Introducción a la Programación con PYTHON

José Rodriguez

Unidad 4 - Entrada-Salida

UNIDAD 4 – ENTRADA/SALIDA ESTANDARD

ÍNDICE

1.- Entrada Estándar

La entrada Standard es, como en la mayoría de lenguajes de programación, el teclado. Python propone dos métodos para introducir datos, el primero es pasarle los datos directamente desde la línea de comandos cuando se invoca al interprete, el segundo método utiliza la función de recogida de datos input().

En el primer método los datos se almacenan en un array cuyo nombre es argv y cuyo primer elemento es el propio nombre del programa que se le envía al intérprete. Este array está incluido en la librería sys y debemos importarlo previamente. Veamos un ejemplo, el fichero 2.py contiene el siguiente código

```
import sys
print sys.argv[0]
```

Si ahora invocamos al interprete pasándole el nombre del programa para ejecutarlo

```
Python 2.sys
```

El resultado sería, obviamente,

```
2.sys
```

Si ahora modificamos el código para incluir un segundo parámetro

```
import sys
print sys.argv[0]
print sys.argv[1]
```

Y llamamos al intérprete pasándole un segundo parámetro

```
Python 2.sys 3
```

El resultado sería

```
2.sys
3
```

En general, la forma de invocar al intérprete es

Python nombreprograma.py p1 p2 p3….

En la cual, los parámetros van seguidos y separados únicamente por un espacio en blanco.

Nota:

Si intentamos utilizar un parámetro que no ha sido pasado a través de la línea de comandos se producirá un error, sin embargo, si pasamos más parámetros de los que realmente necesita el programa no ocurre ningún error, simplemente el parámetro es ignorado.

La forma más sencilla de obtener información por parte del usuario es mediante la función input. Esta función toma como parámetro una cadena a usar como prompt (es decir, como texto a mostrar al usuario pidiendo la entrada) y devuelve una cadena con los caracteres introducidos por el usuario hasta que pulsó la tecla Enter. Veamos un pequeño ejemplo:

```
edad = input("Cual es tu edad? ");
print ("hola, tienes " + edad + " agnos")
```

(No olvides que la codificación de Python no incluye el carácter ñ.)

2.- La Salida estándar.

La salida es, por defecto, la pantalla. Para mostrar datos por pantalla disponemos del comando print () cuya sintaxis más básica es la siguiente:

```
Print ( "Hola Mundo")
```

Print () escribe el texto que se le pasa como argumento entre comillas dobles y genera un salto de línea y retorno de carro.

Así, el script

```
print ("Hola ",)
print ("Mundo")
```

mostrará por pantalla el siguiente texto

```
Hola
Mundo
```

El comando print admite una salida formateada utilizando los parámetros de formato siguientes

Especificador	Formato
%s	Cadena
%d	Entero
%o	Octal
%x	Hexadecimal
%f	Real

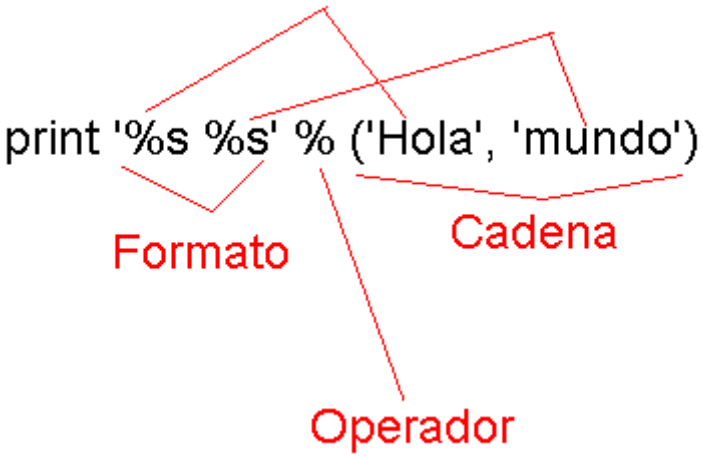
La sintaxis contiene dos campos, el segundo contiene la cadena que queremos escribir y el primero contiene el formato. Ambos están separados por el operador "%".

print '%s' % 'Hola'

Si queremos formatear la salida de varias cadenas concatenadas éstas deben ir encerradas entre paréntesis y en el orden preciso. Los operadores de formateo también deben ir en el mismo orden.

print ('%s %s' % ('Hola', 'mundo'))

Fíjate bien en la estructura:



El formato de salida permite asignar la longitud con la cual se va a representar cada cadena indicando el número de caracteres, por ejemplo

print ('%10s %10s' % ('Hola', 'mundo'))

Dando como resultado

Hola Mundo

La cadena Hola ocupa 10 posiciones y está, evidentemente, alineada a la derecha en esos 10 caracteres.

También podemos incluir caracteres externos a las propias cadenas incluyéndolos en el apartado de formato

```
print ('%10s - %10s' % ('Hola', 'mundo'))
```

daría como resultado

```
    Hola -      Mundo
```

El formato no tiene necesariamente que afectar a toda la cadena, podemos mezclar texto formateado con texto sin formato utilizando una estructura similar a esta

```
print ('Hola %10s' % ('Mundo'))
```

O bien esta

```
print ('%10s Mundo' % ('Hola'))
```

Veamos un último ejemplo que incluya estas características

```
for x in range(1,11):
        print ( 'El cuadrado de %3d es %3d' % (x,x*x))
```

Su resultado sería

```
El cuadrado de   1 es   1
El cuadrado de   2 es   4
El cuadrado de   3 es   9
El cuadrado de   4 es  16
El cuadrado de   5 es  25
El cuadrado de   6 es  36
El cuadrado de   7 es  49
El cuadrado de   8 es  64
El cuadrado de   9 es  81
El cuadrado de  10 es 100
```

Para formatear los números reales el mecanismo es similar pero en este caso hemos de indicar el número de caracteres decimales precedidos por el punto decimal “.”

```
radio=3.0
from math import pi
area=radio*radio*pi
print ('El área del circulo de radio %.3f es %.6f' % (radio, area))
```

El resultado sería

El área del círculo de radio 3.000 es 28.274334