

UNIDADE 2: SINTAXE BÁSICA PARA INSERCIÓN DE CÓDIGO EN PÁXINAS WEB

Índice

1.	Sintaxe básica para inserción de código en páxinas web.	2
1.1	Que é PHP	2
1.2	Scripts en PHP	2
1.3	Comentarios	3
1.4	Variables e identificadores	3
1.5	Variables predefinidas.....	4
1.6	Constantes	4
1.7	Tipos de datos.....	5
1.8	Operadores aritméticos.....	6
1.9	Operadores de asignación	6
1.10	Operadores condicionais	7
1.11	Operadores lóxicos	7

1. Sintaxe básica para inserción de código en páxinas web.

1.1 Que é PHP

PHP (de **H**ypertext **P**reprocesor) é unha linguaxe de programación utilizado para **crear páxinas web dinámicas**.

PHP necesita que un servidor web con capacidade PHP sexa instalado e funcionando para poder executar as páxinas PHP.

O servidor vai a compilar o código PHP e trátalo en tempo real, coa información vindo do cliente web, para regresarlle unha páxina web adaptada, en tempo real.

O servidor ten tamén capacidade de seguir o camiño dun usuario, así como de identificalo.

PHP necesítase cando a páxina web ten que adaptarse en función do usuario, e cando se necesita gardar valores de sesión do usuario.

Existen outros linguaxe que permiten iso, como o **ASP** ou o **JSP**, pero o **PHP ten como vantaxe que é unha linguaxe fácil a aprender**. Os servidores PHP (como os servidores JSP) existen para varios sistemas operativos, entre outros Windows, Linux, e Unix.

PHP é un **estándar de programación Internet** e foi creado en **1994 por Rasmus Lerdorf**.

1.2 Scripts en PHP

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 `?>` 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**.

1.3 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
?>
```

1.4 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
$täyte = '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 adecuado 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.

1.5 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. Estas variables iranse introducindo ao longo das seguintes actividades.

1.6 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;
?>
```

1.7 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(64bits) .
 - **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 = 0x1A; // número hexadecimal (equivalente a 26 decimal)
?>

<?php
$a = 1.234;
$b = 1.2e3;
$c = 1.2e-3;
?>

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

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

$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
?>
```

1.8 Operadores aritméticos

Operador	Función
■ +	Suma dous valores numéricos
■ −	Resta dous valores numéricos
■ *	Multiplíca dous valores numéricos
■ /	Divide dous valores numéricos
■ %	Resto de dividir dous valores
■ ++	Aumenta un valor numérico en unha unidade
■ --	Decrece un valor numérico en unha unidade

1.9 Operadores de asignación

Operador	Función
■ =	Asigna o valor da dereita a variable da esquerda
■ +=	Suma o valor da dereita co da esquerda, e asigna o total o da esquerda
■ -=	Resta o valor da dereita ó da esquerda, e asigna o resultado ó da esquerda
■ *=	Multiplíca o valor da dereita co da esquerda, e asigna o resultado ó da esquerda
■ /=	Divide o valor da esquerda entre o da dereita, e asigna o resultado ó da esquerda
■ %=	Divide o valor da esquerda entre o da dereita, e asigna o resultado do resto ó da esquerda
■ .=	Concatena a cadea da dereita ca da esquerda, e asigna o resultado a da esquerda

1.10 Operadores condicionais

Operador	Función
▪ ==	Comproba se dúas cadeas son idénticas
▪ !=	Comproba se dúas cadeas son diferentes
▪ <	Comproba se o valor de la esquerda é menor que o da dereita
▪ >	Comproba se o valor de la esquerda é maior que o da dereita
▪ <=	Comproba se o valor de la esquerda é menor o igual que o da dereita
▪ >=	Comproba se o valor de la esquerda é maior o igual que o da dereita

1.11 Operadores lóxicos

Operador	Función
▪ and	Se ambos valores se cumplen é verdadeiro
▪ or	Se un ou todos os valores se cumplen é verdadeiro
▪ xor	Se solo un dos valores se cumple é verdadeiro
▪ &&	Idéntico a and
▪	Idéntico a or

Exercicios