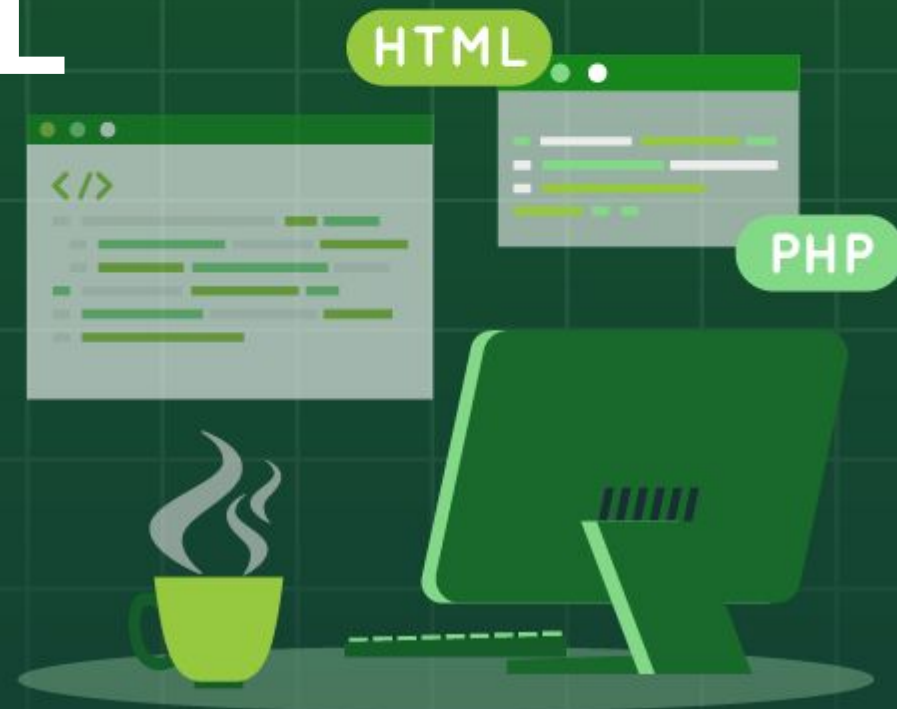



Curso de Integración de PHP con HTML

@RetaxMaster





¿Cómo se renderiza una página web en mi computadora?

Tipos de render

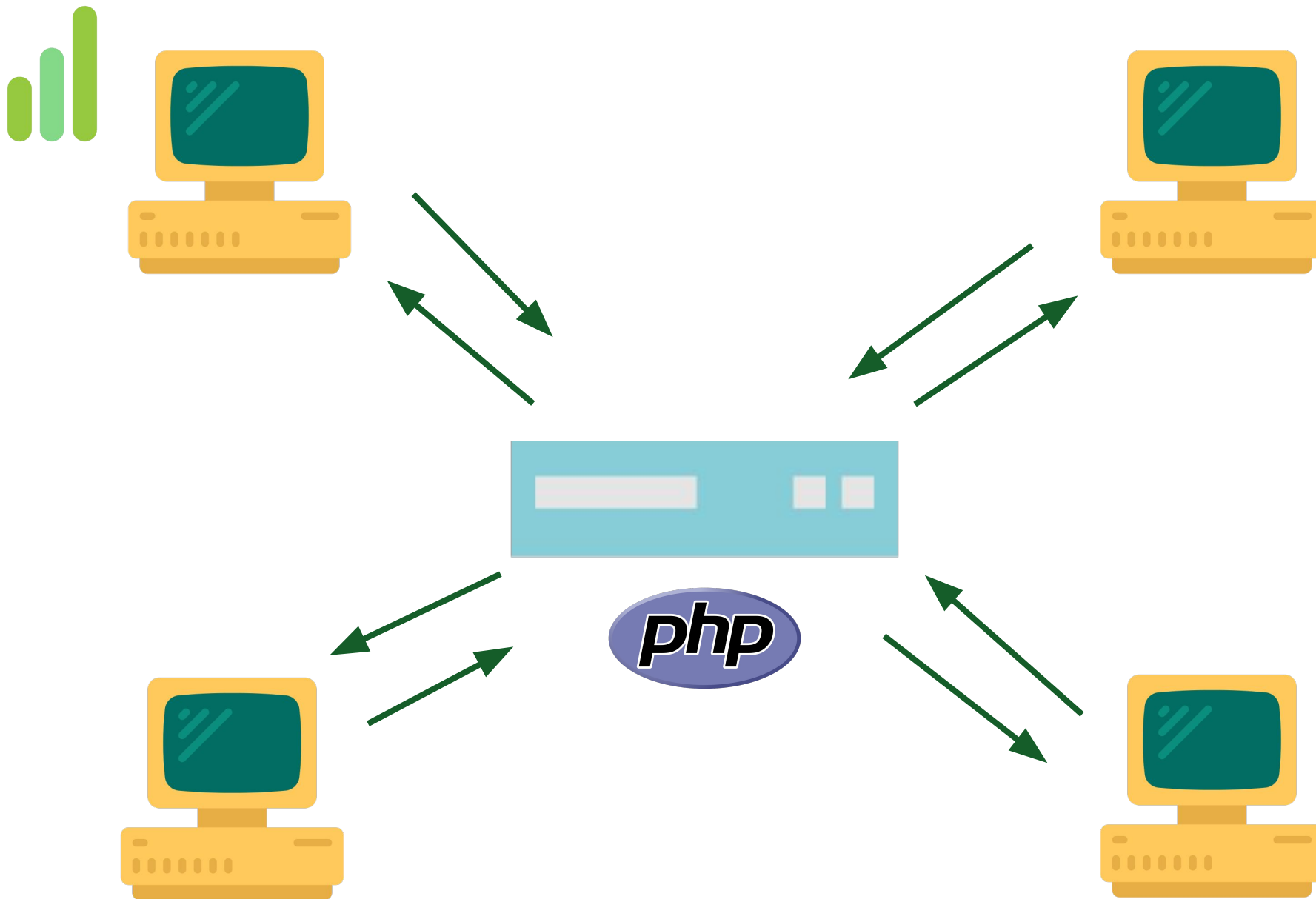




Cliente y servidor

Toda nuestra aplicación está guardada en un servidor, el cual entrega una copia de la misma a cada cliente que la solicite.

Además, el servidor también se encarga de responder cada solicitud del usuario.



Tipos de render

¿SSR? ¿CSR?

¿SSG?





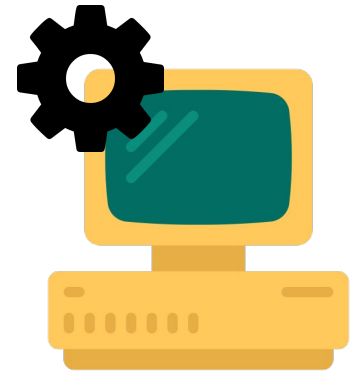
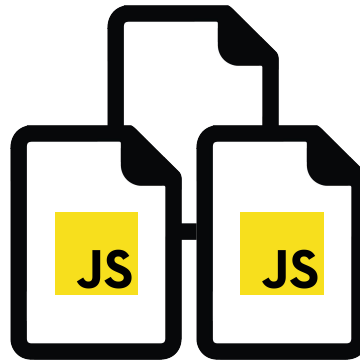
Client Side Rendering (CSR)

Es el tipo de renderizado más común en librerías como React.js o Vue.js. Aquí es el navegador del cliente quien, por medio de código JavaScript, se encarga de armar toda la estructura de la página web.



¿Cómo funciona?

1. El usuario ingresa a una página web.
2. El servidor manda una página web vacía (sin contenido) junto con archivos JavaScript.
3. Dichos archivos hacen peticiones asíncronas al servidor para obtener la información a mostrar.
4. El navegador arma toda la estructura de la página web.





Static Site Generation (SSG)

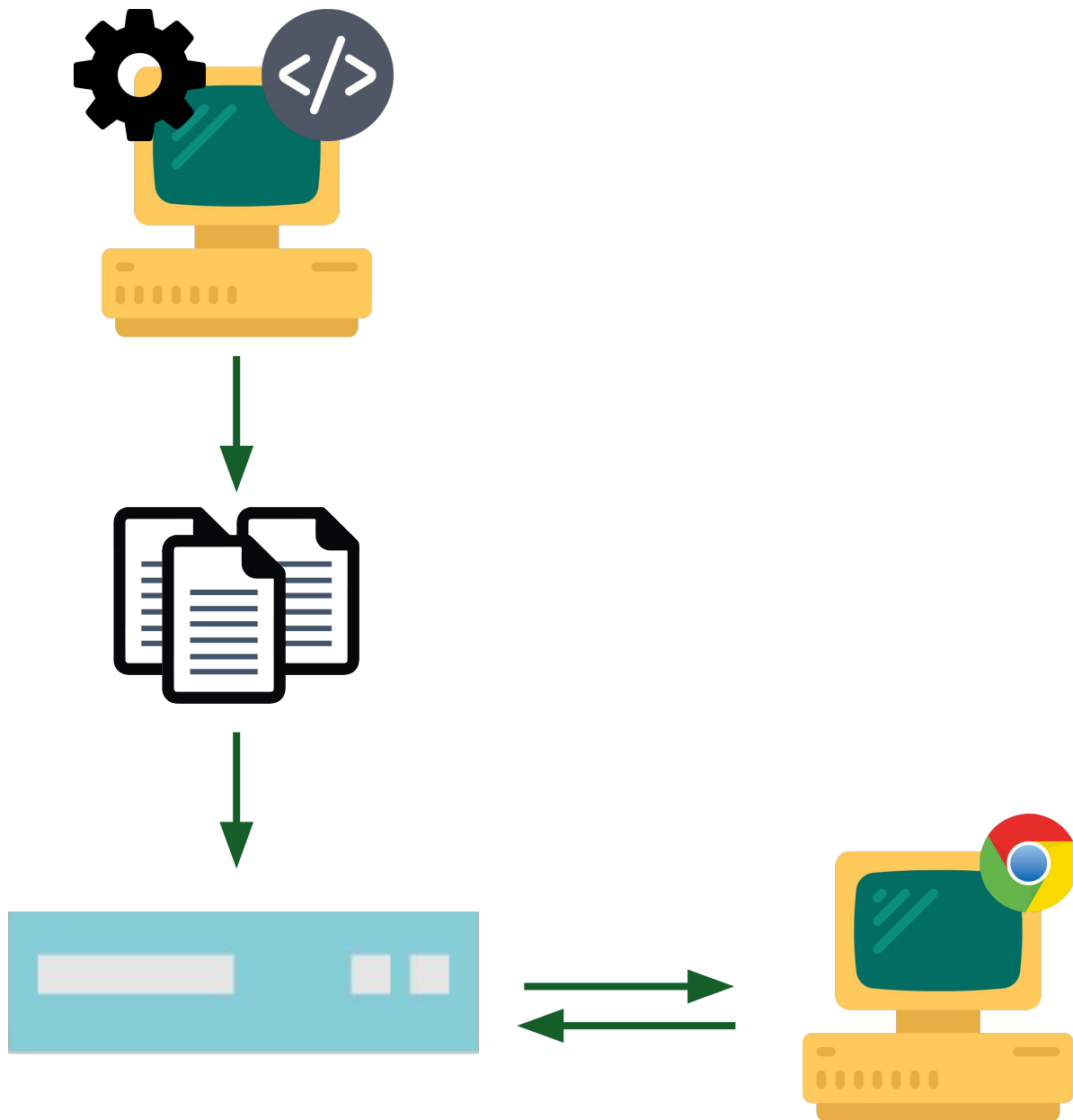
Este tipo de renderizado se hace desde que construimos nuestra aplicación en tiempo de desarrollo. Es común verlo cuando trabajamos con frameworks como Next.js o Nuxt.js.

Todo el HTML es construido en tiempo de *build*, es decir, desde que desarrollamos, lo que hace que no sea necesario un servidor.



¿Cómo funciona?

- Hacemos un *build* de nuestra aplicación.
- El framework en cuestión generará plantillas HTML prerenderizadas.
- En cada petición, nuestro servidor mandará dichas plantillas sin tener que procesar nada.
- La plantilla se renderiza en el navegador.





Server Side Rendering (SSR)

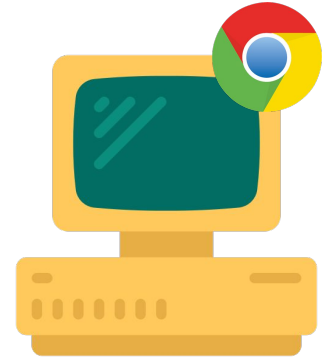
Es el tipo de renderizado más común en la web. Este renderizado se basa en que es nuestro servidor quien procesa todo el código y manda todo el código ya procesado a nuestro navegador.

Es el más común en lenguajes backend.



¿Cómo funciona?

- El usuario hace un request al servidor.
- El servidor ejecuta todo el código backend y genera las plantillas necesarias por cada request.
- La página se envía al navegador.



**Este último
es el rendering
de PHP**





Te doy la bienvenida 🤘



RetaxMaster 🖥️ ❤️
@retaxmaster

Course Director at @platzi ❤️ Fullstack Web Developer, Software Engineer, Backend, Frontend, I love to code!! Making videos to teach code 🛠️

[Traducir la biografía](#)

[linktr.ee/RetaxMaster](#) 📅 Se unió en diciembre de 2011

111 Siguiendo 2.302 Seguidores

Editar perfil



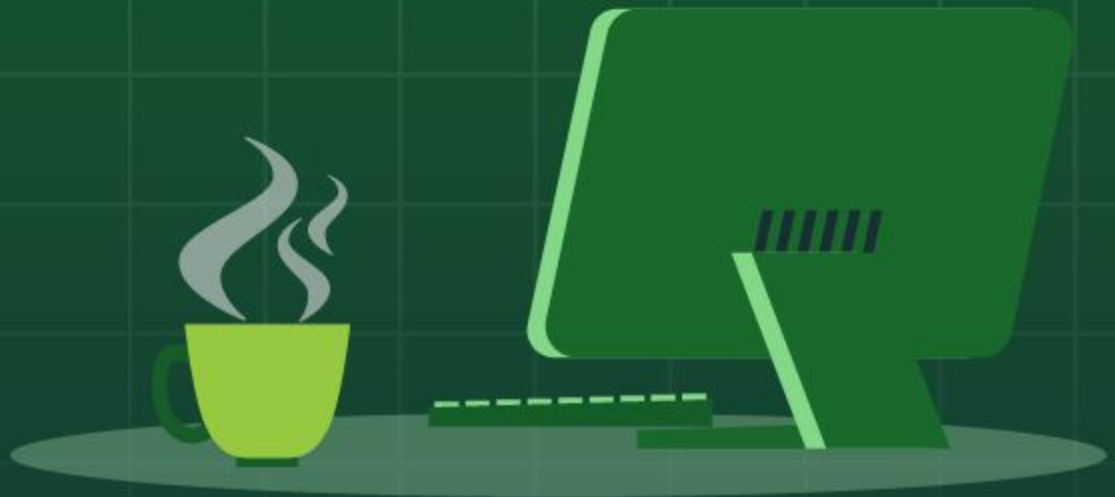
¿Quieres aprende a combinar PHP con HMTL?



End of class

- No eliminar please :3 -

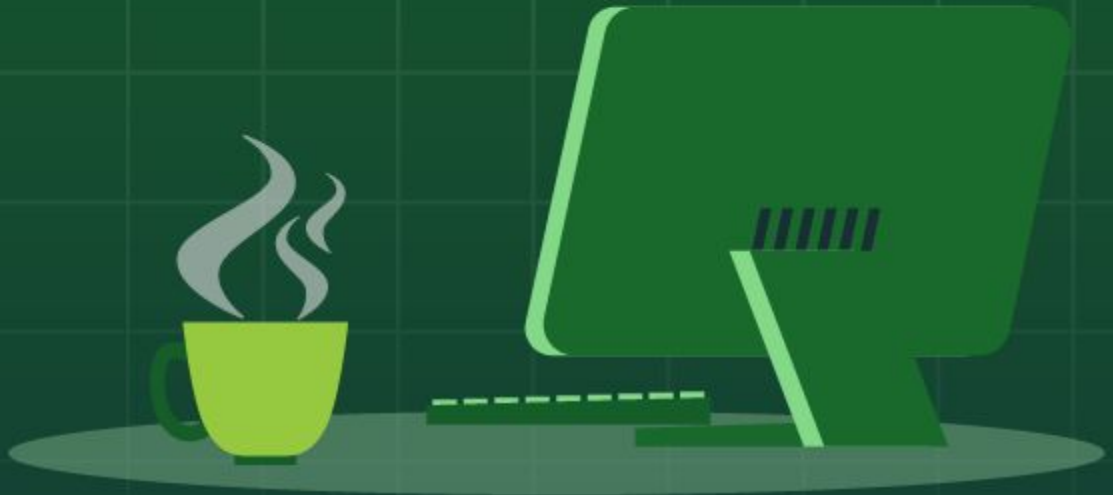
Levantando un servidor local



End of class

- No eliminar please :3 -

PHP como preprocesador de HTML





Consideraciones

Cuando empezamos a combinar archivos PHP con HTML es importante empezar a definir nuestras etiquetas de cierre. De esta forma PHP sabe cuándo está ejecutando código PHP y cuándo está ejecutando HTML.

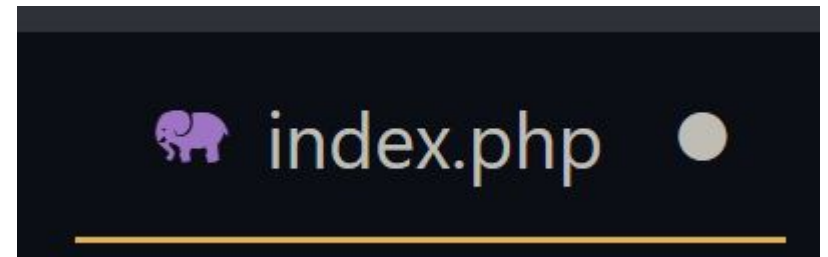
```
<?php
    // Todo nuestro código va aquí
?>

<!DOCTYPE html>
<html lang="es">
    <head>
        <meta charset="utf-8">
        <title>Mi asombrosa página</title>
    </head>
    <body>
        <h1>¡Hola, mundo!</h1>
    </body>
</html>
```



Consideraciones

Aunque estemos trabajando con HTML, nuestros archivos seguirán siendo archivos .php. Es por eso que PHP, a diferencia de otros lenguajes, tiene etiquetas de apertura y cierre, porque un mismo archivo soporta dos lenguajes 😊.



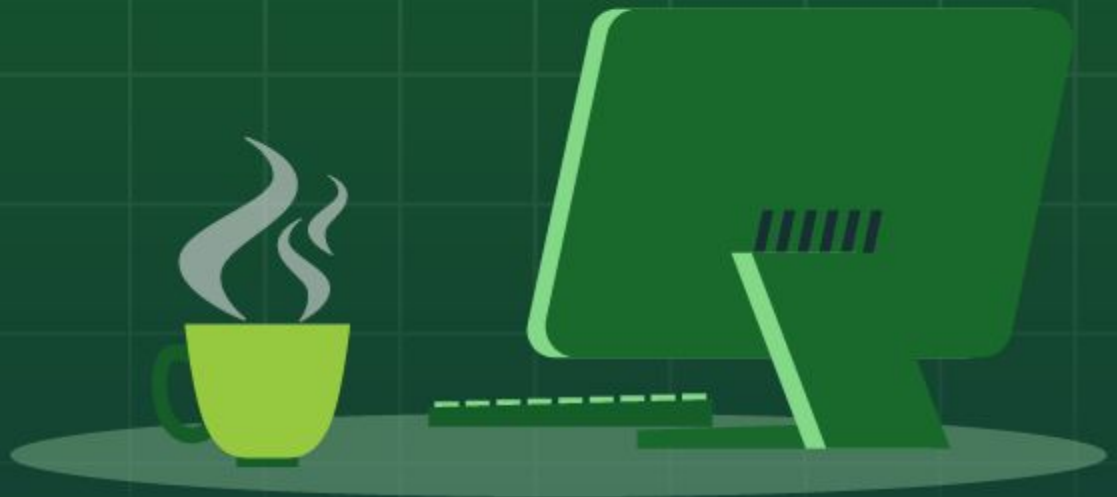
¡Vamos al código!



End of class

- No eliminar please :3 -

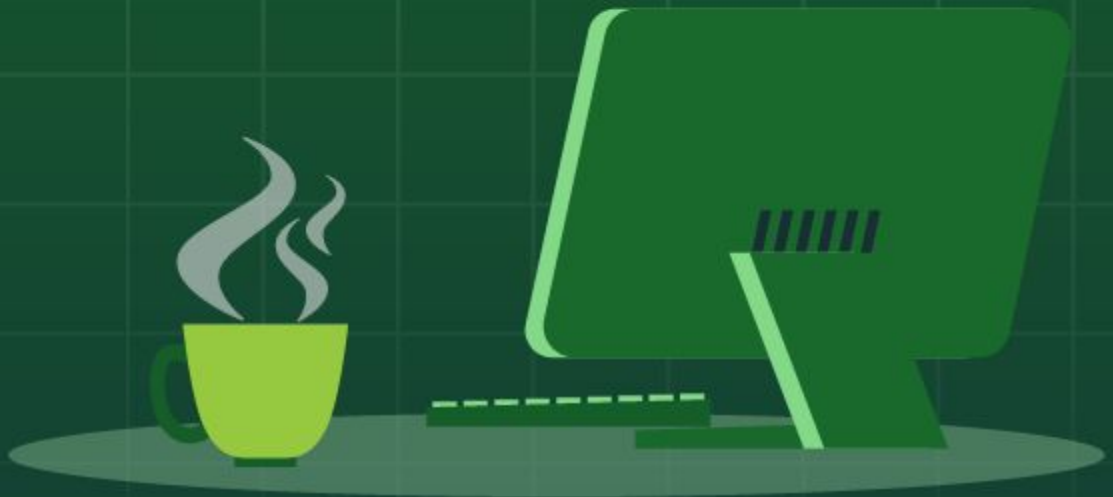
Imprime texto y etiquetas HTML



End of class

- No eliminar please :3 -

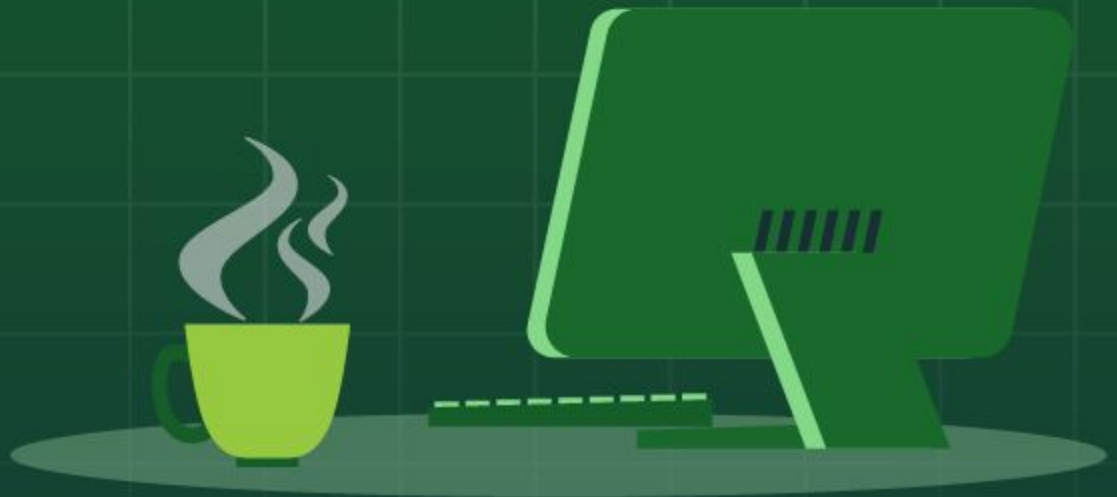
Condicionales \leftrightarrow



End of class

- No eliminar please :3 -

Ciclos



End of class

- No eliminar please :3 -

¿Cómo pasar
variables de PHP
a JavaScript? 🤔





¿Recuerdas qué es el SSR?

PHP preprocesa todo lo que escribas dentro de las etiquetas de PHP. Esto significa que PHP es capaz incluso de preprocesar código JavaScript, permitiéndote pasar tus variables de PHP a JavaScript.

Sí, puedes escribir código JavaScript usando PHP 😏.

¡Vamos al código!





¿Y si quiero pasar variables de JavaScript a PHP?

Puedes, pero tal vez necesites replantearte por qué quieres hacerlo, ya que esto es tan simple como hacer una petición asíncrona al servidor.

Recuerda que PHP solo se ejecuta en tiempo de carga, mientras la página es preprocesada, JavaScript se ejecuta **después de que la página fue cargada**.



Ten en cuenta las buenas prácticas

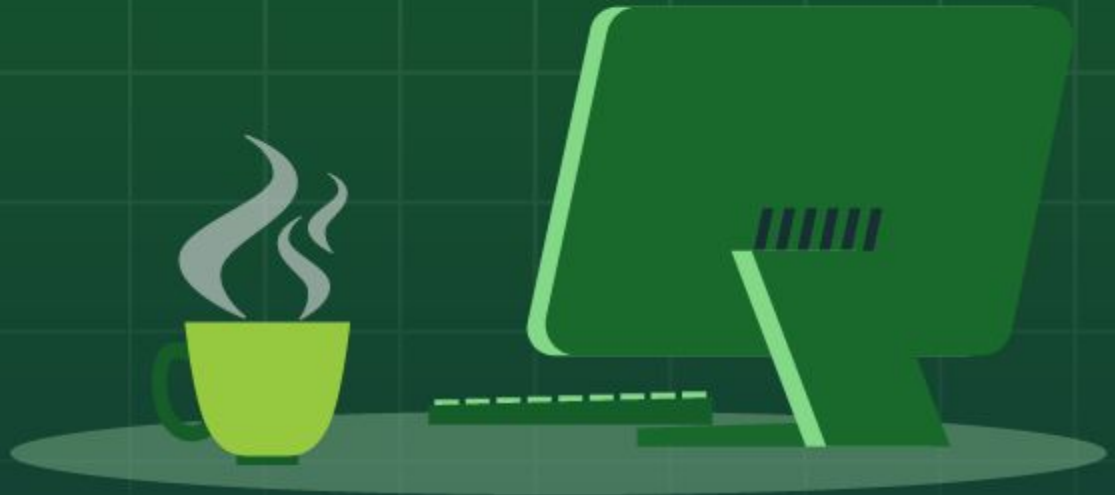
Aunque esto es posible, recuerda tener en cuenta buenas prácticas al momento de escribir código.

Esto lo veremos en la siguiente clase. 😊

End of class

- No eliminar please :3 -

**Evita el código
espagueti** 🍝





```
<?php
```

```
if($a <=> $b){
```

```
    echo '<div class="card">
        <span class="hour">00:00</span>
    ';
```

```
    if ($Cheese == "tool") echo "<UL>
    <LI>uno</LI>
    <LI>dos</LI>
    </UL>";
```

```
    echo '
    </div>';
```

```
}else{echo "<p>No hay resultados</p>";}
```




```
<?php
    $personas = ["Juan", "Pedro", "Luis"];
?>

<!DOCTYPE html>
<html lang="es">
    <head>
        <meta charset="utf-8">
        <title>Mi asombrosa página</title>
    </head>
    <body>

        <ul>
            <?php foreach($personas as $persona): ?>
                <li><?= $persona ?></li>
            <?php endforeach; ?>
        </ul>

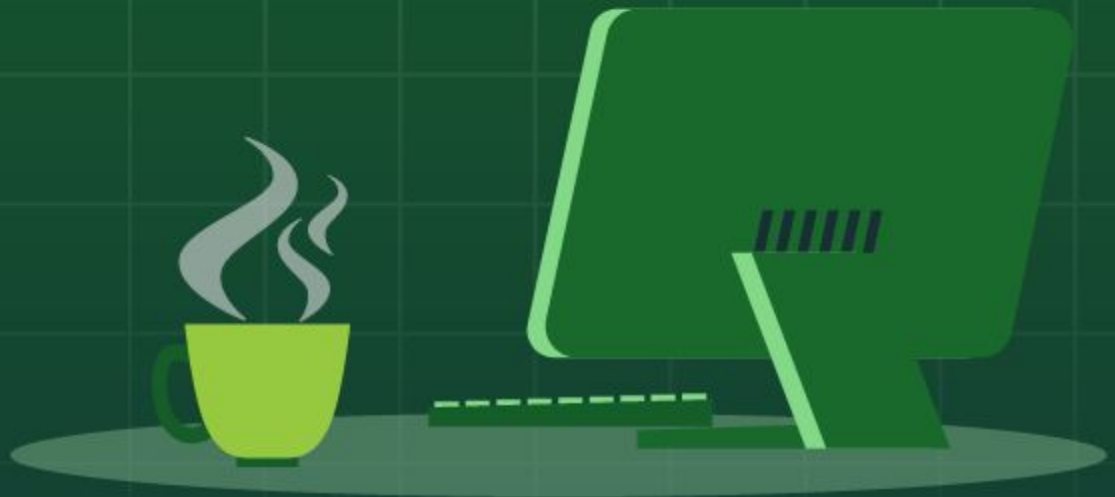
    </body>
</html>
```



Sigue buenas prácticas

- Usa nombres de variables descriptivos.
- No combines lógica PHP con HTML.
- Usa las etiquetas de PHP diseñadas para imprimir en HTML.
- Deja toda la lógica al inicio del documento y solo imprime los resultados en HTML.
- No abuses de la libertad que te da PHP.
- Recomendaciones futuras.

Refactorizando código espagueti



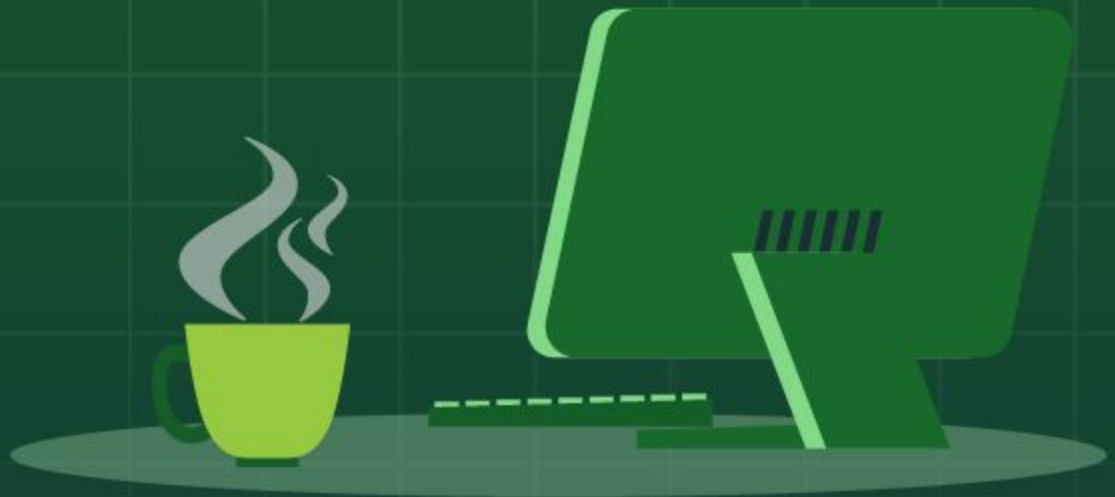
¡Vamos al código!



End of class

- No eliminar please :3 -

¿Cómo obtener una
solicitud al servidor
con PHP? 📧





Ámbito de las variables (scope)

La mayoría de variables de PHP tienen un ámbito simple, es decir, una variable puede estar definida en la mayoría del código, incluso dentro de otros archivos.

Pero existen variables que, sin importar en dónde te encuentres, siempre las tendrás definidas. Estas son llamadas ✨ variables superglobales ✨



Variables superglobales

PHP define variables superglobales a través de las cuales podemos acceder a cierta información desde cualquier parte del código:

- `$_GET`
- `$_POST`
- `$_REQUEST`





Este método permite solicitar información al servidor, pero también nos permite enviar información a través de la URL.

<http://www.platzi.com> 🔍



POST

- Este método permite guardar información. Por ejemplo, podemos recabar datos del usuario desde un formulario y mandarlos a nuestro servidor para procesarlos.
- Los datos se envían por “detrás” de la página, en el cuerpo de la petición.

¡Vamos al código!



End of class

- No eliminar please :3 -

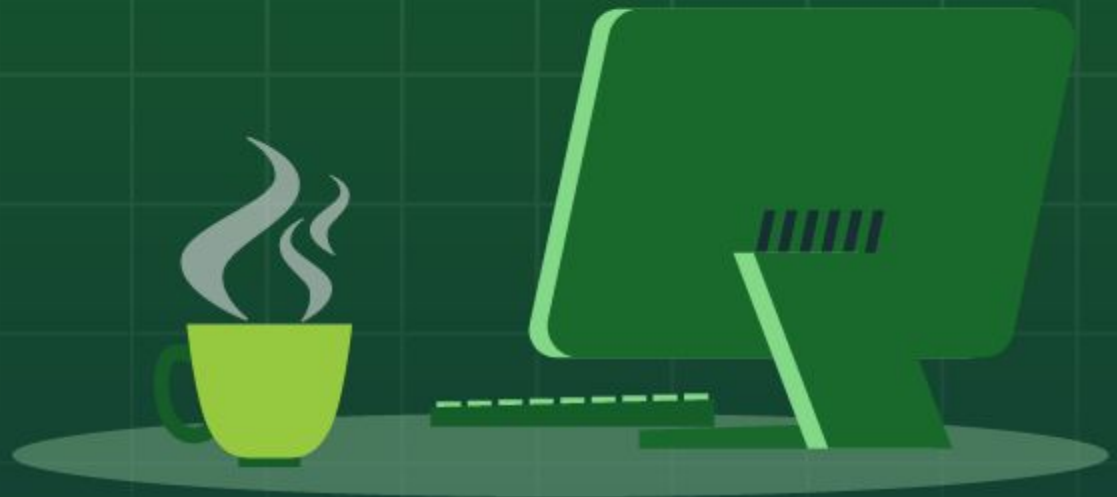
Envío de un formulario a través de GET



End of class

- No eliminar please :3 -

Envío de un formulario a través de POST



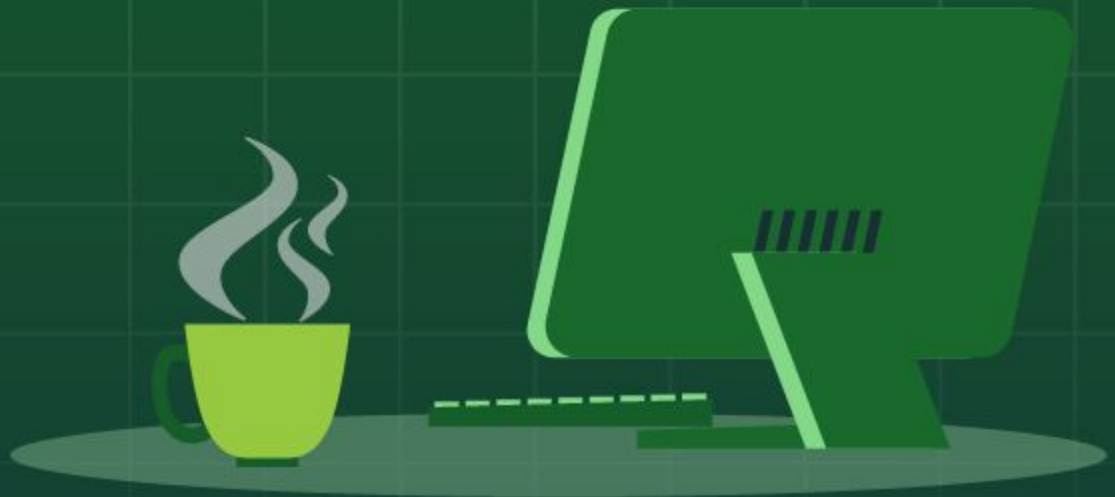
¡Vamos al código!



End of class

- No eliminar please :3 -

Envío de imágenes





Variables superglobales

PHP define variables superglobales a través de las cuales podemos acceder a cierta información desde cualquier parte del código:

- `$_GET`
- `$_POST`
- `$_REQUEST`
- **`$_FILES`**



Cualquier archivo subido y enviado al servidor es guardado en una carpeta temporal que se limpia cada cierto tiempo.

La variable `$_FILES` contiene información en forma de array acerca del archivo que estamos manipulando, incluyendo su ruta a la carpeta temporal.



Multipart Form Data

Por defecto, al enviar formularios en HTML tenemos un tipo de encriptación

“application/x-www-form-urlencoded”, sin embargo, cuando trabajamos con archivos debemos cambiar este tipo a ***“multipart/form-data”***.

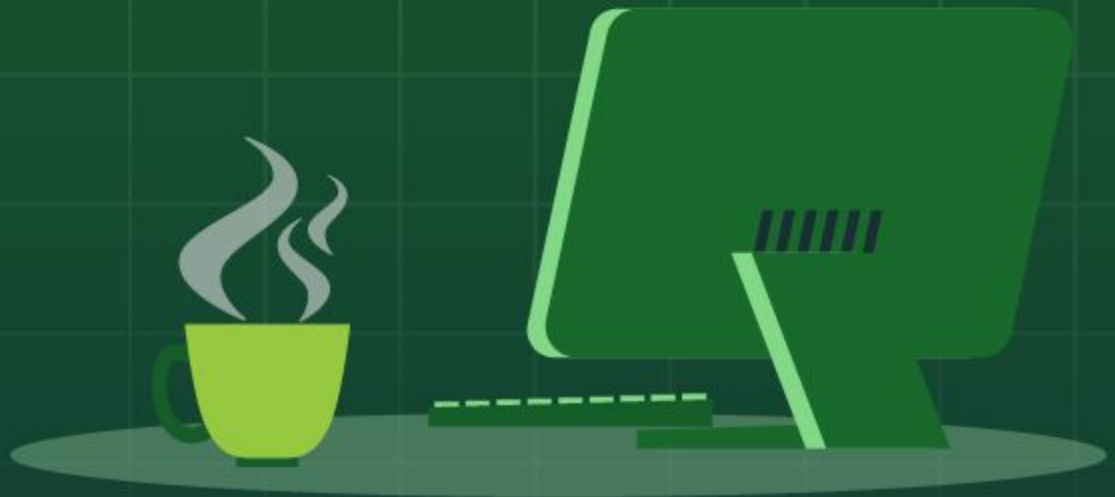
¡Vamos al código!



End of class

- No eliminar please :3 -

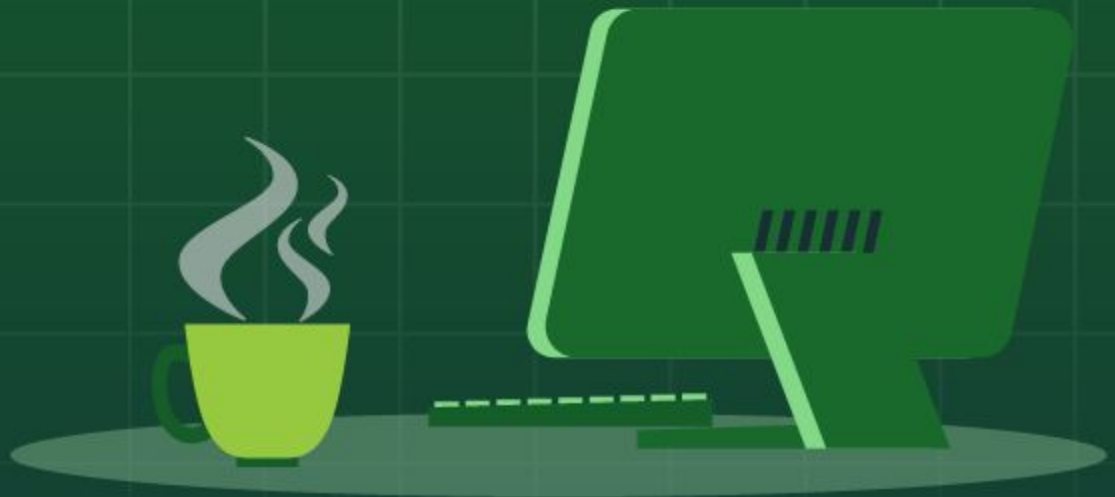
Guardando imágenes en el servidor

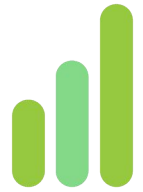


¡Vamos al código!



Tipos de input





El atributo name

El atributo name identificará a tu campo en una solicitud.



```
<input type="text" name="Nombre" />
```

¿Qué tipos de campos puedo mandar?





Input simple

En este input podemos mandar un único campo con un único valor.



```
<input type="text" name="Nombre" />
```



Arreglos

Podemos usar varios inputs para generar un arreglo.



```
<input type="text" name="personas[]" />  
<input type="text" name="personas[]" />  
<input type="text" name="personas[]" />
```



Arreglos asociativos

De igual forma, podemos asignar claves a los arreglos para crear arreglos asociativos.



```
<input type="text" name="persona[nombre]" />  
<input type="email" name="persona[email]" />  
<input type="text" name="persona[telefono]" />
```




Checkbox

Podemos usar inputs de tipo checkbox para mandar al servidor.



```
<input type="checkbox" name="list1" />  
<input type="checkbox" name="list2" />  
<input type="checkbox" name="list3" />
```



Radios

Podemos usar inputs de tipo radio para mandar al servidor.



```
<input type="radio" name="pais" />  
<input type="radio" name="pais" />  
<input type="radio" name="pais" />
```



Múltiples archivos

También podemos mandar múltiples archivos los cuales podremos recibir en forma de arreglo.



```
<input type="file" multiple name="fotos[]" />
```

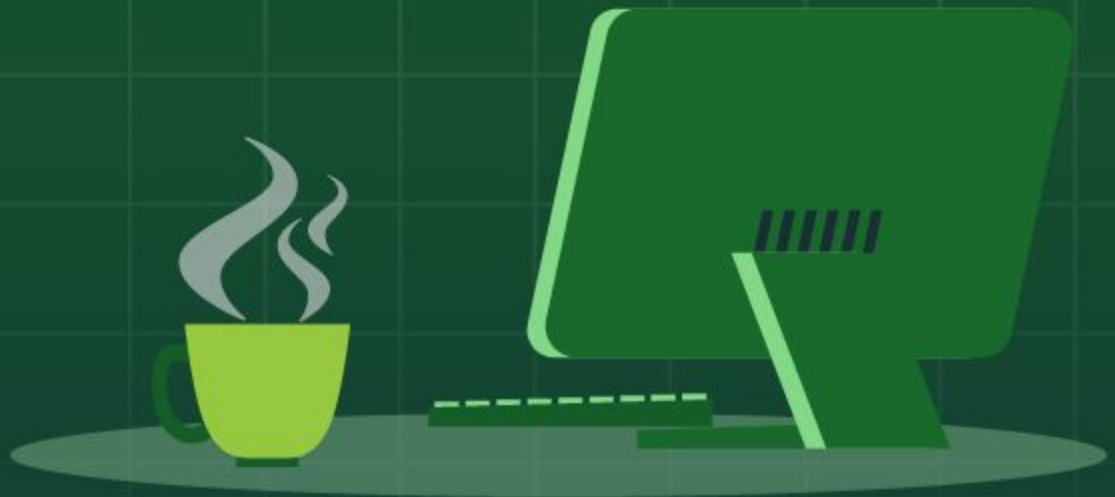
¡Vamos al código!



End of class

- No eliminar please :3 -

**Valida si un
formulario ha
sido enviado**



End of class

- No eliminar please :3 -

Sanitizando datos de mi formulario



End of class

- No eliminar please :3 -

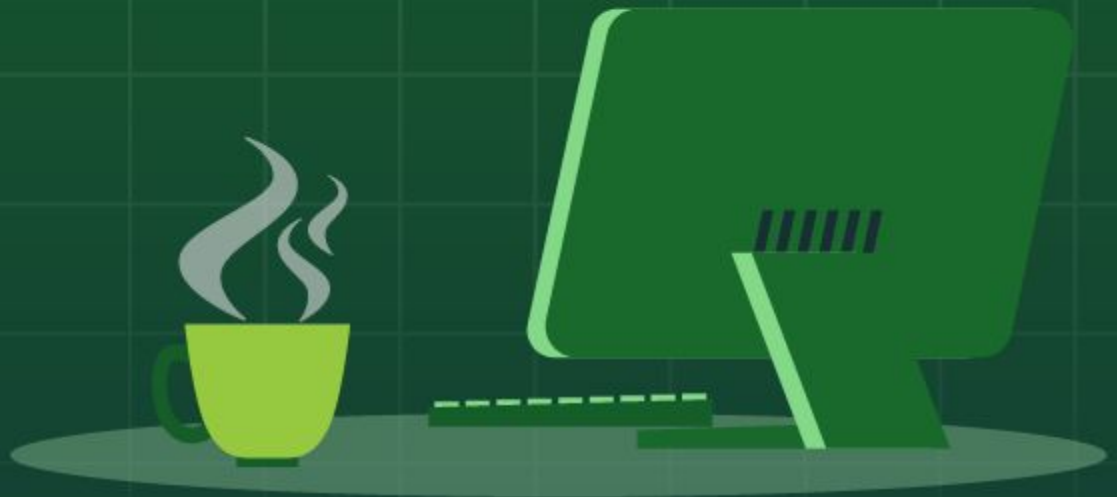
Validando
datos de mi
formulario



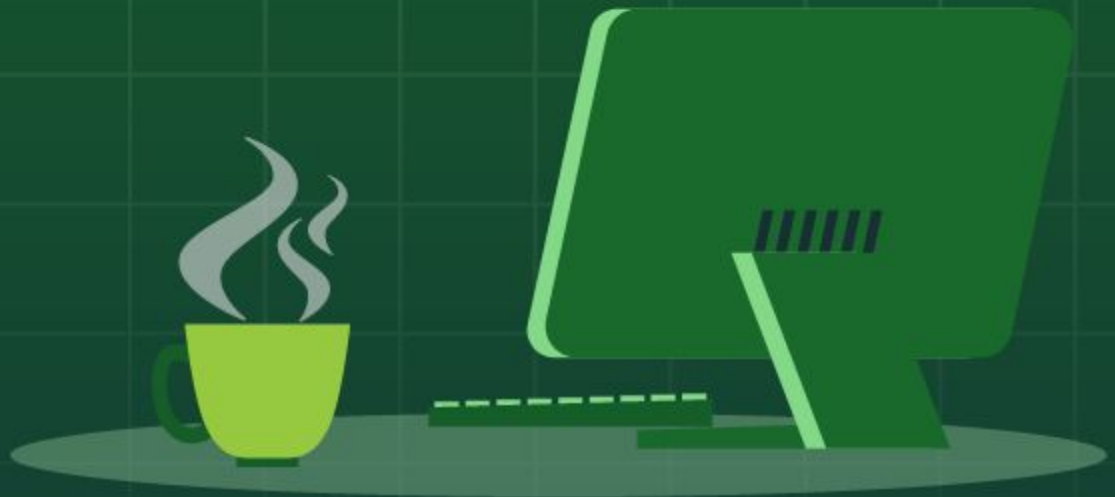
End of class

- No eliminar please :3 -

¡Hora de hacer
un proyecto! 🤠



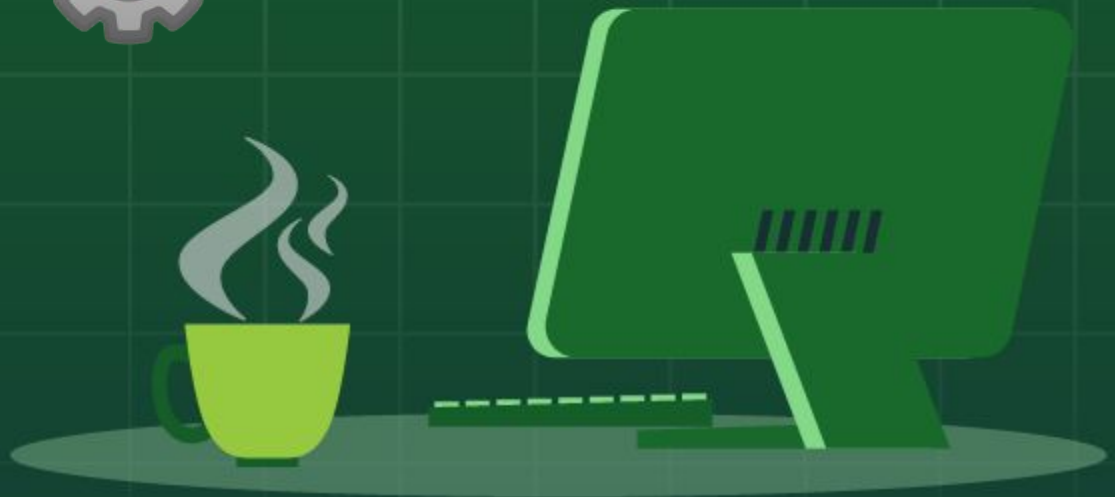
Creando el maquetado



End of class

- No eliminar please :3 -

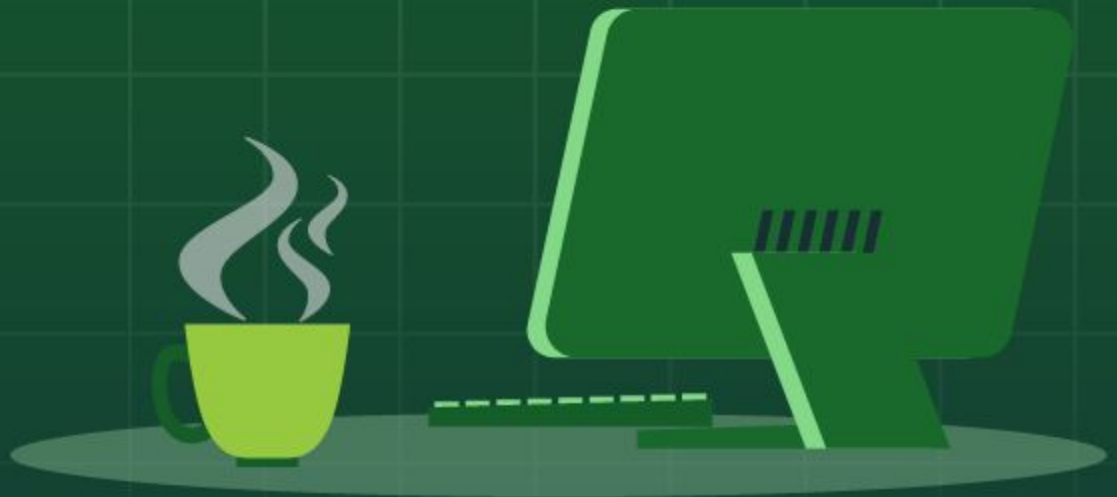
Haciendo la lógica de recepción del formulario



End of class

- No eliminar please :3 -

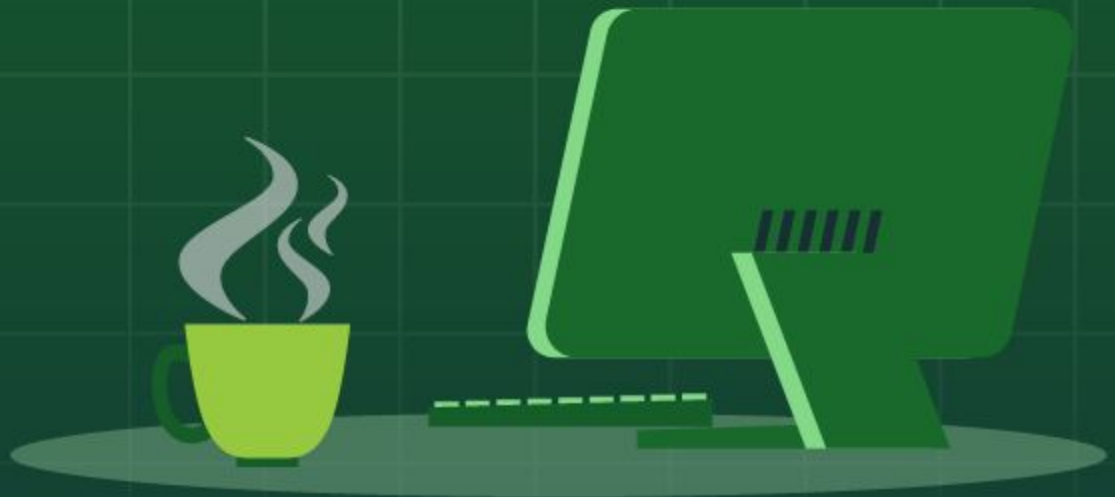
Integrando PHPMailer



End of class

- No eliminar please :3 -

Implementando el servidor de correos de Gmail



**¿Un motor de
plantillas... para un
motor de plantillas?**



End of class

- No eliminar please :3 -

Es tu turno de
poner todo en
práctica 💪





¿Qué aprendiste?

- Cómo se renderizan las páginas web.
- Cómo PHP nos ayuda a preprocesar HTML para crear páginas dinámicas.
- Seguir buenas prácticas.
- Trabajar con formularios.
- Manejar solicitudes simples HTTP.
- Creaste un proyecto.



¿Qué sigue?

¡Es hora de poner en práctica todo lo que has aprendido!

¿Qué ideas de proyectos se te ocurren? Te reto a crearla con todo lo que sabes hasta ahora 😊.



¿Quieres más cursos de PHP?

Aún hay más temas de PHP que nos faltan por aprender...

¿Cómo trabajamos con Cookies y sesiones?

¿Cómo nos conectamos a una base de datos?



 ¡Nos vemos! 



@RetaxMaster



/RetaxMaster



/RetaxMaster



@RetaxMaster

