Concatenação e atribuição .=

```
$a_minha_string .= $nova_adicao;
```

Atribuição multilinha *Heredoc* <<<

```
$a_minha_string = <<< EOT
Vamos colocar todo este texto dentro de uma única string,
e só quando aparecer o character especial acaba EOT;
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – STRINGS

Funções: verificação, comparação e pesquisa

Função	Descrição curta
strlen()	Dado 1 argumento, devolve o comprimento do mesmo
strpos()	Dados 2 argumentos, devolve a posição do texto de pesquisa no texto a pesquisar
strrpos()	Idêntica à anterior, mas começa a pesquisa do final
strcmp()	Dados 2 argumentos, retorna 0 se são ambos equivalentes, caso contrário retorna negativo
strcasecmp()	Idêntica à anterior, mas não considera diferença de maiúsculas/minúsculas
strstr()	Dados 2 argumentos, procura se o primeiro está contido no segundo
strchr()	Idêntica à anterior, mas para caracteres
stristr()	Idêntica à anterior, sendo a comparação independente

PHP – STRINGS

Funções: verificação, comparação e pesquisa

```
$short_string = "Esta string tem 29 caracteres";
print("It does have " . strlen($short_string) . " characters");

$twister = "Peter Piper picked a peck of pickled peppers";
print("A localização de 'p' é " . strpos($twister, 'p') .'<BR>');

$string_a_pesquisar = "showsuponceshowsuptwice";
$string_a_encontrar = "up";
print("Resultado de $string_a_encontrar" . strstr($string_a_pesquisar,
$string_a_encontrar) . "<br>");
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – STRINGS

Funções: substrings e substituição

Função	Descrição curta
chop() ou trim()	Retorna o argumento com os espaços do final cortados
ltrim()	Retorna o argumento com os espaços do lado esquerdo cortados
Trim()	Retorna o argumento com os espaços de ambos os lados cortados
str_replace()	Utilizado para substituir uma string por outra string
substr_replace()	Coloca uma string num índice específico de outra string
str_repeat()	Repete uma string o número de vezes do argumento inteiro

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – STRINGS

Funções: substrings e substituição

```
echo(substr("Retirar o que é preciso e deixar o resto para trás",7, 40));


$primeira_edicao = "Burma é similar à Zâmbia pelo menos num aspeto.”;
$segunda_edicao = str_replace("Burma", "Myanmar", $primeira_edicao);


print(str_repeat("olá ", 3));
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – STRINGS

Funções: maiúsculas/minúsculas

Função	Descrição curta
strtolower()	Retorna o argumento com todos os elementos em minúsculas
strtoupper()	Retorna o argumento com todos os elementos em maiúsculas
ucfirst()	Capitaliza a primeira letra da string
ucwords()	Capitaliza a primeira letra de cada palavra

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – STRINGS

Impressão

```
print()/echo()
printf() // impressão formatada
sprintf() // impressão formatada que devolve a string
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – STRINGS

- Impressão
  - Depois da %, existem 6 elementos para fazer a conversão
    - Um sinal opcional usado para números para indicar se o número é negativo
    - Um caracter opcional de tabulação que é 0 ou espaço
    - Um caracter opcional - para justificar texto à esquerda ou direita
    - Um comprimento mínimo opcional para indicar quantos espaços o valor deve ter no mínimo
    - Um especificador opcional de precisão . seguido de um número
    - Um caracter indicando o tipo de dados do valor: f (double), s (string), d (integer), etc.

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – STRINGS

Impressão

```
<?php
$value = 3.14159;
printf("%f,%10f,%-010f,%2.2f\n",
$value, $value, $value, $value);
?>
```

Devolve:

```
3.141590, 3.141590,3.141590000000000, 3.14
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – TIPOS

Uma variável pode ser de um certo tipo, mas os tipos PHP não são explicitamente definidos

O tipo de variável é determinado pelo contexto em que a variável é usada

Isso significa, portanto, que as variáveis se tornam um tipo quando lhe é atribuído um valor que tem um tipo

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – TIPOS

O PHP converte automaticamente de um tipo para outro, o que pode causar resultados inesperados

Por exemplo, a conversão de *false* para uma *string* pode ter resultado inesperado

```
$value = false;
echo “O valor é $value";
// vai escrever: O valor é
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – TIPOS

Para resolver estes tipos de problemas, o PHP precisa ser instruído como lidar com a variável

```
$value = false;
echo “O valor é " . (int)$value;
// vai escrever: O valor é 0
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – TIPOS

Como podemos ver, para converter uma variável precisamos de usar tipo que você deseja converter antes da variável entre parênteses

Aplica-se então esse tipo à variável que está a ser utilizada

Isto pode ter alguns resultados inesperados

- Por exemplo, converter uma fração num inteiro arredondará a fração para o número inteiro mais próximo

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – TIPOS

Em PHP podemos adicionar uma frase e um número para formar outro número

Isto pode levar a resultados inesperados, é necessário ter cuidado para lidar com isso

```
$value = 1 + "4.5";
// is float with the value 5.5
$value = 1 + "tom";
// is integer with the value 1
$value = 1 + "2 Green Bottles";
// is integer with the value 3
$value = 1 + "Green Bottles 2";
// is integer with the value 3
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – TIPOS

Quando se adiciona uma string a um número se a primeira parte da string é um número isto é então adicionado em conjunto

Quaisquer outros números são ignorados

Se adicionarmos uma string com um float nele a um inteiro o resultado é um float

Novamente a maneira correta de resolver esse problema é detetar e ou converter a variável para o tipo correto

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – TIPOS

Um tipo de variável pode ser identificado usando as funções abaixo

Simplesmente passa-se a variável para a função necessária e ele retornará *true* se ele for desse tipo: esta é uma técnica de programação defensiva

```
$var = "hello";
//Returns true if a variable is a BOOLEAN
is_bool($var)
//Returns true if a variable is a STRING
is_string($var)
//Returns true if a variable is a NUMERIC
STRING
is_numeric($var)
//Returns true if a variable is an INTEGER
is_int($var)
//Returns true if a variable is an ARRAY
is_array($var)
//Returns true if a variable is an OBJECT
is_object($var)
//Returns true if a variable is NULL
is_null($var)
//Returns true if a variable is a FLOAT
is_float($var)
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – CONFIGURAÇÃO

Sempre que desenvolvemos uma aplicação, provavelmente precisaremos de guardar os valores de configuração

Podemos declarar esses valores de configuração estaticamente, mas isso pode tornar o código menos robusto

- Cada vez que necessitarmos de atualizar valores temos que alterar todas as variáveis

O que pode ser um grande problema à medida que o programa vai crescendo

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – CONFIGURAÇÃO

A maneira mais fácil de contornar isto é utilizando a capacidade de configuração incorporada PHP

Para tal basta criar o nosso próprio ficheiro de configuração, neste caso chamado *config.ini*:

```
DatabaseName = "phprocks"
Hostname = "localhost"
Username = "test"
Password = "password123"
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – CONFIGURAÇÃO

Depois pode ser analisado pela função *parse_ini_file* da seguinte forma:

```
$config = parse_ini_file("config.ini");
// Vamos imprimir o nome da BD
echo $config['DatabaseName'];
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – CONFIGURAÇÃO

Se tivermos muitos valores de configuração, isso pode ser dividido em seções para que o ficheiro de configuração ficar organizado:

```
[Database]
DatabaseName = "phprocks"
Hostname = "localhost"
Username = "test"
Password = "password123"
[Site]
SiteTitle = "PHP Rocks! - "
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – CONFIGURAÇÃO

Para utilizar o ficheiro de configuração com seções, apenas precisamos especificar um parâmetro extra na função *parse_ini_file*, como indicado em abaixo:

```
$config = parse_ini_file("config.ini", true);
// Vamos imprimir o nome da BD
echo $config['Database']['DatabaseName'];
```

Isto retorna uma array multidimensional, razão pela qual você devemos primeiro especificar a seção desejada

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – FICHEIROS EXTERNOS

Muitas vezes necessitamos de incluir algo que se repete em várias páginas

Também é frequente querermos colocar algo mais modular, como um controlo, guardado separadamente

A instrução *include* (ou *require*) utiliza todo o texto/código que existe no ficheiro especificado e o copia para o ficheiro que está a utilizar a instrução *include*

Incluir ficheiros é muito útil quando desejamos incluir PHP, HTML ou texto em várias páginas de um site

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – FICHEIROS EXTERNOS

É possível inserir o conteúdo de um ficheiro PHP noutro ficheiro PHP (antes que o servidor o execute), com a instrução *include* ou *require*

As instruções *include* e *require* são idênticas, exceto em caso de falha:

- O *require* produzirá um erro fatal (E_COMPILE_ERROR) e interromperá o script
- O *include* apenas produzirá um aviso (E_WARNING) e o script continuará

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – FICHEIROS EXTERNOS

Portanto, se desejamos que a execução continue e mostre aos utilizadores a saída, mesmo que o ficheiro de inclusão esteja ausente, utilize a instrução *include*

Caso contrário, no caso de FrameWork, CMS, ou uma codificação de aplicativos complexos PHP, utilize sempre a instrução *require* para incluir um ficheiro de chave para o fluxo de execução

Isto ajudará a evitar comprometer a segurança e a integridade da aplicação, apenas no caso de um ficheiro estar acidentalmente ausente

SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – FICHEIROS EXTERNOS

Incluir ficheiros economiza muito trabalho

Isso significa que podemos criar um cabeçalho padrão, rodapé ou ficheiro de menu para todas as suas páginas da web

De seguida, quando o cabeçalho precisa ser atualizado, apenas é preciso atualizar o ficheiro de inclusão de cabeçalho:

```
include 'filename';
// ou
require 'filename';
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – FICHEIROS EXTERNOS

Exemplo:

footer.php

```
<?php
echo ”<p>Copyright &copy; ” . date(”Y”) . ”ua.pt</p>”;
?>
```

index.php

```
<html>
<body>
<h1>Welcome to my home page!</h1>
<p>Some text.</p>
<?php include 'footer.php';?>
</body>
</html>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática

PHP – FICHEIROS EXTERNOS

Exemplo:

menu.php

```
<?php
echo '<a href=”index.php”>Home</a>
<a href=”about.php”>About</a>
<a href=”clients.php”>Clients</a>';
?>
```

index.php

```
<html>
<body>
<div class=”menu”>
<?php include ‘menu.php';?>
</div>
<h1>Welcome to my home page!</h1>
<p>Some text.</p>
</body>
</html>
```


SEBASTIAO@DI.UBI.PT

FACULDADE ENGENHARIA
Departamento de Informática