



PROGRAMACION



ORIENTADA A OBJETOS

CON

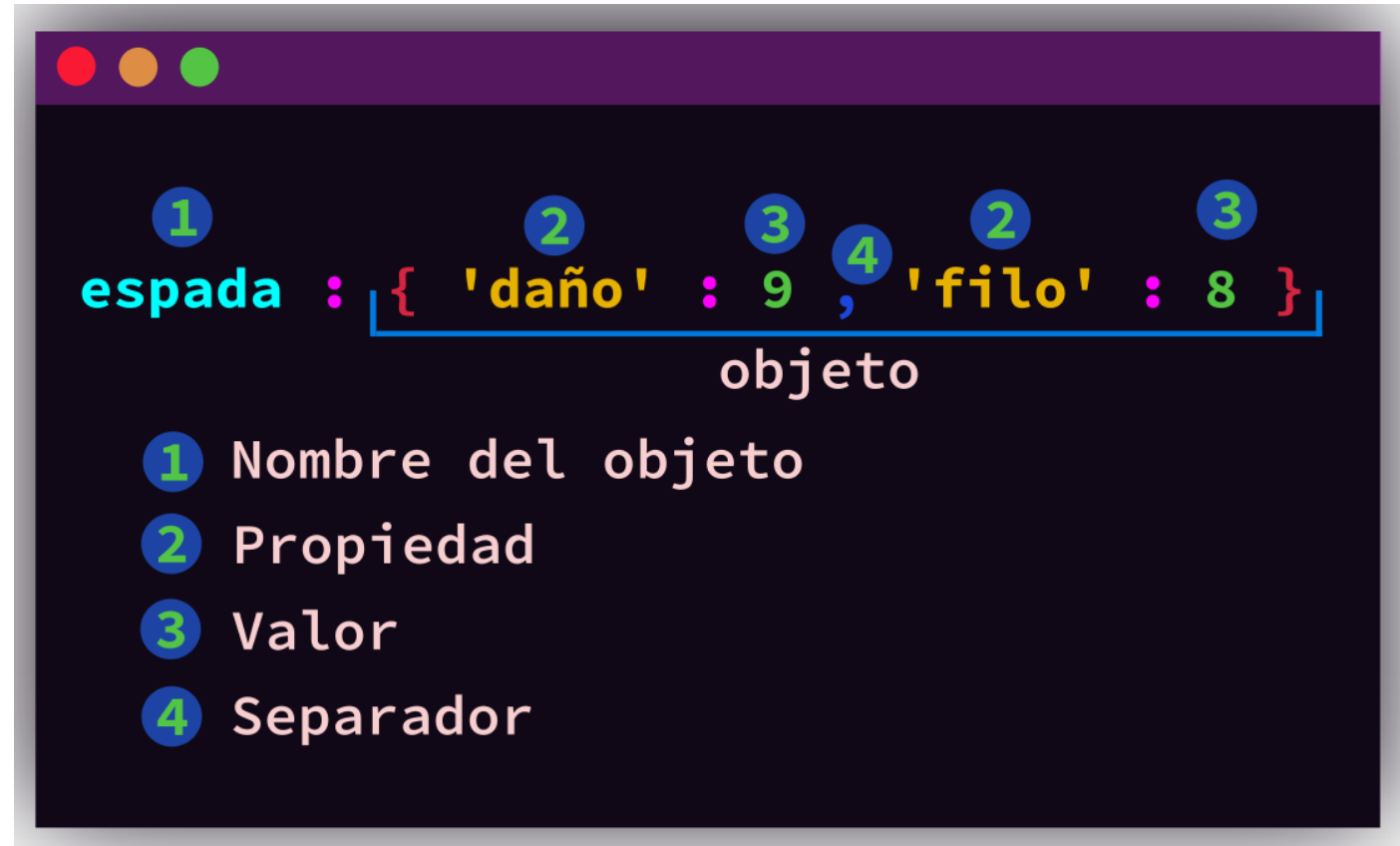
PYTHON



# Concepto

- La programación orientada a objetos es un paradigma de programación, En otras palabras es una forma de resolver problemas siguiendo una metodología en particular en este caso es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real.
- tenemos que aprender a pensar las cosas de una manera distinta para poder escribir nuestros programas en términos de objetos, propiedades y métodos. La POO tiene como principio que todo en la vida es un objeto programable, entonces para empezar a programar con este paradigma tendríamos que empezar con desarrollar nuestro pensamiento basado en objetos.

- Pensar en términos de objetos es pensar de manera natural como lo haríamos en la vida real. Un objeto contiene propiedades que almacenan valores y métodos que le añaden funcionalidad. Por ejemplo al pensar en una espada como un objeto para ser programado diríamos que la espada contiene una serie de características, como podría ser el filo el nombre el daño o el color, Además tiene una serie de funcionalidades o métodos asociados al mismo, como pueden ser cortar, o atacar.



como vimos en la imagen de arriba el objeto espada contiene la propiedad daño con un valor de 9 y filo con un valor de 8, tambien se puede apreciar que las propiedades se separan unas de otras por medio de una coma ",".

supongamos que tenemos un objeto llamado kraken 🐙 el cual es un monstruo marino, este objeto contendrá propiedades como nombre, tipo, vida y resistencia. para crear dicho objeto escribamos el siguiente código:

```
kraken = {  
    'nombre' : 'Kraken',  
    'tipo' : 'marino',  
    'vida' : 12,  
    'resistencia' : 2  
}
```

```
print(kraken['nombre']) >> 'Kraken'  
print(kraken['tipo']) >> 'marino'  
print(kraken['vida']) >> 12  
print(kraken['resistencia']) >> 2
```



**Kraken**

Vida	Tipo
♥♥♥♥♥♥♥♥	<b>Marino</b>
Resistencia	
🛡️🛡️	

# Clases

- Ahora que ya sabemos crear un objeto de manera estática veamos como crearlo a partir de una plantilla o clase
- Una Clase es una plantilla para crear un nuevo objeto donde podemos definir que propiedades y métodos contendrá un objeto .
- pensemos en las clases como plantillas con las que podemos crear uno o varios objetos que tendrán mas o menos la misma estructura. en el siguiente ejemplo crearemos una plantilla para crear espadas las cuales contendrán propiedades tales como nombre, rareza, daño, filo y un método llamado atacar() el cual restara vida a un objetivo dado:

```
class Espada:
```

```
    def __init__(self, nombre, rareza, daño, filo):
```

```
        self.nombre = nombre
```

```
        self.rareza = rareza
```

```
        self.daño = daño
```

```
        self.filo = filo
```

```
    def atacar(self, objetivo):
```

```
        if self.filo > 0 and objetivo['vida'] > 0:
```

```
            objetivo.vida -= self.daño + objetivo['resistencia']
```

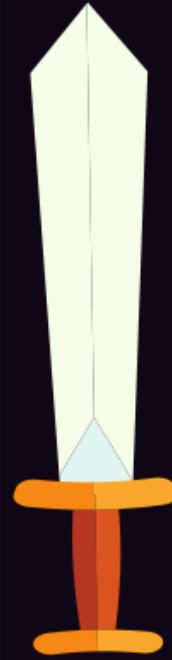
```
            self.filo -= 2
```

```
        elif objetivo['vida'] <= 0:
```

```
            print('El ' + objetivo['nombre'] + ' a muerto!')
```

```
        else:
```

```
            print('La Espada Se a quedado sin filo')
```



**Espada**

**Nombre: ??**

**Rareza: ??**

**Daño: ??**

**Filo: ??**

**Atacar()**

Ahora Podremos crear tantas espadas como queramos!!! y definir su daño, filo ,nombre y rareza así que manos a la obra! A continuación vamos a ver como crear objetos a partir de una clase:



```
#Esta linea de Codigo Crea una Nueva Espada llamada Espada Vikinga
                                nombre      rareza  daño  filo
espadaVikinga = Espada('Espada Vikinga', 'comun', 6, 7)

#Esta linea de Codigo Crea otra llamada Espada Heroica
espadaHeroica = Espada('Espada Heroica', 'rara', 7, 8)

#La vida del kraken es igual a 12
kraken['vida'] = ❤️❤️❤️❤️❤️❤️❤️❤️❤️❤️❤️❤️

espadaVikinga.atacar(kraken)
print(kraken['vida'])

>> ❤️❤️❤️❤️❤️❤️
```



en la imagen que acabamos de ver se instancian dos objetos de la clase Espada llamados espadaVikinga y espadaHeroica en otras palabras se crean dos objetos con la plantilla Espada, y tan solo usando dos lineas de codigo!! y es que el objetivo de las clases es ayudar a reducir el codigo. cada vez que se ejecute la linea espadaVikinga.atacar(Kraken) se le restan 6 puntos de vida al krakeny el filo de la espada vikinga disminuye 2 puntos y cuando el filo de la espada sea 0 o menos de 0 ya atacar() no causara daño alguno al kraken

Linux Lite Terminal

Archivo Editar Ver Terminal Pestañas Ayuda

>>>

>>>

>>> espadaVikinga.atacar(kraken)

>>> kraken['vida']

8

>>> kraken['vida'] = 12

>>> print(kraken['vida'])