



NUEVOS PRODUCTOS



CRÍTICA DE PÁGINA WEB



PRUEBA DE SOFTWARE



BOOKREVIEW



PRUEBA DE APLICACIÓN

EL USO DE PAQUETES Y MÓDULOS

## Programa con éxito

En la tercera parte de esta serie de artículos analizamos las características básicas de la programación en Python. Ya discutimos con anterioridad los módulos en los que almacenábamos sus funciones. En este artículo, damos un paso más allá y observaremos aquellos paquetes que le dan acceso a funciones preprogramadas.

Antes de que echemos un vistazo de cerca a las funciones que ya vienen listas para usar, primero debemos entender cómo están estructuradas las funciones y cómo se pueden instalar en su ordenador. Si desea utilizar funciones precompiladas, encontrará que no siempre podrá descargarlas individualmente. A menudo se empaquetan en diferentes módulos y se conectan mediante paquetes.

### Módulos y paquetes

Primero, debemos distinguir los módulos de los paquetes. Los primeros son, como se explicó en el artículo anterior, los archivos de Python con la extensión ".py". En ellos se pueden almacenar funciones, variables, etcétera para luego recuperarlas y usarlas. En contraste, un paquete es una carpeta que contiene módulos diferentes o paquetes adicionales y cómo se distribuyen en un directorio. Es importante que Python reconozca una carpeta como un

paquete, para ello debe contener un archivo llamado "\_\_init\_\_.py" (más adelante hablaremos de ello).

### Crear módulos

Para aclarar los beneficios y la funcionalidad de los paquetes, crearemos un paquete juntos. Digamos que quiere operar acciones y divisas. También desea ver cómo ha cambiado el precio en su cartera en las últimas 24 horas. Usted creará 2 módulos: "acciones.py" y "monedas.py" (Imagen 1). En ellos deberá colocar la función "course\_change\_display ()" y además, agregará la variable "portfolio". Para simplificar las cosas, imaginaremos que solo tiene una acción y un par de divisas en su cartera. Si ha guardado los 2 módulos con sus nombres respectivos, ya podremos crear un paquete con ellos.

### Crear un paquete

Como se mencionó anteriormente, un paquete es una carpeta. Sin embargo, es necesario que Python sepa encontrarla. En el último artículo, agregamos la carpeta PythonModule a una lista de directorios en donde Python busca los módulos y paquetes. Por ello, también guardaremos el nuevo paquete en esta carpeta. Creemos una nueva carpeta "trading" y empaquetemos allí los 2 módulos creados previamente. Ahora todo lo que necesitamos es crear el archivo "\_\_init\_\_.py". Solo entonces



#### Alan Koska

Alan Koska estudia economía. Además, es el fundador de la iniciativa estudiantil Trading Mannheim y líder del proyecto Onyx, que se especializa en el desarrollo de algoritmos.

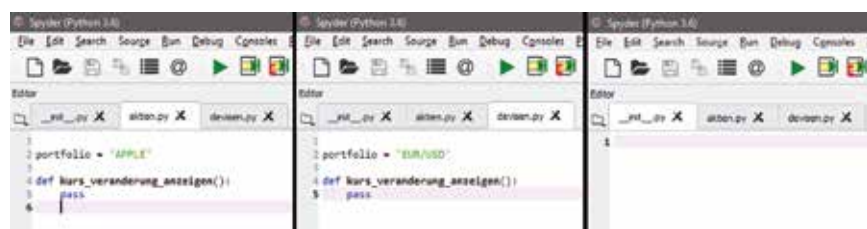


Python reconocerá que la carpeta "trading" es un paquete. Este archivo podrá estar completamente vacío, lo principal es que su nombre sea el correcto. En este archivo también se puede escribir un código de inicialización ya que siempre se ejecuta cuando se lee el paquete. Pero por el momento, sin embargo, no veremos esta capacidad por lo que dejaremos el archivo vacío.

### Acceso a funciones y variables

Si desea utilizar la función que le muestra el cambio de precio de su cartera de acciones, debe usar el comando "import (importar)". También debe nombrar el paquete. En este caso, lo llamaremos: "import trading.shares". Luego podrá acceder a la función con "trading.shares.course\_change\_show ()". Por supuesto, también podrá acceder a lo que esté en su cartera: "trading.shares.portfolio". Si a continuación, importamos el segundo módulo, podrá acceder mediante: "trading.currencias.portfolio", a lo que está en la segunda cartera (Fig. 2). También hay otra forma de importar módulos del paquete al transferir los nombres del módulo al espacio de nombres actual. Es decir, si escribe "portfolio", obtendrá la cartera del módulo que ha importado. Para

### G1 Módulos en Spyder



Crea 2 módulos en Spyder: shares.py y currencias.py (Ilus. 1). En él coloque la función "course\_change\_show ()".

Fuente: Spyder

ello escriba "from trading.shares import \*". Dado que podrá sobrescribir fácilmente las variables existentes, debe tener por ello sumo cuidado. Por ejemplo, si anteriormente escribió "portfolio = 2" y luego lanzó "from trading.shares import \*", la variable "portfolio" se sobrescribirá. Tomará el valor que corresponde a la variable "portfolio" en el módulo "acciones" del paquete "trading".

### Importar paquetes estándar

A través de Anaconda se han instalado varios paquetes estándar, que podrá importar a su Python y usarlos desde allí, de nuevo simplemente con el comando "import". Por ejemplo, si desea importar el paquete predeterminado

**Si desea utilizar la función que le muestra el cambio de precio de su cartera de acciones, debe usar el comando "import (importar)".**

"NumPy" (Numérico Python), simplemente escriba "import NumPy" en Python. NumPy es un paquete que se especializa en computación científica y se ejecuta de manera eficiente. Nos puede servir para tratar con grandes cantidades de datos lo cual es crucial en estos casos.

### Entornos de Python

Antes de atrevernos a instalar paquetes, primero debemos entender los entornos de Python. A veces puede suceder que algunos paquetes solo se ejecuten en versiones antiguas de Python. Usando un nuevo entorno de Python, podrá abrir diferentes versiones de Python en paralelo. Por ejemplo, podrá ejecutar la versión actual de Python 3.7 con paquetes específicos e iniciar la versión anterior de Python 3.5 en paralelo con otros paquetes. Los entornos Python están completamente separados los unos de los otros. Podrá instalar diferentes paquetes de diferentes versiones en diferentes entornos Python; es decir, versiones del software Python.

### Instalar paquetes a través de Anaconda

Si desea usar un paquete que aún no esté instalado a través de Anaconda, deberá hacerlo manualmente utilizando el navegador Anaconda o directamente desde la consola de Windows. Para ello, inicie Anaconda Navigator y vaya a la opción de menú "Entornos". Aquí hay 2 columnas: La columna de la izquierda que muestra los entornos ya creados de Python. En la columna de la derecha, podrá ver los paquetes instalados del entorno de Python seleccionados en la columna de la izquierda, si se ha seleccionado "Instalado" en la columna superior izquierda sobre la columna de la derecha. Para instalar paquetes adicionales, seleccione "All" (todos) en el cuadro de selección superior izquierdo de la columna de la derecha. Entonces podrá ver todos los paquetes instalables. En la columna de la derecha, podrá seleccionar un paquete y hacer clic en el cuadro de la izquierda (junto al nombre del paquete). Verá "Apply (Aplicar)" y "Clear (Borrar)" en la parte inferior de la columna derecha. Presione "Apply" y el paquete que seleccionó se instalará en el entorno de Python. Luego, podrá importar y usar

el paquete como de costumbre a través del comando "import" en Python (Figura 3).

### Instalar paquetes directamente a través de la consola de Windows

Si no desea buscar un paquete en la columna de la derecha, o si no encuentra el paquete, podrá avanzar

#### G2 Importar

##### Importar

**El espacio de nombres permanece**

```
En [2]: import trading.currencies
trading.currencies.course_change_show
trading.currencies.portfolio
```

Currencieschange

```
Fuera[2]: 'EUR/USD'
```

```
En [3]: import trading.currencies
trading.currencies.course_change_show
trading.currencies.portfolio
```

Cambio de acción  
Alteración de moneda

##### Se acepta el espacio de nombres

```
En [4]: de trading.shares import
course_change_show ()
portfolio
```

Cambio de acción

```
Fuera[4]: 'APPLE'
```

##### Importar paquetes estándar

```
En [5]: import numpy
```

Si desea utilizar la función que le muestra el cambio de precios de su cartera de acciones, use el comando "importar": "import trading.shares ". A continuación, podrá acceder a la función mediante trading.shares.course\_change\_show (). Por supuesto, también puede averiguar qué hay en su cartera mediante: " trading.shares.portfolio ". Si a continuación, importar el segundo módulo, podrá visualizar con "trading.currencies.portfolio", lo que está en la segunda cartera.

Fuente: Spyder

**Si desea importar el paquete  
predeterminado "NumPy" (Numérico Python),  
simplemente escriba "import NumPy" en Python.**

**G3 Instalar nuevos paquetes a través de Anaconda Navigator**

Arriba puede ver el navegador Anaconda. En el podrá instalar paquetes adicionales, seleccione "Todos" en el cuadro de selección superior izquierdo de la columna de la derecha. Después verá todos los paquetes instalables. En la columna de la derecha, podrá seleccionar un paquete y hacer clic en el cuadro de la izquierda (junto al nombre del paquete). Verá "Aplicar" y "Borrar" en la parte inferior de la columna derecha. Presione "Aplicar" y el paquete que seleccionó se instalará en el entorno de Python. A continuación, se le mostrará el proceso a través de la consola de Windows. Vuelva a la opción de menú "Entornos". En la columna de la izquierda verá los entornos de Python nuevamente. En su caso, existe la "base (raíz)" predeterminada de Python. A la derecha del nombre del entorno Python hay un triángulo verde en el que podrá hacer clic. Si luego hace clic en "Abrir Terminal", se iniciará el Centro de comando de Windows. En el lado izquierdo verá la entrada "(base)", que le mostrará en qué entorno de Python se encuentra y dónde está todo instalado. Si ahora introduce "Python" en el terminal, inmediatamente podría programar en Python.

Fuente: Anaconda Navigator

un paso más y dirigirse a la consola de Windows. Vuelva a la opción de menú "Entornos". En la columna de la izquierda verá los entornos de Python nuevamente. En su caso, existe la "base (raíz)" predeterminada de Python. A la derecha del nombre del entorno Python hay un triángulo verde en el que podrá hacer clic. Si luego hace clic en "Abrir Terminal", se iniciará el centro de comandos de Windows. En el lado izquierdo verá la entrada "(base)", que le mostrará en qué entorno de Python se encuentra y dónde se instalará todo. Si ahora pone "Python" en el terminal, inmediatamente podría programar en Python. Por ejemplo, use el comando "imprimir" como prueba. Para salir de Python, simplemente escriba "exit ()". Para instalar un paquete debe usar el administrador de paquetes y del entorno "conda". Le permite crear, usar o eliminar entornos de Python e instalar allí paquetes. Al escribir "conda" en la consola, obtendrá una lista de comandos en el terminal que Conda podrá ejecutar. Por ejemplo, si desea instalar el paquete NumPy, el comando es "conda install numpy".

### Conclusión

En esta parte de la serie, aprendió qué es un paquete y cómo se crea o instala. En el siguiente artículo instalaremos el paquete "Zipline" el cual es necesario para realizar pruebas históricas. Más adelante, nos familiarizaremos con la forma de implementar una estrategia de trading con el paquete backtest.