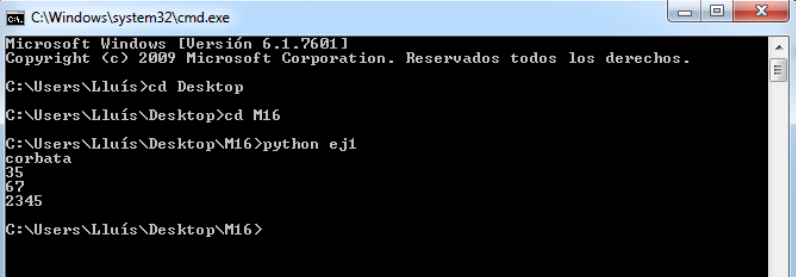


Introducció a Python: Sessió4:Programació Orientada a Objectes

Ejercicio 1

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3
4  class Producto:
5      """Ejemplo de clase con la cantidad y el precio de un producto"""
6
7      def __init__(self, producto, precio, unidades):
8          self.producto = producto
9          self.precio = precio
10         self.unidades = unidades
11
12         def costo_total(self):
13             costo = self.precio * self.unidades
14             return costo
15
16         mi_objeto_producto = Producto("corbata", 35, 67)
17
18         print mi_objeto_producto.producto
19         print mi_objeto_producto.precio
20         print mi_objeto_producto.unidades
21         print mi_objeto_producto.costo_total()
22
```



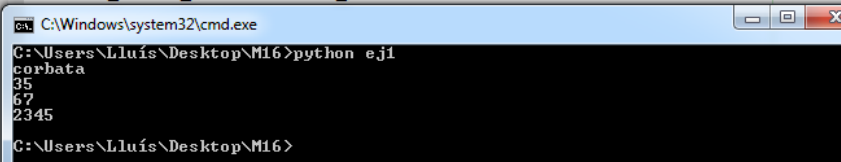
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Lluís>cd Desktop
C:\Users\Lluís\Desktop>cd M16
C:\Users\Lluís\Desktop\M16>python ej1
corbata
35
67
2345
C:\Users\Lluís\Desktop\M16>
```

Tenemos como resultado el nombre del producto, el precio, las unidades y el costo total que son las unidades x el precio.

Poniendo los valores en la función `__init__` nos sale lo mismo. No hace caso de los parámetros que añadimos en `mi_objeto_producto`

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3
4  class Producto:
5      """Ejemplo de clase con la cantidad y el precio de un producto"""
6
7      def __init__(self, producto, precio, unidades):
8          self.producto = "corbata"
9          self.precio = 35
10         self.unidades = 67
11
12         def costo_total(self):
13             costo = self.precio * self.unidades
14             return costo
15
16         mi_objeto_producto = Producto("corbata2", 35, 67)
17
18         print mi_objeto_producto.producto
19         print mi_objeto_producto.precio
20         print mi_objeto_producto.unidades
21         print mi_objeto_producto.costo_total()
22
```



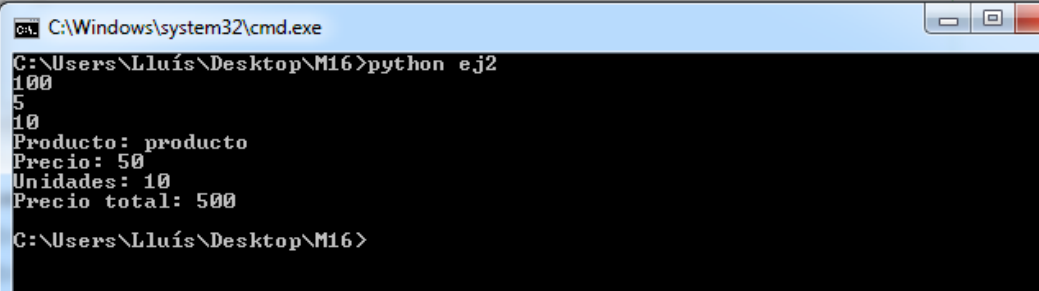
```
C:\Windows\system32\cmd.exe
C:\Users\Lluís\Desktop\M16>python ej1
corbata
35
67
2345
C:\Users\Lluís\Desktop\M16>
```

Los docstring son cadenas especificadas en el código fuente que se utiliza como comentarios, para documentar un segmento específico de código. Se utiliza triple “.

Las pep son propuestas de mejora de Python. Es información de la comunidad sobre un tema, pueden ser revisados y comentados por la comunidad Python.

Ejercicio 2

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3
4  class Producto:
5      """Ejemplo de clase con la cantidad y el precio de un producto"""
6
7      def __init__(self, producto, precio, unidades):
8          self.producto = "producto"
9          self.precio = precio
10         self.unidades = unidades
11
12         def __costo_total(self):
13             costo = self.precio * self.unidades
14             return costo
15
16         def nuevo_precio(self, precio):
17             self.precio = precio
18
19         def agrega(self, cantidad):
20             self.unidades = self.unidades + cantidad
21
22         def saca(self, cantidad):
23             if cantidad <= self.unidades:
24                 self.unidades = self.unidades - cantidad
25             else:
26                 print "No hay suficientes"
27
28         def informe(self):
29             print "Producto: " + self.producto
30             print "Precio: " + str(self.precio)
31             print "Unidades: " + str(self.unidades)
32             print "Precio total: " + str(self.__costo_total())
33
34         mi_producto1 = Producto("Pantalon", 100, 6)
35         mi_producto2 = Producto("Camiseta", 50, 5)
36
37         print mi_producto1.precio
38         print mi_producto2.unidades
39
40         mi_producto2.agrega(5)
41         print mi_producto2.unidades
42
43         mi_producto2.informe()
44
45
```



```
C:\Windows\system32\cmd.exe
C:\Users\Lluís\Desktop\M16>python ej2
100
5
10
Producto: producto
Precio: 50
Unidades: 10
Precio total: 500
C:\Users\Lluís\Desktop\M16>
```

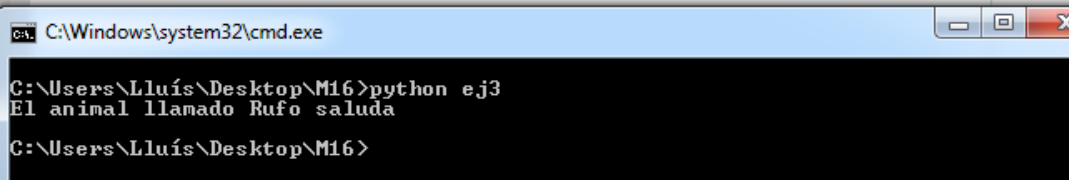
Primero imprimí el precio del producto uno, después las unidades del producto 2, después las unidades del producto 2 tras añadir 5, por último un informe del producto 2.

Comprobacion del error:

```
C:\Users\Lluís\Desktop\M16>python ej2
100
5
10
Producto: producto
Precio: 50
Unidades: 10
500
Precio total: None
Traceback (most recent call last):
  File "ej2", line 46, in <module>
    mi_producto2.__costo_total()
AttributeError: Producto instance has no attribute '__costo_total'
C:\Users\Lluís\Desktop\M16>
```

Ejercicio 3

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3
4  class Animal:
5      """Ejemplo de clase con la cantidad y el precio de un producto"""
6
7      def __init__(self,nombre,patas):
8          self.nombre = nombre
9          self.patas = patas
10
11     def saluda(self):
12         print "El animal llamado " + str(self.nombre) + " saluda"
13
14     class Perro(Animal):
15         """Clase hija para mostrar la herencia"""
16         # Simplemente, no hacemos nada
17         pass
18
19     mi_mascota = Perro("Rufo",4)
20
21     mi_mascota.saluda()
22
```



```
C:\Windows\system32\cmd.exe
C:\Users\Lluís\Desktop\M16>python ej3
El animal llamado Rufo saluda
C:\Users\Lluís\Desktop\M16>
```

Dentro de la clase Animal encontramos dos funciones, una para guardar nombre y numero de patas y otra para imprimir una frase en la cual recogemos el nombre.

Ejercicio 4

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

class Animal:
    """Clase base para mostrar la herencia"""

    def __init__(self,nombre,patas):
        self.nombre = nombre
        self.patas = patas

    def saluda(self):
        print "El animal llamado " + str(self.nombre) + "saluda"

class Perro(Animal):
    """Clase hija para mostrar la herencia"""

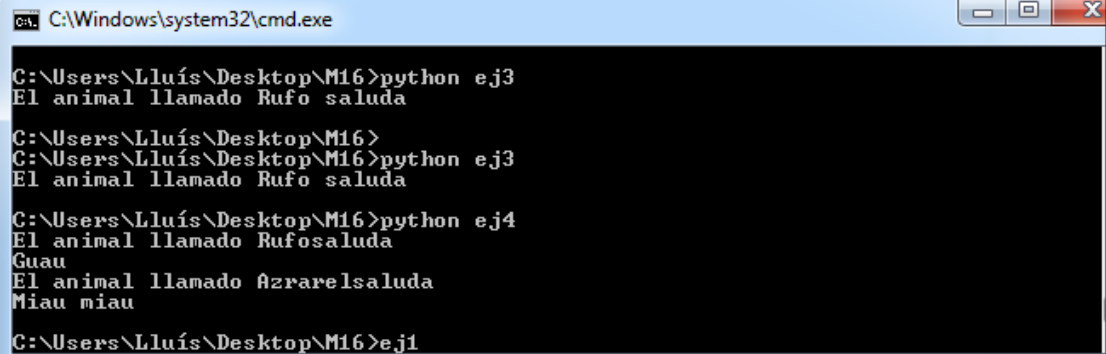
    def ladra(self):
        print "Guau"

class Gato(Animal):
    """Clase hija para mostrar la herencia"""

    def maulla(self):
        print "Miau miau"

mi_mascota = Perro("Rufo",4)
mi_mascota.saluda()
mi_mascota.ladra()

mi_otra_mascota = Gato("Azrarel",4)
mi_otra_mascota.saluda()
mi_otra_mascota.maulla()
```



```
C:\Windows\system32\cmd.exe

C:\Users\Lluís\Desktop\M16>python ej3
El animal llamado Rufo saluda

C:\Users\Lluís\Desktop\M16>python ej3
El animal llamado Rufo saluda

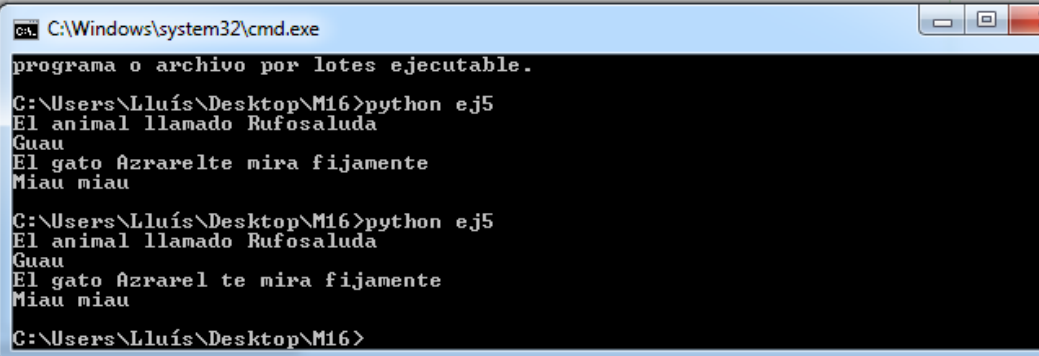
C:\Users\Lluís\Desktop\M16>python ej4
El animal llamado Rufosaluda
Guau
El animal llamado Azrarelsaluda
Miau miau

C:\Users\Lluís\Desktop\M16>ej1
```

Usamos la funcion saluda con los dos animales, y con el perro llamamos a la funcion ladra de la classe perro, y con el gato llamamos a la funcion maulla de la classe gato.

Ejercicio 5

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3
4  class Animal:
5      """Clase base para mostrar la herencia"""
6
7      def __init__(self,nombre,patas):
8          self.nombre = nombre
9          self.patas = patas
10
11     def saluda(self):
12         print "El animal llamado " + str(self.nombre) + "saluda"
13
14     class Perro(Animal):
15         """Clase hija para mostrar la herencia"""
16
17         def ladra(self):
18             print "Guau"
19
20     class Gato(Animal):
21         """Clase hija para mostrar la herencia"""
22
23         def maulla(self):
24             print "Miau miau"
25
26         def saluda(self):
27             print "El gato " + str(self.nombre) + "
28                 te mira fijamente"
29
30     mi_mascota = Perro("Rufo",4)
31     mi_mascota.saluda()
32     mi_mascota.ladra()
33
34     mi_otra_mascota = Gato("Azrarel",4)
35     mi_otra_mascota.saluda()
36     mi_otra_mascota.maulla()
```



```
C:\Windows\system32\cmd.exe
programa o archivo por lotes ejecutable.
C:\Users\Lluís\Desktop\M16>python ej5
El animal llamado Rufosaluda
Guau
El gato Azrarelte mira fijamente
Miau miau
C:\Users\Lluís\Desktop\M16>python ej5
El animal llamado Rufosaluda
Guau
El gato Azrarel te mira fijamente
Miau miau
C:\Users\Lluís\Desktop\M16>
```

Imprime la función saluda de la clase animal y la función ladra de la clase perro, después imprime la función saluda y maulla de la clase gato.

Ejercicio 6

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3
4  class Animal:
5      """Clase base para mostrar la herencia"""
6
7      def __init__(self,nombre,patas):
8          self.nombre = nombre
9          self.patas = patas
10
11      def saluda(self):
12          print "El animal llamado " + str(self.nombre) + "saluda"
13
14  class Amigo:
15      """Clase base para mostrar la herencia"""
16
17      def __init__(self, nombre):
18          self.nombre = nombre
19
20      def salir(self,num):
21          if num == 0:
22              print "Vamos a pasear"
23          elif num == 1:
24              print "Vamos a jugar"
25          else:
26              print "Vamos al parque"
27
28  class Perro(Animal,Amigo):
29      """Clase hija para mostrar la herencia"""
30
31      def ladra(self):
32          print "Guau"
33
34  mi_mascota = Perro("Rufo",4)
35  mi_mascota.saluda()
36  mi_mascota.salir(1)
37
```

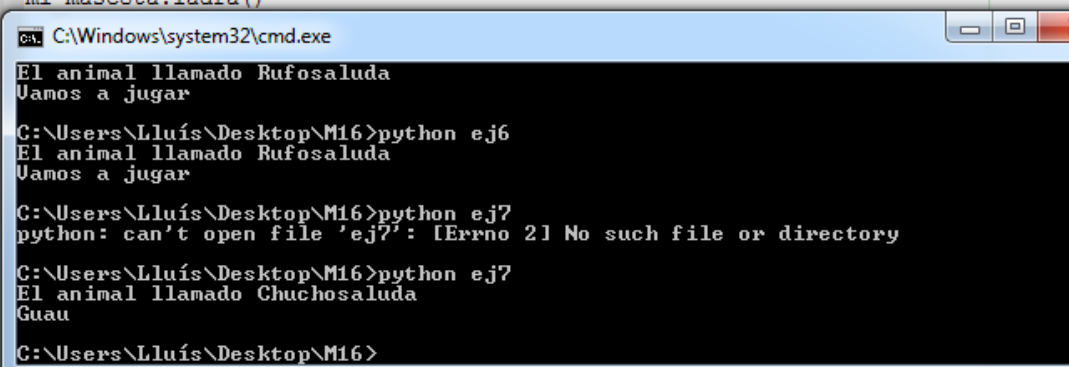
C:\Windows\system32\cmd.exe

```
C:\Users\Lluís\Desktop\M16>python ej6
El animal llamado Rufosaluda
Vamos a jugar
```

Se ejecuta la función saluda de la clase animal, y la función salir de la clase amigo.

Ejercicio 7

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3
4  class Animal:
5      """Clase base para mostrar la herencia"""
6
7      def __init__(self,nombre,patas):
8          self.nombre = nombre
9          self.patas = patas
10
11     def saluda(self):
12         print "El animal llamado " + str(self.nombre) + "saluda"
13
14
15     class Perro(Animal):
16         """Clase hija para mostrar la herencia"""
17
18         def __init__(self, nombre):
19             Animal.__init__(self, nombre, 4)
20             self.sonido = "Guau"
21
22         def ladra(self):
23             print self.sonido
24
25     mi_mascota = Perro("Chucho")
26     mi_mascota.saluda()
27     mi_mascota.ladra()
28
```



```
C:\Windows\system32\cmd.exe
El animal llamado Rufosaluda
Vamos a jugar

C:\Users\Lluís\Desktop\M16>python ej6
El animal llamado Rufosaluda
Vamos a jugar

C:\Users\Lluís\Desktop\M16>python ej7
python: can't open file 'ej7': [Errno 2] No such file or directory

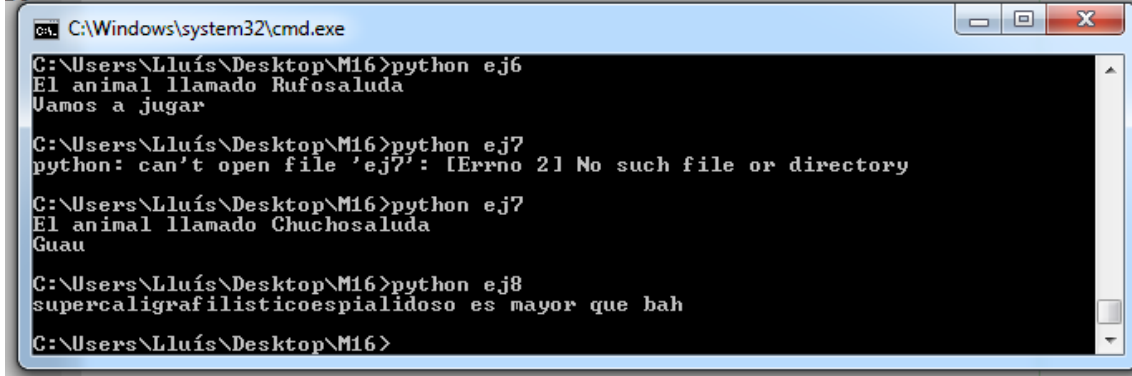
C:\Users\Lluís\Desktop\M16>python ej7
El animal llamado Chuchosaluda
Guau

C:\Users\Lluís\Desktop\M16>
```

Se ejecuta la función saluda de la clase animal, y se ejecuta la función ladra de la clase Perro.

Ejercicio 8

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-
3
4  class Palabra:
5      """Clase para mostrar el metodo __cmp__"""
6
7      def __init__(self, contenido):
8          self.contenido = contenido
9
10     def __cmp__(self, otro):
11         if self.contenido > otro.contenido:
12             return 1
13         elif self.contenido < otro.contenido:
14             return -1
15         else:
16             return 0
17
18
19     larga = Palabra("supercaligrafilisticoespialidoso")
20     corta = Palabra ("bah")
21
22     if (larga > corta):
23         print larga.contenido + " es mayor que " + corta.contenido
24     else:
25         print larga.contenido + " NO es mayor que " + corta.contenido
26
```



```
cmd: C:\Windows\system32\cmd.exe
C:\Users\Lluís\Desktop\M16>python ej6
El animal llamado Rufosaluda
Vamos a jugar
C:\Users\Lluís\Desktop\M16>python ej7
python: can't open file 'ej7': [Errno 2] No such file or directory
C:\Users\Lluís\Desktop\M16>python ej7
El animal llamado Chuchosaluda
Guau
C:\Users\Lluís\Desktop\M16>python ej8
supercaligrafilisticoespialidoso es mayor que bah
C:\Users\Lluís\Desktop\M16>
```

Compara las dos string y dependiendo de cual es mas grande hace un print o otro.

Comprobacion tres el cambio:


```
#!/usr/bin/python
# -*- coding: utf-8 -*-

class Palabra:
    """Clase para mostrar el metodo __cmp__"""

    def __init__(self, contenido):
        self.contenido = contenido

    def __cmp__(self, otro):
        if self.contenido < otro.contenido:
            return 1
        elif self.contenido > otro.contenido:
            return -1
        else:
            return 0

larga = Palabra("supercaligrafilisticoespialidoso")
corta = Palabra ("bah")

if (larga > corta):
    print larga.contenido + " es mayor que " + corta.contenido
else:
    print larga.contenido + " NO es mayor que " + corta.contenido
```

C:\Windows\system32\cmd.exe

C:\Users\Lluís\Desktop\M16>python ej8
supercaligrafilisticoespialidoso NO es mayor que bah