**Super()**

Pada kali ini kita akan masuk pada super untuk inheritance. Lihat kode berikut :

class Hero:

    def \_\_init\_\_(self, name, health):

*self*.name = name

*self*.health = health

class Hero\_intelligent(Hero):

    def \_\_init\_\_(self, name):

*self*.name = name

*self*.health = 100

class Hero\_strength(Hero):

    def \_\_init\_\_(self, name):

*self*.name = name

*self*.health = 200

lina = Hero\_intelligent("lina")

axe = Hero\_strength("axe")

print(lina.name, " ", lina.health)

print(axe.name, " ", axe.health)

dari kode diatas, kita bisa melihat banyak hal yang diulang. Dalam dunia programming hal ini tidak seharusnya terjadi. Seharusnya kita bisa membuat def \_\_init\_\_ yang ada di super class Hero bisa juga dipakai di class turunannya. Berikut caranya :

ubah kode berikut yang ada di class turunan :

*self*.name = name

*self*.health = 100

menjadi kode berikut :

def \_\_init\_\_(self, name):

        Hero.\_\_init\_\_(*self*, name, 100)

Ketika anda menjalankannya, maka outputnya akkan sama

Anda juga bisa menggunakan super, berikut caranya :

def \_\_init\_\_(self, name):

        super().\_\_init\_\_(name, 100)

**penjelasan :**

**super().\_\_*init\_\_*** = kita mengambil method \_\_init\_\_ yang berada di class Supernya ( disini Hero yang dimaksud dengan class super ). Kita tak perlu menggunakan **self** di super.

Berikut contoh lainnya :

class Hero:

    def \_\_init\_\_(self, name, health):

*self*.name = name

*self*.health = health

    def showInfo(self):

        print(f"{*self*.name} dengan helath : {*self*.health}")

class Hero\_intelligent(Hero):

    def \_\_init\_\_(self, name):

        super().\_\_init\_\_(name, 100)

        super().showInfo()

class Hero\_strength(Hero):

    def \_\_init\_\_(self, name):

        super().\_\_init\_\_(name, 200)

        super().showInfo()

lina = Hero\_intelligent("lina")

axe = Hero\_strength("axe")

anda juga sebenarnya bisa menambahkan self di dalam method yang akan anda panggil tapi itu opsional.