



PHP

Clase 4

PHP

Funciones



Permiten separar en partes nuestro código a fin de:

- Organizar e identificar secciones.
- Reutilizar partes en lugar de repetir.
- Corregir, cambiar y refactorizar de forma más simple.






Funciones

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

    echo sumar(82, 98);
?>
```



Funciones

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

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

PHP

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

PHP

Scope



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

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

No produce salida!

PHP

Scope



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

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

PHP

Scope



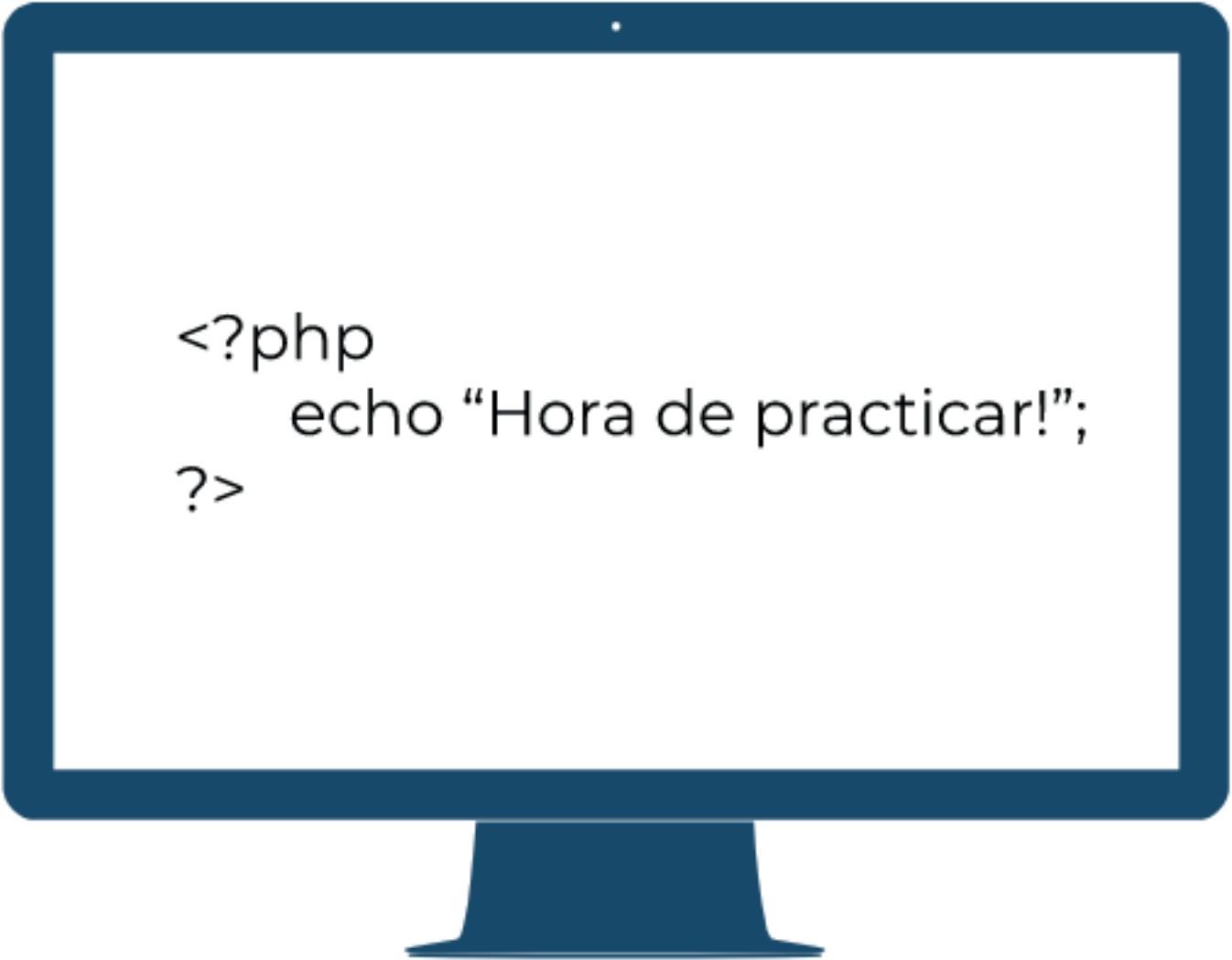
```
<?php
    function test() {
        $a = 0;
        echo ++$a;
    }
    test();
    test();
    test();
?>
```


Static

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

¡A practicar!

Ejercicios **1** y **2**



```
<?php  
    echo "Hora de practicar!";  
?>
```

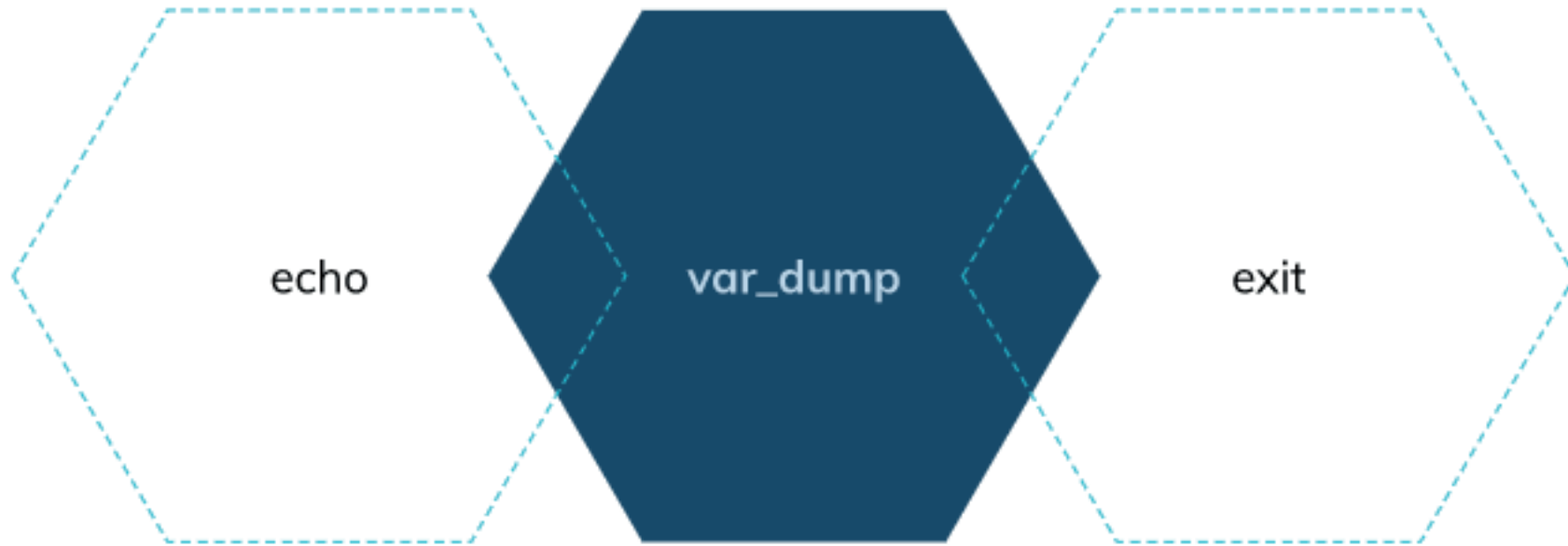


Inclusión de archivos

- include
- include_once
- require
- require_once

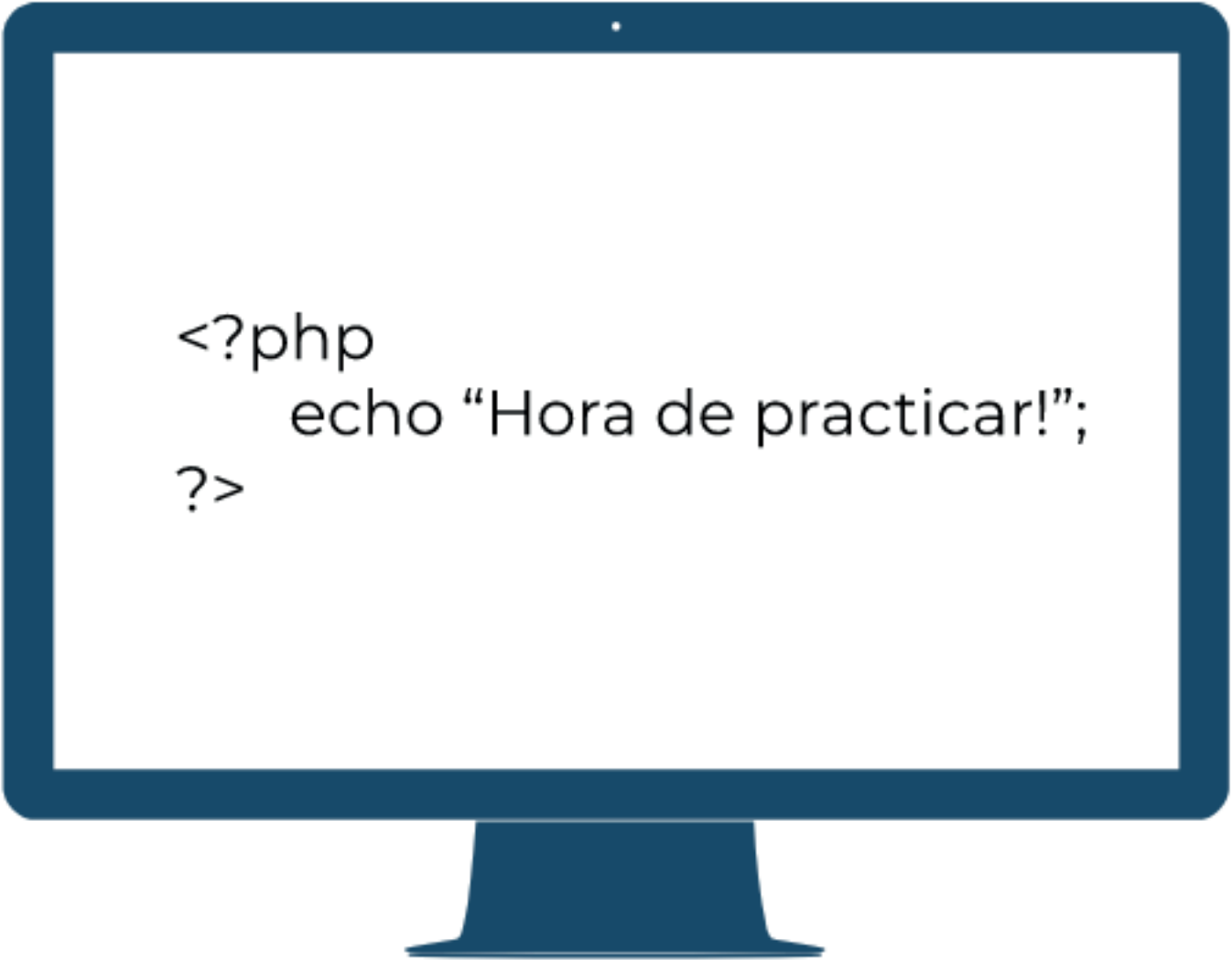
```
<?php  
    require("soporte.php");  
?>
```

¿Cómo hacemos debug?



¡A practicar!

Ejercicios del **3** al **7**



```
<?php  
    echo "Hora de practicar!";  
?>
```

PHP

Scripting



Y teniendo todos estos recursos...

¿para qué nos puede servir PHP?

PHP

Scripting



Y teniendo todos estos recursos...

¿para qué nos puede servir PHP?

- Para hacer sitios web.

PHP

Scripting



Y teniendo todos estos recursos...

¿para qué 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**.

(El “comando” es en realidad, la ejecución del intérprete)



Scripting

OPCIONES del comando

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

> **php -r 'echo "Hola Mundo";'**

-a nos permite acceder al modo interactivo:

> **php -a** <enter>

> echo "Hola Mundo";

-f nos permite ejecutar un archivo:

> **php -f ejercicio1.php**



Scripting - readline()

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

Uso:

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

Scripting - Argumentos

La variable **\$argv** es un array (convenientemente llenado por PHP para nuestro deleite) que contiene los argumentos utilizados al correr el script:

php saludar.php Roberto Sanchez

```
<?php
    echo 'Hola ' . $argv[1] . ' ' . $argv[2];

?>
```


¿Y \$argv[0]?



Scripting - Opciones

La función **getopt()** nos permite obtener las **opciones** utilizadas para ejecutar nuestro script:

php example.php -f "value for f" -v -a --required value --optional="optional value" --option



Scripting - Opciones

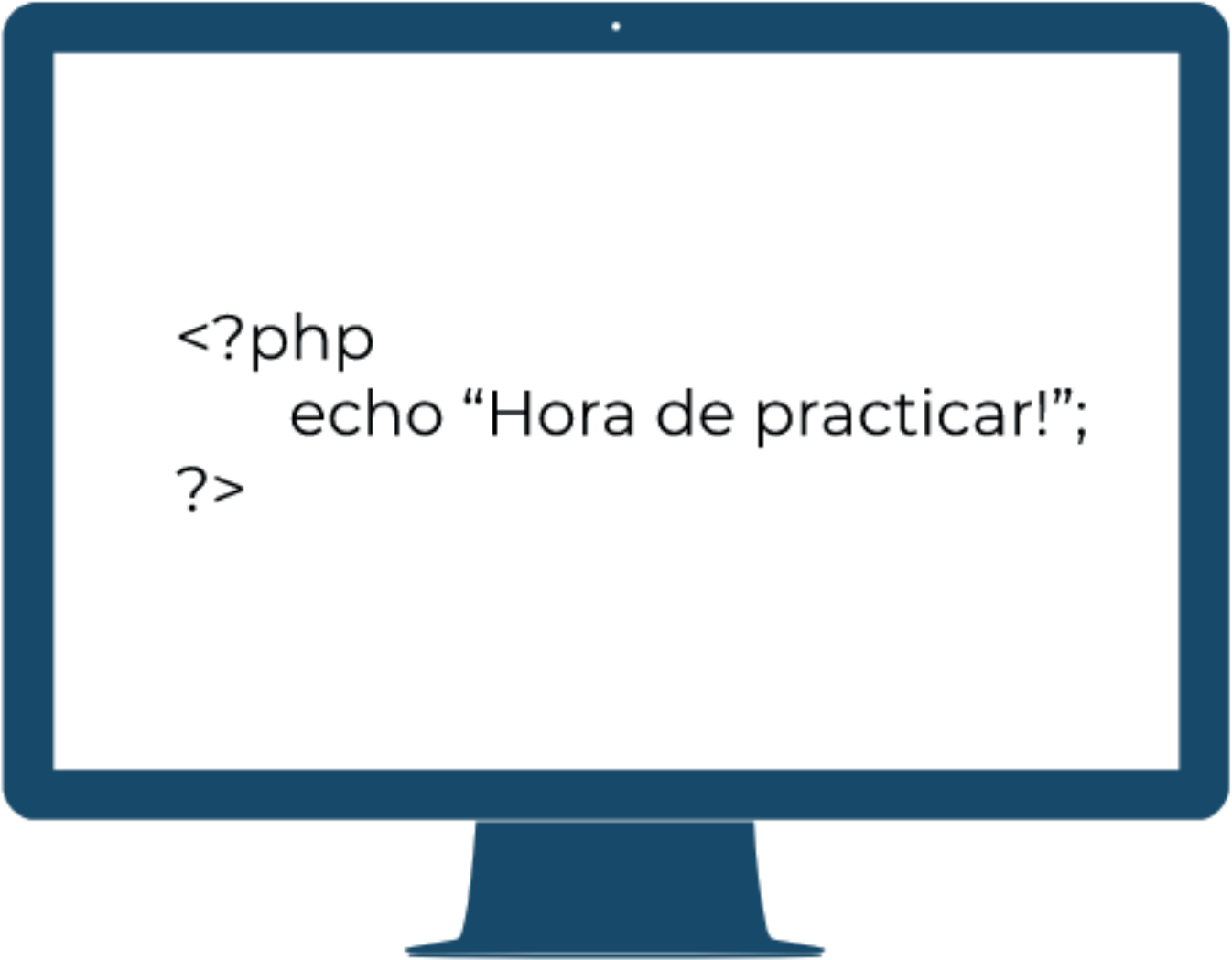
```
<?php
    $short = "";
    $short.= "f:"; // Requerido
    $short.= "v::"; // Opcional
    $short.= "abc"; // Booleanas: Si están son true.

    $long = [
        "required:", // Requerida
        "optional::", // Opcional
        "option", // Booleana
        "opt", // Booleana
    ];

    $options = getopt($short, $long);
    var_dump($options);
?>
```

¡A practicar!

Ejercicio 8



```
<?php  
    echo "Hora de practicar!";  
?>
```