Sesión 4
Funciones, clases, paquetes y módulos

Rafael Cabañas de Paz
Curso: Introducción a Python
Almería, 10 Abril 2019

¿Por qué?

CDTIME

- Las **funciones, clases y paquetes** son bloques de código que realizan una serie de tareas.

  - Permiten la reutilización de código
  - La codificación se reduce en pequeñas tareas más simples
  - Facilitan el mantenimiento

```
l1 = [1,2,3,4]
for i in range(len(l1)-1):
    for j in range(i+1,len(l1)):
        if l1[i]>l1[j]:
            aux = l1[i]
            l1[i] = l1[j]
            l1[j] = aux

l2 = [13,4,1,21]

for i in range(len(l2)-1):
    for j in range(i+1,len(l2)):
        if l2[i]>l2[j]:
            aux = l2[i]
            l2[i] = l2[j]
            l2[j] = aux

l3 = [2,3,4,10,11]

for i in range(len(l3)-1):
    for j in range(i+1,len(l3)):
        if l3[i]>l3[j]:
            aux = l3[i]
            l3[i] = l3[j]
            l3[j] = aux
```

```
def ordenar(x):
    for i in range(len(x)-1):
        for j in range(i+1,len(x)):
            if x[i]>x[j]:
                aux = x[i]
                x[i] = x[j]
                x[j] = aux


l1 = ordenar([1,2,3,4])
l2 = ordenar([13,4,1,21])
l3 = ordenar([2,3,4,10,11])
```

- Python built-in

- pypi.org

CDTIME

- Las definiciones de pueden tener parámetros opcionales

Parámetro obligatorio

```
def potencia(base, exponente=2):
    return base**exponente
```

Parámetro opcional

Definición

```
In [11]: potencia(2,10)
Out[11]: 1024

In [12]: potencia(2)
Out[12]: 4
```

Llamadas

Funciones

CDTIME

Definición

```
def resta(a,b):
    return a-b
```

Llamadas

Parámetro keyword

```
In [3]: resta(a = 10, b = 5)
Out[3]: 5

In [4]: resta(b = 5, a = 10)
Out[4]: 5

In [5]: resta(10, b = 5)
Out[5]: 5

In [6]: resta(a = 10, 5)
  File "<ipython-input-6-d47775773d0b>", line 1
    resta(a = 10, 5)
                  ^
SyntaxError: positional argument follows keyword argument
```

Parámetro posicional

CDTIME

- Para devolver multiples valores utilizaremos las **tuplas**

Definición

```
def ultimo_y_anterior(n):
    return n-1, n-2
```

Llamadas

```
In [21]: t = ultimo_y_anterior(10)
         print(t)
```

```
(9, 8)
```

```
In [22]: a, b = ultimo_y_anterior(10)
         print(a)
         print(b)
```

```
9
8
```

```
In [23]: a, _ = ultimo_y_anterior(10)
         print(a)
```

```
9
```

CDTIME

- Número arbitrario de parámetros posicionales

Definición

```
def print_list(*args):
    for elem in args:
        print(elem)
```

Llamadas

```
In [25]: print_list()
```

```
In [26]: print_list(1,2,3)
1
2
3
```

```
In [27]: print_list(1,2,3, "abc")
1
2
3
abc
```

Clases

- Las *clases* definen un tipo genérico con **datos + funciones (métodos)**
- Las instancias de estas clases se denominan *objetos*
- Cada objeto puede tener distintos datos

```python
class Persona():
    def __init__(self, nombre):
        self.nombre = nombre

    def saluda(self):
        print(f"Hola mundo, soy {self.nombre}")
```

```
In [29]: p1 = Persona("Juan")
         p2 = Persona("Pepe")

In [30]: p1.saluda()
         p2.saluda()

Hola mundo, soy Juan
Hola mundo, soy Pepe
```

- Los métodos siempre tendrán el parámetro *self*

Paquetes y módulos

pandas

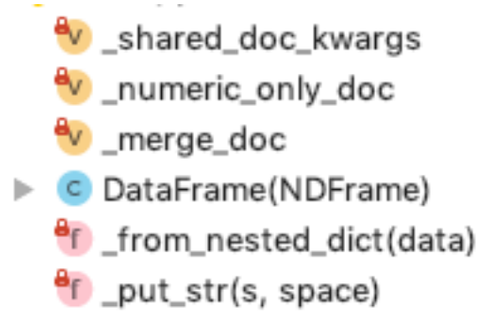
Paquete

core

Sub-paquete

frame.py

Módulo



Componentes

```
import pandas.core.frame
pandas.core.frame.DataFrame()
```

```
import pandas
pandas.DataFrame()
```

```
import pandas.core.frame as pdf
pdf.DataFrame()
```

```
import pandas as pd
pd.DataFrame()
```