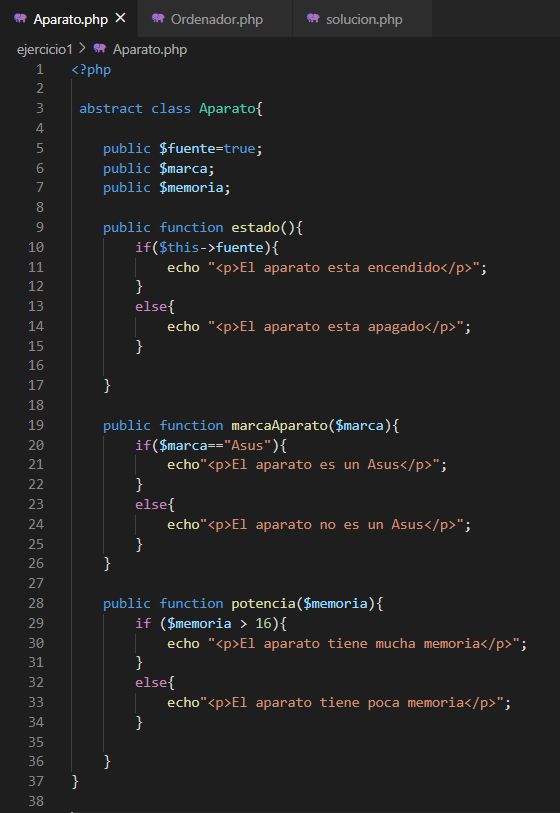
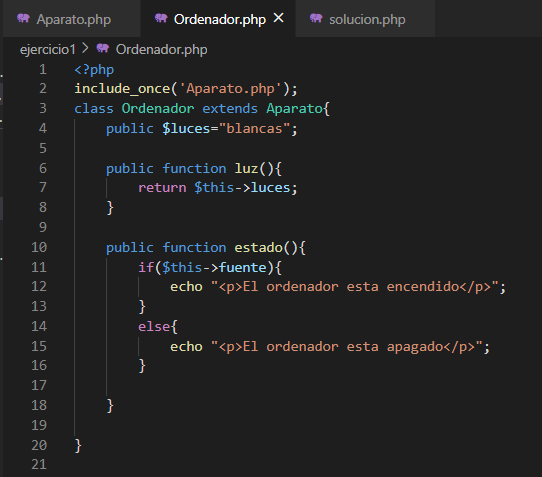
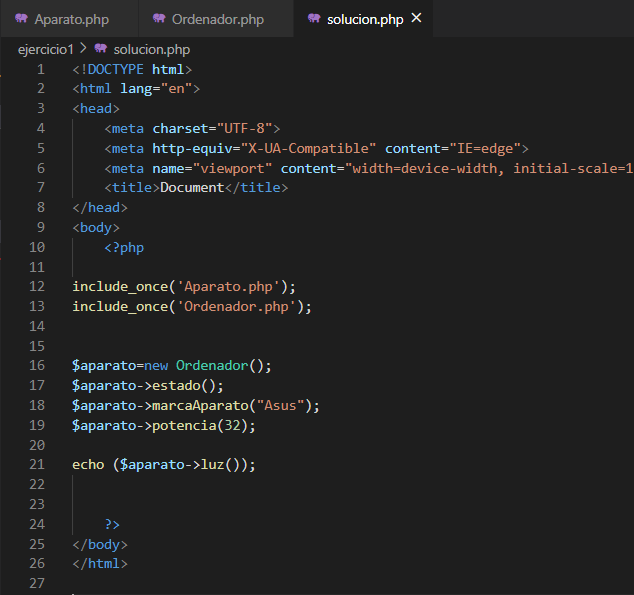
1. Ejercicio herencias.

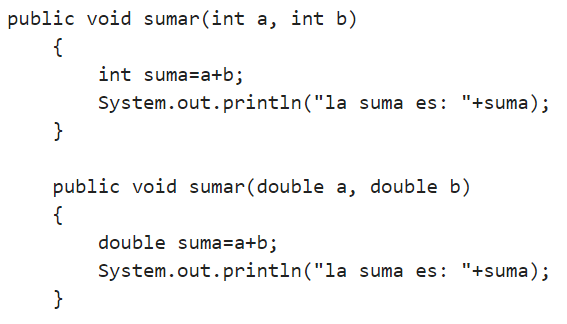






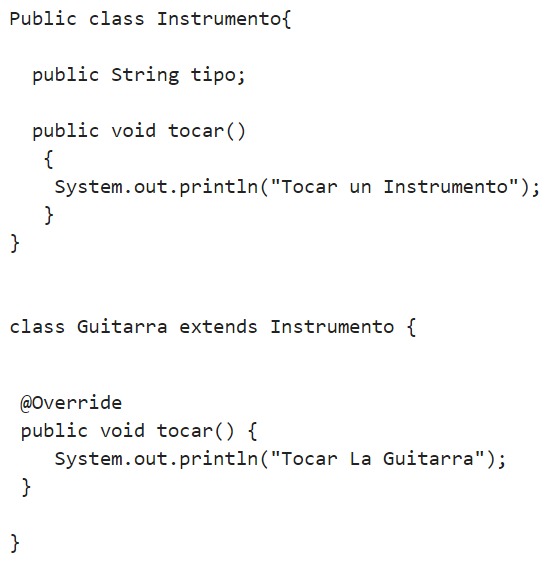
Sobrecargar: declarar métodos que se llamen igual pero que reciban parámetros diferentes. Tenemos dos métodos que se llaman igual, pero con diferentes argumentos.

Ejemplo: crear dos métodos y los llamamos sumar, pero al primero le introducimos los parámetros int $a, int $b y al segundo double $a, double $b, accederemos a cada método dependiendo de los parámetros que introducimos.

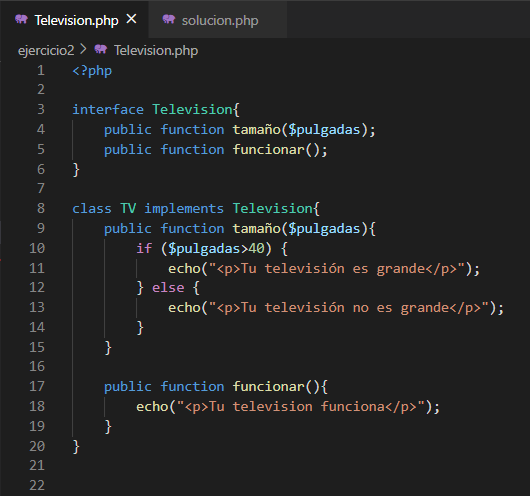


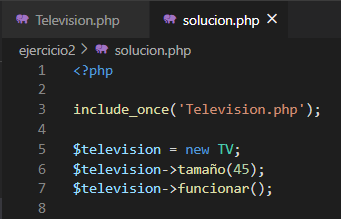
Sobreescribir: en la clase hija un método que se llama igual que la clase hija. La forma por la cual una clase que hereda puede redefinir los métodos de su clase Padre, de esta manera puede crear nuevos métodos con el mismo nombre de su superClase.

Ejemplo: creamos la clase Instrumento y creamos un método llamado tocar(), que nos devuelve (“Tocar un Instrumento”). Ahora creamos una clase hija llamada Guitarra que herede de Instrumento y creamos un método con el mismo nombre (tocar()) podemos cambiar y poner que nos devuelva por ejemplo (“Tocar la Guitarra”).

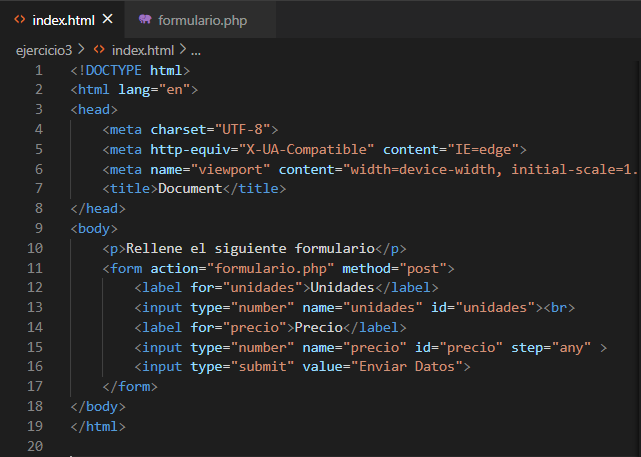


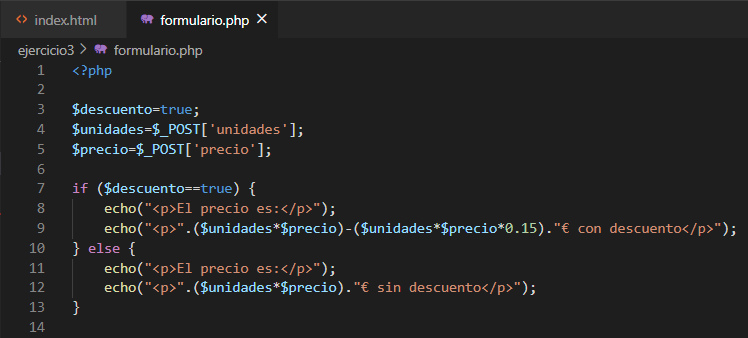
1. Clase interface.



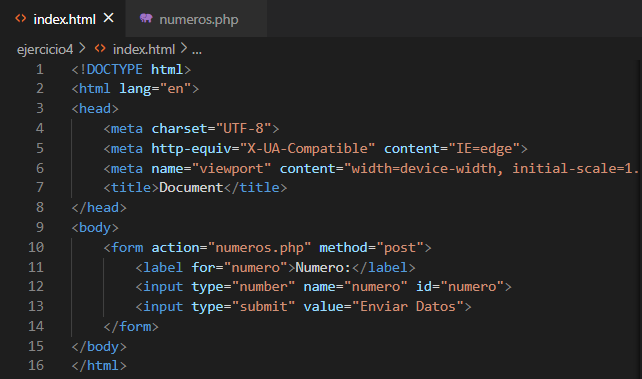


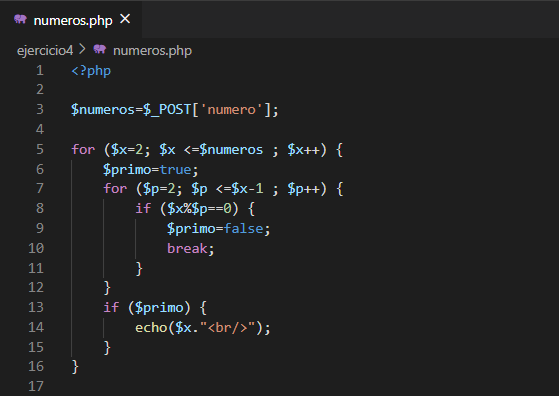
1. Unidades, precio y descuento.





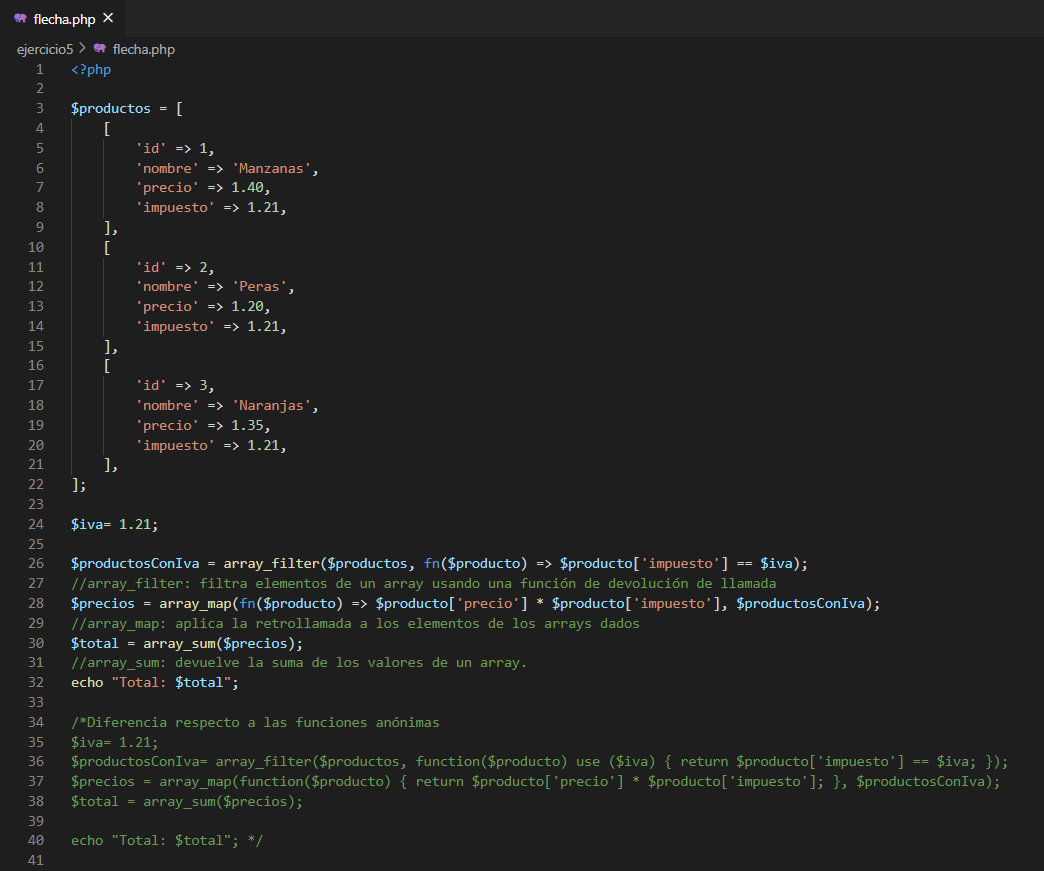
1. Números primos.





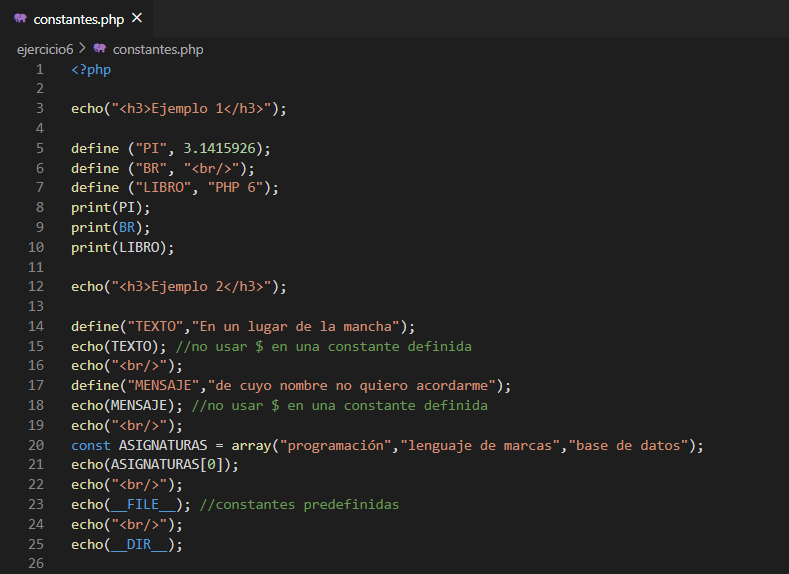
1. Explica con un ejemplo un caso de función flecha y demuestra cómo mejora el código de la programación tradicional.

Las funciones flechas son una forma más corta de escribir funciones anónimas.



1. Utiliza un ejemplo con constantes y explica qué diferencia tiene con una variable "clásica".

Las constantes se diferencian de una variable clásica que el valor almacenado permanece idéntico, es decir, permanece siempre constante, hasta que el intérprete de PHP termine de procesar nuestra página. Es imposible definirles un valor mediante un operador de asignación, lo que facilita que no alteremos su valor durante toda su utilización, ya que siempre almacenará el mismo valor. Nombre en mayúsculas para diferenciar.



1. Comenta qué son los operadores unarios, binarios y terciarios. Realiza un ejemplo con cada uno.

Operadores unarios: son los cuales toman sólo un valor. Ejemplos de ello son los operadores lógicos de negación (!) y de incremento (++), ya que necesitan únicamente una variable para trabajar.

Operadores binarios: éstos toman dos valores y son los más numerosos. Ejemplos de ello son los operadores de suma (+) y resta (-), que necesitan dos variables para poder realizar la operación.

Operadores ternarios: son los que necesitan tres valores para poder funcionar.

