



PHP

Clase 1

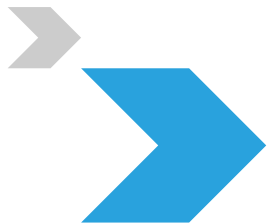
PHP

**PHP Hypertext
Preprocessor**



PHP es un lenguaje de programación de código abierto, que se ejecuta del lado del servidor, lo que lo hace muy adecuado para el desarrollo web.





¿Por qué PHP?

Simplicidad

PHP es un lenguaje simple pero poderoso, haciéndolo sencillo de aprender.

Costo

PHP gratis y libre para usar.

Soporte

La primera versión de PHP se lanzó en 1994. La versión actual es la 7.2, habiendo salido en Enero de 2018.

Comunidad

En [packagist](#) hay más de 170.000 paquetes.

Masividad

Actualmente PHP es ampliamente usado en la web.

De hecho...



244.000.000

sitios

se estima que están hechos en PHP.

<http://php.net/usage.php>



Algunos sitios usando PHP

- Facebook.com
- Wikipedia.org
- Flickr.com
- Pinterest.com
- Mailchimp.com

Y muchos más!



PHP

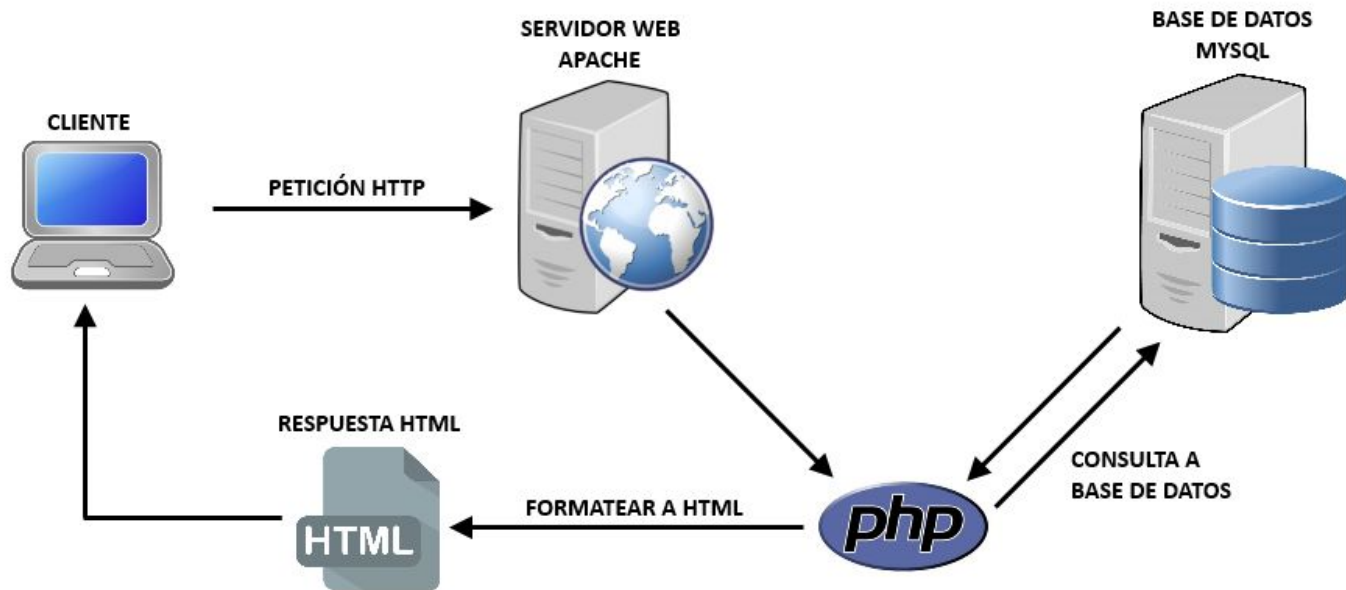
PHP Hypertext
Preprocessor



Es un lenguaje
que se ejecuta
del lado del
servidor



Cliente Servidor



Php es un lenguaje...



Interpretado

PHP procesa tu script en una serie de instrucciones (llamadas opcodes) cada vez que se accede. Las instrucciones se ejecutan línea por línea hasta el final del script.



Embebido

Extiende la funcionalidad del HTML, proporcionándole **dinamismo y haciendo que sea ideal para la web.**

Php es un lenguaje...



Multiplataforma

Puede emplearse en todos los sistemas operativos principales y admite la mayoría de servidores web de hoy en día.



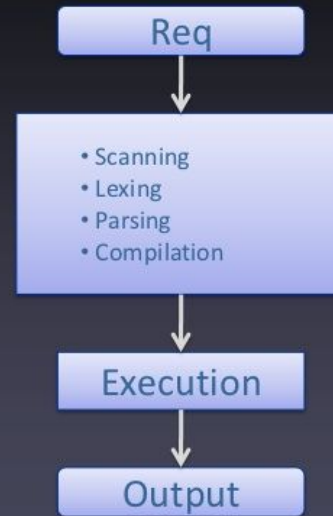
Multiparadigma

Se puede programar utilizando el paradigma de procedimientos y permite las técnicas de programación orientada a objetos y funcional.

¿Cómo funciona?

PHP Flow

- **Scanning and Lexing**
(Plain Text to Tokens)
- **Parsing**
(Tokens to Expressions)
- **Compilation**
(Translated Expressions to Opcode)
- **Execution**
(Opcode stacks are processed)



PHP

Ejecución



Un pedido a un servidor que ejecuta PHP puede retornar como respuesta HTML, CSS, JS, imágenes o cualquier tipo de formato **que pueda entender el navegador.**



PHP

Ejecución



```
<html>
```

```
  <body>
```

```
    <?php
```

```
      echo date("Y-m-d");
```

```
    ?>
```

```
  </body>
```

```
</html>
```





Variables

Una variable es un espacio reservado en memoria y puede considerarse como un nombre dado a una caja imaginaria en la que se puede colocar cualquier valor.

Pueden cambiar de contenido mientras se ejecuta un programa. Según el lenguaje, tendrán (o no) restricciones en cuanto a los tipos de datos que pueden almacenar.

Variables

```
<?php
    $nombre = "Juan";
    $apellido = "Perez";
    $casado = false;
    $edad = 35;

    echo $nombre;
    var_dump($apellido);
?>
```

Variables

```
<?php
    $nombre = "Juan";
    $apellido = "Perez";
    $casado = false;
    $edad = 35;

    $nombre = 53;

    echo $nombre;
    var_dump($apellido);
?>
```

¡Tipos dinámicos!

PHP

Tipos de datos I



- **Strings**
 - Caracteres o cadenas de texto. Pueden comenzar con comilla simple o doble pero no se pueden alternar.
- **Integer**
 - Números enteros. Positivos o negativos.
- **Float**
 - Números decimales. Se utiliza el punto y no la coma. Positivos o negativos.

¡A practicar!

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



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

PHP

Tipos de datos II



- **Boolean**
 - Expresa un valor de verdad. Puede ser **TRUE**
 - **FALSE**.
- **NULL**
 - Representa una variable sin valor:

`$prueba = null;`

`echo $prueba;`

False

Otros valores que evalúan a **FALSE**:

- 0
- "0"
- ""
- NULL
- []

PHP

Tipos de datos III



- **Array**

```
<?php
    $miArray = [];
    $miArray[0] = "Hola";
    $miArray[1] = "Chau";

    var_dump($miArray);
?>
```

PHP

Tipos de datos III



- **Array**

```
<?php
    $miArray = [];
    $miArray[0] = "Hola";
    $miArray[37] = "Chau";

    var_dump($miArray);
?>
```

PHP

Tipos de datos III



- **Array**

```
<?php
    $miArray = [];
    $miArray[] = "Hola";
    $miArray[] = "Chau";

    var_dump($miArray);
?>
```

Sobreescribir datos

- **Array**

```
<?php
    $miArray = [];
    $miArray[] = "Hola";
    $miArray[] = "Chau";
    $miArray [0] = 100;

    var_dump($miArray);
?>
```

Arrays dentro de Arrays

- **Array**

```
<?php
    $miArray = [];
    $miArray[0] = [];
    $miArray[1] = "Hola";

    $miArray[0][] = "Chau";

    var_dump($miArray);
?>
```


Arrays dentro de Arrays

- **Array**

```
<?php
    $miArray = [];
$miArray[0] = []; // sin esta línea...
    $miArray[1] = "Hola";

$miArray[0][] = "Chau"; // cambiamos esta...
    $miArray[0] = ["Chau"]; // por esta!
    var_dump($miArray);
?>
```

Inicialización

- **Array**

```
<?php
    $miArray = [];
    $miArray = ["hola", "chau", 0, 42];
    $miArray = [
        0 => "hola",
        1 => "chau",
        2 => 0,
        3 => 42
    ];
```

Arrays asociativos

- **Array**

```
<?php
    $auto = [];
    $auto["color"] = ["Negro", "Verde"];
    $auto["marca"] = "Ford";
    $auto["anio"] = 1992;

    var_dump($miArray);
?>
```

Arrays asociativos

- **Array**

```
<?php
    $auto = [];
    $auto["color"] = ["Negro", "Verde"];
    $auto["marca"] = "Ford";
    $auto["anio"] = 1992;
    $auto[] = "Prueba";

    var_dump($auto);
?>
```

Arrays asociativos

- **Array**

```
<?php
    $auto = [
        "color" => ["Negro", "Verde"],
        "marca" => "Ford"
    ];

    var_dump($auto);
?>
```

¡A practicar!

Ejercicios del **3** al **5**



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

PHP

Operadores



- **Numéricos**

- +
- -
- *
- /
- %
- **

- **Strings**

- .

Operadores

```
<?php
    $nombre = "Juan";
    $apellido = "Perez";

    echo "Me llamo " . $nombre . " " . $apellido;
?>
```


Operadores

```
<?php
    $nombre = "Juan";
    $apellido = "Perez";

    echo "Me llamo $nombre $apellido"
?>
```

¡Solo vale con comillas dobles!

PHP

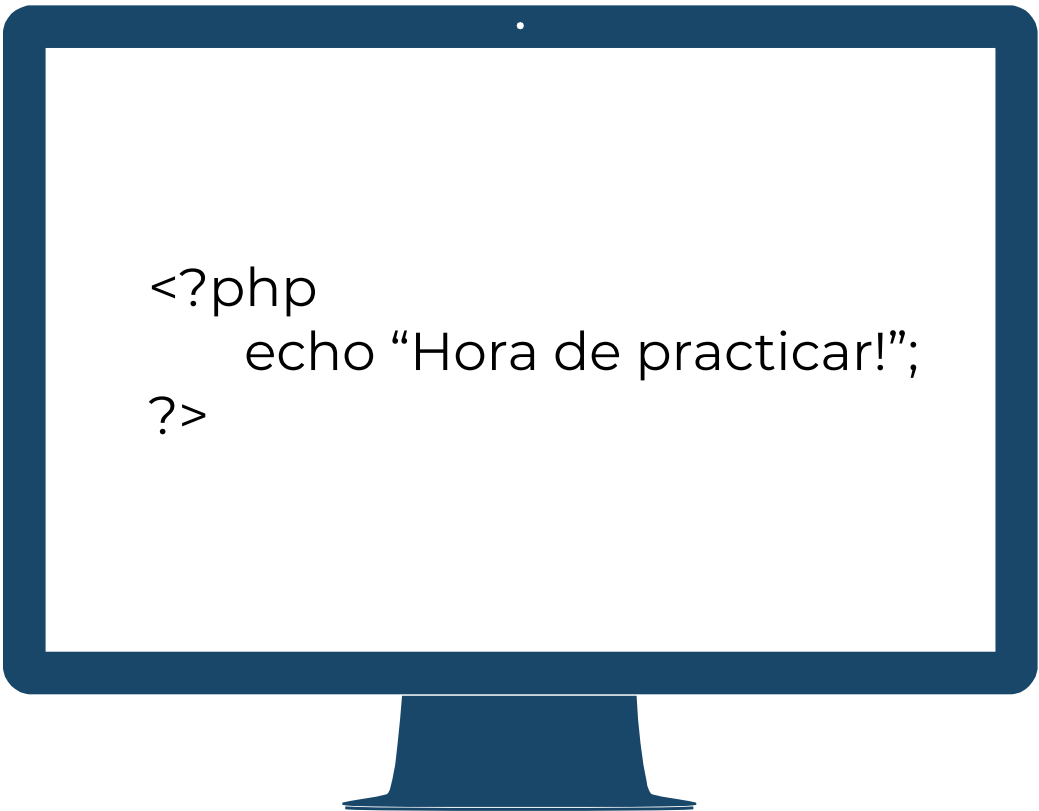
Caracteres escapados



- `\n` (sólo para uso en consola)
- `\"`
- `\\`
- `\$`

¡A practicar!

Ejercicios **6** y **7**



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



¡Gracias!



¿Preguntas?