Desarrollo Web en Entorno Servidor

3.- PHP Avanzado. Funciones.

IES Severo Ochoa

Índice

- Funciones definidas por el usuario
- Paso por valor y referencia
- Alcance
  - Variables globales
- Funciones tipadas
- Anónimas / Closures
- Bibliotecas de funciones
  - `include` / `require`

Función

```
function nombreFuncion($par1, $par2, ...) {
    // código
    return $valor;
}

$resultado = nombreFuncion($arg1, $arg2, …);
```

- El paso de parámetros se realiza por valor → copia

Ejemplo función

```php
<?php
function diaSemana() {
  $semana = array(
    "lunes", "martes", "miércoles",
    "jueves", "viernes", "sábado", "domingo"
  );
  $dia = $semana[rand(0, 6)];
  return $dia;
}

$diaCine = diaSemana();
echo "El próximo $diaCine voy al cine.";
?>
```

Parámetros por referencia

- **&param**
- No se hace copia
  - Se pasa la dirección de memoria de la variable

```php
<?php
function duplicarPorValor($argumento) {
  $argumento = $argumento * 2;
  echo "Dentro de la función: $argumento.<br>";
}
function duplicarPorReferencia(&$argumento) {
  $argumento = $argumento * 2;
  echo "Dentro de la función: $argumento.<br>";
}

$numero1 = 5;
echo "Antes de llamar: $numero1.<br>";
duplicarPorValor($numero1);
echo "Después de llamar: $numero1.<br>";
echo "<br>";

$numero2 = 7;
echo "Antes de llamar: $numero2.<br>";
duplicarPorReferencia($numero2);
echo "Después de llamar: $numero2.<br>";
?>
```

Parámetros por defecto

- Asignar valores en la declaración
- Dejar el argumento en blanco

```php
<?php
function obtenerCapital($pais = "todos") {
  $capitales = array("Italia" => "Roma",
  "Francia" => "Paris",
  "Portugal" => "Lisboa");

  if ($pais == "todos") {
    return array_values($capitales);
  } else {
    return $capitales[$pais];
  }
}

print_r(obtenerCapital());
echo "<br/>";
echo obtenerCapital("Francia");
?>
```

Parámetros variables

- `$arrayArgs = func_get_args();`
  - Obtiene un array con los parámetros
- `$cantidad = func_num_args();`
  - Obtiene la cantidad de parámetros recibidos
- `$valor = func_get_arg(numArgumento);`
  - Obtiene el parámetro que ocupa la posición `numArgumento`.

- Estas funciones no se pueden pasar como parámetro a otra función → guardar previamente en una variable

Ejemplo parámetros variables

ies severo ochoa

```php
<?php
function sumaParametros() {
  if (func_num_args() == 0) {
    return false;
  } else {
    $suma = 0;

    for ($i = 0; $i < func_num_args(); $i++) {
      $suma += func_get_arg($i);
    }

    return $suma;
  }
}

echo sumaParametros(1, 5, 9); // 15
?>
```

Alcance

- Las variables definidas fuera de las funciones tienen alcance **global** → accesibles desde cualquier función
- Los parámetros de una función y las variables declaradas dentro de una función (se conocen como variables locales) sólo son accesibles desde dentro de la misma función → alcance de **función**
  - En caso de conflicto, tienen prioridad las locales

Variable local vs global

```php
<?php
function miCiudad() {
  $ciudad = "Elche";
  echo "Dentro de la función: $ciudad.<br>";
}
$ciudad = "Alicante";
echo "Antes de la función: $ciudad.<br>";
miCiudad();
echo "Después de la función: $ciudad.<br>"
?>
```

```php
<?php
function miCiudad() {
  global $ciudad;
  $ciudad = "Elche";
  echo "Dentro de la función: $ciudad.<br>";
}
$ciudad = "Alicante";
echo "Antes de llamar: $ciudad.<br>";
miCiudad();
echo "Después de llamar: $ciudad.<br>"
?>
```

Recuerda


NO UTILIZÉIS VARIABLES GLOBALES DENTRO DE UNA FUNCIÓN
Si hace falta, pasadlas como parámetro

Funciones tipadas

- PHP 7
- Permite definir los tipos de los parámetros y de los valores devueltos.
  - `int, float, string, bool, object, array`
- Si queremos comprobación estricta

  ```
  declare(strict_types=1);
  ```

  Se pone en la primera línea

Ejemplo funciones tipadas

```php
<?php
declare(strict_types=1);

function suma(int $a, int $b) : int {
  return $a + $b;
}

$num = 33;
echo suma(10, 30);
echo suma(10, $sum);
?>
```

Funciones variable

- Permite asignar una función a una variable.
  - Nombre de la función entre comillas.
- Si una variable tiene paréntesis, PHP buscará una función con su valor.

```php
<?php
$miFuncionSuma = "suma";
echo $miFuncionSuma(3,4); // llama a suma
?>
```

Funciones anónimas

```php
<?php
$anonima = function() {
  echo "Hola";
};
$anonima();

// Uso de variables externas al closure
$mensaje = "Hola";
$miClosure = function() use ($mensaje) {
  echo $mensaje;
};
$miClosure();

// Uso de parámetros
$holaPHP = function ($arg) use ($mensaje) {
  echo $mensaje." ".$arg;
};
$holaPHP("PHP");
?>
```

Biblioteca de funciones

- Agrupación de funciones en un archivo.
  - Permite reutilizar el código
- Incluir con:
  - `include(archivo); / include_once(archivo);`
  - `require(archivo); / require_once(archivo);`
- Si no encuentra el archivo, `require` lanza un error fatal, `include` lo ignora
- Las funciones `_once` sólo se cargan una vez, evita bucles

Ejemplo biblioteca

```php
<?php
function suma(int $a, int $b) : int {
  return $a + $b;
}

function resta(int $a, int $b) : int {
  return $a - $b;
}
?>
```

biblioteca.php

```php
<?php
include_once("biblioteca.php");
echo suma(10,20);
echo resta(40,20);
?>
```

Ejemplo plantilla

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title><?= $titulo ?></title>
</head>
<body>
```

encabezado.php

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

pie.html

```
<?php
  $titulo = "Página con includes";
  include("encabezado.php");
?>
<h1><?= $titulo ?></h1>
<?php
  include("pie.html");
?>
```

¿Alguna pregunta?