Entrada y Salida

Lectura de datos de teclado. Raw_input()

Podemos leer datos desde el teclado, de forma interactiva. La forma de hacerlo es utilizar la función **raw_input()**.

Esta función hace lo siguiente: detiene la ejecución del programa y espera a que el usuario escriba un texto y pulse la tecla de retorno de carro; en ese momento prosigue la ejecución y la función devuelve una cadena con el texto que tecleó el usuario.

Si deseas que el dato sea un valor flotante, debes transformar la cadena devuelta por raw_input utilizando la función **float**. Lo mismo con los valores enteros, utilizando la función **int**.

La función raw_input acepta un argumento: una cadena con el mensaje que debe mostrar.

Posibles llamadas:

- `x = raw_input()`
- `x = float(raw_input())`
- `x = int(raw_intput())`
- `x = raw_input('Dato = ')`

Presentación de datos por pantalla: `print`

Para presentar algo por pantalla, basta utilizar la instrucción `print`

```
In [5]: a = 5
        print a
```

```
5
```

Podemos presentar varios datos separados por comas.

```
In [6]: print a, 2*a, a*a
```

```
5 10 25
```

También podemos presentar mensajes de texto, entre comillas.

```
In [7]: print a,'al cuadrado es',a*a
```

```
5 al cuadrado es 25
```

Podemos evitar el salto de línea con una coma al final.

Salida con formato

Para mejorar las presentaciones, podemos dar distintos formatos a la salida.

```
In [16]: a = 0.5
```

```
a = 0.5
i = 1
while i < 11:
    print a,'elevado a',i,'es',a**i
    i+=1
```

```
0.5 elevado a 1 es 0.5
0.5 elevado a 2 es 0.25
0.5 elevado a 3 es 0.125
0.5 elevado a 4 es 0.0625
0.5 elevado a 5 es 0.03125
0.5 elevado a 6 es 0.015625
0.5 elevado a 7 es 0.0078125
0.5 elevado a 8 es 0.00390625
0.5 elevado a 9 es 0.001953125
0.5 elevado a 10 es 0.0009765625
```

In [17]:
```
a = 0.5
i = 1
while i < 11:
    print '%f elevado a %d es %f' %(a,i,a**i)
    i+=1
```

```
0.500000 elevado a 1 es 0.500000
0.500000 elevado a 2 es 0.250000
0.500000 elevado a 3 es 0.125000
0.500000 elevado a 4 es 0.062500
0.500000 elevado a 5 es 0.031250
0.500000 elevado a 6 es 0.015625
0.500000 elevado a 7 es 0.007812
0.500000 elevado a 8 es 0.003906
0.500000 elevado a 9 es 0.001953
0.500000 elevado a 10 es 0.000977
```

In [20]:
```
a = 0.85
i = 1
while i < 11:
    print '%4.2f elevado a %2d es %5.3f' %(a,i,a**i)
    i+=1
```

```
0.85 elevado a  1 es 0.850
0.85 elevado a  2 es 0.722
0.85 elevado a  3 es 0.614
0.85 elevado a  4 es 0.522
0.85 elevado a  5 es 0.444
0.85 elevado a  6 es 0.377
0.85 elevado a  7 es 0.321
0.85 elevado a  8 es 0.272
0.85 elevado a  9 es 0.232
0.85 elevado a 10 es 0.197
```

Aparte de %d y %f existe la marca %s para cadenas

In [21]:
```
c = 'X'
i = 0
while i < 10:
    print 'La cadena es', c
    c = c + 'Y'
    i += 1
```

```
La cadena es X
La cadena es XY
La cadena es XYY
La cadena es XYYY
La cadena es XYYYY
La cadena es XYYYYY
La cadena es XYYYYYY
La cadena es XYYYYYYY
La cadena es XYYYYYYYY
La cadena es XYYYYYYYYY
```

In [22]:

```
c = 'X'
i = 0
while i < 10:
    print 'La cadena es %10s' %(c)
    c = c + 'Y'
    i += 1
```

```
La cadena es          X
La cadena es         XY
La cadena es        XYY
La cadena es       XYYY
La cadena es      XYYYY
La cadena es     XYYYYY
La cadena es    XYYYYYY
La cadena es   XYYYYYYY
La cadena es  XYYYYYYYY
La cadena es XYYYYYYYYY
```

In [ ]: