Metaclases

y la metaprogramación

¡Las clases también son objetos!

Tal cual como `def` define una función y las funciones son objetos, `class` define una clase y las clases son objetos.

Si las clases son objetos…
¿De dónde provienen?

Metaclases

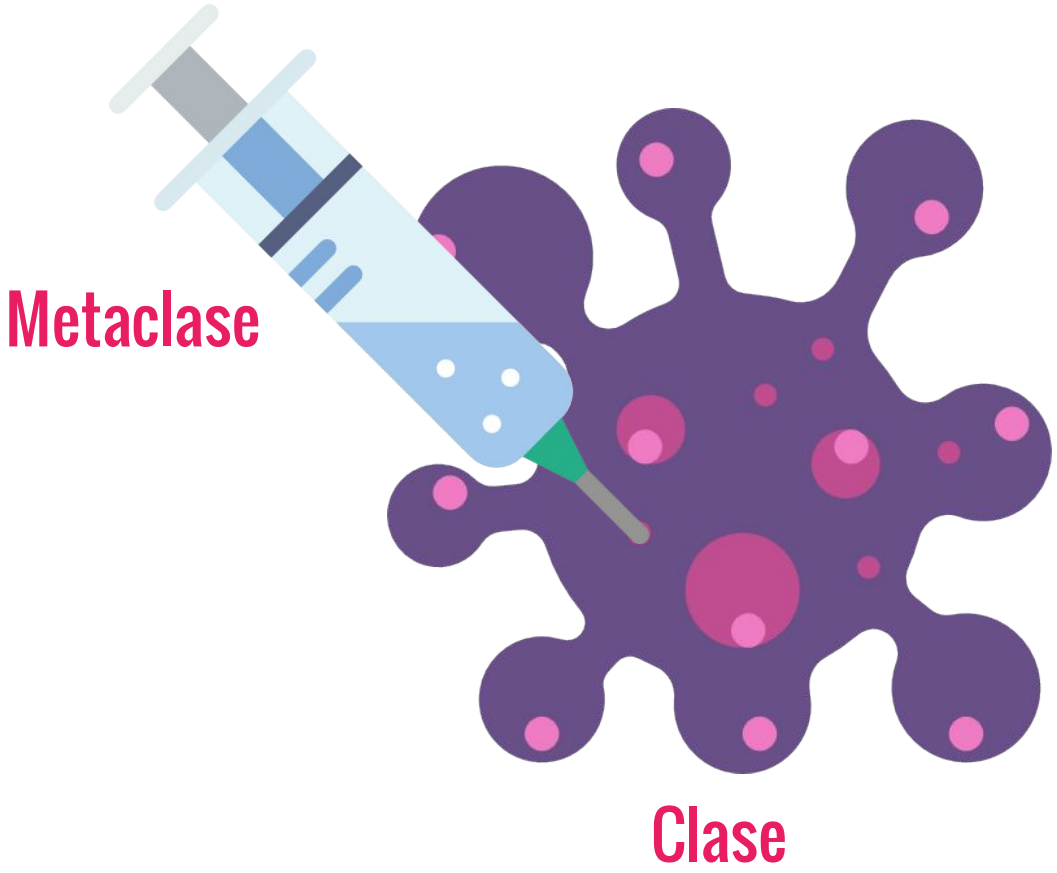
LAS CLASES DE LAS CLASES

¿Qué es una metaclase?

Una clase para instanciar clases, nada más.

¡CIENCIA!

¡Nos permiten controlar el nacimiento de una clase!



Es como la herencia, pero NO.

Cuando una clase hereda de otra, adopta a su superclase y añade mayor funcionalidad.

Cuando una clase nace a partir de una metaclase, la metaclase en cuestión trata a la clase como una instancia y la clase no hereda nada de la metaclase.

¿Cómo puedo crear una metaclase?

~~__prepare__~~

__new__

__init__

__call__

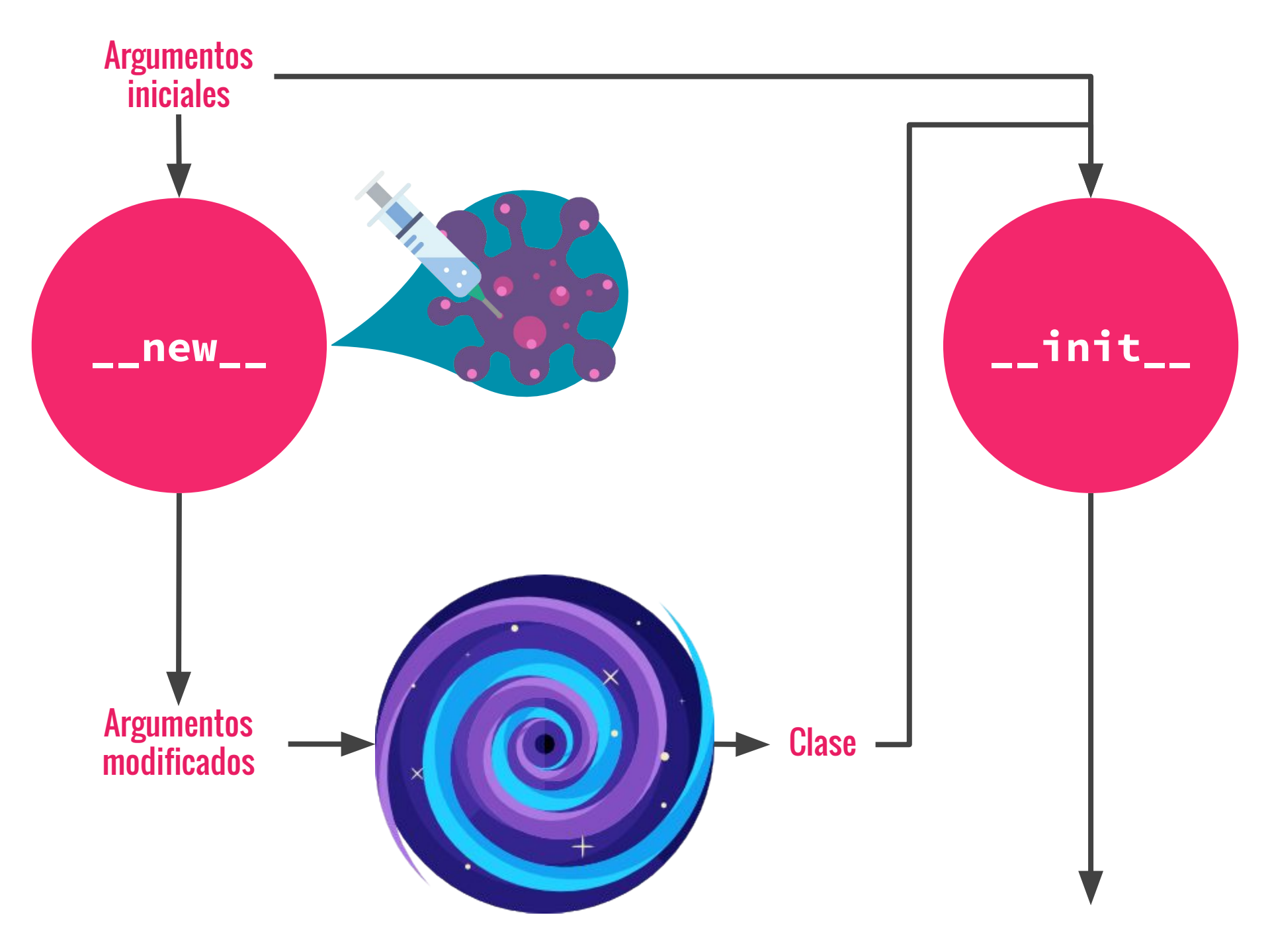
Estructura básica de una metaclase

```
class Metaclase(type):

    def __new__(mcs, name, bases, attrs):
        return super().__new__(mcs, name, bases, attrs)

    def __init__(cls, name, bases, attrs):
        return super().__init__(name, bases, attrs)

    def __call__(cls, *args, **kwargs):
        return super().__call__(*args, **kwargs)
```


Argumentos iniciales
__new__
__init__
Argumentos modificados
Clase

¿Y `__call__`?

¿Cuándo es llamado el método `__call__` en una clase común y corriente?