





```
<?php
```

```
$arrayCosas = [ [ "Latptop", "Teléfono",  
"Auriculares"], [ "Zapatillas", "Remera"], [  
"Futbol", "Beisbol" ] ];
```

```
//¿Como es array?
```

```
//¿Le podemos agregar alguna key que nos señale  
el significado de los datos?
```

```
//¿Cómo recorremos el array?
```





# Objetivos

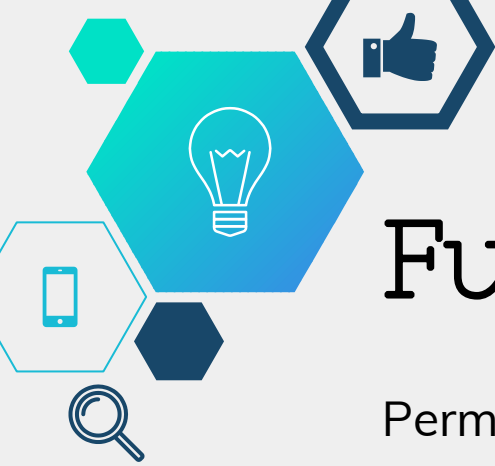
- ◇ Realizar y ejecutar funciones usando o no argumentos.
- ◇ Manejar el ámbito de las variables.
- ◇ Incrustar archivos dentro de nuestro programa.
- ◇ Ejecutar php en línea de comandos.





# Funciones





# Funciones

Permiten separar en partes nuestro “programa” a fin de:

- Organizar e identificar secciones en nuestro código.
- Reutilizar partes en lugar de repetir código.
- Corregir, cambiar y refactorizar código de forma más fácil.
- Puede utilizar argumentos para procesarlos.
- Usualmente las funciones retornan datos.



# Funciones

```
<?php
    function ejemplo($arg1, $arg2, ... , $argN = null) {
        echo "Función de ejemplo.\n";
        return $valor_devuelto;
    }
?>
```



# Funciones

```
<?php
    function sumar($num1, $num2) {
        return $num1 + $num2;
    }

    echo sumar(82, 98);

?>
```



# Funciones

```
<?php
    function saludar($nombre, $apellido = "") {
        return $nombre . $apellido;
    }

    $saludo = saludar("Juan", "Perez");
    $saludo2 = saludar("Ricardo");

?>
```





# Ámbito de Variables ó Scope





# Scope

El ámbito de una variable es el contexto dentro del que la variable está definida. La mayor parte de las variables PHP, sólo tienen un ámbito simple.

PHP tiene 2 alcances posibles:

- ◇ Global
- ◇ Local





# Scope

```
<?php  
    $a = 1; //ámbito global
```

```
function test() {  
    echo $a; //ámbito local  
}  
test();
```

```
?>
```

No produce salida!





# Scope

```
<?php  
    $a = 1; //ámbito global
```

```
function test() {  
    global $a;  
    echo $a;  
}  
test();  
?>
```





# Static

```
<?php
    function test() {
        $a = 0;
        echo ++$a;
    }
    test();
    test();
    test();
?>
```





# Static

```
<?php
    function test() {
        static $a = 0;
        echo ++$a;
    }
    test();
    test();
    test();
?>
```





## ¡A practicar!

Ejercicios del 1 al 2

```
<?php
Function Practica($desde, $hasta){
    return 'hacer ejercicios desde el'.
        $desde.' al '. $hasta;
}

echo Practica(1, 2);
?>
```





Si queremos ordenar  
y agregar nuestras  
funciones a otro  
programas?







# Inclusión de Archivos

- ◇ include
- ◇ include\_once
- ◇ require
- ◇ require\_once

```
<?php  
    require("soporte.php");  
?>
```





¿Cómo es que varias páginas tengan el mismo encabezado y pie de página ?

Encabezado y Menú.

Contenido de mi sitio.

Pie de Página






# Dividimos contenido.

// archivo header.php

// archivo index.php

//archivo footer.php





# Incorporamos el contenido.

```
// archivo index.php
```


```
Include ('header.php')
```

```
// archivo header.php
```

```
// Contenido de mi sitio.
```

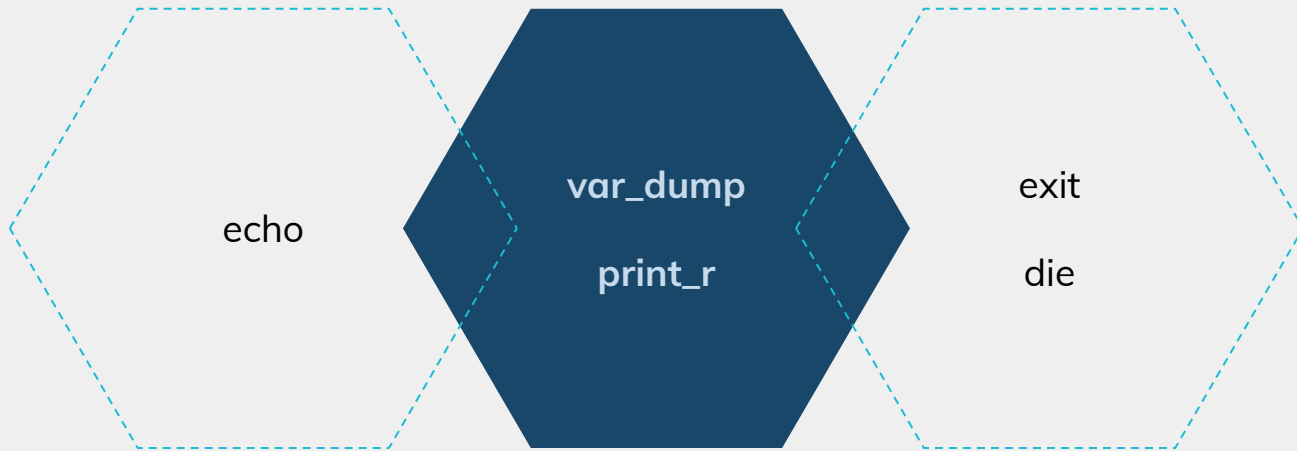
```
Include ('footer.php')
```

```
//archivo footer.php
```





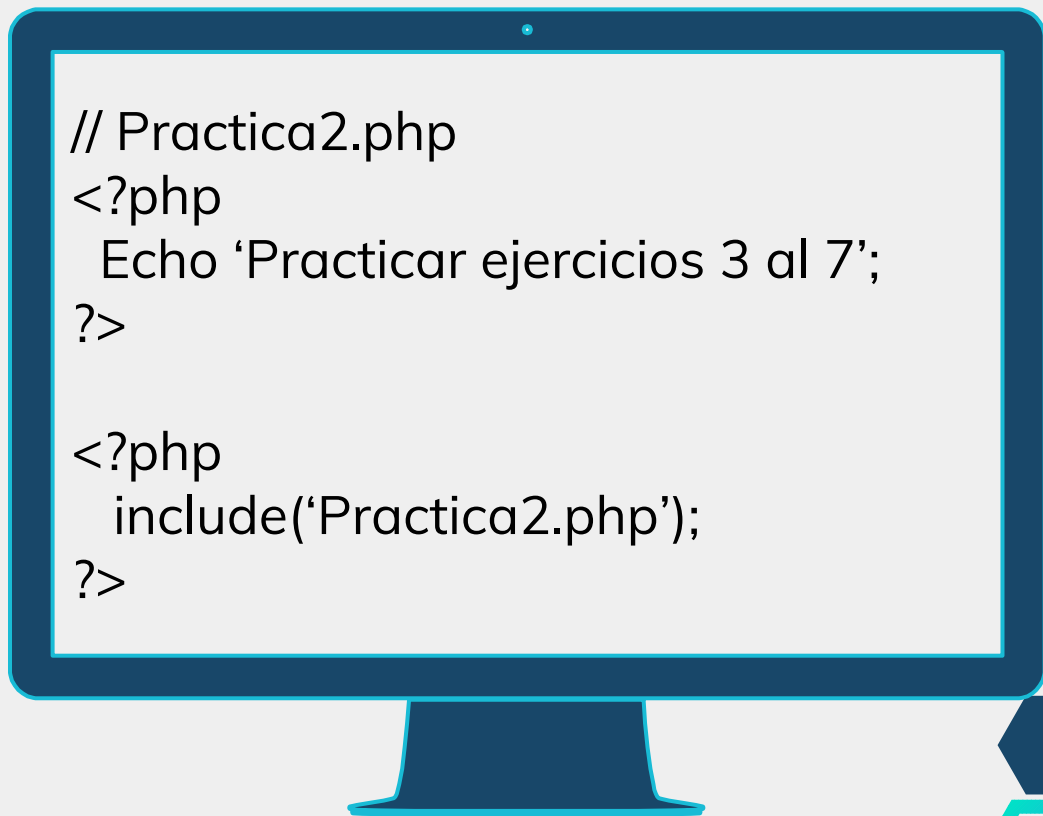
# ¿Cómo hacemos Debug?





# ¡A practicar!

Ejercicios del 3 al 7





# Scripting

Y teniendo todos estos recursos...

¿para que nos puede servir PHP?



# Scripting

Y teniendo todos estos recursos...

¿para que nos puede servir PHP?

- Para hacer sitios web.





# Scripting

Y teniendo todos estos recursos...

¿para que nos puede servir PHP?

- Para hacer sitios web.
- ¡Para procesar información en otros formatos!



# Scripting

Para ejecutar procesos internos muchas veces no utilizamos un navegador ni como punto de entrada ni como punto de salida. Entre otras cosas podemos usar una **consola**.

Para esto utilizaremos el comando **php**



# Scripting

**php -r** nos permite ejecutar código definido en la misma línea:

```
php -r 'echo "Hola Mundo";'
```



# Scripting

**php -a** nos permite acceder al modo interactivo

**php -f** nos permite ejecutar un archivo:

**php -f** scripting.php



# readline()

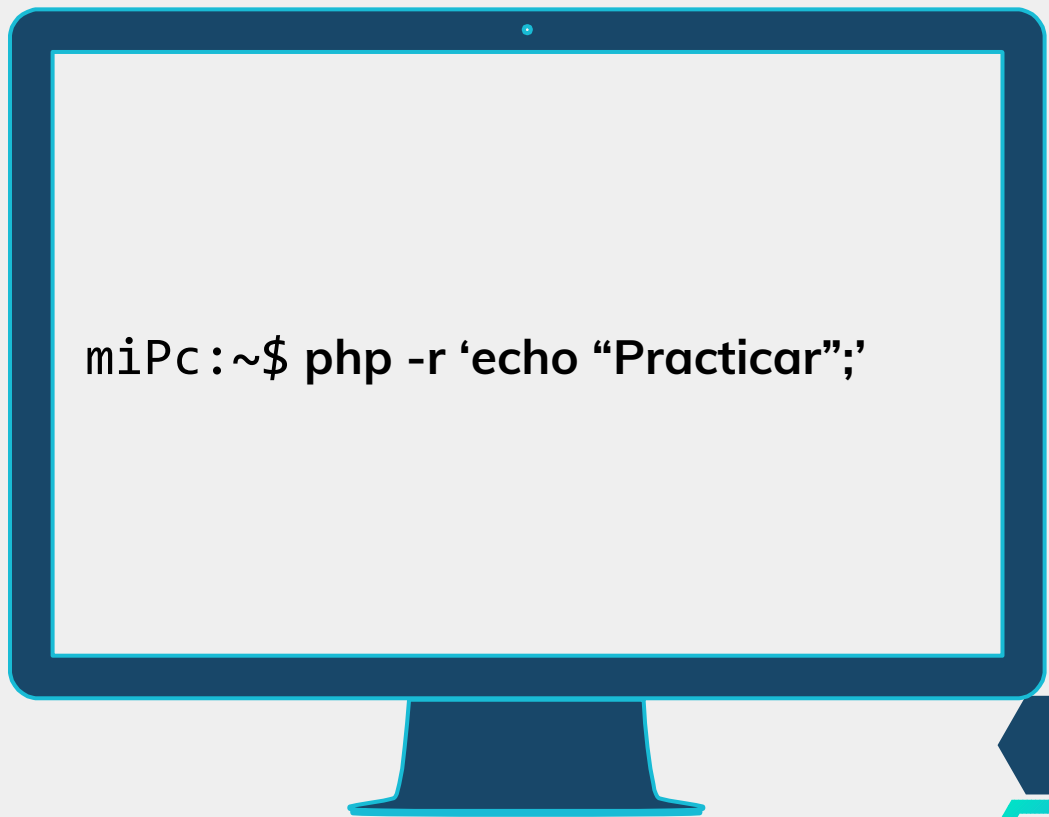
La función **readline()** nos permite interactuar con el usuario desde la terminal para pedirle información

Uso:

```
$nombre = readline("Ingrese su nombre: ");
```



¡A practicar!





¿Que  
aprendimos?

