

# Funciones en PHP



# Subprogramación

Las funciones las usamos como elementos para la organización del código.

- Ordenar el código
- Jerarquizar
- Modularizar
- Reutilizar el código

Las funciones son muy útiles en programación ya que nos permiten agregar código para realizar alguna tarea. Las funciones son una manera de distribuir nuestro código y mantener ordenado el mismo. También nos ayudan a no escribir varias veces el mismo código, ya que al estar agrupado en una función, podemos llamarla las veces que nos haga falta.



# Funciones

Si tenemos que realizar una acción que conlleva varias líneas de código, las programamos y ya está

¿Qué ocurre si esa acción la quiero repetir 5 veces, o 10 o un número indeterminado de veces?

La forma correcta es hacerlo mediante una función. El código se escribe una sola vez pero puede ser invocado tantas veces como queramos o necesitemos.



# Partes de una función

```
function nombre ($arg1,$arg2,...) {  
    instrucciones;  
    return $valor;  
}
```

Donde tenemos como partes las siguientes:

- nombre: Es el nombre que le damos a la función. No puede ser una palabra reservada del sistema.
- \$arg1,\$arg2: Son los argumentos que le pasamos a la función. Puede no recibir ninguno a recibir varios.
- El cuerpo de la función está compuesto por instrucciones, que desarrollan la función.
- return: Instrucción que permite devolver el resultado de la función. Puede haber varios return en la función pero hay que tener en cuenta que después de ejecutar uno de ellos se acaba la función y ya no se ejecuta nada más.
- Las funciones anónimas son aquellas que no tienen nombre. Sólo se usan una vez. Las veremos más adelante con Laravel.



# Parámetros

- Los parámetros son pasados por valor por defecto
- Se antepone & si queremos pasarlos por referencia
- En la propia línea podemos poner valores por defecto

```
function saluda(&$nombre, $formalismo = "Buenos días") {  
    if ($nombre == "") {  
        $nombre = "amigo";  
    }  
    echo $formalismo . " " . $nombre;  
}
```



# Resultado de la función

Se devuelve el resultado de la función con la palabra reservada **return** seguida de una variable.

- Si se omite el **return**, la función devuelve **NULL**.
- Cuando se ejecuta, se acaba la función.
- Podemos devolver múltiples valores con un array.



# Ámbito de las variables

Las variables creadas dentro de una función, son locales a la función. Eso significa que desde fuera de la función no se puede acceder a las mismas.

Las variables creadas fuera de la función son desconocidas para la función. Si las pasamos como parámetros entonces si las reconoce.

Disponemos como opción el definir variables globales, que son conocidas en todo el ámbito de la aplicación. No es aconsejable salvo excepciones porque nos cargamos la modularidad y la posibilidad de reutilización del código. Para definir variables globales anteponemos a la definición de la variable la palabra reservada **global** o podemos usar el array **\$GLOBALS**.