UAlg ISE
UNIVERSIDADE DO ALGARVE
INSTITUTO SUPERIOR DE ENGENHARIA

Tecnologias Web

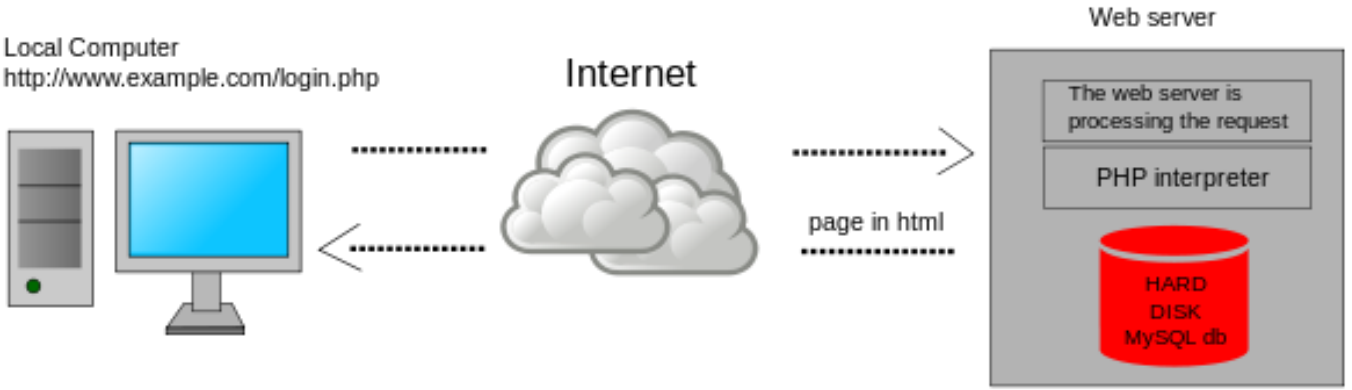
8ª AULA, 8 de abril

Roberto Lam, Instituto Superior de Engenharia,
Universidade do Algarve
rlam@ualg.pt

2023/24

Server Side Scripting

Server vrs Client processing



Compilado vrs Interpretado

http://en.wikipedia.org/wiki/Server-side_scripting

Server Side Scripting

Algumas linguagens (SSS):

ASP (*.asp)
ActiveVFP (*.avfp)
ASP.NET (*.aspx)
C via CGI (*.c, *.csp)
ColdFusion Markup Language (*.cfm)
Java via JavaServer Pages (*.jsp)
JavaScript using Server-side JavaScript (*.ssjs, *.js)
Lua (*.lp *.op)
Perl CGI (*.cgi, *.ipl, *.pl)
PHP (*.php) - Open Source Scripting
Python, e.g. via Django (*.py)
Ruby, e.g. Ruby on Rails (*.rb, *.rbw)
SMX (*.smx)
Lasso (*.lasso)
WebDNA (*.dna,*.tpl)
Progress WebSpeed (*.r,*.w)

PHP

Motivação

Gratuito,

Código aberto

Modular,

Multiplataforma (sistemas operativos),

Bases de dados: Oracle, Sybase, PostgreSQL, InterBase, MySQL, MSSQL, etc.

Protocolos: IMAP, SNMP, NNTP, POP3, HTTP, LDAP, XML-RPC, SOAP, REST.

PHP

Instalação PHP

http://php.net/manual/pt_BR/install.php

http://www.php.net/manual/pt_BR/tutorial.firstpage.php

```
<html>
 <head>
  <title>PHP Teste</title>
 </head>
 <body>
 <?php echo "<p>Olá Mundo</p>"; ?>
 </body>
</html>
```

http://localhost/exemplo1.php
http://10.11.143.247/~aXXXX/exemplo1.php

PHP

Sintaxe basica

Tags definição código PHP

Os recursos, incluido em HTML, que contenham código PHP no seu interior devem possuir as marcas: **<?php …. ?>**

```
<?php
    /* comentário */
    // linha comentário
    // as variáveis devem ser antecedidas do caracter $

?>
```

PHP

Tipos de dados

Escalares: **boolean**, **integer**, **float** e **string**.
Compostos: **array** e **object**.
Especiais: **resource, NULL e referência.**

Exemplo2.

```
<html>
<head><title>2º Exemplo</title></head>
<body>
<?php
$HR=true;
if($HR==true)
        echo “<hr>\n”;

$a=5.0;
$b=&$a; // b é uma variável referência à variável a
 ?>
</body
</html>
```

PHP

Lógico (boolean)

A conversão de um tipo distinto ao lógico pode ser convertido ao lógico através da conversão explícita.

```
$b=5;
$c=(bool) $b;
```

Valores considerados falso (pelo PHP):

integer	0
float	0.0
string	vazia
vector	sem elementos
objecto	sem elementos
NULL	

PHP

integer (inteiro)

A sua capacidade de armazenamento é 2147483647 (dependendo da arquitectura do sistema). Os inteiros podem ser especificados como decimal, octal e hexadecimal.

```
$b=124;   // decimal positivo
$c=-1234; // decimal negativo
$d=0123; // octal
$e=0x1A; // hexadecimal
```

float (real)

A sua capacidade de armazenamento é aproximadamente 1,80e+3082147483647 (64 bits, dependendo da arquitectura do sistema). Os reais podem ser especificados do seguinte modo:

```
$b=4.124; // real 4.124
$c=1.2e3; // real 1200
$d=7E-10; // real 0.0000000007
```

PHP

string (vector de caracteres)

O tipo string, ou vector de caracteres, podem armazenar caracteres de 8 bits. As cadeias de caracteres não tem limite de comprimento e podem ser especificadas entre plicas, aspas e marcas. Tal como a linguagem C, as strings podem possuir caracteres especiais de formatação

Sequência	Significado
\n	Nova linha
\r	Caracter de retorno de linha
\t	Caracter de tabulação
\\	Barra invertida
\$	Cifrão
\”	Aspas

```
$MeuNome=”Ricardo”;   // aspas
$MeuApelido=’Silva’;      // plicas

printf(“Este é o meu nome:$meuNome $MeuApelido”;
```

PHP


```
// exemplo de uma string delimitada por marca

$str=<<<FIM
Exemplo de uma string
Limitada por
uma
marca
FIM;
```

Acesso às strings (vector de caracteres)

Operador de índice **[ ]**. O primeiro caracter da string é referenciado pela posição 0. Existem um conjunto de funções de manipulação de *strings* (ver manual do PHP).

PHP

Exemplo3.

```
<html>
<head><title>3º Exemplo</title></head>
<body>
<?php
$str=”String de exemplo”;
echo “o primeiro caracter é: $str[0]”;
echo “o terceiro caracter é: $str[2]”;
echo “o último caracter é: $str[strlen($str)-1]”;
?>
</body
</html>
```

Arrays (vectores)

```
$a=array(1=>’um’, 2=>’dois’, 3=>’três’);

$a=array(’cor’=>’verde’, ’forma’=>’oval’, ’nome’=>’melão’, 4);
```

É idêntico a:

```
$a[’cor’]= ’verde’; $a[’sabor’]= ’oval’; $a[’nome’]= ’melão’;
$a[]= 4; // atribuida 1ª chave inteira vazia
```

PHP

Objectos (classes)

Exemplo4.

```
<?php
class teste{
function testar(){
        echo “Em teste”;
}
}
$x= new teste;
$x->testar();
?>
```

Resources (recursos)

```
$db=mysql_connect($srv, $user, $pass);
$interpretadorXML=xml_parser_create();
```

PHP

Variáveis predefinidas

Vector	Descrição
$GLOBALS	Contém uma referência para cada var. que está disponível de forma global ao prog. em execução
$_SERVER	Var's referentes ao servidor e algumas do cliente, nomeadamente IP e porto do cliente.
$_GET	Var's obtidas através do método GET, HTTP
$_POST	Var's obtidas através do método POST, HTTP
$_COOKIE	Var's obtidas a partir dos cookies via HTTP
$_FILES	Var's referentes à manipulação de ficheiros
$_ENV	Var's ambiente
$_REQUEST	Var's fornecidas pelo cliente
$_SESSION	Var's associadas à sessão

PHP

Variáveis globais

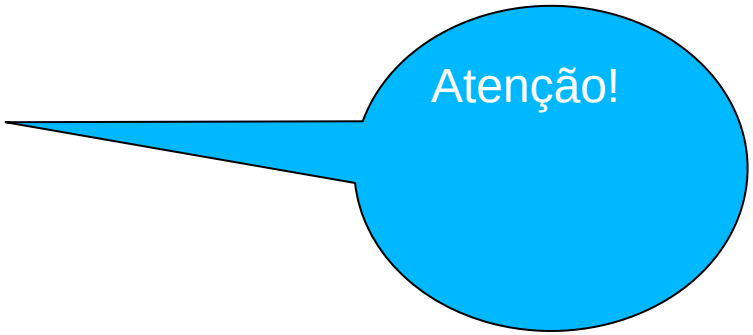
Exemplo5.

```
<?php
$a=2;
$b=5;

function soma1(){
global $a, $b;
        $b=$a+$b;
}

function soma2(){
        $GLOBALS[‘b’]=$GLOBAL[‘a’]+ $GLOBAL[‘b’];
}

soma1();
echo $b.’****’;
soma2();
echo $b;
?>
```



PHP

Variáveis de variáveis

```
$a='Olá';
$$a='Mundo';  // $($a)
echo "$a $Olá"; // "$a $($a)
```

Variáveis exteriores ao PHP

```
<form action="dados.php" method="post">
    Nome: <input type="text" name="NomeUt"> <br/>
    Correio <input type="text" name="email"> <br/>
    <input type="submit" name="Enviar">
</form>
```

dados.php

```
<?php
echo $_POST['NomeUt'];
echo $_POST['email'];
?>
```

PHP

Sintaxe PHP

Expressões

atribuição

```
$b=6.55;
$a=$b++;
$c=$a+=10;
$d=duplica($a);
$d= ($a=4)+ 20;
```

comparação

```
$b>6.55
$a>=$b++
$c==$a
```

Operadores

Operadores aritméticos

*****, **+**, **-**, **/**, **%**.

* multiplicação
+ soma
- subtracção
/ divisão
% módulo

PHP

Operadores bit a bit

&, **|**, **^**, **~**, **<<**, **>>**.

& conjunção
| disjunção inclusiva
^ disjunção exclusiva
~ negação
<< deslocamento à esquerda
>> deslocamento à direita

Operadores relacionais

==, **===**, **!=**, **<>**, **!==**, **>**, **<**, **>=**, **<=**

== igual
=== idêntico (mesmo tipo)
!= diferente
<> diferente
!== não idêntico
> maior
< menor
>= maior ou igual
<= menor ou igual

PHP

Operadores lógicos

and, **or**, **xor**, **!**, **&&**, **||**

and conjunção
or disjunção inclusiva
xor disjunção exclusiva
! negação
&& conjunção
|| disjunção inclusiva

Operadores de strings

O PHP possui um operador de concatenação de strings “.”. Para a manipulação de *strings* existem conjunto de funções (ver manual PHP).

```
$b=”Olá”;
$b=$b . “ “ . “Mundo”;
echo $b;

$a=”Olá”;
$a .= “ Mundo”;
```

http://www.php.net/
http://www.phpbuilder.com/
http://www.phpfreaks.com/

PHP

Operadores de vectores

O operador **+** permite a união de dois vectores onde as chaves duplicadas não são sobrepostas.

```
$a=array('a'=>'maçã', 'b'=>'banana');
$b=array('a'=>'pera', 'b'=>'morango', 'c'=>'cereja');
$c=$a+$b;
var_dump($c); // visualiza o conteúdo do vector C
```

Estruturas de controlo

```
if( expressão1){ .. }
if( expressão1){ .. }else{...}
if( expressão1){ .. }elseif( expressão2){...}else{...}
```

```
if($a>$b){
        printf("%s"," a é maior que b");
elseif($a==$b){
                printf("%s"," a é igual a b");
        }else{
                printf("%s"," a é menor que b");
}
```

PHP

switch(expressão1){ case : .. }

```
switch($op){
        case 1:
printf(“i é igual a 1”);
break;
        case 2:
printf(“i é igual a 2”);
break;
case 3:
printf(“i é igual a 3”);
break;
        }
```

while(expressão1){ .. }

```
$a=1;
while($a<=10){
        print($a++);
      }
```

PHP

do{ ....}while(expressão1);

```
$a=1;
do{
        print($a++);
}while($a<=10);
```

for(expressão1; expressão2; expressão3){ ....}

```
for($i=1; $i<=10; $i++){
        echo $i;
}
```

foreach($vector as $valor){ ....}
foreach($vector as $chave =>$valor){ ....}

```
$a=array(1, 2, 3, 4);

foreach($a as $v){
        echo “Valor: $v”;
}
```


```
$a=array('um'=>1,'dois'=> 2,'três'=> 3,'dezassete'=> 17);

foreach($a as $k =>$v){
        echo "Valor \$a[$k]: $v.\n";
}
```

instrução break

Interrompe a execução do fluxo.

instrução require e include

Possibilitam a inclusão de ficheiros externos ao ficheiro em processo.

PHP

Funções

```
function exemplo($arg1, $arg2, $arg3,.....,$argN){
        echo “função de exemplo\n”;

        $valor=$arg2+$arg2;
        return $valor;
}
```

As funções só podem devolver uma única variável, podendo ela ser de qualquer tipo, incluindo **array** ou **objecto**.

PHP

Argumentos por referencia

```
function str_add(&$cadeia_caracteres){

        $cadeia_caracteres.=“ função de exemplo com arg. ref.”;

 }

$cad=”isto é um exemplo”;
str_add($cad);
echo $cad;
```

Argumentos por defeito

```
function str_a($org, $tipo=”com leite”){

        return “fazer uma chávena de café de $org, $tipo”;

}

echo str_a(“de Timor”);
echo str_a(“do Brasil”, “expresso”);
```

Hacking in JavaScript

www.hackertest.net

Tentar aceder ao nível 5

Perguntas?