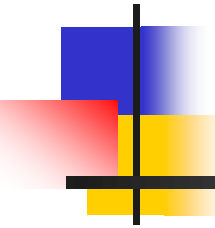


4. Funciones en PHP

Profesor: Alejandro Amat Reina

Funciones propias

- Para crear tus propias funciones, deberás usar la palabra function:

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

- Para invocar la función:

```
$resultado = suma(5, 7);
```

- Si una función no tiene una sentencia return, devuelve null al finalizar.

Valores por defecto en los parámetros

- Podemos indicar valores por defecto para los parámetros
- Si cuando llamamos a la función no indicamos el valor de un parámetro se tomará el valor por defecto indicado.

```
function precio_con_iva($precio, $iva=0.21)
{
    return $precio * (1 + $iva);
}
$precio = 10;
$precio_iva = precio_con_iva($precio);
```

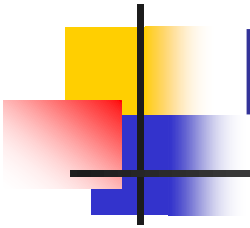
- Puede haber más de un parámetro con valor por defecto, pero siempre tienen que estar al final.

Paso de parámetros por referencia

- Por defecto los parámetros se pasan por valor.
- Para pasar un parámetro por referencia añadiremos el símbolo & delante de su nombre.

```
function precio_con_iva(&$precio, $iva=0.18)
{
    $precio *= (1 + $iva);
}
```

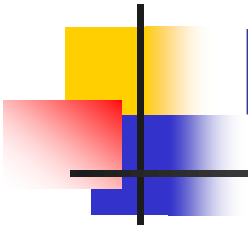
Declaraciones de tipo (Type Hinting)



- Las funciones obligan a que los parámetros sean de cierto tipo
- Si el valor dado es de un tipo incorrecto, se generará un error
- Debe anteponerse el nombre del tipo al nombre del parámetro
- Se puede hacer que una declaración acepte valores NULL si el valor predeterminado del parámetro se establece a NULL

Tipos válidos

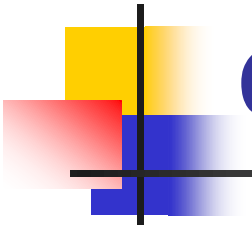
Tipo	Descripción	Versión de PHP mínima
nombre de clase/interfaz	El parámetro debe ser una *instanceof* del nombre de la clase o interfaz dada.	PHP 5.0.0
self	El parámetro debe ser una *instanceof* de la misma clase donde está definido el método. Esto solamente se puede utilizar en clases y métodos de instancia.	PHP 5.0.0
array	El parámetro debe ser un array.	PHP 5.1.0
callable	El parámetro debe ser un callable válido.	PHP 5.4.0
bool	El parámetro debe ser un valor de tipo boolean.	PHP 7.0.0
float	El parámetro debe ser un número de tipo float.	PHP 7.0.0
int	El parámetro debe ser un valor de tipo integer.	PHP 7.0.0
string	El parámetro debe ser un string.	PHP 7.0.0



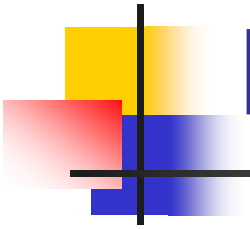
Ejemplo

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

$resultado = suma(5, 3);
```

Declaraciones de tipo de devolución



- Añadido en PHP 7
- Especifican el tipo del valor que serán devuelto desde una función
- Están disponibles los mismos tipos de la tabla anterior



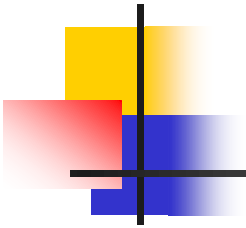
Ejemplo

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

$resultado = suma('5', 3);
```

Tipificación estricta

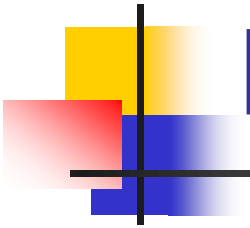
- PHP fuerza a los valores de un tipo erróneo a ser del tipo escalar esperado si es posible
- Ejemplo:
  - Una función que espera un string y recibe un int obtendrá una variable de tipo string
- En el modo estricto esto no se permite
- Si los tipos no coinciden nos dará error
- Excepción: se puede pasar un int a una función que espere un float
- El modo estricto se habilita para cada fichero php
- Habilitar el modo estricto:

```
declare(strict_types=true);
```



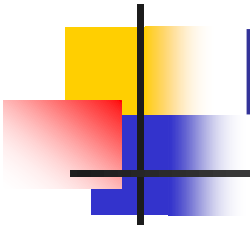
Ejemplo

```php
declare(strict_types=true);

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

$resultado = suma('5', 3);
```



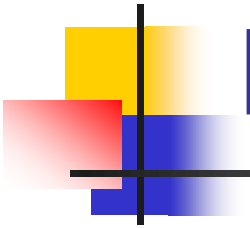
Ejercicio

- Queremos crear una función llamada insert que nos genere una sentencia insert into en sql.
- Para ello la función recibirá dos parámetros:
  - El nombre de la tabla
  - Un array asociativo que contendrá los nombres y valores de los campos de la tabla.
- La sentencia resultante tendrá la siguiente forma:
  - 'insert into nombre_tabla (nombres campos separados por comas) values (nombres campos separados por comas con el carácter ':' delante)
- De momento no haremos nada con los valores de los campos.
- Ayuda:
  - utiliza las funciones sprintf, implode y array_keys.



Ejercicio

- A partir del ejercicio anterior crea otra función que reciba los mismos parámetros más un parámetro booleano para indicar si queremos generar la query con los nombres de los campos o no.
- El parámetro tendrá el valor true por defecto.
- Si su valor es true generará la query igual que en el ejercicio anterior, pero si es false la generará así:
  - ‘insert into nombre_tabla values (nombres campos separados por comas con el carácter ‘:’ delante)

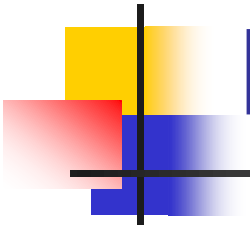


Ejercicio

- Repite el ejercicio anterior con los siguientes cambios:
  - La cadena resultante se pasará por referencia.
  - Pasaremos la cadena de la siguiente forma:
    - insert into tabla (campos) values (valores)
  - Dentro de la función sustituiremos lo siguiente:
    - El texto tabla por el nombre de la tabla.
    - El texto campos por los nombres de los campos separados por comas
    - El texto valores por los nombres de los campos separados por comas y el carácter ‘:’ delante.

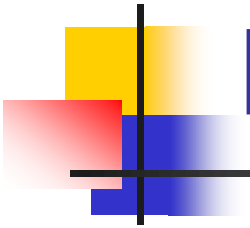
Variables superglobales

- Variables internas predefinidas de PHP
- Pueden usarse desde cualquier ámbito
- Son arrays asociativos que contienen un conjunto de valores
  - $_SERVER. Contiene información sobre el entorno del servidor web y de ejecución.
  - $_GET, $_POST y $_COOKIE contienen las variables que se han pasado al script actual utilizando, respectivamente, los métodos GET (parámetros en la URL), HTTP POST y Cookies HTTP
  - $_REQUEST junta en uno solo el contenido de los tres arrays anteriores, $_GET, $_POST y $_COOKIE.
  - $_ENV contiene las variables que se puedan haber pasado a PHP desde el entorno en que se ejecuta.
  - $_FILES contiene los ficheros que se puedan haber subido al servidor utilizando el método POST.
  - $_SESSION contiene las variables de sesión disponibles para el guion actual.
- http://es.php.net/manual/es/language.variables.superglobals.php



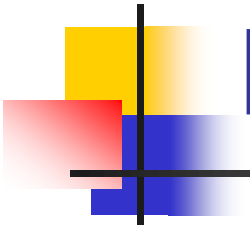
Ejercicio

- Crea una función que devuelva la uri de la página actual eliminando el carácter ‘/’ inicial.



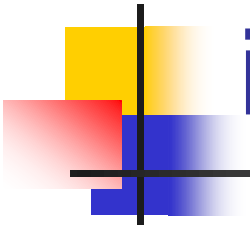
Ejercicio

- Crea una función como la anterior que elimine el querystring de la uri si lo tuviera.



Ejercicio

- Crea una función que nos devuelva el método por el que se ha solicitado la página actual (GET, POST, etc.).



isset y unset

- isset determina si una variable o variables están definidas y no son NULL
- unset destruye las variables especificadas
- Si una variable ha sido eliminada con unset(), isset() devolverá FALSE
- isset devolverá false si la variable ha sido definida como NULL
- Si una variable contiene una cadena vacía isset devolverá true
- Si son pasados varios parámetros, isset() devolverá true únicamente si todos los parámetros están definidos
- isset se utiliza mucho para comprobar si se ha recibido un parámetro por GET

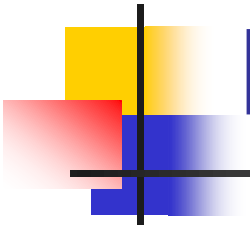
Nuevos operadores de PHP 7

- Operador nave espacial <=> (Spaceship operator)

```
($a <=> $b)  ->  ($a < $b) ? -1 : (($a > $b) ? 1 : 0)
```

- Operador de fusión de null ??

```
$a = $b ?? "default_value";  ->  $a = isset($b) ? $b : "default_value";
```



Ejercicio

- Crea una página que compruebe si ha recibido un parámetro llamado número
- En caso de haberlo recibido mostrará por pantalla si el número es mayor, menor o igual que 10
- NOTA: Utiliza los nuevos operadores de PHP 7 <=> y ??