



PHP | Funciones

Vamos a hablar ahora de las funciones. Si ya has realizado el módulo de Javascript, las funciones en PHP son prácticamente iguales a Javascript, con algunas particularidades.

Las funciones es la manera de agrupar funcionalidad y poder reutilizar código. Nos evita el tener código repetido por nuestro código. Simplemente definimos nuestra función y luego la llamamos desde nuestro código.

Las funciones pueden recibir argumentos o parámetros, y pueden devolver o no un valor.

El mejor ejemplo y el más sencillo puede ser una función que te diga si una persona es mayor de edad en base a la fecha que se le pasa como parámetro.

En PHP la sintaxis para definir una función es con la palabra reservada **function**.

```
<?php
function esMayorEdad($fechaNacimiento) {
    // aquí irá nuestro código para identificar si es mayor de edad o no.
}
```

Para poder llamar a funciones que están en otros ficheros, es necesario primero incluirlas. Es decir si la función **esMayorEdad** estuviese en el fichero **funciones.php**, debemos de incluir el fichero antes:

```
<?php
include 'funciones.php'; //ya podemos llamar a las funciones
esMayorEdad('25-03-2008');
```

Si queremos que la función devuelva un valor, solamente tendremos que devolverlo con la palabra reservada **return**

```
<?php
function suma($op1, $op2) {
    return $op1 + $op2;
}
```

El valor de retorno puede ser cualquier cosa, un array, un string, un booleano, etc...

Siempre que PHP encuentra una línea de **return**, sale de la función devolviendo el valor indicado, sin ni siquiera ejecutar el código que haya después de la línea del return.

Argumentos por defecto

PHP permite en las funciones que haya argumentos por defecto, que en el caso de que no se pasen a la hora de llamar a la función tomarán el valor que se haya configurado por defecto.

NOTA: Puede tener más un argumento con valores por defecto, pero siempre tienen que ser los últimos de la función.

Es decir, lo siguiente está bien:

```
<?php
function saludar($persona, $mensaje = 'hola'){
    //contenido de la función
}
```

Pero esto otro es incorrecto:

```
<?php
function dameTemperatura($grados = 'celsius', $ciudad){
    //contenido de la función
}
```

Para llamar por ejemplo a la función saludar se podría hacer de estas 2 maneras:

```
<?php
saludar('Sandra', 'A las muy buenas');
```

```
saludar('Javi');
```

En el segundo ejemplo, al entrar en la función `saludar`, el valor de `$mensaje` sería **'hola'**, mientras que en el primero sería **'A las muy buenas'**

Declaración de tipos:

Es mucho más fácil para un desarrollador entender el código cuando se le da información extra, y los tipos es una manera de asegurarte del valor que recibirá tu función.

A los argumentos se les puede indicar el tipo que tienen, y también aplica para el valor que devuelven, pero siempre es opcional el ponérselo o no.

Para añadirle los tipos a la función ***dameTemperatura*** sería así:

```
<?php
function dameTemperatura(string $grados = 'celsius', string $ciudad):float{
    //contenido de la función
}
```

En este caso la función recibe 2 argumentos de tipo cadena, y devuelve un flotante (ej: **27.5**)

La tipificación estricta la daremos en el Bootcamp.