

Primera Parte Proyecto: Uso y Definición de Funciones

Luis Garreta

March 16, 2015

1 Funciones Integradas de Python

Una función integrada de python (*built-in* function) es la que python tiene por defecto para ser usadas sin necesidad de importar ningún módulo. Estas funciones en realidad pertenecen al módulo “*__builtins__*” (doble rayita-baja). Otras funciones del mismo módulo las puede ver llamando a la función `dir` desde el prompt:

```
>>> dir (__builtins__)
['abs', 'all', 'any', 'apply', 'basestring', 'bin', 'bool', 'buffer', '
 bytearray', 'bytes', 'callable', 'chr', 'classmethod', 'cmp', '
 coerce', 'compile', 'complex', 'copyright', 'credits', 'delattr', '
 dict', 'dir', 'divmod', 'enumerate', 'eval', 'execfile', 'exit', '
 file', 'filter', 'float', 'format', 'frozenset', 'getattr', '
 globals', 'hasattr', 'hash', 'help', 'hex', 'id', 'input', 'int', '
 intern', 'isinstance', 'issubclass', 'iter', 'len', 'license', '
 list', 'locals', 'long', 'map', 'max', 'memoryview', 'min', 'next',
 'object', 'oct', 'open', 'ord', 'pow', 'print', 'property', 'quit',
 'range', 'raw_input', 'reduce', 'reload', 'repr', 'reversed', '
 round', 'set', 'setattr', 'slice', 'sorted', 'staticmethod', 'str',
 'sum', 'super', 'tuple', 'type', 'unichr', 'unicode', 'vars', '
 xrange', 'zip']
```

Puede usar la función *help* () para ver la ayuda de cada función (qué hace, qué parámetros lleva, y qué retorna). Por ejemplo, digitando en el prompt de python: `>>> help (abs)`, imprime

```
>>> help (abs)
abs(...)
abs(number) -> number
Return the absolute value of the argument.
```

Lo que nos dice que *abs* calcula y retorna el valor absoluto del número que se le ingrese como parámetro. Por ejemplo:

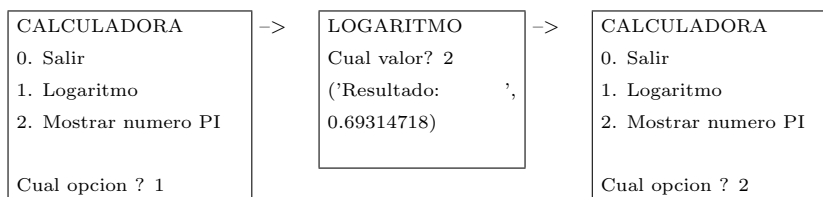
```
>>> abs (-2.3)
2.3
```

2 Problemas

Construya un programa para cada uno de los siguientes problemas. Cada programa debe presentar al usuario un menu con las diferentes opciones que el usuario selecciona de forma continua hasta que escoge la opción “0” (cero) que termina toda la ejecución del programa. Cada opción ejecuta lo necesario e imprime el resultado que se solicita y termina, para luego volver a mostrar el menu de opciones.

Cada opción del menu debe llamar una funcion creada por usted que ejecuta lo que se pide en la opción. Su función puede realizar lo que se solicita implementando directamente la tarea (algoritmo) o usando funciones ya creadas de los diferentes módulos de Python (Ej. *math*, *string*, *etc*).

Aquí está un ejemplo con un menú de dos opciones para el segundo problema de funciones matemáticas (Programa 1):



Programa 1.

Escriba un programa en python que use unas de sus funciones integradas: *min*, *max*, *range*, *abs*, *round* y *pow* para imprimir las siguientes respuestas. Construya un solo programa para imprimir estos resultados.

1. El valor absoluto de -15.5
2. Me redondee el número 3.456789123 a dos decimales
3. Imprima el mínimo y máximo de la secuencia [1.5, 6.7, 4.3, -12.3]
4. Construya una secuencia de 0 a 20 de 1 en 1
5. Construya una secuencia de 0 a 30 de 3 en 3
6. Ordene la secuencia [1,6,3,8,2,4] de forma ascendente y descendente.

Programa 2.

Escriba un programa que actúe como calculadora científica para operaciones entre números. El programa imprime un menú con mínimo 5 opciones (al menos una que involucre dos operandos). El usuario selecciona una de ellas, y el programa ejecuta lo necesario para esa opción y muestra el resultado. Es decir si la operación requiere un solo número, lo solicita (lee) y después imprime el resultado. Si involucra dos números, los lee, ejecuta la operación e imprime el resultado.

Si quiere, implementa las funciones o puede usar las que ya están implementadas en el módulo *math* de python. Un ejemplo de la interacción se presentó en la figura anterior.

Programa 3

Escriba un programa (también con menu y opciones para cada operación) que use las funciones del módulo *string* para ejecutar las siguientes operaciones con la secuencia:

“gcagtgactaagaattccacctaactacctgggagagaggaaataaactatatcgcagcagtaaggaggtggaatgg”

1. Cuantas ocurrencias de la subsecuencia **“gagg”** están en la secuencia principal?
2. Cúal es la posición inicial donde está la primera ocurrencia de la subsecuencia anterior (**“gagg”**)
3. Cuál es el contenido de **g+c** en la secuencia.
4. Cambie la secuencia a letras mayúsculas.
5. Reemplace las timinas por uracilos.
6. Un paso muy simplista para identificar posibles genes es buscar marcos de lectura abierta (ORFs, de sus siglas en inglés). Y un paso en esta búsqueda es determinar regiones de la secuencia que no contengan codones de parada. De la secuencia anterior, encuentre todos los posibles ORFs tanto en la hebra positiva como en la hebra negativa, es decir encuentre las subregiones que estén delimitadas por el codon de parada **“TAA”** e imprimalas.

Programa 4

El número de aves que se agrupan en diferentes sitios del parque de los Farallones de Cali ha sido contado por otros compañeros suyos y registrado en la siguiente table (datos completos están en el archivo *“datos-aves.csv”*):

Sitio	Número
PicoLoro	28
PicoLoro	32
PicoLoro	1
.....
PicoPance	145
PicoPance	27
PicoPance	36
.....
PicoAguila	122
PicoAguila	87
PicoAguila	36
.....
PicoSimio	0
PicoSimio	5
PicoSimio	55
.....

Realize un programa que cargue estos datos desde un archivo externo (“datos-aves.csv”), y los cargue en dos listas: listaSitios y listaConteo. Con los datos cargados en estas dos listas, el programa debe dar las opciones para responder las siguientes preguntas:

1. Cuantas aves fueron contadas en total?
2. Cuantas aves fueron contadas en la sexta muestra?
3. Cuantos sitios fueron muestreados y cuales son?
4. Cúal fue el promedio de aves muestreadas?
5. Cual es el total y el promedio de aves por sitio