





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.



Funciones

```
<?php
```

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

```
    sumar(15, 28);
```

```
?>
```



Funciones

```
<?php
```

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

```
echo sumar(15, 28);
```

```
?>
```



Funciones

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

```
$saludo = saludar("Ana", "Pérez");
echo $saludo; // Hola Ana Pérez
echo saludar("Ricardo"); // Hola Ricardo
```

```
?>
```



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
```

```
$persona = "Ana Pérez"; // ámbito global
```

```
function saludo() {
```

```
    return $persona; // ámbito local
```

```
}
```

```
echo saludo();
```

```
?>
```

¿Que imprime el **echo**?





Scope

```
<?php
```

```
$persona = "Ana Pérez"; // ámbito global
```

```
function saludo() {
```

```
    return $persona; // ámbito local
```

```
}
```

```
echo saludo();
```

```
?>
```

undefined variable





Scope

<?php

```
$persona = "Ana Pérez"; // ámbito global
```

```
function saludo() {
```

```
    global $persona; // usamos la variable GLOBAL
```

```
    return $persona; // ámbito local
```

```
}
```

```
echo saludo();
```

?>

Imprime: Ana Pérez





Static

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

¿Qué imprime el echo?





Static

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

Siempre imprime 1 ¿por qué?





Static

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

Ahora, ¿qué imprime el echo?





Static

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

Imprime:

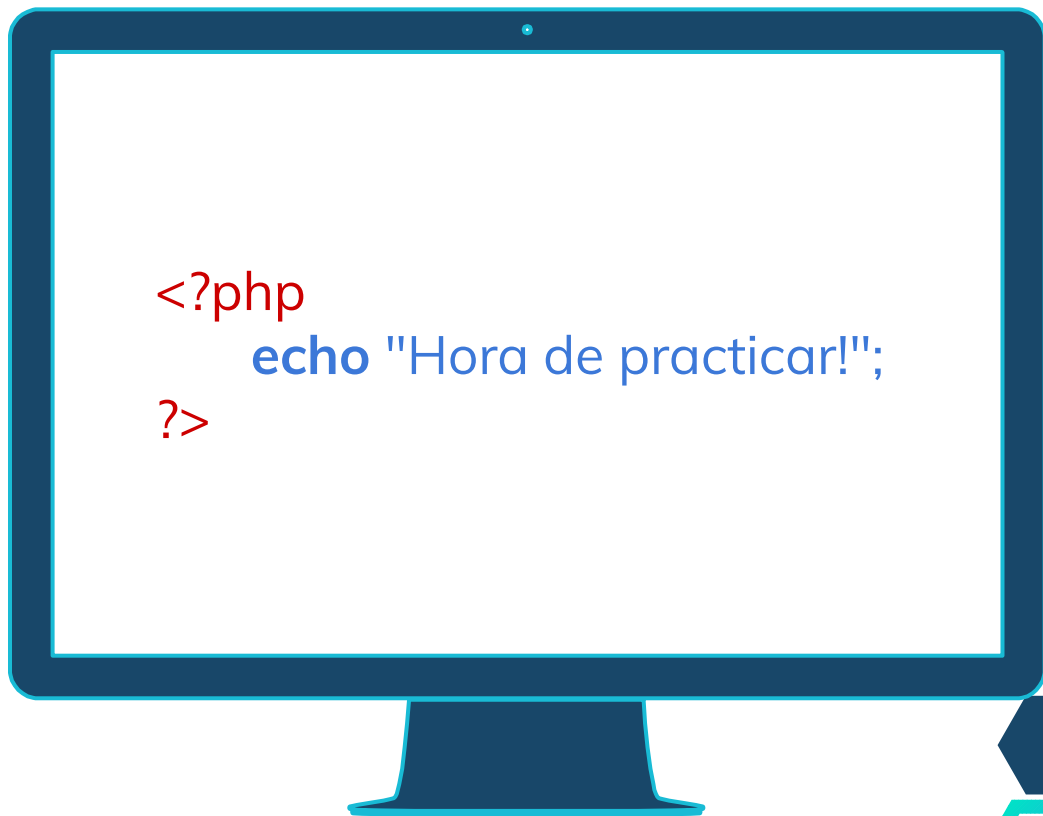
1
2
3





¡A practicar!

Ejercicios 1 y 2





Inclusión de Archivos

PHP nos permite modularizar nuestra aplicación de una manera bastante prolija.

Pues nos da un set de *funciones nativas* que nos permiten hacer inclusión de archivos de manera dinámica.

Todo con la finalidad de evitar la repetición de código y así mismo insertar lógica dentro de nuestra Aplicación web.





Inclusión de Archivos

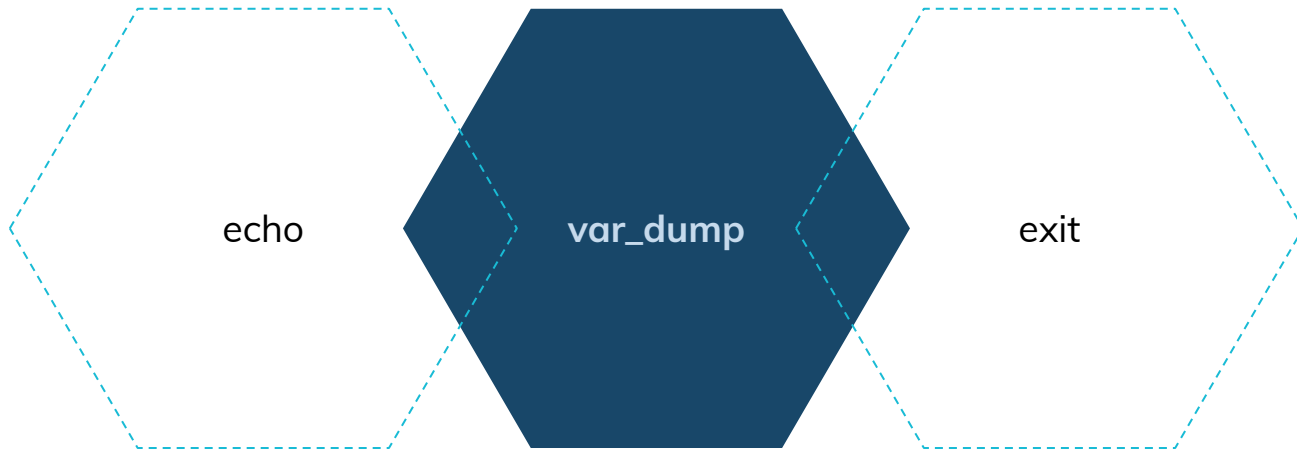
- ◇ `include` // incluye el archivo varias veces
- ◇ `include_once` // incluye el archivo una vez
- ◇ `require` // requiere el archivo varias veces
- ◇ `require_once` // requiere el archivo una vez

```
<?php  
    require "includes/header.php";  
?>
```





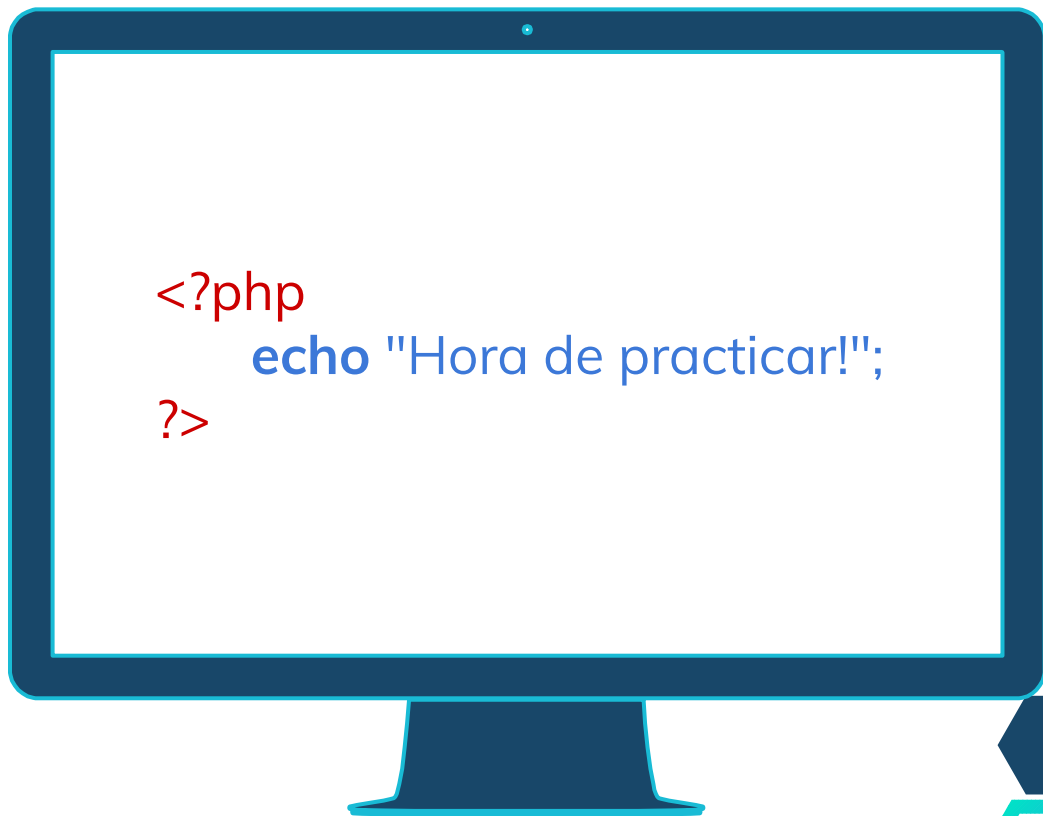
¿Cómo hacemos Debug?





¡A practicar!

Ejercicios del 3 al 7





Gracias!

¿Preguntas?

