



# Curso de Python

## 6 – Paquetes

Ramón Invarato Menéndez

Ricardo Moya García



# Paquetes

## Bibliotecas en Python

- Python tiene muchos paquetes pre-importados (los que hemos usado hasta ahora) de los métodos **Built-in** más usados: `range()`, `sorted()`, `list()`, `str()`, `int()`, etc.
- Otros paquetes que no están pre-importados y hay que importarlos a mano. Algunos vienen en Python y no están importados ya que no se usan tanto, para prevenir sobrescripciones no intencionadas (habría que conocer todos para no sobrescribirlos por accidente) y para sobrecargar menos el código (menos tiempo en ejecución). Otros son paquetes de terceros que hay que instalar previamente y tienen la misma causalidad que estos últimos de Python



- Tipos de paquetes:
  - Built-in (incorporadas)
  - Propias
  - De terceros (instalables con pip)
- Ubicación de los paquetes:
  - Global: Python/Lib
  - Local: En cualquier sitio

# Estructura de un Paquete

- Módulo: Cada **script** (fichero) de Python dentro de un Paquete (con las clases, funciones o variables a importar)
- Paquete: Directorio con una colección de módulos y el fichero `__init__.py`
- Directorio con scripts: Directorio con scripts de Python (sin fichero `__init__.py`)
- Distribución: **paquete** listo para **instalar** (con PIP, por ejemplo)
- Biblioteca (en Python): **Paquete** o **script** de Python preparado para importar.



# Importar paquetes (Import)

- Los paquetes en Python se importan con la palabra reservada "from" seguido del nombre del paquete a importar, luego la palabra reservada "import" para importar los métodos, variables o clases específicos (o todos con "\*")
- También se puede importar usando el espacio de nombres de la forma "import" seguido del nombre del paquete. Habrá que utilizar los métodos/variables/clases con el paquete delante (ejemplo: paquete.método() )
- Se suele importar al principio del fichero, aunque también se puede importar en mitad del código (esto último viene bien cuando se quiere utilizar un paquete, pero el código puede funcionar igual sin el paquete mediante otros métodos)

## Pseudo-código

```
from paquete import funcion, variable, clase
```

```
# Usar el método del paquete  
funcion()
```

## Pseudo-código (importar todo)

```
from paquete import *
```

```
# Usar el método del paquete  
funcion()
```

## Pseudo-código (espacio de nombres)

```
import paquete
```

```
# Usar el método con el espacio de nombres  
paquete.funcion()
```

## Nuestro propio paquete

- Ejemplo de importar desde el fichero "trabajando.py" la función "función\_modulo1" que tiene dentro el fichero "Modulo1.py".
- Para importar en este ejemplo se escribirá: **from** + espacio + **punto** (que indica la ruta actual, no escribir este punto estarías apuntando a directorio de Python con las bibliotecas instaladas) + el nombre del "paquete" + **punto** + "módulo" a importar + espacio + **import** + espacio + "la función a importar"



# Instalar paquetes de terceros con PIP

## pip install paquete

- Administrador de paquetes PIP

Instalación:

1. Crear un fichero llamado "**get-pip.py**" con el contenido de: <https://bootstrap.pypa.io/get-pip.py>
2. Ejecutar ese fichero "get-pip.py" con Python:

Consola

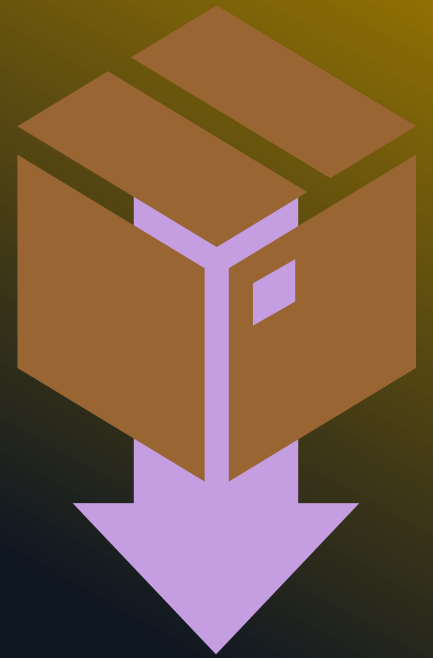
```
python get-pip.py
```

- Repositorio oficial de Python: <https://pypi.org/>

- Instalar un paquete (ejemplo para instalar el paquete Pandas):

Consola

```
pip install pandas
```



# Random

## Aleatorio

- `random()`: Devuelve un float en el rango [0,0, 1.0)
- `randrange(max)`: Devuelve un int [0, max)
- `uniform(min, max)`: Devuelve un float [min, max)
- `choice(lista)`: Devuelve un elemento al azar de la lista

### Código (importar)

```
from random import random, randrange, choice
```

### Código

```
resultado = random()
```

print

### Consola

0.28627298510709775

### Código

```
resultado = randrange(11)
```

print

### Consola

9

### Código

```
resultado = choice(["Cara", "Cruz"])
```

print

### Consola

Cruz

# Math

## Matemáticas (extensión de las fórmulas básicas)

- `ceil(num)`: redondea num (float) hacia arriba
- `floor(num)`: redondea num (float) hacia abajo
- `sqrt(num)`: raíz cuadrada de num
- `factorial(num)`: factorial de num
- `exp(num)`: e elevado a num
- `log(num, base)`: logaritmo de una base de num
- `pow(num, potencia)`: potencia de num (igual que: `num ** potencia`)
- `cos(num)`: Coseno de num
- `sin(num)`: Seno de num
- `tan(num)`: Tangente de num
- `pi`:  $\pi = 3,141592\dots$
- `e`:  $2,718281\dots$

### Código (importar)

```
from math import ceil, floor, sqrt, factorial
```

#### Código

```
resultado = ceil(10.5)
```

print

#### Consola

11

#### Código

```
resultado = floor(10.5)
```

print

#### Consola

10

#### Código

```
resultado = sqrt(9)
```

print

#### Consola

3.0

#### Código

```
resultado = factorial(5)
```

print

#### Consola

120

$$5! = 1 * 2 * 3 * 4 * 5 = 120$$



# Datetime

## Código

```
fecha_actual = datetime.now()
```

## Código (importar)

```
from datetime import datetime, timedelta
```

## Código

```
fecha = datetime(year=2019, month=12, day=31,  
                 hour=23, minute=59, second=59)
```

## Código

```
fecha = datetime.strptime("2019-12-31 23:59:59",  
                           "%Y-%m-%d %H:%M:%S")
```

print

## Consola

2019-12-31 23:59:59

## Fechas

- `now()`: devuelve la fecha actual
- `utcnow()`: devuelve la fecha UTC actual
- `datetime(...)`: inicializa un datetime
- `strptime(fecha, formato)`: obtiene el datetime desde un string con un formato de fecha fijo
- `strftime(formato)`: Convierte el datetime a un string con un formato dado
- `timedelta(tiempo)`: cantidad de tiempo

## Código

```
str_fecha = fecha.strftime("%d/%m/%Y %H:%M")
```

print

## Consola

31/12/2019 23:59

## Código

```
fecha_un_mes_mas = fecha + timedelta(days=31)
```

print

## Consola

2020-01-31 23:59:59

## Código

```
es_mayor = fecha < fecha_un_mes_mas
```

print

## Consola

True

¡GRACIAS!



Web: <https://jarroba.com/>

Ramón Invarato Menéndez

*Linked-in*

<https://www.linkedin.com/in/rinvarato/>

*Github*

<https://github.com/Invarato>

Ricardo Moya García

*Linked-in*

<https://www.linkedin.com/in/phdricardomoya>

*Github*

<https://github.com/RicardoMoya>