**Constructor \_init\_()**

***\_\_init\_\_ = inisialisasi / inisialisasion***

Di episode sebelumnya saat kita membuat object sederhana, itu tidak terlalu mempresentasikan object tapi itu lebih mengarah ke prosedural. Berikut cara membuat object yang benar :

Di dalam python adalah yang di sebut dengan magic keyword, beriku contoh salah satunya :

class Hero:

    def \_\_init\_\_(self):

        print("Hello World")

hero1 = Hero()

ketika menjalankan kode di atas akan menghasilkan output berikut :

***Hello World***

Ini akan sangat berguna ketika kita memasukkan argument. Berikut contohnya jika saya memasukkan argument :

class Hero:

    def \_\_init\_\_(self, x):

        print("Hello World", x)

hero1 = Hero(10)

maka akan mengasilkan output berikut :

***Hello World 10***

Fungsi dari self adalah si hero1. Contoh nya di episode kemarin ketika kita membuat name kita akan melakukan cara berikut :

hero1.name = 'Saber'

sekarang tidak perlu, kita bisa menggunakan cara berikut :

class Hero:

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

*self*.name = inputName

hero1 = Hero("saber")

print(hero1.name)

ini akan membuat kode kita lebih ringkas. Ini tidak hanya sebatas untuk satu inputan saja, kita bisa menaruh sebanyak yang kita butuhkan. Berikut contohnya jika saya melengkapi yang kemarin :

class Hero:

    def \_\_init\_\_(self, inputName, inputHealth, inputPower, inputArmor):

*self*.name = inputName

*self*.health = inputHealth

*self*.power = inputPower

*self*.armor = inputArmor

hero1 = Hero("saber", 4600, 6800, 1100)

berikut cara untuk mengetahui apa saja atribut yang di miliki oleh sebuah object :

print(hero1.\_\_dict\_\_)

berikut outputnya :

***{'name': 'saber', 'health': 4600, 'power': 6800, 'armor': 1100}***