

In [1]:

```
#Else comprehension
l = [x if x > 3 else 2 for x in range(5)]
l
```

Out[1]:

```
[2, 2, 2, 2, 4]
```

In [ ]:

```
#Todo en Python son objetos

x = [2, 4]
x.append(2) #Append es una función de clase lista en Python
```

In [3]:

```
#Creación de la clase ser humano
class humano:
    ojos = 2 #Aunque dependen del ser humano que sea estos valores son por defecto
    piernas = 2
    nombre = None #En este caso depende completamente del ser humano como no lo sabemos p
onemos None
    edad = None

#Si queremos crear a Juan como humano

juan = humano()
print(juan.ojos)
juan.nombre = "Juan"
print(juan.nombre)
```

```
2
Juan
```

In [7]:

```
#funcion constructor de la clase: __init__ y debe llevar el atributo self

class humano:
    def __init__(self, o, n, b, p, e): #Aquí hay que meter todos los argumentos que yo q
uiero mostrar a la hora de crear un nuevo ser humano
        self.ojos = o
        self.nombre = n
        self.brazos = b
        self.piernas = p
        self.edad = e

juan = humano(o=2, n="Juan", b=2, p=2, e=30) #Juan es una instancia de humano
print(juan.ojos)
pepe = humano(o=1, n="Pepe", b=2, p=3, e=2)
print(pepe.piernas)
```

```
2
3
```

In [8]:

```
juan is pepe
```

Out[8]:

```
False
```

In [9]:

```
juan == pepe
```

Out[9]:

False

In [11]:

```
juan2 = humano(o=2, n="Juan", b=2, p=2, e=30)
juan == juan2 #Porque cada uno tiene su instancia, es como si fuesen dos gemelos pero cada uno es unico, tiene su instancia
```

Out[11]:

False

In [12]:

```
juan is juan2 #Porque cada uno tiene su instancia, es como si fuesen dos gemelos pero cada uno es unico, tiene su instancia
```

Out[12]:

False

In [47]:

```
class Perro:
    def __init__(self, patas, cola, raza, peso, edad, nombre): #Este es el constructor
        self.patas = patas
        self.colas = cola
        self.raza = raza
        self.peso = peso
        self.edad = edad
        self.nombre = nombre

    def cumple_edad(self):
        self.edad = self.edad + 1 #Self es como llamar a Perro el que hayamos creado, es decir, rudolf o baxter
        print("Feliz cumple")

rudolf = Perro(patas=4, cola=True, raza="Pastor Alemán", peso=25, edad=10, nombre="Rudolf")
print(rudolf.edad)
rudolf.edad = rudolf.edad + 1
print(rudolf.edad)

baxter = Perro(patas=4, cola=True, raza="Labrador", peso=12, edad=3, nombre="Baxter")
baxter
```

10

11

Out[47]:

<\_\_main\_\_.Perro at 0x17a1836df40>

In [48]:

```
print(baxter.edad)
baxter.cumple_edad()
print(baxter.edad)
```

3

Feliz cumple

4

In [24]:

```
mc = humano(o=2, n="MC", b=2, p=2, e=51)
print(mc)
```

<\_\_main\_\_.humano object at 0x0000013D15C9AF40>

In [25]:

```
def f1():  
    return 2
```

```
print(f1) #El resultado significa donde está ubicado en la memoria
```

```
<function f1 at 0x0000013D15B87AF0>
```

In [26]:

```
print(f1()) #Hasta que no llamemos a la función no pone el resultado
```

2

In [28]:

```
class Pelota:
```

```
    def __init__(self, velocidad=0, posicion=0, diametro=0):  
        self.velocidad = velocidad  
        self.posicion = posicion  
        self.diametro = diametro  
  
    def golpear(self, fuerza): #cuanta mas fuerza mas velocidad, ¿como hacerlo para que  
cambie velocidad?  
        velocidadantigua = self.velocidad  
        posicionantigua = self.posicion  
        self.velocidad = self.velocidad + fuerza  
  
        if self.posicion == 0 or self.velocidad == 0:  
            self.posicion = self.posicion + self.velocidad*1.5  
  
        else:  
            self.posicion = self.posicion * self.velocidad  
  
        print("Velocidad:", self.velocidad - fuerza, "-->", self.velocidad) #dos formas d  
e imprimir velocidad antigua  
        print("Velocidad:", velocidadantigua, "-->", self.velocidad) #dos formas de impr  
imir velocidad antigua  
        print("Posición:", posicionantigua, "-->", self.posicion)  
  
    def parar(self):  
        print("Parar:", self.velocidad, "-->", 0)  
        self.v = 0  
  
    def rajar(self):  
        self.diametro = self.diametro - 5  
        print("Rajar:", self.diametro)
```

In [29]:

```
pelota_tenis = Pelota(velocidad=20, diametro=10)  
pelota_tenis.golpear(fuerza=8)  
pelota_tenis.golpear(fuerza=80)  
pelota_tenis.parar()  
pelota_tenis.rajar()
```

```
Velocidad: 20 --> 28  
Velocidad: 20 --> 28  
Posición: 0 --> 42.0  
Velocidad: 28 --> 108  
Velocidad: 28 --> 108  
Posición: 42.0 --> 4536.0  
Parar: 108 --> 0  
Rajar: 5
```

In [1]:

```
#Coger apuntes de Gabriel - class lista  
class Lista:
```

```
def __init__(self, elementos):
    self.elementos = elementos

def append(self, value):
    self.elementos.append(value)
    return self
```

```
l = [2, 3, 5, 7]
d = l.append(29)
print(l)
```

```
[2, 3, 5, 7, 29]
```

In [3]:

```
l = Lista(elementos=[2, 3, 5, 7])
l = l.append(29)
print(l.elementos)
```

```
[2, 3, 5, 7, 29]
```

In [36]:

```
class Humano:

    def __init__(self, vida, fuerza, armadura, n_dientes, esta_vivo=True): #True tiene que estar al final
        self.vida = vida
        self.armadura = armadura
        self.fuerza = fuerza
        self.esta_vivo = esta_vivo
        self.n_dientes = n_dientes

    def atacar(self, troll):
        vida_antigua = troll.vida
        troll.vida = troll.vida - (self.fuerza - troll.armadura)
        print("Vida Troll:", vida_antigua, "-->", troll.vida)
        if troll.vida <= 0:
            troll.esta_vivo = False

class Troll:
    #True tiene que estar al final
    def __init__(self, vida, fuerza, armadura, n_dientes, esta_vivo=True): #True tiene que estar al final
        self.vida = vida
        self.armadura = armadura
        self.fuerza = fuerza
        self.esta_vivo = esta_vivo
        self.n_dientes = n_dientes

    def atacar(self, humano):
        vida_antigua = humano.vida
        humano.vida = humano.vida - (self.fuerza - troll.armadura)
        print("Vida Humano:", vida_antigua, "-->", humano.vida)
        if humano.vida <= 0:
            humano.esta_vivo = False
```

In [37]:

```
Leo = Humano(vida=100, fuerza=5, armadura=3, n_dientes=27)

troll = Troll(vida=100, fuerza=8, armadura=1, n_dientes=42)
```

In [41]:

```
import time

def empezar_pelea(contrincantel, contrincante2):
    while True:
```

```
contrincante1.atacar(troll=contrincante2)
time.sleep(2)
contrincante2.atacar(humano=contrincante1)
if contrincante1.esta_vivo == False:
    print("Troll ganó")
    break
elif contrincante2.esta_vivo == False:
    print("Humano ganó")
    break
print("-----")
```

```
empezar_pelea(contrincante1=Leo, contrincante2=troll)
```

```
Vida Troll: 96 --> 92
Vida Humano: 100 --> 93
-----
Vida Troll: 92 --> 88
Vida Humano: 93 --> 86
-----
Vida Troll: 88 --> 84
Vida Humano: 86 --> 79
-----
Vida Troll: 84 --> 80
Vida Humano: 79 --> 72
-----
Vida Troll: 80 --> 76
Vida Humano: 72 --> 65
-----
Vida Troll: 76 --> 72
Vida Humano: 65 --> 58
-----
Vida Troll: 72 --> 68
Vida Humano: 58 --> 51
-----
Vida Troll: 68 --> 64
Vida Humano: 51 --> 44
-----
Vida Troll: 64 --> 60
Vida Humano: 44 --> 37
-----
Vida Troll: 60 --> 56
Vida Humano: 37 --> 30
-----
Vida Troll: 56 --> 52
Vida Humano: 30 --> 23
-----
Vida Troll: 52 --> 48
Vida Humano: 23 --> 16
-----
Vida Troll: 48 --> 44
Vida Humano: 16 --> 9
-----
Vida Troll: 44 --> 40
Vida Humano: 9 --> 2
-----
Vida Troll: 40 --> 36
Vida Humano: 2 --> -5
Troll ganó
```

In [ ]: