

# Sesión 4

Sergio Lizana

## 2. Activitats de la Sessió 4.

### Activitats:

- Executar el programa. Explicar els resultats obtinguts.

```
sergio@sergio:~/Escritorio/Introducción Python/02- ejercicios/Sesión 4$ python ej01.py
corbata
35
67
2345
```

El programa muestra el nombre del producto asignado a la clase. El producto, el precio, la cantidad y el precio total de todas las unidades juntas.

- Modificar el programa definint al començament de la classe els atributs amb el valor inicial adient.

```
# -*- coding: utf-8 -*-

class Producto:
    "Ejemplo de clase con la cantidad y el precio de un prodducto"

    def __init__(self, producto, precio, unidades):
        self.producto = "corbata"
        self.precio = 35
        self.unidades = 67

    def costo_total(self):
        costo = self.precio * self.unidades
        return costo

mi_objeto_producto = Producto(producto,precio,unidades)

print mi_objeto_producto.producto
print mi_objeto_producto.precio
print mi_objeto_producto.unidades
print mi_objeto_producto.costo_total()
```

- Explicar per a què serveixen els docstrings. (Buscar a la web)

Un docstring es una cadena literal que se pone como primera instrucción en una definición de un módulo, función, clase o método. Se usa como un comentario para documentar un segmento específico del código.

- Explicar que és una PEP: Python Enhancement Proposal. (Buscar a la web)

El PEP contiene el índice de todas las propuestas de mejora de Python. Los números PEP los asignan los editores de PEP y no se cambian.

## 2. Definició i utilització d'una classe. Mètodes privats:

```
# -*- coding: utf-8 -*-  
  
class Producto:  
    "Ejemplo de clase con la cantidad y el precio de un producto"  
  
    def __init__(self, producto, precio, unidades):  
        self.producto = producto  
        self.precio = precio  
        self.unidades = unidades  
  
    def __costo_total__(self):  
        costo = self.precio * self.unidades  
        return costo  
  
    def nuevo_precio(self, precio):  
        self.precio = precio  
  
    def agrega(self, cantidad):  
        self.unidades = self.unidades + cantidad  
  
    def saca(self, cantidad):  
        if cantidad <= self.unidades:  
            self.unidades = self.unidades - cantidad  
        else:  
            print "No hay suficientes"
```

### Activitats:

- Executar el programa. Explicar els resultats obtinguts.

```
sergio@sergio:~/Escritorio$ python3 02- ejercicios/Sesión 02- ejercicios.py  
100  
5  
10  
Productos: camiseta  
Precio50  
Unidades10  
Precio total500
```

Muestra los precios de los productos que se han introducido. Además se añaden 5 unidades del segundo producto gracias al método agrega(). Luego muestra las unidades del segundo producto, dando como resultado 10.

Para acabar, printa el nombre, el precio, las unidades, y el coste de las unidades totales.

- Des del programa principal, cridar directament al mètode `__costo_total` i comprovar que es produeix un error.

```
sergio@sergio:~/Escritorio/Introducción Python/02- ejercicios/Sesión 02- ejercicios.py$ python3 02- ejercicios.py  
100  
5  
10  
Productos: camiseta  
Precio50  
Unidades10  
Precio total500  
Traceback (most recent call last):  
  File "ej02.py", line 46, in <module>  
    mi_producto2.__costo_total()  
AttributeError: Producto instance has no attribute '__costo_total'
```

### 3. Herència

#### Activitats:

**Executar el programa. Explicar els resultats obtinguts.**

```
sergio@sergio:~/Escritorio/Intro  
El animal llamado Rufo saluda
```

El programa imprime la frase de la imagen gracias a las variables creadas. A la variable `mi_mascota` se le asocia la clase `Perro`, que ésta hereda de `Animal`. `Animal` contiene los atributos `Nombre` y `patas`, por eso cuando se llama a la función `saluda`, se imprime la frase.

### 4. Herència:

#### Activitats:

**Executar el programa. Explicar els resultats obtinguts.**

```
sergio@sergio:~/Escritorio/Intro  
El animal llamado Rufo saluda  
Guau  
El animal llamado Azrael saluda  
miau miau
```

El programa hace lo mismo que el anterior, però se le han añadido unas variables diferentes. Se crean las variables `mi_mascota` a la que se le ha atribuido la clase `Perro`, que hereda de `Animal`, que tiene como atributos `Nombre` y `patas`. Al llamar a la función `saluda` se imprime la frase «El animal llamado Rufo saluda» y «Guau». Lo nuevo aquí es la variable `mi_otra_mascota`, que se le atribuye a la clase `Gato`, que hereda de `animal`. Al llamar a la función `saluda` y `maulla`, hace que se imprima la frase «El animal llamado Azrael saluda» «Miau miau»

### 5. Herència amb sobreescritura de mètodes:

#### Activitats:

**Executar el programa. Explicar els resultats obtinguts.**

```
sergio@sergio:~/Escritorio/Introdu  
El animal llamado Rufo saluda  
Guau  
El gato Azrael te mira fijamente  
miau miau
```

El programa hace lo mismo que el anterior, però se le han añadido unas variables diferentes. Ahora tenemos que el gato `saluda`, que a la hora de imprimir da preferencia a la función específica de la clase y no a la que hereda.

## 6. Herència múltiple:

### Activitats:

**Executar el programa. Explicar els resultats obtinguts.**

```
sergio@sergio:~/Escritorio/Intro  
El animal llamado Rufo saluda  
Vamos a jugar
```

Tenemos la clase Animal de antes, però añadimos una clase «Amigo». La variable mi\_mascota llama a la función salir de «Amigo» al cual se le ha pasado un 1 como parámetro, por eso imprime la frase «Vamos a jugar»

## 7. Cridar mètodes (sobreescrips) de la superclasse:

### Activitats:

**Executar el programa. Explicar els resultats obtinguts.**

```
sergio@sergio:~/Escritorio/Intro  
El animal llamado chucho saluda  
Guau
```

Esta vez tenemos dentro de la clase Perro unas nuevas variables. A la variable mi\_mascota se le pasa como parámetro «chucho» que está atribuida a la clase Perro, que hereda de Animal y al llamar a la función saluda, se imprime la frase. También se imprime la palabra «Guau» asociada a la variable sonido, porque la clase hija define su propio método init.

## 8. Altres mètodes predefinits de les classes.

### Activitats:

• **Executar el programa. Explicar els resultats obtinguts.**

```
sergio@sergio:~/Escritorio/Introducción Python/02-  
supercalifragilisticoespialidoso es mayor que bah
```

Se comparan las dos cadenas que se le han pasado a la clase Palabra gracias a la función \_\_cmp\_\_. Si se quitara esta función no funcionaría, ya que ésta compara el objeto con otro objeto. Si devuelve 0 es que los objetos son iguales, si devuelve -1 si el objeto self es más pequeño que el otro objeto y si devuelve 1 la comparativa es al contrario.