UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE PSICOLOGÍA

CURSO DE INTRODUCCIÓN A LA PROGRAMACIÓN EN *PYTHON*

JUNIO 2017

Edgar de Jesús Vázquez Silva, Facultad de Ingeniería

Índice

1. Introducción

Este documento tiene la finalidad de ayudar a los alumnos de psicología a entender la estructura de un programa en Python, crear un programa básico, además de conocer algunas librerías útiles, por ejemplo: generación de gráficos.

2. Conociendo Python

Como estudiantes, profesores o curiosos de la programación hemos de buscar un lenguaje de programción que nos sea útil según nuestras necesidades. Es importante mencionar que existen muchos, y muy diversos, lenguajes de programación: R, Ruby, Java, Python, C, C++, por mencionar algunos. Cada uno de ellos ofrece características que lo hacen destacar según las necesidades del desarrollador.

Para comenzar a programar en Python necesitaremos una interfaz que nos facilite la tarea de generar Çódigos"(programas computacionales), lo más recomendable para las personas que no han tenido un acercamiento a un lenguaje de programación es utilizar una *interfaz gráfica conocida como IDLE.* Otra opción es utilizar un editor de texto como SublimeText, NotePad++ ó Atom, y ejecutarlo desde la consola. Se recomienda al lector consultar la guía de instalación del IDLE Spyder, desarrollado por los alumnos del LAB 25. `https://github.com/Lab25UNAM/PAPIME2016/blob/master/guia_Instalacion.pdf`

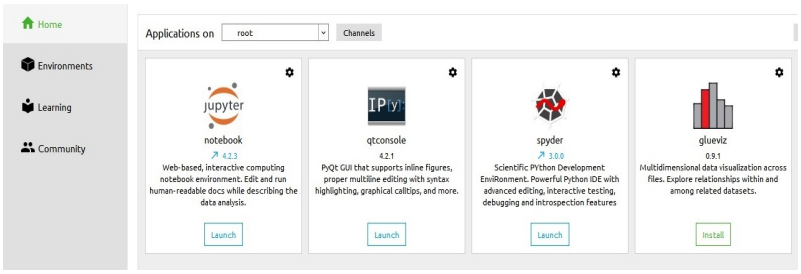
En los siguientes ejemplos, las entradas y salidas son distinguidas por la presencia o ausencia de los prompts ¿: para reproducir los ejemplos, debes escribir todo lo que esté después del prompt, cuando este aparezca; las líneas que no comiencen con el prompt son las salidas del intérprete.

Una vez instalado la suite de desarrollo Anaconda, buscaremos el entorno de desarrollo *Spyder.* Con esto estaremos listos para comenzar a desarrollar programas en lenguaje de programación Python. ***Referencia a archivo de instalación en el git.

2.1. Escribiendo un hola mundo

Una práctica muy conocida en el mundo de programación es comenzar a escribir un programa de bienvenida al lenguaje de programación. Para ello seguiremos estos sencillos pasos:

- Buscar Anaconda en en menú inicio del sistema.
- Abrir el IDE Spyder.
- Una vez en editor de Spyder, escribir el siguiente comando.




```
1        print "Hola mundo...."
2
```

Con lo cual observaremos en la terminal del editor el mensaje anteriormente escrito.

3. Variables

Una de las características más poderosas en un lenguaje de programación es la capacidad de manipular variables. Se entiende como variable el nombre dado a un valor.

La sentencia de asignación crea nuevas variables y les da valores:

```
>>> mensaje = " Que  Onda?"
>>> n = 17
>>> pi = 3.14159

```

Este ejemplo hace tres asignaciones. La primera asigna la cadena "¿Que Onda?.ª una nueva variable denominada mensaje. La segunda le asigna el entero 17 a n, y la tercera le da el valor decimal 3.14159 a pi.

3.1. Tipos de variables

Hasta este punto ya hemos declarado algunas variables; sin embargo es importante conocer que tipo de valores podemos asignar a estas variables. Dentro de los tipos de datos principales que podemos utilizar en Python estan:

- Boleanos (bool): Son tipos de datos que solo pueden tomar dos valores True o False (Verdader o Falso). Este tipo de variables nos pueden ayudar a guardar información que responda a una pregunta de afirmación. Por ejemplo, ¿Una persona es mayor de edad?, ¿Aún hay leche en el refrigerador? ¿Una paloma ha presionado el botón azul?, etc.
- Enteros (int): Se refiere a los tipos de datos que podemos representar con un numero entero. Por Ejemplo el numero de hermanos que una persona puede tener.
- Flotantes (float): Los tipos de datos flotantes se refieren a valores numéricos con punto decimal, Por ejemplo para almacenar el valor de la estatura de una persona utilizaremos una varible de tipo flotante.
- Cadenas (string): Son tipos de datos que se refieren a un conjunto de caracteres. Este tipo de datos son muy utilizados para manejo de textos, por ejemplo despliegue de mensajes o información.

- Arreglos (Arrays): Este tipo de datos en realidad es una extensión de los anteriores, y se refiere a la forma en que organizamos los datos. Un arreglo es un acomodo especial de datos de forma contigua, a la cual accedemos mediante un indice. Por ejemplo, si queremos guardar las calificaciones de 25 alumnos es más conveniente hacerlo en un arreglo de flotantes llamado çalificaciones.^{en} lugar de declarar 25 variables de tipo flotante.

Muchos de los ejemplos de este manual, incluso aquellos ingresados en el prompt, incluyen comentarios. Los comentarios en Python comienzan con el carácter numeral, , y se extienden hasta el final físico de la línea. Un comentario quizás aparezca al comienzo de la línea o seguidos de espacios blancos o código, pero sin una cadena de caracteres. Ya que los comentarios son para aclarar código y no son interpretados por Python, pueden omitirse cuando se escriben ejemplos.

```
1 #Ejemplo de tipos de datos y la declaracion de variables
2 #Este es un comentario, debe comenzar con el simbolo #
3
4 >>> paloma_boton_azul = True    #Variable tipo booleana
5 >>> numero_mascotas = 3         #Variable de tipo entero
6 >>> promedio = 7.56             #Variable de tipo flotante
7 >>> Saludo = "Hola extra o.. :)"#Variable de tipo cadena
8
9     #Variable de tipo arreglo
10 >>> calificaciones = [8.7, 9.7, 8.9, 8.1, 7.5, 6.8]
11
```

4. Estructuras de control condicionales

Una sentencia de control condicional es una estructura de programación que nos permite evaluar si una expresión es verdadera. Este tipo sentencias se emplean para responder una pregunta y tomar una desición en caso que esta sea verdadera o falsa.

4.1. if-else

La más simples de estas estructuras es la sentencia *if*. Esta sentencia evalua una condicion presentada.

```
if expresion_1 == expresion_2:
    hacerAlgo()
```

Para hablar de estructuras de control de flujo en Python, es imprescindible primero, hablar de identación. **¿Qué es la identación?** En un lenguaje informático, la identación es lo que la sangría al lenguaje humano escrito (a nivel formal).

Así como para el lenguaje formal, cuando uno redacta una carta, debe respetar ciertas sangrías, los lenguajes informáticos, requieren una identación. No todos los lenguajes de programación, pero en el caso de Python, la identación es obligatoria, ya que de ella, dependerá su estructura.

Una identación de 4 (cuatro) espacios en blanco, indicará que las instrucciones identadas, forman parte de una misma estructura de control.

```
if expresion_1 == expresion_2:
    leerVariable          #Esto pasar  si la comparacion
    compararVariable      #de varibles es verdadera
    imprimirMensaje
    multiplicarVariables

imprimirMensajeFinal      #Esto ya no forma parte de la
                          #estructura   if
```

4.2. if-else anidados

5. Ciclos condicionales

5.1. Ciclo while

5.2. ciclo for

6. Definición de funciones

7. Estructuras definidas

7.1. Listas ordenadas

7.2. Diccionarios

8. Librerías

8.1. numpy

8.2. matplotlib

8.3. scipy