

## A2\_T3. Funcions

Cualquier programa que no sea trivial estará dividido en varios módulos o subprogramas con el fin de que sea legible. Estos bloques que podrán definirse en distintos ficheros se llaman funciones.

PHP evidentemente cuenta con soporte para funciones definidas por el usuario y también proporciona, a través de su biblioteca de extensiones, una gran colección de APIs que pueden realizar diversas tareas.

### Funciones definidas por el usuario en PHP

La definición de funciones en PHP utiliza la siguiente sintaxis:

```
<?php
function prueba($a, $b, $c) {
    echo "Función de prueba:\n";
    echo '$a=' . $a . ' $b=' . $b . ' $c=' . $c . "\n";

    return $a + $b + $c;
}
?>
```

Algunas características de las funciones en PHP son:

- Pueden declararse en cualquier punto. Es decir, se puede escribir una función dentro de cualquier estructura como un **if** o incluso dentro de otra función.
- Todas las funciones son **globales**. Es decir, siempre es posible llamar a una función aunque esta haya sido **declarada** en el **interior** de otra función.
- En general es posible invocar a una **función** que **todavía no** ha sido **declarada**, salvo cuando se trata de funciones que han sido declaradas en el interior de una estructura condicional u otra función. En ese caso, sólo se puede invocar a las funciones después de que hayan sido declaradas.

Respecto a la lista de argumentos, las funciones de PHP soportan tanto listas de argumentos de longitud variable como argumentos predeterminados, en este último caso los argumentos predeterminados deben declararse después de los no

predeterminados. El pase de parámetros por defecto se realiza por valor pero es posible hacerlo por referencia utilizando el & al declarar el argumento. En la lista de argumentos de una función se pueden utilizar los corchetes para especificar que algunos argumentos son opcionales.

Puede utilizarse una variable para contener el nombre de la función a invocar. Esto da lugar a las llamadas [funciones variables](#), que también se pueden utilizar al llamar a un método de una clase.

## **Biblioteca de extensiones de PHP**

En PHP las funciones se agrupan en el interior de extensiones que conforman APIs con un propósito específico (por ejemplo manipulación de gráficos con [GD](#) o cálculos con números de precisión arbitraria con [GMP](#)). Algunas de estas extensiones forman parte del núcleo de PHP y no son opcionales, siempre están disponibles. Otras están formadas por ficheros externos que pueden instalarse o no, y algunas dependen de bibliotecas externas.

En la documentación de PHP existen listados de extensiones categorizados [por función](#) o [por localización](#).

## **Dividiendo el código PHP en diferentes ficheros**

Para mantener el código de una aplicación ordenado es posible utilizar diferentes ficheros que podrán ser incluidos o requeridos e interpretados allí donde haga falta.

Las sentencias [include](#) y [require](#) permiten importar e interpretar un fichero externo, que incluso puede descargarse desde una url http. La diferencia entre ambas, es que la primera presentará una advertencia cuando el fichero no pueda ser incluido (porque no se encuentra) mientras que la segunda provocará un error y detendrá la ejecución.

También existen las sentencias [include\\_once](#) y [require\\_once](#) que incluyen e interpretan únicamente el fichero si no se ha hecho previamente. Es decir, aunque se invoque varias veces la carga de un fichero externo esta operación solo se hace una vez.

La directiva del núcleo [include\\_path](#) especifica la lista de directorios en los que se buscarán los ficheros a incluir por las

M09 - UF1

UF1: LLENGUATGES DE GUIONS DE SERVIDORS

A2: PROGRAMACIÓ WEB DEL SERVIDOR



sentencias anteriores. De todos modos es posible especificar la ruta relativa o absoluta para evitar utilizar `include_path`.