



Módulos y paquetes

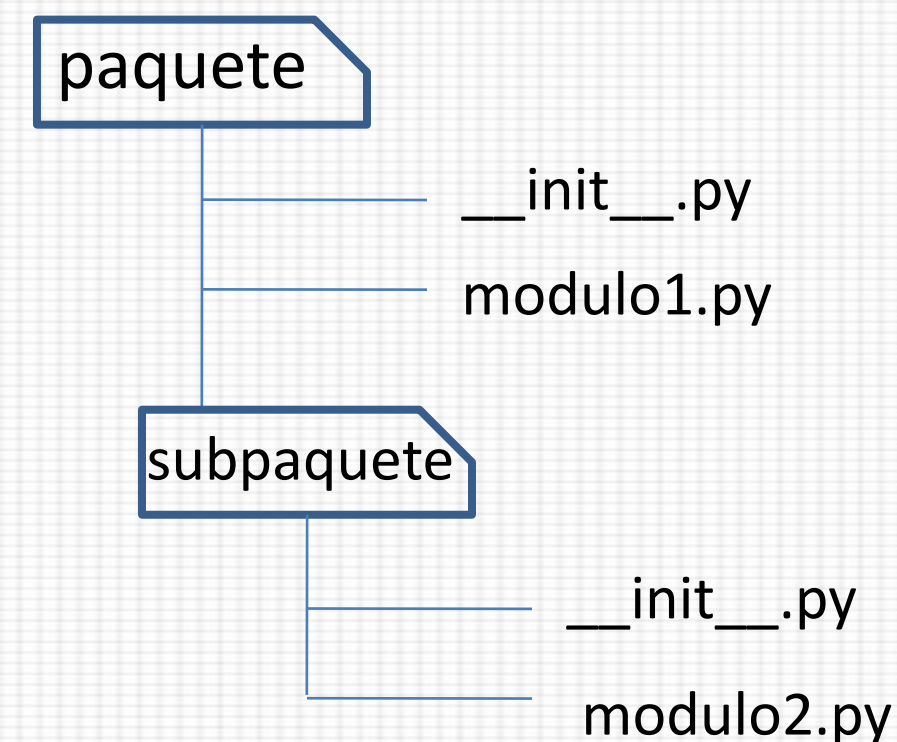
Paquetes

Índice

- Concepto de paquete
- El archivo `__init__.py`
- Distribución de paquetes
- Instalación de paquetes

Concepto de paquete

- Agrupación de módulos bajo un mismo directorio
- Directorio que incluye una serie de módulos y otros subpaquetes
- Cada paquete incluye un archivo `__init__.py`, que puede estar vacío pero que sirve para indicar que se trata de un paquete.
- Estructura:



El archivo `__init__`

- Puede estar vacío:

modulo1.py

```
def fun1():  
    print("Fun 1")
```

uso del módulo desde fuera
del paquete



uso.py

```
from paquete import modulo1  
modulo1.fun1()
```

- O incluir importaciones de módulos internos:

`__init__.py`

```
from paquete.modulo1 import *
```

modulo1.py

```
def fun1():  
    print("Fun 1")
```

uso del módulo desde fuera
del paquete



uso.py


```
import paquete  
paquete.fun1()
```


Distribución de paquetes

- Un paquete con sus correspondientes módulos y subpaquetes se puede empaquetar en un único archivo para facilitar su distribución
- Se debe crear un archivo `setup.py` en la carpeta que contiene al paquete con la siguiente estructura:

```
from setuptools import setup
setup(
    name="nombre_archivo",
    version="1.0",
    description="Este es un paquete de ejemplo",
    author="Antonio Martin",
    author_email="xxxxx@gmail.com",
    packages=['paquete1'],
    scripts=[]
)
```

Lista de paquetes
y subpaquetes



- Se genera el paquete con:

```
> c:\Users\usuario\Anaconda3\python setup.py sdist
```


Instalación de paquetes

- Al generar el archivo de distribución, se genera un archivo .zip en la subcarpeta del proyecto.
- Para instalar el paquete y que pueda utilizarse desde cualquier parte, utilizaremos pip, el comando habitual para realizar instalaciones en python, indicando la ruta del archivo a instalar:

```
> pip install nombre_archivo.zip
```