

---

Produced by [the CoderDojo Foundation](#) // [@CoderDojo](#)

1

PHP está diseñado para generar páginas web dinámicas sobre servidores web. Por ello lo primero que debemos saber cómo ejecutar nuestro código. Hay varias opciones:

- **Editor online:** Hay varias opciones, por ejemplo tenemos Cloud 9 (<http://c9.io>), que nos permite editar y ejecutar nuestro código desde la misma herramienta.
- **Servidor externo:** Tendrás que instalar PHP, Apache y un servidor FTP para permitir que cada ninja suba su código y poder ejecutarlo.
- **Servidor en local:** Otra alternativa es usar un servidor LAMP, WAMP o XAMP instalado en el ordenador personal de cada ninja.

Si usas Windows o Mac también puedes usar Docker Kitematic y buscar la instancia "php-nginx". Docker es un entorno que te permite ejecutar servidores virtualizados en tu ordenador, con todo listo y configurado.



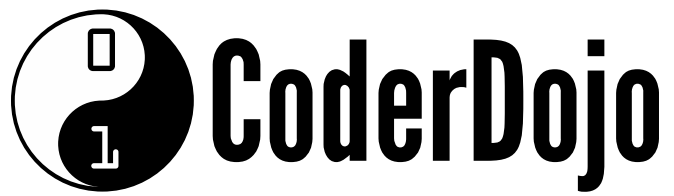
---

Produced by [the CoderDojo Foundation](#) // [@CoderDojo](#)

2

Dependiendo de la opción que escojas puedes necesitar algunas herramientas adicionales.

- **Editor online:** tienes todo lo que te hace falta, así que ¡a comenzar!.
- **Servidor externo:** Necesitarás un cliente FTP para conectarte al servidor y poder subir tus archivos. También te hará falta un editor de código. Te recomendamos Atom ([atom.io](#))
- **Servidor local:** Tendrás que poner tus archivos en una carpeta determinada, que dependerá del sistema que hayas escogido y del sistema operativo. También te hará falta aun editor de código.



Produced by the CoderDojo Foundation // @CoderDojo

3

Como ejemplo te mostraremos cómo ejecutar un sencillo script en en Cloud 9, que es la opción que recomendamos, porque será la que te permita empezar más rápidamente.

- Crea una cuenta en [Cloud 9](http://c9.io) (<http://c9.io>)
- Entra en la plataforma
- Crea un nuevo proyecto (workspace) y escoge la plantilla (template) "PHP, Apache & ...". Puedes nombrarlo con el nombre que prefieras.
- Verás a la izquierda que hay varios archivos, incluido uno llamado `hello-world.php`. Ábrelo.
- Arriva a la derecha verás un botón de ejecutar (play). Haz clic para ejecutar tu proyecto.
- Verás un mensaje abajo indicando que Apache se ha iniciado y te mostrará un url (dirección web) donde poder ver el resultado de tu programa. Copia este enlace en una nueva pestaña.

¡Felicidades! Acabas de ejecutar tu primer script de PHP.

Veamos cómo podemos crear una web interactiva con este entorno.

# SUSHI - INICIACIÓN A PHP



## RESUMEN DEL PROYECTO

Card 2 of 9  
I'm Learning: Php

Produced by [the CoderDojo Foundation](#) // [@CoderDojo](#)

1

¿Estás ilusionado por aprender un nuevo lenguaje de programación? Veamos primero algunos datos importantes sobre PHP.

- PHP se ha estado usando desde su aparición en 1995
- PHP está instalado en más de 244 millones de páginas web y 2 millones de servidores
- el 75% de las páginas de la Web 2.0 están construidas con PHP (como Facebook, Flickr, Wikipedia y Yahoo!)

2

Esto es lo que vamos a aprender.

- Algunos elementos básicos de HTML, ya que habitualmente PHP y HTML deben trabajar juntos.
- Generar algunas cadenas de texto ( **strings** ) con PHP.
- Hacer algunas operaciones matemáticas programando.
- Aprender algunos trucos para hacer tareas repetitivas con ayuda de los bucles.
- Recoger información por medio de formularios web.

3

A través de esta serie de Sushi Cards vas a aprender a programar un juego de "Adivina el número".

# SUSHI - INICIACIÓN A PHP



## RESUMEN DEL PROYECTO

Card 2 of 9  
I'm Learning: Php

Produced by [the CoderDojo Foundation](#) // [@CoderDojo](#)

4

Analicemos algunos detalles a partir del proyecto que se genera cuando comienzas con Cloud 9.

```
<html><body>
<?php
// A simple web site in Cloud9 that runs through
Apache
// Press the 'Run' button on the top to start the
web server,
// then click the URL that is emitted to the Output
tab of the console

echo 'Hello world from Cloud9!';

?>
</body>
</html>
```

5

Para indicar a PHP que debe ejecutar un archivo debemos asegurarnos de que tiene extensión `.php`

El código PHP Code se muestra junto con el html rodeándolo por medio de las etiquetas `<?php ... ?>`.

El código PHP puede ponerse en cualquier parte del archivo siempre que lo rodeemos de las etiquetas `<?php ?>`.

6

Las cadenas que aparecen entre los caracteres `<` `>` se llaman **etiquetas HTML**. Aprenderás algunas de ellas. Por ahora te basta saber que nos permiten organizar la información en una página web.

# SUSHI - INICIACIÓN A PHP



## RESUMEN DEL PROYECTO

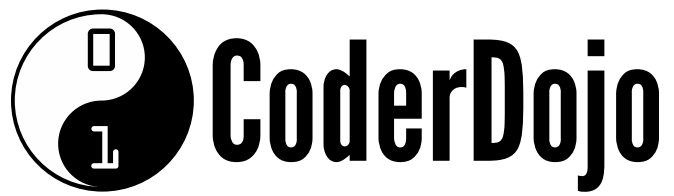
Card 2 of 9  
I'm Learning: Php

---

Produced by [the CoderDojo Foundation](#) // [@CoderDojo](#)

7

Habrás visto en el ejemplo que se utilizan los caracteres `//`. Esto indica que estamos creando un comentario, útil para ti y otros programadores que vean tu código. El intérprete de PHP ignorará todo lo que aparezca tras los caracteres de comentario.



Produced by [the CoderDojo Foundation](#) // [@CoderDojo](#)

1

Es el momento de mostrar al usuario las instrucciones de nuestro juego, así sabrá cómo jugar.

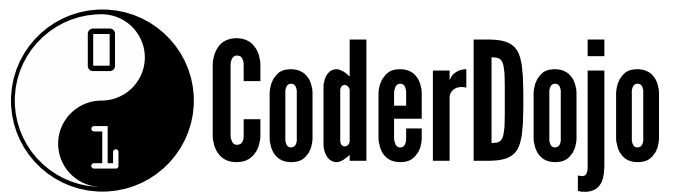
2

Para mostrar texto en **PHP** debemos usar la función `echo`.

La función `echo` se utiliza para mostrar un valor, en nuestro caso en el resultado de la página web. Puedes ver en el ejemplo original que el texto que se mostraba era 'Hello world from Cloud9'. Prueba a cambiar este texto y [recarga](#) la página usando la dirección que Cloud 9 te ha dado anteriormente.

En PHP cada línea de código debe terminar usando el carácter `;`. Esto indica al lenguaje que has terminado de escribir la instrucción.

Si te olvidas de usar este carácter se producirá un error sintáctico (**SYNTAX errors**) en PHP.



Produced by the CoderDojo Foundation // @CoderDojo

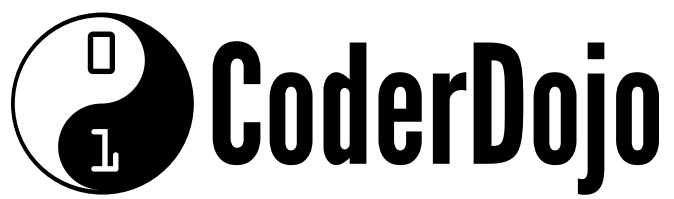
3

Ahora que has aprendido a mostrar instrucciones en el juego, prueba a cambiar la línea de la función `echo` con el siguiente código.

```
echo "<h1>iAdivina el número!</h1>";  
echo "Escoge un número entre 1 y 10.";
```

Acabas de mostrar 2 líneas en tu página web. Habrás notado que la apariencia de la primera línea es diferente a la segunda. Esto es porque estamos usando `h1`, una **etiqueta html** para crear títulos.





Produced by the CoderDojo Foundation // @CoderDojo

4

Añade otra línea para dejar al mundo claro que tu creaste esta página web.

```
echo "<p><code>" . "Creado por un ninja genial" .  
</code></p>";
```

Aquí puedes ver un par de cosas interesantes. La línea la hemos construido con diferentes **strings**, cada una de ellas delimitada por el carácter `"`. Hemos usado el carácter `.` para unir esas **strings** en una sola. Esto que acabas de aprender será importante en las siguientes tarjetas, pero por ahora debes saber que esto te permite separar un texto en diferentes líneas.

Hay otra **etiqueta html** muy útil, la etiqueta `p` indica que estamos creando un párrafo.

Fíjate que puedes usar tanto el carácter ``` como `"` para crear **strings**. Hay diferencias de usar una u otra, pero por ahora puedes usarlas indistintamente.

5

Si todo ha ido bien, veremos en la pantalla nuestros mensajes.

6

Si aparece algún error vuelve a comprobar tu código. Recuerda que cada línea debe acabar con punto y coma (`;`).



Produced by the CoderDojo Foundation // @CoderDojo

1

Las variables nos permiten guardar datos para usarlos después. En este ejemplo lo usaremos en la línea donde nos damos crédito.

2

Vamos a separar la línea y crear una variable que contenga nuestro nombre.

```
$autor = "un ninja genial";  
echo "Hecho por " . $autor;
```

En PHP se utiliza el carácter **\$** al inicio del nombre de las variables.

3

Intenta cambiar el texto de la variable. ¿Qué crees que va a pasar?

4

Las variables pueden ser de diferente tipo, además de texto (**string**), puede usar números (**integer**).

Algunos ejemplos de tipos de variable son:

- **string** - "Ma"
- **integer** - 1
- **boolean** - true / false

Estas otras no las usarás en el tutorial, pero así conoces otros tipos de variables.

- **arrays** - array(1,2,3,4,5)
- **double** - 1.000000002



Produced by [the CoderDojo Foundation](#) // [@CoderDojo](#)

5

No hay nada diferente que tengas que hacer cuando muestras el valor de diferentes tipos de variables. Vamos a introducir un par más y hacer una pequeña modificación en el código.

```
$numero_menor = 1;  
$numero_mayor = 10;  
echo "Escoge un número entre {$numero_mayor} y  
{$numero_menor}.";
```

Puedes ver que hay diferentes modos de mostrar una **variable**. Además del método que ya conoces de usar el `.` puedes mostrar el valor de una variable poniéndola dentro de una cadena como se muestra `"{$variable}"`.



Produced by [the CoderDojo Foundation](#) // [@CoderDojo](#)

1

Vamos a ampliar la lógica de nuestro juego usando variables. El jugador debe adivinar un número que es seleccionado aleatoriamente por el ordenador. Creemos una variable y asignemos el valor **aleatorio**.

```
$secreto = rand(1, 10);
```

Estamos usando una **función** para obtener un número aleatorio entre 1 y 10 `rand(1,10)`. La función acepta dos números como **parámetros** que nos permite indicar el número inferior y superior entre los que escoger el número.

Cuando usas una **función** debes conocer qué tipo de variables y cuantas recibe la función. Si escribes mal los parámetros, recibirás un aviso (**Warning**) indicándote el error.

2

Prueba a cambiar los parámetros por las variables que creaste anteriormente y muestra el valor de `$secreto` para ver su valor.

Cada vez que recargues la página verás que este número es diferente.

```
$secreto = rand($numero_menor, $numero_mayor);
```



Produced by [the CoderDojo Foundation](#) // [@CoderDojo](#)

3

¿Te has fijado en que ya has estado usando otra función durante los ejemplos? **echo** es una función que te permite imprimir cadenas y variables. De hecho puedes usar esta función de estas dos formas:

```
echo $secreto;  
echo($secreto);
```

4

Las funciones normalmente hacen cosas complicadas y te devuelven un valor como resultado. Así que es interesante conocer algunas funciones para hacer cosas potentes en tus programas. En el siguiente ejemplo vamos a mostrar en qué momento se generó la página web.

```
$fecha = date("r");  
echo "Mi hora actual es {$fecha}";
```

El parámetro **"r"** le dice a la función **date** cómo queremos ver la fecha. Hay un montón de opciones, por ejemplo puedes pedir que te devuelva solamente la hora. Puedes ver los detalles en <http://php.net/manual/es/function.date.php>



Produced by [the CoderDojo Foundation](#) // [@CoderDojo](#)

1

En esta tarjeta vamos a explicar cómo pasar información desde las páginas al servidor. Los dos métodos habituales son GET y POST.

- **GET** Si pasamos la información por GET los datos están en la dirección web, por lo que podemos guardarla en los marcadores para usarla cuando queramos.
- **POST** Este método pasar la información de una forma más segura.

Vamos a usar el método **GET** para mandar al servidor el número que hemos adivinado.

2

Nuestra aplicación podrá leer estos datos usando la variable `$_GET` que es un array (una especie de lista de datos).

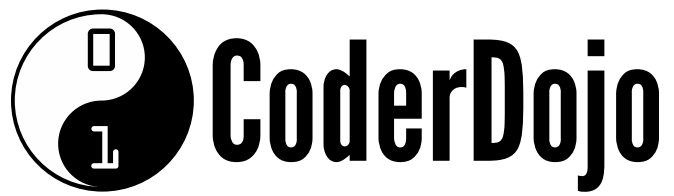
Estas variables se inicializan automáticamente para que puedas usarlas.

Añade esta línea a tu código y recarga la página.

```
print_r($_GET);
```

Ahora añade esto al final de la [url](#) de tu página web: `/hello-world.php?adivinar=3`. De esta forma creas una nueva variable con valor 3. Ahora verás que en tu página web se muestra algo como esto:

```
Array ( [adivinar] => 3 )
```



Produced by [the CoderDojo Foundation](#) // [@CoderDojo](#)

3 La variable `adivinar` es una clave en el array `$_GET`. Esto significa que los valores están asociados a claves dentro de la variable. Las claves pueden ser números o cadenas.

4 Puedes leer el valor de la variable de la siguiente forma.

```
$adivinar = $_GET['adivinar'];
```

5 Ahora puedes imprimir el valor del número seleccionado por el ordenador y por el usuario.

```
echo "<p>Tu predicción es {$adivinar}, el número  
correcto es {$secreto}</p>";
```

6 Para mejorar la usabilidad puedes añadir un campo de formulario para introducir el valor. Añade este código tras la etiqueta `?>`.

```
<form method="get">  
  <input name='adivinar'>  
</form>
```

Ahora puedes introducir el valor que quieres predecir y pulsar intro.



Produced by [the CoderDojo Foundation](#) // [@CoderDojo](#)

1

Podemos lograr mucho más de nuestro juego si le permitimos tomar decisiones, aquí es donde entran en juego los **Condicionales**.

Los bloques condicionales se componen de expresiones y sentencias. Las expresiones son evaluadas y si el resultado es evaluado como **TRUE** (verdadero) entonces las sentencias se ejecutan.

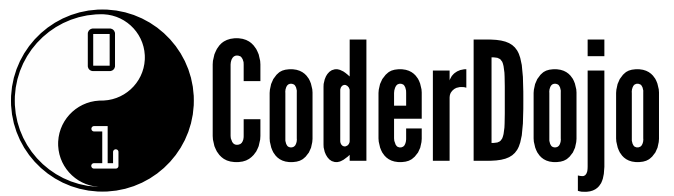
```
<?php
    if($adivinar == $secreto) {
        echo "¡Has encontrado el número!";
    }
?>
```

2

A veces debes ofrecer una alternativa cuando la expresión condicional no se cumple. En estos casos podemos usar la palabra reservada **else**. Esto indica que si la primera expresión es **false** (falsa), la secuencia del bloque que a continuación será ejecutada.

```
<?php
    if($adivinar == $secreto) {
        echo "¡Has encontrado el número!";
    } else {
        echo "¡Fallaste, vuelve a intentarlo!";
    }
?>
```





Produced by the CoderDojo Foundation // @CoderDojo

3

Por último, a veces quieres hacer diferentes comprobaciones. En estos casos usamos la palabra `elseif`, que nos permite evaluar otra expresión y ejecutar el bloque a continuación si se evalúa como cierto.

```
<?php
    if($adivinar == $secreto) {
        echo "¡Has encontrado el número!";
    } elseif($guess > $higher_number) {
        echo "Prueba con un número más pequeño";
    } else {
        echo "¡Fallaste, vuelve a intentarlo!";
    }
?>
```

Puedes usar tantos `elseif` como necesites, siempre deben aparecer entre el `if` inicial y el `else` final.



Produced by [the CoderDojo Foundation](#) // [@CoderDojo](#)

1

Por ahora el juego sólo da una oportunidad para adivinar el número. Esto se debe a que cada vez que recargamos la página el número secreto cambia. Pero podemos pasarlo entre diferentes ejecuciones y así tendremos varias oportunidades. Modifica tu formulario para que sea como éste:

```
<form method="post">
  <input type="hidden" name="secreto" value="<?
php echo $secreto; ?>" />
  <input name='adivinar' />
</form>
```

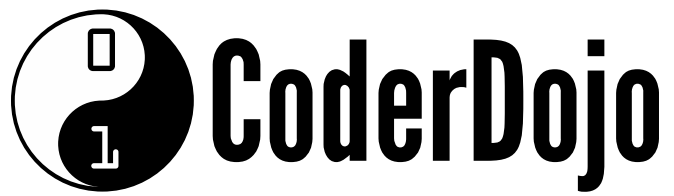
Fíjate en que hemos añadido un nuevo campo del tipo **hidden**, que permite mandar datos del formulario sin mostrarlos. También hemos cambiado el método de envío por **POST** para que no se puedan ver los parámetros en la url.

2

Con este cambio también debemos modificar el modo en que se recupera o genera el número secreto.

```
$secreto = $_POST['secreto'];
if(!$secreto) {
  $secreto = rand($numero_menor, $numero_mayor);
}
$adivinar = $_POST['adivinar'];
```

La instrucción **if** comprueba si la variable se ha pasado desde el formulario, si no el número se genera.



Produced by [the CoderDojo Foundation](#) // [@CoderDojo](#)

3

Si quieres limitar el número de intentos a un número determinado puedes crear otra variable y usar un poco de matemáticas para ir reduciendo el número de intentos.

```
$intentos = $_POST['intentos'];  
if(!is_null($intentos)) {  
    $intentos = $intentos - 1;  
} else {  
    $intentos = 3;  
}  
echo "Tienes {$intentos} intentos disponibles.";
```

4

Recuerda que debes añadir este nuevo campo al formulario.

```
<input type="hidden" name="intentos" value="<?php  
echo $intentos; ?>" />
```

# SUSHI - INICIACIÓN A PHP



## ÚLTIMOS AJUSTES

Card 9 of 9  
I'm Learning: Php

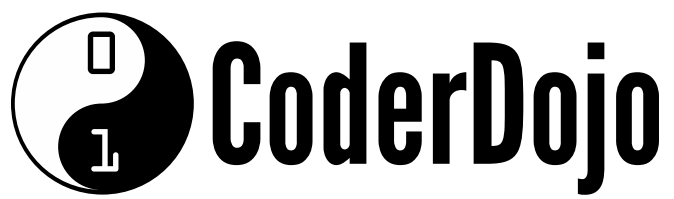
Produced by the CoderDojo Foundation // @CoderDojo

1

Aquí tienes todo el código del juego.

```
<html>
<body>
  <?php
    echo "<h1>iAdivina el número!</h1>";
    $n_menor = 1;
    $n_mayor = 10;

    // set game variables
    $secreto = $_POST['secreto'];
    if(!$secreto) {
        $secreto = rand($n_menor, $n_mayor);
    }
    $adivinar = $_POST['adivinar'];
    $intentos = $_POST['intentos'];
    if(!is_null($intentos)) {
        $intentos = $intentos - 1;
    } else {
        $intentos = 3;
    }
    echo "Tienes {$intentos} intentos.";
    if ($adivinar) {
        echo "<p>Tu número es {$adivinar}. ";
        if($adivinar == $secreto) {
            echo "¡Has encontrado el número!";
        } elseif($adivinar > $n_mayor) {
            echo "Intenta un número menor.";
        } else {
            echo "¡Prueba de nuevo!";
        }
        echo "</p>";
    }
  }
```



Produced by the CoderDojo Foundation // @CoderDojo

2

```
        } else {  
            echo "Escoge un número entre {$n_menor} y  
{$n_mayor}.";  
        }  
    ?>  
    <form method="post">  
        <input type="hidden" name="secreto" value="<?  
php echo $secreto; ?>" />  
        <input type="hidden" name="intentos" value="<?  
php echo $intentos; ?>" />  
        <input name='adivinar' />  
    </form>  
</body>  
</html>
```

3

Esto es lo que has aprendido en la serie de cartas.

- **Salida de texto** - Hablar con los usuarios.
- **Variables** - Recordar algunos valores para usarlos después.
- **Combinar cadenas de texto** - Combinar diferentes cadena de texto.
- **GET / POST** - Cómo leer información que se envía al servidor.
- **Expresiones condicionales** - Comprobar condiciones y ejecutar código si las condiciones son ciertas.

4

Hay algunas cosas que aún no están terminadas. Por ejemplo deberías impedir que el usuario siga adivinando números cuando no queden intentos..