

Cookbook Python

August 24, 2014

1 Empezando

Para abrir Python escribimos en la terminal `python` lo que arroja lo siguiente:

```
python
Python 2.7.4 (default, Sep 26 2013, 03:20:26)
[GCC 4.7.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

`>>>` indica que estamos dentro de python y podemos empezar a escribir. Por ejemplo:

```
>>> 1
>>> 1
```

Es muy útil escribir aritmetica e.g:

```
>>> 3*5
>>> 15
```

Si quiero dividir dos numeros tengo que especificar que son decimales de lo contrario el resultado sera aproximado al entero mas cercano. Para especificar que un número es un decimal utilizamos el punto así:

```
>>> 40/25.0
>>> 1.6
```

otra operación importante es el operador modulo (%) que se define como el sobrante de la division del numero a entre b veces.

```
>>> 10%3
>>> 1
```

ya que 3 puede estar 3 veces en 10 y hay un sobrante de 1.

Con el fin de escribir programas lo mejor será abrir un editor de texto:

```
emacs holamundo.py
```

Acá es donde escribiremos todos nuestros programas.

Nuestro primer programa sera imprimir `Hola mundo`, para eso escribimos lo siguiente:

```
print 'hola mundo'
```

Las comillas sencillas o dobles se pueden usar para indicar que imprimimos strings (texto), Para ejecutar el programa escribimos en la terminal:

```
python holamundo.py  
Hola mundo
```

2 Variables:

Que tipo de variables tneemos en python?, Para esto abramos un documento `emacs variables.py` & y escribamos:

```
a = 1  
b = 1.0  
c = "hola"  
  
print a, b, c  
print type(a), type(b), type(c)
```

Esto arrojaría el siguiente resultado:

```
1 1.0 hola  
<type 'int'> <type 'float'> <type 'str'>
```

Las variables tipo `int` guardan números enteros, mientras que las tipo `float` guaradn flotantes que son numeros relaes y las tipo `str` guardan string (texto).

3 Condicionales (if y While)

Podemos controlar los programas poniendo condiciones en el. Por ejemplo si queremos imprimir algo solo si alguna condicion cumple, por ejemplo si $a == b$.

```
a = 2.0  
b = 3.0  
if (a==b):  
    print a, "es igual que", b  
elif (a>b):  
    print a, "es mayor que ". b  
else:  
    print b, "es mayor que", a
```

En este caso el resultado de este programa será:

```
3.0 es mayor que 2.0
```

La sintaxis de `if` es: entre parentesis la condición (En programación `==` es igual, `>` es mayor o igual, `<` es menor o igual) seguido por los dos puntos `(:)` y luego indexado (espaciado) lo que se quiera que vaya dentro de la condicion. `elif` funciona como la contraccion de `else if`.

Tambien podemos poner condiciones usando `while`

```
i = 0
while (i<10):
    print i
    i += 1
```

Esto va a producir la siguiente cuenta regresiva:

```
0
1
2
3
4
5
6
7
8
9
```

4 Iteración

Otra forma de controlar el programa es mediante iteraciones esto lo hacemos mediante un **for**:

```
for i in range(5):
    print "hola"
```

Lo que arrojará lo siguiente:

```
hola
hola
hola
hola
hola
```

5 Listas

En muchos casos es útil, crear listas con diferentes tipos de variables. Por ejemplo:

```
Lista = [1.0, 3, "hola"]
```

Si queremos imprimir toda la lista hacemos:

```
print Lista
```

Podemos llamar a un elemento en particular de la lista solo conociendo su posición, En python las posiciones empiezan desde cero, es decir que en este caso el primer elemento de la lista 1.0 lo podemos llamar así:

```
print Lista[0]
```

Y así los siguientes elementos:

```
Lista[1]  
Lista[2]
```

imprimirán respectivamente 3 y 'hola'

Si queremos imprimir los elementos 1 y 2 podemos usar:

```
print Lista[0:2]
```

Esta notación significa imprima todos los elementos desde el [0] hasta antes del elemento [2]

6 Arreglos