

★ Python 3 para impacientes ★

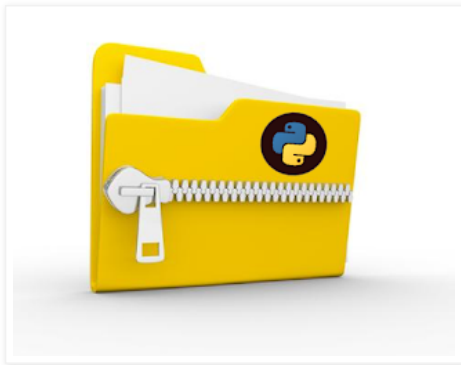


"Simple es mejor que complejo" (Tim Peters)

| | | | | | |
|--------|---------|---------|---------|------------|-------|
| Python | IPython | EasyGUI | Tkinter | JupyterLab | Numpy |
|--------|---------|---------|---------|------------|-------|

viernes, 30 de octubre de 2015

Empaquetar y desempaquetar archivos con shutil



El módulo **shutil** cuenta, desde Python 3.2, con funciones de alto nivel para empaquetar y desempaquetar archivos basadas en los módulos **zipfile** y **tarfile**.

Las operaciones que se pueden realizar son las siguientes:

- empaquetar y desempaquetar archivos con distintos formatos (zip, tar, etcétera);
- obtener una lista de formatos permitidos;
- y registrar nuevos formatos o suprimir formatos existentes de un sistema.

Empaquetar un directorio: `make_archive()`

La función **make_archive()** se utiliza para empaquetar o comprimir el contenido de un directorio. Dicha función devuelve el nombre del archivo creado, que será del tipo **zip**, **tar**, etc.

```
shutil.make_archive(base_name, format[, root_dir[, base_dir[,
    verbose[, dry_run[, owner[, group[, logger]]]]]])
```

Argumentos de la función:

- **base_name**: ruta y nombre del paquete a crear.
- **format**: formato del archivo: **zip**, **tar**, **bztar**. El formato **xztar** es soportado desde Python 3.5.
- **root_dir**: directorio a empaquetar (raíz).
- **base_dir**: directorio a empaquetar incluyendo **path**.
- **dry_run**: Si su valor es **True** no se creará el paquete, pero todas las operaciones que se ejecuten se registrarán en un archivo **log**.
- Los valores de los argumentos **owner** (propietario) y **group** se asignarán al paquete. Si se omiten, se asignarán los valores actuales del propietario y grupo.
- **Logger**: por lo general se corresponde con una instancia de **logging.Logger** que permitirá registrar todas las operaciones que se realicen durante el archivado.

Tanto el argumento **root_dir** como **base_dir** tienen por defecto asignado el **path** del directorio de trabajo actual.

Ejemplo: Empaqueta (o comprime) una carpeta

El siguiente ejemplo muestra el modo de empaquetar (o comprimir) una carpeta de fotografías llamada **"carpeta-fotos"** utilizando el formato **"zip"**. El archivo a crear se llamará **"viaje.zip"**.

```
import shutil

archivo_zip = shutil.make_archive("viaje", "zip", "carpeta-fotos")
print("Creado el archivo:", archivo_zip)
```

Buscar

Python para impacientes

[Python](#)
[IPython](#)
[EasyGUI](#)
[Tkinter](#)
[JupyterLab](#)
[Numpy](#)

Anexos

[Guía urgente de MySQL](#)
[Guía rápida de SQLite3](#)

Entradas + populares

[Dar color a las salidas en la consola](#)

En Python para dar color a las salidas en la consola (o en la terminal de texto) existen varias posibilidades. Hay un método basado ...

[Instalación de Python, paso a paso](#)

Instalación de Python 3.6 A finales de 2016 se produjo el lanzamiento de Python 3.6. El propósito de esta entrada es mostrar, pas...

[Añadir, consultar, modificar y suprimir elementos en Numpy](#)

Acceder a los elementos de un array. [], [,]. ... Acceder a un elemento de un array. Para acceder a un elemento se utiliz...

[Variables de control en Tkinter](#)

Variables de control Las variables de control son objetos especiales que se asocian a los widgets para almacenar sus valore...

[Cálculo con arrays Numpy](#)

Numpy ofrece todo lo necesario para obtener un buen rendimiento cuando se trata de hacer cálculos con arrays. Por como está concebido...

[Tkinter: interfaces gráficas en Python](#)

Introducción Con Python hay muchas posibilidades para programar una interfaz gráfica de usuario (GUI) pero Tkinter es fácil d...

[Operaciones con fechas y horas. Calendarios](#)

Los módulos datetime y calendar amplían las posibilidades del módulo time que provee funciones para manipular expresiones de ti...

[Convertir, copiar, ordenar, unir y dividir arrays Numpy](#)

Esta entrada trata sobre algunos métodos que se utilizan en Numpy para convertir listas en arrays y viceversa; para copiar arrays d...

[Tkinter: Tipos de ventanas](#)

Ventanas de aplicación y de diálogos En la entrada anterior tratamos los distintos gestores de geometría que se utilizan para di...

[El módulo random](#)

El módulo random de la librería estándar de Python incluye un conjunto de funciones

Ejemplo: Empaqueta carpeta incluyendo rutas

Para incluir el **path**, en este caso desde el directorio de trabajo actual, de todos los archivos y directorios en el archivo "viaje.zip":

```
archivo_zip = shutil.make_archive("viaje",
                                  "zip",
                                  base_dir = "carpeta-fotos")
```

La ruta de **base_dir** puede ser relativa al directorio de trabajo actual, como en el ejemplo, o absoluta (por ejemplo: /home/usuario/carpeta-fotos).

Ejemplo: Empaqueta carpeta y registra operaciones en un archivo log

En el siguiente ejemplo se registran en el archivo "viaje.log" todas las operaciones que se realizan durante la creación del paquete:

```
import shutil, logging
logging.basicConfig(level=logging.DEBUG,
                    format='%(asctime)s : %(levelname)s : %(message)s',
                    filename = 'viaje.log',
                    filemode = 'w',)
logging.info("Inicio del proceso")
archivo_zip = shutil.make_archive("viaje",
                                  "zip",
                                  base_dir = "carpeta-fotos",
                                  logger=logging)
logging.info("Fin del proceso")
```

viaje.log:

```
2015-10-30 20:44:01,147 : INFO : Inicio del proceso
2015-10-30 20:44:15,167 : INFO : creating 'viaje.zip' and adding 'carpeta-fotos' to it
2015-10-30 20:44:15,199 : INFO : adding 'carpeta-fotos/foto5.jpg'
2015-10-30 20:44:15,224 : INFO : adding 'carpeta-fotos/foto4.jpg'
2015-10-30 20:44:15,248 : INFO : adding 'carpeta-fotos/foto2.jpg'
2015-10-30 20:44:15,272 : INFO : adding 'carpeta-fotos/foto1.jpg'
2015-10-30 20:44:15,295 : INFO : adding 'carpeta-fotos/foto3.jpg'
2015-10-30 20:44:29,212 : INFO : Fin del proceso
```

Desempaquetar un directorio: unpack_archive()

La función **unpack_archive()** del módulo **shutil** se emplea para desempaquetar o descomprimir un archivo **zip**, **tar**, etcétera.

```
shutil.unpack_archive(filename[, extract_dir[, format]])
```

Argumentos de la función:

- **filename**: nombre del archivo comprimido (puede incluir el **path**).
- **extract_dir**: nombre del directorio destino donde se descomprimirá el archivo. Si no se indica, se asumirá el directorio de trabajo actual.
- **format** es el formato de archivo comprimido: **zip**, **tar**, **gztar** o cualquier otro formato de descompresión registrado con la función **register_unpack_archive()**.

Ejemplo: Desempaqueta (o descomprime) en el directorio actual

El siguiente ejemplo muestra la forma de descomprimir el archivo "viajes.zip". Todo su contenido será extraído en el directorio de trabajo actual.

```
archivo_zip = shutil.unpack_archive('viaje.zip')
```

Ejemplo: Desempaqueta en otro directorio diferente

Para extraer los archivos en otro directorio diferente al actual (por ejemplo. "fotografías"):

```
archivo_zip = shutil.unpack_archive('fotos-viaje.zip', 'fotografias')
```

Si el archivo fue comprimido utilizando el argumento **base_dir** todos los **path** de los archivos y directorios serán conservados durante la descompresión o desempaquetado.

que permiten obtener de distintos modos números a...

Archivo

octubre 2015 (2) ▼

python.org**pypi.org****Sitios**

- [ActivePython](#)
- [Anaconda](#)
- [Bpython](#)
- [Django](#)
- [Flask](#)
- [Ipython](#)
- [IronPython](#)
- [Matplotlib](#)
- [MicroPython](#)
- [Numpy](#)
- [Pandas](#)
- [Pillow](#)
- [PortablePython](#)
- [PyBrain](#)
- [PyCharm](#)
- [PyDev](#)
- [PyGame](#)
- [Pypi](#)
- [PyPy](#)
- [Pyramid](#)
- [Python.org](#)
- [PyTorch](#)
- [SciPy.org](#)
- [Spyder](#)
- [Tensorflow](#)
- [TurboGears](#)

Obtener formatos permitidos para empaquetar: `get_archive_formats()`

La función `shutil.get_archive_formats()` devuelve una lista con los formatos permitidos para empaquetar registrados en el sistema; siendo cada elemento de la secuencia devuelta una tupla con el contenido: (nombre, descripción).

Ejemplo: Obtiene formatos permitidos para empaquetar

```
shutil.get_archive_formats()
```

Salida:

```
[('bztar', 'bzip2'ed tar-file'),  
( 'gztar', 'gzip'ed tar-file"),  
( 'tar', 'uncompressed tar file'),  
( 'zip', 'ZIP file')]
```

Es posible registrar nuevos formatos o proporcionar un compresor propio con la función `register_archive_format()`.

Obtener formatos permitidos para desempaquetar: `get_unpack_formats()`

Devuelve una lista de todos los formatos registrados para desempaquetar. Cada elemento de la secuencia devuelto será una tupla con el contenido siguiente: (nombre, extensión, descripción).

Ejemplo: Obtiene formatos permitidos para desempaquetar

```
shutil.get_unpack_formats()
```

Salida:

```
[('bztar', ['.bz2'], 'bzip2'ed tar-file"),  
( 'gztar', ['.tar.gz', '.tgz'], 'gzip'ed tar-file"),  
( 'tar', ['.tar'], 'uncompressed tar file'),  
( 'zip', ['.zip'], 'ZIP file')]
```

Es posible registrar nuevos formatos o proporcionar un desempaquetador propio con la función `register_unpack_format()`.

Otras funciones relacionadas

Para registrar en el sistema un nuevo formato para empaquetar:

```
shutil.register_archive_format(name,  
                               function[, extra_args[, description]])
```

Para suprimir del sistema un formato existente para empaquetar:

```
shutil.unregister_archive_format(name)
```

Para registrar en el sistema un nuevo formato para desempaquetar:

```
shutil.register_unpack_format(name, extensions,  
                              function[, extra_args[, description]])
```

Para suprimir del sistema un formato existente para desempaquetar:

```
shutil.unregister_unpack_format(name)
```

[Ir al índice del tutorial de Python](#)

Publicado por Pherkad en [17:11](#)



Etiquetas: [Python3](#)

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)

2014-2020 | Alejandro Suárez Lamadrid y Antonio Suárez Jiménez, Andalucía - España
. Tema Sencillo. Con la tecnología de [Blogger](#).