

funciones en php

conjunto de ordenes agrupadas por un nombre en concreto

instrucciones que se pueden reutilizar cuantas veces se requiera

sintaxis

```
function nombreFuncion ($parametros) {  
    // bloque de instrucciones  
}
```

```
nombreFuncion($parametros);
```

// code...

```
function muestraNombre(){  
    echo "jared\n";  
    echo "kenneth\n";  
    echo "cris\n";  
    echo "rodry\n";  
}  
muestraNombre();
```

todas las tablas de multiplicar

```
function tabla($numero) {  
    echo "<h3>Tabla de multiplicar del numero: $numero</h3>";  
  
    for ($i=0; $i <= 10; $i++) {  
        $operacion = $numero*$i;  
        echo "$numero x $i = $operacion <br/>";  
    }  
}  
  
for($i=0; $i <= 10; $i++){  
    tabla($i);  
}
```

parametros

vemos como una funcion recibe parametros que son variables, que le entregan valores para ue la funcion opere.

Estos parametros son obligatorios

```
function calculadora($numero1, $numero2) {
    $suma = $numero1 + $numero2;
    $resta = $numero1 - $numero2;
    $multiplicacion = $numero1 * $numero2;
    $division = $numero1 / $numero2;

    echo "Suma: $suma\n";
    echo "Resta: $resta\n";
    echo "Multiplicacion: $multiplicacion\n";
    echo "Division: $division\n";
    echo "<hr/>\n";
}
calculadora(10,5);
calculadora(12,3);
calculadora(42,2);
calculadora(6,3);
calculadora(19,7);
calculadora(8,2);
```

parametros opcionales

son parametros que ya traen por default un valor

```
function calculadora($numero1, $numero2, $negrita = false) {
    $suma = $numero1 + $numero2;
    $resta = $numero1 - $numero2;
    $multiplicacion = $numero1 * $numero2;
    $division = $numero1 / $numero2;

    if($negrita != false){
        echo "<h1>"
    }

    echo "Suma: $suma\n";
    echo "Resta: $resta\n";
    echo "Multiplicacion: $multiplicacion\n";
    echo "Division: $division\n";
    echo "<hr/>\n";

    if($negrita != false){
```

```
        echo "</h1>"
    }

}

calculadora(10,5,true);
calculadora(12,3);
calculadora(42,2,true);
calculadora(6,3);
calculadora(19,7,true);
calculadora(8,2);
```

return

normalmente no deberia imprimir nada desde una funcion, por buenas practicas, ya que se debe devolver un valor.

atravez del return.

```
function calculadora($numero1, $numero2, $negrita = false) {
    $suma = $numero1 + $numero2;
    $resta = $numero1 - $numero2;
    $multiplicacion = $numero1 * $numero2;
    $division = $numero1 / $numero2;

    $cadenaTexto = "";

    if($negrita){
        $cadenaTexto .= "<h1>";
    }

    $cadenaTexto .= "Suma: $suma\n";
    $cadenaTexto .= "Resta: $resta\n";
    $cadenaTexto .= "Multiplicacion: $multiplicacion\n";
    $cadenaTexto .= "Division: $division\n";
    $cadenaTexto .= "<hr/>\n";

    if($negrita){
        $cadenaTexto .= "</h1>";
    }

    return $cadenaTexto;
}

echo calculadora(10,20,true);
```

funciones dentro de otras funciones

Este ejemplo podria ser tonto, pero es la base para entender como una funcion recibe otra funcion, y muy util cuando una funcion debe hacer muchas cosas.

```
// funciones que reciben funciones

function getNombre ($nombre) {
    $texto = "Su nombre es <strong>$nombre </strong>";
    return $texto;
}

function getApellido($apellido) {
    $texto = "y su apellido es <strong>$apellido </strong>";
    return $texto;
}

function devuelveNombre ($nombre, $apellido) {
    $texto = "<h1>".getNombre($nombre).getApellido($apellido)."</h1>";
    return $texto;
}

echo devuelveNombre("Jared", "Latorre")."<br/>";
echo getNombre('Paco')."<br/>";
```

ademas estas funciones son reutilizables.

variables globales y locales

Scope de las variables

1. *variables globales*

- Son las que se declaran fuera de una funcion, y estan disponibles dentro del código.

1. *variables locales*

- Son declaradas dentro de la funcion y estan disponibles solo en la funcion.
- solo estarian disponibles sacandolas con return

```
$frase = " ni los genios son tanto, ni los mediocres tampoco ";

echo $frase;
```

```
function holaMundo() {  
    global $frase;  
    echo "<h1> $frase </h1>";  
  
    $year = 2021; //scope local  
    echo "<h1>$year</h1>";  
    return $year; //al retornarla si la dejara imprimir  
  
}  
holaMundo();
```

con la palabra reservada **global** podemos hacer uso de la variable global dentro de una funcion.

funciones en variables

se guarda la funcion en una variable

```
function buenasDias () {  
    return "<h1>Hola! buenos dias</h1>";  
}  
function buenasTardes () {  
    return "<h1>Hola! como ha estado la comida</h1>";  
}  
function buenasNoches () {  
    return "<h1>¿Te vas a dormir ya? Buenas noches!!</h1>";  
}  
  
$horario = 'Noches';  
  
$miFuncion = "buenas".$horario;  
  
echo $miFuncion();
```

De esta manera guardamos el string con el nombre de la funcion, y luego ejecutamos esta variable con los parentesis, ejecutando la funcion.

```
$jiji = 'buenasTardes';  
echo $jiji();
```

funciones predefinidas

son funciones que ya trae php, algunas

debugging

1. `var_dump();` // datos de la variable, por pantalla *fechas*
2. `date('D-d M Y');` // muestra fecha, hay muchas variantes, buscar echo `date('D-d M Y')."\n";` //
Mon-17 May 2021

hora

3. `time();` // timestamp en formato unix

funciones matematicas

sacar raices

1. `sqrt(10);` // raiz cuadrada

random

2. `rand(10,40);` // rango aleatorio
3. `rand();` // numero aleatorio x

pi

4. `pi();` // numero pi

redondear un numero decimal

5. `round(7.8915632)` // redondea

rangos

6. `rand(0,10)."\n";` // rango aleatorio entre 0 y 10
7. `rand()."\n";` // numero aleatorio

funciones internas del lenguaje

1. `gettype($var);` // dice que tipo de variable es
2. `is_` // para comprobar datos de una variable con `is`

```
is_string($var)
is_float($var)
is_int($var)
```

si la respuesta es 1 es true, si no da nada, es false.

3. isset(\$edad) // comprobar si existe la variable
4. trim // limpia los espacios del contenido de la variable
5. unset // eliminar variables o indices de arrays

```
$year = 2000;  
unset($year); // elimino la variable
```

6. empty // comprueba si la variable esta vacia

```
$texto = '';  
if (empty($texto)) {  
    echo "No hay dato";  
} else {  
    echo "Contiene dato";  
}
```

7. strlen // caracteres de un string

```
$caracteres = "123456";  
echo strlen($caracteres);
```

8. strpos // encuentra caracteres o palabras

```
$pelicula = "La vida es bella";  
echo strpos($pelicula, "vida");
```

3 la encuentra desde la posicion 3 de la cadena de texto

9. str_replace("vida","moto",\$pelicula) // remplazar contenido
10. strtolower(\$pelicula) // a minusculas
11. strtoupper(\$pelicula) // a mayusculas