



Aprenda a programar con Python 3

Entrada/Salida y trabajo con ficheros

Carolina Mañoso, Ángel P. de Madrid y Miguel Romero

Índice

◆ Entrada estándar

◆ Salida estándar

◆ Archivos

- Abrir y cerrar
- Lectura
- Escritura
- Mover el puntero
- Lectura desde la web

◆ Referencias



Entrada estándar

- ◆ La forma más sencilla de obtener información por parte del usuario es mediante la función `input()`.
 - Esta función toma como parámetro una cadena, para usar como texto a mostrar al usuario, pidiendo la entrada y devuelve una cadena con los caracteres introducidos por el usuario hasta que pulsa la tecla `Enter`.

```
nombre =input("introduce tu nombre:")  
print("hola", nombre)
```

- Si lo que se necesita es otro tipo de datos debemos usar la función conversora:

```
edad = int(input('Teclear edad: ')) # entrada de entero  
peso = float(input('Teclear peso: ')) # entrada de float
```

Salida estándar (1/2)

- ◆ La función `print()` se utiliza para mostrar información en la salida estándar que, normalmente, se corresponde con la pantalla de un ordenador.
 - La forma más sencilla es escribir el nombre de la función y pasarle como parámetro la cadena.
 - Después de escribir la cadena el puntero se sitúa en la siguiente línea de la pantalla. Si queremos que continúe en la misma línea finalizamos con el parámetro `end` que evita el salto de línea al final de la salida
 - Si queremos varias cadenas las separamos por comas
 - Podemos añadir caracteres especiales `/n`, `/t`
 - El operador `+` concatena cadenas.

Salida estándar (2/2)

```
print('Python') # muestra: Python
dis_bilbao = 385 # asigna valor a variable
print('Distancia Madrid-Bilbao', dis_bilbao, 'km')
print('Distancia Madrid-Bilbao' + str(dis_bilbao)+' km')
```

```
# muestra la cadena en varias líneas
print('Hola\n¿qué tal?\nos gusta el curso?')
```

```
# ejecuta varios print() en una misma línea
print('Continuará...', end=' ')
print('otro día')
```

```
# ejecuta varios print() en distintas líneas
print('Continuará...')
print('otro día')
```

```
# inserta una línea en blanco
print()
```

Archivos (1/9)

◆ Los ficheros en Python son objetos de tipo `file` creados mediante la función `open()`.

- Esta función toma como parámetros el nombre del archivo seguido, opcionalmente, por el modo o tipo de operación a realizar y la codificación que tendrá el archivo.

```
f = open('nombre_archivo', 'modo')
```

- Si no se indica el tipo de operación el archivo se abrirá en modo de lectura y si se omite la codificación se utilizará la codificación actual del sistema.
- Si no existe la ruta del archivo o se intenta abrir para lectura un archivo inexistente se producirá una excepción del tipo `IOError`.

```
Objf=open('/home/archivo.txt')
```

```
Objf=open('/home/archivo.txt', 'r')
```

```
Objf=open('/home/archivo.txt', mode='r', encoding='utf-8')
```

Archivos (2/9)

- El modo de acceso puede ser cualquier combinación de:
 - ♦ `'r'`: read, lectura. Abre el archivo en modo lectura. El archivo tiene que existir previamente, en caso contrario se lanzará una excepción de tipo `IOError`.
 - ♦ `'w'`: write, escritura. Abre el archivo en modo escritura. Si el archivo no existe se crea. Si existe, sobrescribe el contenido.
 - ♦ `'a'`: append, añadir. Abre el archivo en modo escritura. Se diferencia del modo `'w'` en que en este caso no se sobrescribe el contenido del archivo, sino que se comienza a escribir al final del archivo.
 - ♦ `'b'`: binary, binario.
 - ♦ `'r+'`: permite lectura y escritura simultáneas.
- Una vez terminemos de trabajar con el archivo debemos cerrarlo con el método `close()`

Archivos (3/9)

- ◆ Para la lectura de archivos se utilizan los métodos `read()`, `readline()` y `readlines()`:
 - El método `read()` devuelve una cadena con el contenido del archivo o bien el contenido de los n primeros bytes, si se especifica el tamaño máximo a leer.

```
archivo = open('archivo.txt','r') # abre archivo en modo lectura
cadena1 = archivo.read(9) # lee los 9 primeros bytes
cadena2 = archivo.read() # lee la información restante
```



Archivos (4/9)

- El método `readline` sirve para leer las líneas del archivo **una por una**, es decir, devuelve el contenido desde el puntero hasta donde se encuentra un carácter de nueva línea, incluyendo ese carácter.

```
>>>archivo = open('archivo.txt','r') #abre archivo para
lectura
>>>cadena1 = archivo.readline() # ... lee primera línea
>>>cadena2 = archivo.readline() # ... lee segunda línea
```

- Para leer líneas de un archivo se puede iterar sobre el objeto archivo.

```
archivo = open('archivo.txt','r') # abre archivo en modo
lectura
for linea in archivo:
    print(linea, end=' ')
```



Archivos (5/9)

- El método `readlines` funciona leyendo todas las líneas del archivo y devolviendo una lista con ellas

```
#leemos todas las lineas
archivo = open('archivo.txt','r')
lineas = archivo.readlines()# lee todas la líneas a una lista
numlin = 0# inicializa un contador
for linea in lineas:# recorre todas los elementos de la lista
    numlin += 1# incrementa en 1 el contador
    print(numlin, linea)#muestra contador y elemento de lista,línea
archivo.close()# cierra archivo
```



Archivos (6/9)

- ◆ Para la escritura se utilizan los métodos `write()` y `writelines()`:
 - `f = write('cadena')` escribe en el archivo una cadena de texto que toma como parámetro. Devuelve el número de caracteres escrito.
 - El método `writelines()` escribe una lista de cadenas de texto indicando las líneas que queremos escribir
- ◆ La declaración `with` tiene la característica que el archivo es cerrado de forma automática cuando el bloque termina, incluso si ha habido una excepción:

```
with open('nombre_archivo', 'modo') as variable:  
    bloque
```

```
with open('archivo.txt', 'r') as archivo:  
    lineas = archivo.readlines()
```

Archivos (7/9)

```
cadena1 = 'Datos'# declara cadena1
cadena2 = 'Secretos'# declara cadena2
archivo = open('datos1.txt','w')# abre archivo para escribir
archivo.write(cadena1 + '\n')# escribe cadena1 con salto de línea
archivo.write(cadena2)# escribe cadena2 en archivo
archivo.close()# cierra archivo
```

```
lista = ['lunes', 'martes', 'miercoles', 'jueves', 'viernes']
# declara lista
archivo = open('datos2.txt','w')# abre archivo en modo escritura
archivo.writelines(lista)# escribe toda la lista en el archivo
archivo.close() #cierra archivo
```



Archivos (8/9)

- ◆ Para mover el puntero de lectura/escritura a una posición determinada está el método `seek()`
 - Toma como parámetro un número que indica el desplazamiento. También se le puede indicar desde donde queremos que se haga el desplazamiento: 0 desde el principio, 1 posición actual y 2 final.
 - El método `tell()` determina la posición actual del puntero

```
archivo = open('datos2.txt','r') #abre archivo en modo r
archivo.seek(5) # mueve puntero al quinto byte
cadena1 = archivo.read(5) # lee los siguientes 5 bytes
print(cadena1) # muestra cadena
print(archivo.tell()) # muestra posición del puntero
archivo.close() # cierra archivo
```



Archivos (9/9)

◆ Para leer un archivo de la web necesitamos el módulo

`urllib.request`

- Utilizamos la función `urlopen(url)`.

```
import urllib.request
url = 'http://www.xtals.org/residue.dat'
with urllib.request.urlopen(url) as webpage:
    for line in webpage:
        print(line)
```



Referencias

- ◆ La documentación oficial, *Python Tutorial Release 3.5.2*. Guido van Rossum and the Python development team. Python Software Foundation. Junio 25, 2016.
<https://docs.python.org/3/download.html>
- ◆ *Learning Python 5th edition*. Mark Lutz. O'Reilly 2013
- ◆ *Practical Programming: An Introduction to computer Science using Python 3*. P. Gries, J. Campbell & J. Montoyo. Edited by Lynn Beighley. 2nd Edition. The Pragmatic Bookshelf. 2013
- ◆ *Learning Python with Raspberry Pi*. Alex Bradbury, Ben Everard. Wiley. 2014
- ◆ Tutorial: *Python 3 para impacientes*. Antonio Suárez Jiménez. 2014
<http://python-para-impacientes.blogspot.com.es/p/indice.html>
- ◆ *Raspberry Pi, User Guide*. Eben Upton, Gareth Halfacree. Wiley. 2012

Aviso



Aprenda a programar con Python 3 by C. Mañoso, A. P. de Madrid, M. Romero is licensed under a [Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional License](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Esta colección de transparencias se distribuye con fines meramente docentes.

Todas las marcas comerciales y nombres propios de sistemas operativos, programas, hardware, etc. que aparecen en el texto son marcas registradas propiedad de sus respectivas compañías u organizaciones.