Sintaxis Basica de PHP

Etiquetas PHP

Un script PHP comienza:

```
<?php
    //codigo
?>
//archivo.php
```

o sino:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Pagina</title>
</head>
<body>
    <?php

    ?>
</body>
</html>
```

Impresion

PHP tiene varias funciones prerdefinidas para dar salida al texto. Pero las usuales son:

```
<?php
    echo "Hola mundo";
    print("Lo de arriba X2");
?>
```

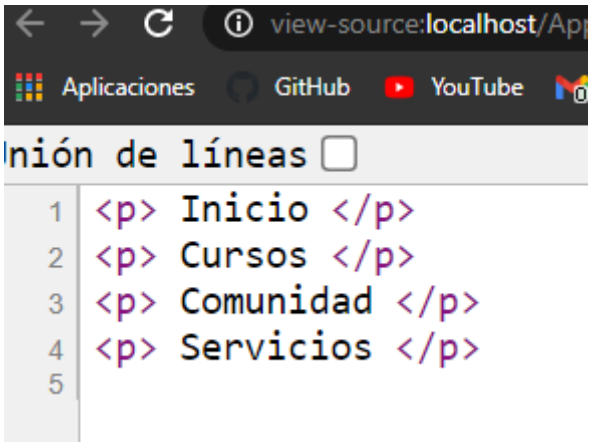
echo

Las etiquetas HTML pueden ser anadidas a texto en la declaracion `echo`

```php
<?php
    echo "<strong> texto en negritas </strong>";
?>
```

como se usaria?

```php
<?php
    $variable = ["Inicio", "Cursos", "Comunidad",
"Servicios"];
    foreach ($variable as $value) {
        echo "<p> $value </p>\n";
    }
?>
```



Comentarios

```php
<?php
    //comentario de una linea
    /*
        comentario multilineas
    */
?>
```

Variables

Es un lenguaje dinamicamente tipado, declaran variables sin necesidad de definir el tipo (el interprete infiere el tipo).

Las variables son creadas al momento que se le asigna un valor por primera vez.

```
<?php
    $nombre_variable = valor; //definicion
?>
```

Reglas de variables:

- un nombre de variable debe comenzar con una letra o un guion bajo.
- No puede comenzar con un numero
- los nombres de variables son sensibles a mayusculas.

Tipos de Datos

Tipos de datos soportados por PHP:

- String (cadena de texto)
- Integer (entero)
- Float (flotantes)
- Boolean
- Array (arreglos, matrices)
- Object (objeto)
- NULL
- Resource (recurso)

```php
<?php
    $numeroA = 12; //Integer
    $numeroB = -55.235; //Float
    $texto = "Hola mundo"; //String
    $bandera = true; //Boolean

    //impresion
    echo "entero: $numeroA";
    echo "flotante: ".$numeroB;
    echo ($texto);
    echo $bandera;
?>
```

Constantes

Son variables que no pueden ser modificadas o eliminar su definicion una ves han sido definidas.

Funcion:

```php
<?php
    define(nombre_variable, valor, case-insensitive);

    //creando constante con nombre sensible a mayusculas
    define("PI", 3.141592654);
    echo PI; //imprime valor
    echo pi; //error
?>
```

case-insensitive(insensibilidad a mayusculas): especifica si el nombre es insensible a mayusculas. por defecto es **false**.

NO es necesario utilizar el signo dolar($) antes del nombre de la constante.

Operadores

Operadores Aritmeticos

Operador	Nombre	Ejemplo
+	Adicion	`echo $x + $y`
-	Sustraccion	`echo $x - $y`
*	Multiplicacion	`echo $x * $y`
/	Division	`echo $x / $y`
%	Modulo	`echo $x % $y`
++	Incremento	`$x++`
– –	Decremento	`$y--`

Operadores de Comparacion

Nombre	Ejemplo	Resultado
Mayor que	$a > $b	true si $a es mayor que $b false en caso contrario
Menor que	$a < $b	true si $a es menor que $b false en caso contrario
Mayor o igual que	$a >= $b	true si $a es mayor o igual que $b false en caso contrario
Menor o igual que	$a <= $b	true si $a es menor o igual que $b false en caso contrario
Diferente	$a <> $b ó $a != $b	true si $a es diferente a $b false en caso contrario
Idéntico o estrictamente igual	$a === $b	true si $a es igual a $b y son del mismo tipo false en caso contrario
No idéntico o estrictamente distinto	$a !== $b	true si $a no es igual a $b o no son del mismo tipo false en caso contrario
Igual	$a == $b	true si $a es igual a $b false en caso contrario

```
12 == "12"; //true - igualdad de valores
12 === "12"; //false - igualdad de valores y tipo
```

Operadores Logicos

Operadores lógicos

Ejemplo	Nombre	Resultado
$a and $b	And (y)	**TRUE** si tanto *$a* como *$b* son **TRUE**.
$a or $b	Or (o inclusivo)	**TRUE** si cualquiera de *$a* o *$b* es **TRUE**.
$a xor $b	Xor (o exclusivo)	**TRUE** si *$a* o *$b* es **TRUE**, pero no ambos.
! $a	Not (no)	**TRUE** si *$a* no es **TRUE**.
$a && $b	And (y)	**TRUE** si tanto *$a* como *$b* son **TRUE**.
$a || $b	Or (o inclusivo)	**TRUE** si cualquiera de *$a* o *$b* es **TRUE**.

Arreglos

Un arreglo es una variable especial, la cual puede almacenar mas de un valor al mismo tiempo.(Incluso de tipo diferentes).

```php
$nombres = array("David", "Julio", "John"); //arreglo de string

echo $nombres[1]; //Julio
```

Arreglos Asociativos

Son arreglos que utilizan nombres que pueden asignar como **claves**.

Hay dos formas de crear este arreglo:

```php
//forma 1
$personas = array(
    //Clave => Valor
    "David" => "27",
    "Julio" => "21",
    "John" => "42"
);

//forma 2
$personas["David"] = "27";
```

```php
$personas["Julio"] = "21";
$personas["John"] = "42";

echo $personas["David"]; //27
```

Estructuras de Control

Condicionales [if/else | switch]

```php
if(condicion)
{
    //paso A
}
elseif(condicion)
{
     //paso B
}
else
{
     //paso C
}

//-------------------------------------------

switch(opcion){
    case valor1:

    break;

    case valo2:

    break;

    default:
}
```

Blucles(Ciclos)

```php
//CICLO WHILE
while(condicion)
{

}

//CICLO DO/WHILE
do{

}while(condicion);

//Ciclo FOR
for ($i=0; $i < valorMax; $i++)
{

}

//CICLO FOREACH
//iterar sobre colecciones
$nombres = array("David", "Julio", "John");

foreach($array as $iterador)
{

}
$personas = array(
    //Clave => Valor
    "David" => "27",
    "Julio" => "21",
    "John" => "42"
);
//para Arreglos Asociados
foreach($array as $clave => $valor)
{
    $clave // David     $valor "27"
}
```

Funciones

Es un bloque de declaraciones que puede ser utilizado repetidamente en un programa.

Las funciones no sera ejecutada inmediatamente cuando una pagina es cargada sino que sera ejecutada por una invocacion de la funcion.

Declaracion:

```
function nombre_funcion()
{
    //bloque de codigo
    return $valor;
}

//ejemplo simple
//declaracion
function sayHello($nombre="World")
{
    echo "Hello $nombre";
}

//llamada de la funcion
sayHello("Wardell"); //Hello Wardell
sayHello(); //Hello World
```

@Uzziel.