

# LENGUAJES y HERRAMIENTA PARA CIENCIAS DE DATOS I

Librería estándar de Python III  
Compresión de datos



**UCO**  
ONLINE

# Librerías

- Comprimir un archivo o fuente de datos
  - ◆ zlib y gzip
  - ◆ Bz2
- Combinar varios archivos en uno
  - ◆ tarfile
  - ◆ zipfile



# Módulo zlib

- Funciones comprimir o descomprimir
  - ◆ *compress(bytes)*
  - ◆ *decompress(bytes)*

```
>>> import zlib
>>> datos = b'Cadena de prueba\n'*5
>>> print(datos, len(datos))
b'Cadena de prueba\nCadena de prueba\nCadena de prueba\nCadena de prueba\nCadena
de prueba\n' 85
>>> comprimido = zlib.compress(datos)
>>> print(comprimido, len(comprimido))
b'\x9c$NLI\xcdKTHIU(( *MMJ\xe4r\xa6\x82\x00\x00\xe6`\x1d\x07' 28
>>> descomprimido = zlib.decompress(comprimido)
>>> print(descomprimido, len(descomprimido))
b'Cadena de prueba\nCadena de prueba\nCadena de prueba\nCadena de prueba\nCadena
de prueba\n' 85
>>> █
```

# Módulo zlib

- Funciones para calcular suma de comprobación
  - ◆ *adler32()*
  - ◆ *crc32()*

```
>>> cksum = zlib.adler32(datos)
>>> cksum
3865058567
>>> cksum = zlib.crc32(datos)
>>> cksum
1407368743
```

# Módulo gzip

- Interfaz simple para comprimir/descomprimir archivos
  - ◆ Usa el módulo zlib para la compresión de los datos
- Clase GzipFile
  - ◆ open()
  - ◆ compress()
  - ◆ decompress()

```
1 import gzip
2
3 with gzip.open('datos.zip', 'wb') as f:
4     f.write(b'Información a comprimir')
5
6 with gzip.open('datos.zip', 'rb') as f:
7     leido = f.read()
8     print('Información en el fichero:', leido.decode())
```

# Módulo bz2

- Interfaz comprimir/descomprimir archivos
  - ◆ Algoritmo de compresión bzip2
- Clase BZ2File
  - ◆ Lee y escribir archivos comprimidos
- Clases Bz2Compressor y Bz2Decompressor
  - ◆ Compresión y descompresión de flujos de datos
- Funciones compress() y decompress()
  - ◆ Compresión y descompresión de un bloque de datos

# Módulo bz2

- Creación y lectura de un archivo comprimido

```
1 import bz2
2
3 with bz2.BZ2File('datos.bzip', 'wb') as f:
4     f.write(b'Información a comprimir')
5
6 with bz2.BZ2File('datos.bzip', 'rb') as f:
7     leido = f.read()
8     print('Información en el fichero:', leido.decode())
```

# Módulo tarfile

- Lectura y escritura archivos tar
  - ◆ `is_tarfile()`
  - ◆ `open()`
  - ◆ `getnames()`
  - ◆ `extractfile()`
  - ◆ `extract()`
  - ◆ `extractall()`
  - ◆ `add()`

```
1 import tarfile
2
3 ficheros = ['datos1.txt', 'datos2.txt', 'datos3.txt']
4
5 # Crea un fichero tar con datos*.txt
6 with tarfile.open('datos.tar', 'w') as f:
7     for fich in ficheros:
8         f.add(fich)
9
10 if tarfile.is_tarfile('datos.tar'):
11     with tarfile.open('datos.tar', 'r') as f:
12         print(f.getnames())
13         f.extractall('extraidos2')
```



# Módulo zipfile

- Manipular archivos comprimidos zip

- ◆ is\_zipfile()
- ◆ Clase ZipFile
  - namelist()
  - infolist() / getinfo()
  - write()
  - read()

```
1 import zipfile
2
3 ficheros = ['datos1.txt', 'datos2.txt', 'datos3.txt']
4
5 # Crea un fichero zip con datos*.txt
6 with zipfile.ZipFile('datos.zip', 'w') as f:
7     for fich in ficheros:
8         f.write(fich)
9
10 # Si es un fichero zip muestra el contenido de los ficheros
11 if zipfile.is_zipfile('datos.zip'):
12     with zipfile.ZipFile('datos.zip', 'r') as f:
13         print(f.namelist())
14         for fich in f.namelist():
15             print(f'{fich} -> {f.read(fich)}')
```

