

Material para a formación profesional inicial

Familia profesional	IFC	Informática e comunicacións
Ciclo formativo	CSIFC01	Administración de sistemas informáticos en rede
Grao		Superior
Módulo profesional	MP0376	Implantación de aplicacións web
Unidade didáctica	UD05	Introdución á linguaxe PHP e á súa sintaxe básica.
Actividade	A03	Sintaxe básica da linguaxe PHP.
Autores		Víctor Alfredo Pascual Vázquez

© 2013 Xunta de Galicia.

Consellería de Cultura, Educación e Ordenación Universitaria.

Este traballo foi realizado durante unha licenza de formación retribuída pola Consellería de Cultura, Educación e Ordenación Universitaria e ten licenza Creative Commons BY-NC-SA (recoñecemento - non comercial - compartir igual). Para ver unha copia desta licenza, visitar a ligazón <http://creativecommons.org/licenses/by-nc-sa/3.0/es/>.

Índice

1.Introdución á linguaxe PHP e á súa sintaxe básica 3

1.1Introdución	3
1.2Actividade	3
Sintaxe básica da linguaxe	3
Comentarios	4
Variables e identificadores	4
Variables predefinidas	5
Constantes	5
Tipos de datos	5
Operadores e expresións	7
Arrays e cadeas	9
Cadeas	9
Arrays	10

2.Materials 12

2.1Textos de apoio ou de referencia	12
2.2Recursos didácticos	12

1. Introducción á linguaxe PHP e á súa sintaxe básica

1.1 Introducción

Na presente actividade trátaranse os seguintes conceptos e manexo de destrezas sobre a linguaxe PHP:

- Descrición da sintaxe básica da linguaxe.
- Uso de variables e identificadores.
- Descrición dos tipos de datos.
- Manexo de operadores e expresións.
- Utilización de cadeas e arrays.

1.2 Actividade

Sintaxe básica da linguaxe

Os scripts PHP son executados no servidor web, e o HTML resultante é enviado de volta ao cliente. Un script PHP pode ser colocado en calquera parte do documento HTML. O script PHP comeza ca etiqueta de apertura `<?php` en termina coa etiqueta de peche `?>`. Todo o que estea fora destas etiquetas de PHP será ignorado polo intérprete.

Estrutura xeral dun script PHP

```
<?php
    // Código PHP
?>
```

A extensión por defecto para os arquivos PHP é `.php`. Un ficheiro PHP contén normalmente etiquetas HTML e scripts PHP. Neste caso trataríase dun ficheiro PHP de contido mixto, aínda que poden conter soamente código PHP. O exemplo que segue contén un exemplo dun ficheiro simple PHP, cun script que envía o texto Ola Mundo! ao navegador.

Estrutura dun script PHP dentro dun documento HTML

```
<!DOCTYPE html>
<html>
<body>

<h1>Miña primeira páxina PHP</h1>

<?php
    echo "Ola Mundo!";
?>

</body>
</html>
```

O intérprete PHP cando intercepta unha etiqueta de peche `</p>` comeza a imprimir calquera cousa que atope ata que dea con outra etiqueta de apertura (salvo que estea nunha instrución condicional, pero isto se verá máis adiante).

Exemplo de código mixto

```
<p>Isto vai ser ignorado por PHP e mostrado polo navegador.</p>
<?php echo 'Mentres que isto vai ser interpretado.'; ?>
<p>Isto tamén será ignorado por PHP e mostrado polo navegador.</p>
```

Para imprimir bloques de código grandes, é máis eficiente abandonar o modo intérprete de PHP para enviar todo o texto a través de `echo`.

Cada liña en PHP debe finalizar cun punto e coma, que é un separador que permite distinguir entre distintas instrucións.

Comentarios

Os comentarios permiten incrustar anotacións lexibles ao programador no código fonte. Os comentarios a nivel de liña comezan por `//` ou `#`. Os comentarios a nivel de bloque comezan por `/*` e terminan por `*/`. No seguinte exemplo amósanse os dous tipos de comentario:

Exemplo de comentarios en PHP

```
<?php
echo 'Isto é unha proba'; // Isto é un comentario estilo C++ dunha soa liña
/* Isto é un comentario multiliña
   e outra liña de comentarios */
echo 'Isto é outra proba';
echo 'Una proba final'; # Isto é un comentario estilo consola de una sola liña
?>
```

Variables e identificadores

Unha variable é un contedor para almacenar información, e se representa mediante un identificador. En PHP os identificadores de variable represéntanse cun signo dólar seguido do nome da variable. O nome da variable é sensible a minúsculas e maiúsculas. Os nomes das variables deben seguir unhas regras que consisten en que, basicamente, un nome de variable válido ten que comezar por unha letra ou un carácter de subliñado, seguido de calquera número de letras, números ou caracteres de subliñado. A continuación móstrase un exemplo de variables.

Exemplo de variables en PHP

```
<?php
$x=5;
$y=6;
$z=$x+$y;
echo $z;

$var = 'Roberto';
$Var = 'Juan';
echo "$var, $Var"; // imprime "Roberto, Juan"

$4site = 'aínda non'; // inválido; comeza cun número
$_4site = 'aínda non'; // válido; comeza cun carácter de subliñado
$äyte = 'mansikka'; // válido; 'ä' é ASCII (Estendido) 228
?>
```

A unha variable pódesele asignar un valor directamente ou o resultado de avaliar unha expresión. Por defecto, as variables sempre se asigna por valor, aínda que tamén poden asignarse por referencia, é dicir, a variable referencia a outra variable e os cambios na nova variable afecta a orixinal e viceversa.

Exemplo de variables por valor e por referencia

```
<?php
$foo = 'Bob';          // Asigna o valor 'Bob' a $foo
$bar = &$foo;          // Referencia $foo via $bar.
$bar = "O meu nome é $bar"; // Modifica $bar...
echo $bar;
echo $foo;             // $foo tamén se modifica.
?>
```

En PHP non hai instrución para declarar unha variable. A variable é creada no momento en que se usa por vez primeira. PHP converte a variable ao tipo de dato adecuando dependendo do valor que se lle asigne.

Non é necesario inicializar variables en PHP, así a todo, é unha boa práctica. As variables non inicializadas teñen un valor de acordo ao seu tipo dependendo do contexto en que son usadas.

Variables predefinidas

PHP proporciona unha gran cantidade de variables predefinidas a calquera script que se execute. Moitas de estas, non obstante, dependen do servidor que se estea executando, versión ou configuración deste. Estás variables iranse introducindo ao longo das seguintes actividades.

Constantes

Unha constante é un valor que non pode ser alterado durante a execución dun programa. Ten asignado un identificador que segue as mesmas regras que os identificadores de variable, salvo que non comeza polo signo \$. Habitualmente, o identificador das constantes escríbese en maiúsculas. As constantes defínense mediante a función define(), e a partir da versión 5.3.0 de PHP mediante a palabra reservada const. No exemplo seguinte defínense e utilízanse dúas constantes.

Definición e uso de constantes

```
<?php
define("CONSTANT", "Ola mundo.");
echo CONSTANT; // mostra "Hola mundo."
echo Constant; // mostra "Constant" y provoca un error.
?>

<?php
// Funciona desde PHP 5.3.0
const CONSTANT = 'Ola Mundo';

echo CONSTANT;
?>
```

Tipos de datos

PHP é unha linguaxe debilmente tipada xa que o tipo da variable se corresponde co tipo do valor que almacena. Nunha linguaxe fortemente tipada sería necesario declarar o tipo e o nome da variable antes de poder utilizala.

PHP soporta oito tipos primitivos:

- Catro tipos escalares.
 - boolean: expresa un valor de verdade que pode ser TRUE ou FALSE.
 - integer: representa un enteiro, é dicir, un número do conxunto $Z = \{\dots, -2, -1, 0, 1, 2, \dots\}$.
 - float: números en punto flotante, tamén coñecidos como flotantes, dobres ou números reais. PHP utiliza o formato de dobre precisión IEEE 754.
 - string: representa series de caracteres. Estudaranse en detalle máis adiante.
- Dous tipos compostos.
 - array: representa un conxunto de valores accesibles mediante un índice ou clave. Estudaranse máis adiante en detalle.
 - object: representa obxectos.
- Dous tipos especiais.
 - resource: representa referencias a recursos externos.
 - null: representa unha variable sen valor. NULL é único valor posible do tipo null.

No seguinte código amósase a utilización dos devanditos tipos de datos.

Exemplo de uso dos tipos de datos

```
<?php
$foo = True; // asigna o valor TRUE a $foo
?>

<?php
$a = 1234; // número decimal
$a = -123; // un número negativo
$a = 0123; // número octal (equivalente a 83 decimal)
$a = 0x1A; // número hexadecimal (equivalente a 26 decimal)
?>

<?php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
?>

<?php
$txt="Ola mundo!";
echo $txt;
?>

<?php
$cars=array("Volvo","BMW","Toyota");
echo "Gústame " . $cars[0] . " , " . $cars[1] . " e " . $cars[2] . " .";
?>

<?php
class foo
{
    function do_foo()
    {
        echo "Facendo foo.";
    }
}

$bar = new foo;
$bar->do_foo();
?>

<?php
$var = NULL;
```

```
?>
```

Como xa se mencionou, PHP non soporta a declaración explícita de tipos na declaración de variables. O tipo da variable se determina polo contexto no que se emprega a variable, polo tanto se produce unha asignación automática de tipo. Existe a posibilidade de forzado de tipos, indicando o nome do tipo desexado entre parénteses antes da variable que se queira forzar. No seguinte código móstrase un exemplo disto.

Forzado de tipos

```
<?php
$foo = 10; // $foo é un integer
$bar = (boolean) $foo; // $bar é un boolean
?>
```

Operadores e expresións

Unha expresión é unha combinación de constantes, variables ou funcións (veranse máis adiante na seguinte actividade). En PHP case todo o que se escribe son expresións, e constitúen as distintas instrucións do código fonte. Expresións básicas poden ser de asignación, aritméticas ou lóxicas, dependendo dos operadores que se utilicen.

Un operador é un elemento de programa que se aplica a uno ou varios operandos nunha expresión ou instrución.

Os operadores aritméticos permiten construír expresións aritméticas. Amósanse na seguinte táboa.

Operadores aritméticos en PHP				
Operador	Nome	Descrición	Exemplo	Resultado
$x + y$	Suma	Suma de x e y	$2 + 2$	4
$x - y$	Resta	Diferenza entre x e y	$5 - 2$	3
$x * y$	Multipliación	Produto de x por y	$5 * 2$	10
x / y	División	Cociente de x entre y	$15 / 5$	3
$x \% y$	Módulo	Resto da división de x entre y	$5 \% 2$	1
$-x$	Negación	O oposto de x	-2	

Os operadores de asignación permiten expresións nas que se adxudica un valor a unha variable. Amósanse na seguinte táboa.

Operadores de asignación en PHP		
Asignación	Equivalente a	Descrición
$x = y$	$x = y$	O operando da esquerda obtén o valor da expresión da dereita
$x += y$	$x = x + y$	Suma
$x -= y$	$x = x - y$	Resta
$x *= y$	$x = x * y$	Produto
$x /= y$	$x = x / y$	División
$x \% = y$	$x = x \% y$	Módulo

Os operadores de incremento/diminución permiten aumentar/diminuír o valor dunha variable. Amósanse na seguinte táboa.

Operadores de incremento/diminución en PHP		
Operador	Nome	Descrición
<code>++ x</code>	Pre-incremento	Incrementa x nunha unidade e despois devolve o valor de x
<code>x ++</code>	Post-incremento	Devolve o valor de x e despois incrementa x nunha unidade
<code>-- x</code>	Pre-diminución	Decrementa x nunha unidade e despois devolve o valor de x
<code>x --</code>	Post-diminución	Devolve o valor de x e despois diminuír x nunha unidade

Os operadores de comparación permiten comparar valores e devolven verdadeiro ou falso. Amósanse na seguinte táboa.

Operadores comparación en PHP			
Operador	Nome	Descrición	Exemplo
<code>x == y</code>	Igual	Verdadeiro se x é igual a y	<code>5==8</code> devolve falso
<code>x === y</code>	Idéntico	Verdadeiro se x é igual a y, e ambos son do mesmo tipo	<code>5=== "5"</code> devolve falso
<code>x != y</code>	Distinto	Verdadeiro se x non é igual a y	<code>5!=8</code> devolve verdadeiro
<code>x <> y</code>	Distinto	Verdadeiro se x non é igual a y	<code>5<>8</code> devolve verdadeiro
<code>x !== y</code>	Non idéntico	Verdadeiro se x non é igual a y, ou ambos non son do mesmo tipo	<code>5!== "5"</code> devolve verdadeiro
<code>x > y</code>	Menor que	Verdadeiro se x é maior que y	<code>5>8</code> devolve falso
<code>x < y</code>	Maior que	Verdadeiro se x é menor que y	<code>5<8</code> devolve verdadeiro
<code>x >= y</code>	Maior ou igual que	Verdadeiro se x é maior ou igual a y	<code>5>=8</code> devolve falso
<code>x <= y</code>	Menor ou igual que	Verdadeiro se x é menor ou igual a y	<code>5<=8</code> devolve verdadeiro

Os operadores lóxicos permiten construír expresións lóxicas, que son aquelas que teñen como resultado verdadeiro ou falso. Amósanse na seguinte táboa.

Operadores lóxicos en PHP			
Operador	Nome	Descrición	Exemplo
<code>x and y</code>	And	Verdadeiro se tanto x como y son verdadeiros	<code>x=6</code> <code>y=3</code> <code>(x<10 and y>1)</code> devolve verdadeiro
<code>x or y</code>	Or	Verdadeiro se algún ou ambos son verdadeiros	<code>x=6</code> <code>y=3</code> <code>(x==6 or y==5)</code> devolve verdadeiro
<code>x xor y</code>	Xor	Verdadeiro se algún dos dous é verdadeiro, pero falso se os son ambos	<code>x=6</code> <code>y=3</code> <code>(x==6 xor y==3)</code> devolve falso
<code>x && y</code>	And	Verdadeiro se tanto x coma y son verdadeiros	<code>x=6</code> <code>y=3</code> <code>(x<10 && y>1)</code> devolve verdadeiro
<code>x y</code>	Or	Verdadeiro se algún ou ambos son verdadeiros	<code>x=6</code> <code>y=3</code> <code>(x==5 y==5)</code> devolve falso
<code>! x</code>	Not	Verdadeiro se x é falso	<code>x=6</code> <code>y=3</code> <code>!(x==y)</code> devolve verdadeiro



Realice a tarefa 1 correspondente á construción de expresións en PHP.

Arrays e cadeas

Mentres unha cadea representa unha secuencia de caracteres, un array representa un conxunto de elementos cuxo tipo non ten que ser carácter. Estudaranse ambos tipos a continuación debido á súa importancia e frecuente presenza nos códigos fonte PHP.

Cadeas

Unha cadea, ou string, é unha serie de caracteres. Dado que en PHP cada carácter se representa cun byte, soamente soporta un conxunto de 256 caracteres, e polo tanto non ten soporte nativo Unicode.

Pódense especificar mediante comiñas simples ou comiñas dobres. Cando se especifican con comiñas simples os únicos caracteres de escape serán \\ (representa o literal \) e \' (representa o literal '). A continuación móstranse exemplos de cadeas.

Exemplo de cadeas en PHP

```
<?php
echo 'Isto é unha cadea sinxela';

echo 'Tamén pode incluír novas liñas en
cadeas desta forma xa que é
correcto facelo así';

// Resultado: Arnold unha vez dixo: "I'll be back"
echo 'Arnold unha vez dixo: "I\'ll be back"';

// Resultado: Borrou C:\*.??
echo 'Borrou C:\\*.??';

// Resultado: Borrou C:\*.??
echo 'Borrou C:\*.??';

// Resultado: As variables $stampou se $expandiran
echo 'As variables $stampou se $expandiran';
?>
```

O único operador de string en PHP a concatenación. Representase cun . e se usa para unir dous cadeas. Pódense concatenar cadeas con código HTML, por exemplo ca etiqueta
 para que se amosen fragmentos da cadea en distintas liñas. A continuación móstrase un exemplo de concatenación.

Exemplo de concatenación de dúas cadeas

```
<?php
$txt1="Ola mundo!";
$txt2="Que día tan bo!";
echo $txt1 . " " . $txt2;
// Resultado: Ola mundo! Que día tan bo!

echo "<br/>".$txt1."<br/>".$txt2;
/*Resultado: Ola mundo!
           Que día tan bo!*/
?>
```

Soe ser útil a función `strlen()` que devolve o número de caracteres dunha cadea. No seguinte exemplo móstrase o seu uso.

Exemplo de uso da función `strlen()`

```
<?php
echo strlen("Ola mundo!");
// Resultado: 10
?>
```

Existen máis funcións nas librerías de PHP para o manexo de strings. Iranse introducindo en posteriores actividades na medida en que fagan falla.



Realice a tarefa 2 correspondente ao manexo de cadeas en PHP.

Arrays

Un array é un tipo especial de variable que pode conter máis dun valor ao mesmo tempo. Defínense mediante o uso da función `array()`. En PHP existen tres tipos de arrays:

- Arrays indexados: os elementos indéxanse cun número. Exemplo:

Exemplo de arrays indexados

```
<?php
$cars=array("Volvo","BMW","Toyota");
echo "Gústame os coches " . $cars[0] . " , " . $cars[1] . " e " . $cars[2] . " .";
?>
```

- Arrays asociativos: os elementos indéxanse con claves. Exemplo:

Exemplo de arrays asociativos

```
<?php
$age=array("Uxío"=>"35","Xoán"=>"37","Miguel"=>"43");
echo "Uxío ten " . $age["Peter"] . " anos.";
?>
```

- Arrays multidimensionais: son arrays que conteñen á súa vez outros arrays. Exemplo:

Exemplo de arrays multidimensionais

```
<?php
// Array de dúas dimensións:
$familias = array
(
    "Vázquez"=>array
    (
        "Carmen",
        "Pilar",
        "Juan"
    ),
    "García"=>array
    (
        "Manuel"
    ),
    "López"=>array
    (
        "Nuria",
```

```
"Patricia",  
"Alberto"  
)  
);  
  
echo "Es " . $familias["Vázquez"][2] .  
" parte de la familia Vázquez?";  
  
// Resultado: Es Juan parte da familia Vázquez?  
?>
```

Para obter a lonxitude dun array, é dicir, o número de elementos dun array, úsase a función `count()`. A continuación amósase un exemplo de uso desta función.

Exemplo de uso da función `count()`

```
<?php  
$coches=array("Volvo","BMW","Toyota");  
echo count($coches);  
  
//Resultado: 3  
?>
```



Realice a tarefa 3 correspondente á definición de arrays en PHP.

2. Materiais

2.1 Textos de apoio ou de referencia

- <http://www.php.net/>, documentación sobre a linguaxe PHP.
- <http://www.w3schools.com/php/default.asp>, documentación sobre a linguaxe PHP.

2.2 Recursos didácticos

- Ordenador persoal, con software de virtualización.
- Sistemas operativo Windows.
- Software XAMPP tal e como se indica na actividade e nas tarefas correspondentes.
- Netbeans.