


Reutilizar código de una clase

Traits

Antes de continuar...

Hemos visto que es posible reutilizar código mediante:

- **Require/Include:** Carga de scripts dentro de archivos
 - **Namespace:** Agrupar partes de código bajo un mismo nombre
 - **Extends:** Herencia simple donde una clase (hija) hereda atributos y/o métodos de otra (padre)
 - **Abstract:** Definir clases abstractas que pueden contener declaración o implementación a heredar
 - **Interface:** Colección de métodos abstractos declarados que deben ser implementados.
- 

Traits vs Namespace

Traits replica el funcionamiento de Namespace; da un nombre específico a una clase (con sus atributos y métodos).

Esta “clase” con nombre se puede emplear dentro de la declaración de otra clase mediante “use”, de forma similar a los alias de Namespace.

Diferencia principal;

- Namespace da nombre a un **fragmento de código** - Trait nombra **una clase**
- Namespace se usa en la lógica del **programa** - Trait se usa **dentro de la clase**



Características de las “clases” tipo Traits

Replican el comportamiento de una clase abstracta: se usa para declarar o implementar atributos o métodos que puedan reutilizar otras clases.

- Se declara como **Trait** en lugar de Class.
- No son instanciables. No tiene constructor.
- Se deben emplear con **Use** dentro de la clase que desea reutilizar el código.
- Los atributos de un Trait se comportan como propios de la clase que los use.
- Los métodos pueden ser declarativos mediante **abstract**
- O tener implementación “por defecto” con **default**.
- Una clase puede usar múltiples Traits.

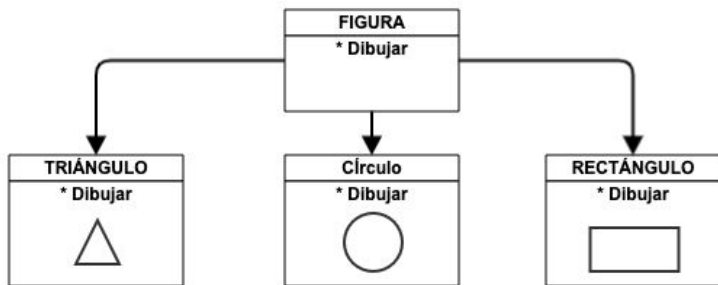


Polimorfismo mediante Traits

Mediante las penúltimas propiedades listadas:

- Los métodos pueden ser declarativos mediante **abstract**
- O tener implementación “por defecto” con **default**.

Tenemos que estos métodos se pueden “sobreescribir”, por lo que **Trait es otra herramienta para tener polimorfismo** (mismo nombre de método, diferente resultado)



Herencia múltiple mediante Traits

Mediante la última propiedad listada:

- Una clase puede usar múltiples Traits.

Se puede conseguir replicar el funcionamiento de **herencia múltiple en PHP**.

