

Curso Básico de PHP

Carlos Gómez
@RetaxMaster





¡Te doy la bienvenida!

Soy Carlos (RetaxMaster) y hago parte del equipo de Course Directors de Platzi en la facultad de Computer Science. Tal vez me recuerdes de aportes en otros cursos como este:

**RetaxMaster**
Platzi Team · Hace 9 meses


138

Ohh si PHPCito!!

En PHP se pueden hacer muuuuuchas cosas con los arrays, pero en serio, **MUCHAS**, en PHP puedes declarar arrays usando corchetes `[]` o usando la función `array()` que es mucho más elegante.

En PHP puedes declarar los arreglos en múltiples líneas, y además puedes usar una cosa llamada "trailing commas", que es básicamente dejar una coma al final del último valor del array, por ejemplo:

```
$weather = array(  
    "Bogota" => "cold",  
    "Monteria" => "hot",  
    "Medellin" => "mild", // <- Este es el trailing  
    comma  
);
```



¿Qué necesitas saber?

- Cómo ejecutar PHP.
- Tipos de datos.
- Tipos de operadores.



¿Qué necesitas saber?



Curso Básico de PHP: Instalación, Fundamentos y Operadores

★★★★★ 149 Opiniones

BÁSICO

NUEVO

¡Aprende PHP, el lenguaje de programación para backend presente en el 80% de sitios web! Inicia tu ruta de aprendizaje como PHP Developer y desarrolla tus primeros algoritmos con tu profesor Retax Master.

- Comprende el flujo para desarrollar aplicaciones web
- Instala PHP en cualquier sistema operativo
- Configura tu entorno de desarrollo con herramientas como XAMPP
- Domina la sintaxis básica, tipado y operadores en PHP

CONTINUAR APRENDIENDO

AGREGAR A MI RUTA



Carlos Eduardo Gómez
García

Course Director, Faculty
Computer Science en Platzi



Hora de meterle esteroides

- ¿Qué estructuras de datos existen en PHP?
- ¿Cómo puedo tomar decisiones?
- ¿Cómo puedo repetir código?
- ¿Cómo puedo reutilizar código?



¿Quieres ver más sobre PHP?



1 0
1 0 1
1 0 1

¿Qué son los arreglos?

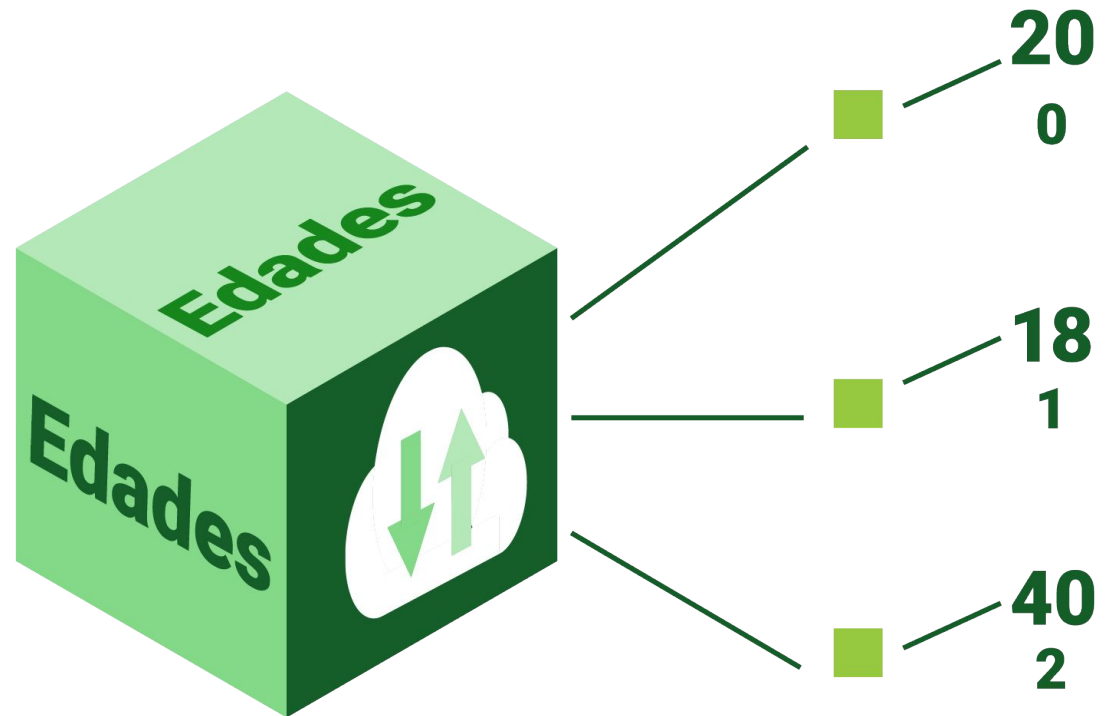
1
0





¿Qué son los arreglos?

Una variable que puede guardar múltiples valores.





¡Vamos al código!



End of class

- No eliminar please :3 -

1 0
1 0 1
1 0 1

Arreglos asociativos

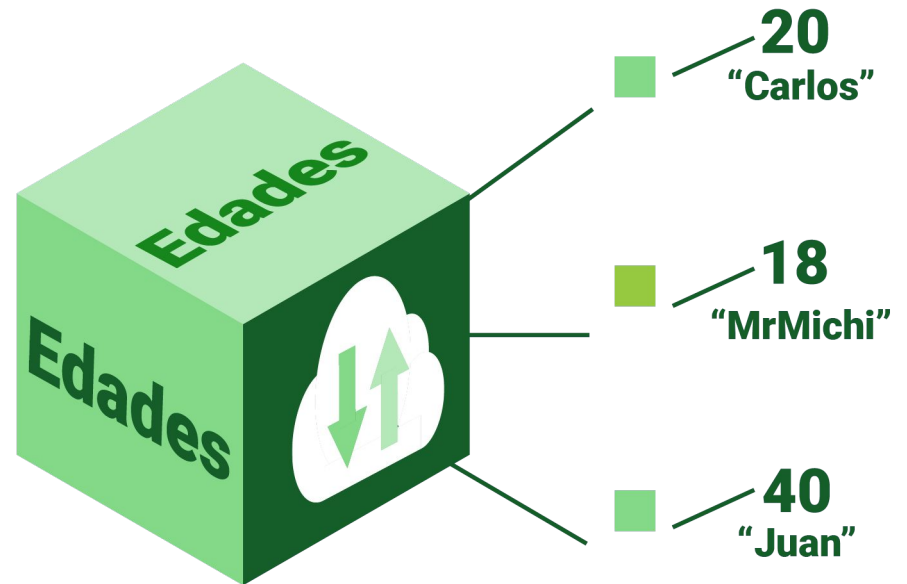
1
0





Arreglos asociativos

Una variable que puede guardar múltiples valores y puedes acceder a ellos mediante una palabra.





¡Vamos al código!



End of class

- No eliminar please :3 -

1 0
1 0 1
1 0 1

Manipulando arreglos

1
0





¡Vamos al código!



End of class

- No eliminar please :3 -

1 0 1
1 0 1
1 0 1

Destructuring en PHP

1
0





¡Vamos al código!



End of class

- No eliminar please :3 -

1 0 1
1 0 1

El operador de nave espacial

1
0





Operador de nave espacial





¡Vamos al código!



End of class

- No eliminar please :3 -

1 0 1
1 0 1
1 0

Reto: Escuela de michis

1
0



End of class

- No eliminar please :3 -

1 0 1
1 0 1
1 0 1

Aprende a tomar decisiones con if y else

1
0





¿Qué son las condicionales?

Muchas veces nos vemos en la necesidad de tomar una decisión u otra dependiendo de lo que el usuario elija. Las condicionales nos ayudan a elegir cuál bloque de código se debe ejecutar.



if... else

La sentencia if nos ayuda a declarar una condición. Si la condición se cumple entonces se ejecuta el código del if; si no se cumple, se ejecuta el código del else.



Sintaxis



```
if (/* Condición */) {  
    // Si la condición se cumple, se ejecuta esto.  
}  
else {  
    // Si no se cumple, entonces se ejecuta esto.  
}
```



¿Recuerdas las tablas de verdad?

Un `if` **siempre** requerirá de un valor booleano. Por eso es que podemos poner **true** o **false** directamente.

```
if (true) {  
    // Si la condición se cumple, se ejecuta esto.  
}  
else {  
    // Si no se cumple, entonces se ejecuta esto.  
}
```




¡Vamos al código!



End of class

- No eliminar please :3 -

1 0 1
1 0 1
1 0 1

Cómo organizar tu código con switch





¿Qué es switch?

Switch es una estructura de control que nos ayuda a elegir qué bloque de código ejecutar según el valor de alguna variable. Es una forma más fácil de tomar decisiones.



¿Cuándo debo usar switch?

Úsalo cuando...

- ✓ Necesites decidir entre un caso u otro.
- ✓ Quieras una forma fácil de plantear diversas situaciones.
- ✓ Usarlo no signifique sacrificar la legibilidad de tu código.

No lo uses cuando...

- ✗ Necesites hacer comparaciones complejas.
- ✗ Quieras verificar un rango de números.
- ✗ Necesites poner grandes bloques de código y/o tu código se vuelva ilegible.



Sintaxis

```
switch (/* Variable a evaluar */) {  
  
    case 1:  
        // Código para el caso 1  
        break;  
  
    case 2:  
        // Código para el caso 2  
        break;  
  
    case 3:  
        // Código para el caso 3  
        break;  
  
    default:  
        // Código por si ninguno de los casos se ejecuta.  
  
}
```




¡Vamos al código!



End of class

- No eliminar please :3 -

1 0 1
1 0 1

Reto: ¿Puedo
retirar mis
donaciones de
Twitch?

1
0



End of class

- No eliminar please :3 -

1 0
1 0 1
1 0 1

El ciclo while



1
0



¿Qué es un bucle?

Un bucle es una estructura de control que nos ayuda a repetir un bloque de código las veces que lo necesitemos.

- Bucle
- Contador
- Iteración



¿Cómo funciona el bucle while?

- El bucle while, al igual que el if, recibe una condición booleana. Mientras que esa condición sea true, el ciclo se va a repetir. Dentro del while nosotros podemos cambiar el valor de nuestra variable para detener el ciclo.
- Por eso se dice que este es un ciclo indefinido.



¿Cómo funciona el bucle while?





Sintaxis



```
while (/* Condición */) {  
    // Aquí va todo el código que se va a repetir.  
}
```




¡Vamos al código!



End of class

- No eliminar please :3 -

¿Do... While?





¿Qué hace el ciclo Do... While?

- Este ciclo es exactamente igual que el ciclo while, también recibe una condición y puedes cambiar el valor de tu variable desde dentro del ciclo.
- La diferencia es que este ciclo se ejecuta al menos una vez, sin importar si la condición es verdadera o falsa.
- También es un ciclo indefinido.



Sintaxis



```
do {  
    // Aquí va todo el código que se va a repetir.  
}  
while (/* Condición */);
```



¡Vamos al código!



End of class

- No eliminar please :3 -

1 0
1 0 1
1 0 1

El ciclo for



1
0



El ciclo for

Este ciclo nos permite definir una serie de condiciones, desde dónde queremos comenzar hasta dónde queremos terminar.

Nosotros le decimos explícitamente cuantas iteraciones hará, es por eso que decimos que este es un ciclo definido.



Sintaxis

```
for (/* Parámetros iniciales */; /* Condición */; /* Acciones */) {  
    // Este es el código que se repetirá  
}
```



¡Vamos al código!



1 0
1 0 1
1 0 1

El ciclo foreach



1
0



El ciclo foreach

La ventaja de este ciclo es que nos permite recorrer cualquier elemento que sea iterable sin tener que contar los elementos del mismo. Él por sí solo detecta cuántos elementos tiene dicho elemento.



Sintaxis básica

```
foreach ($iterable as $valor) {  
    // Código a repetir por cada valor  
}
```



Sintaxis obteniendo las llaves del elemento iterable

```
foreach ($iterable as $llave => $valor) {  
    // Código a repetir por cada valor  
}
```




¡Vamos al código!





Break y continue

También somos capaces de detener un ciclo o de avanzar a la siguiente iteración. Con estas palabras reservadas podemos tener más control sobre la ejecución de un ciclo.



¡Vamos al código!



End of class

- No eliminar please :3 -

1 0 1
1 0 1

Reto: ¿Cuántos caminos hay para llegar al mismo punto?

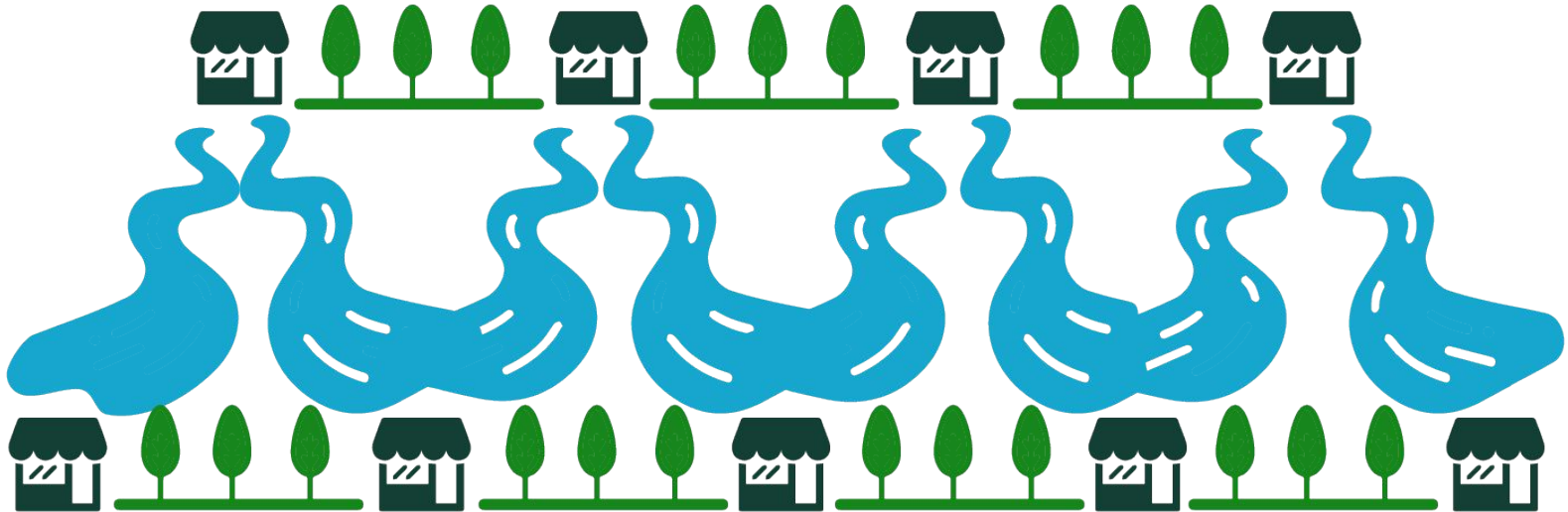
1
0





Reto

¿De cuántas formas puedo ir de la tienda 1 a la tienda n sin retroceder en el camino? Es decir, yendo siempre de izquierda a derecha.





¡Felicidades!

Ahora ya sabes en qué debes enfocarte al momento de resolver un problema, sin embargo...



La sucesión de Fibonacci

La sucesión de Fibonacci es una sucesión definida por recurrencia. Esto significa que para calcular el siguiente número de la sucesión se necesitan el número actual y el número anterior.

Es decir...

0 - 1 - 1 - 2 - 3 - 5 - 8 - 13 ...

End of class

- No eliminar please :3 -

1 0
1 0 1
1 0 1

Aprende a reutilizar tu código

Funciones

1
0





¿Qué es una función?

Podemos verlas como un chef mágico al cual tú le das los ingredientes y te devuelve un pastel ya hecho.



cocinar()





¿Qué es una función?

También podemos verlas como “ponerle nombre a un bloque de código”.



```
echo "Poniendo los huevos";  
echo "Poniendo la harina";  
echo "Poniendo la leche";  
echo "Mezclando los ingredientes";  
echo "Metiéndolos al hornor";  
  
echo "¡Pastel listo!";
```




¿Qué es una función?

En resumen, una función nos sirve para separar todo un bloque de código, ponerle un nombre, y **utilizarla cuantas veces queramos** sin preocuparnos por cómo trabaja por dentro.



¿Qué es una función?

Nos ayuda a tener código más ordenado, reutilizable y fácil de entender.

Separar tu código en funciones para después reutilizarlo **es buena práctica**. Copiar y pegar código **no lo es**.



¡Vamos al código!



1 0 1
1 0 1

Parámetros en las funciones

Funciones

1
0



End of class

- No eliminar please :3 -

1 0 1
1 0 1

Profundizando en parámetros

Funciones

1
0



End of class

- No eliminar please :3 -

1 0 1
1 0 1

Las funciones son cajas mágicas

Uso de return

1
0





Retornar valores de una función

¿Recuerdas que nuestro chef debía devolvernos un pastel ya cocinado? Retornar un valor significa decirle **explícitamente** a la función qué es lo que debe devolver.

Esto lo hacemos con la palabra reservada “return” y usualmente guardamos ese valor en una variable.



Retornar valores de una función

Es como si el llamado a nuestra función se convirtiera en el resultado que deseamos por arte de magia.

```
$resultado = suma(6, 4);
```



¡Vamos al código!



End of class

- No eliminar please :3 -

1 0 1
1 0 1
1 0 1

¡Expande tu arsenal de funciones!

1
0



End of class

- No eliminar please :3 -

1 0 1
1 0 1
1 0

Reto: completa el código



1
0

End of class

- No eliminar please :3 -

1 0 1
1 0 1
1 0

¡Hagamos un juego con PHP!

1
0



End of class

- No eliminar please :3 -

1 0 1
1 0 1
1 0 1

Cómo interactúa PHP con HTML

1
0





PHP es HTML con esteroides



PHP fue diseñado con el propósito de preprocesar HTML, es decir, de hacer operaciones como si fuera un lenguaje de programación.



PHP es HTML con esteroides



Esto le permite imprimir páginas HTML dinámicas. Sí, PHP puede imprimir directamente HTML y podemos usar todas sus ventajas para generar documentos dinámicos.



¡Vamos al código!





Ten en cuenta las buenas prácticas

Usualmente, no es bueno mezclar tanta lógica de PHP con HTML. Hacer esto vuelve nuestro código ilegible, por lo que la mejor solución es separar nuestra lógica de nuestra vista.

Pero esto lo veremos en otro curso. 😊

End of class

- No eliminar please :3 -

Carlos Gómez @RetaxMaster

Carlos Gómez

@RetaxMaster





En este curso aprendiste...

- Arreglos y cómo manipularlos.
- Toma de decisiones con condicionales.
- Repetir código con bucles.
- Reutilizar código con funciones.
- Combinar PHP con HTML.
- ¡Hicimos un proyecto!