**MANUAL DE PROGRAMADOR**

Huaman Ancco Yuver

Quintana Olarte Elizabeth

***Universidad Nacional José María Arguedas***

**Cubo de Rubik en Python**

**Introducción**

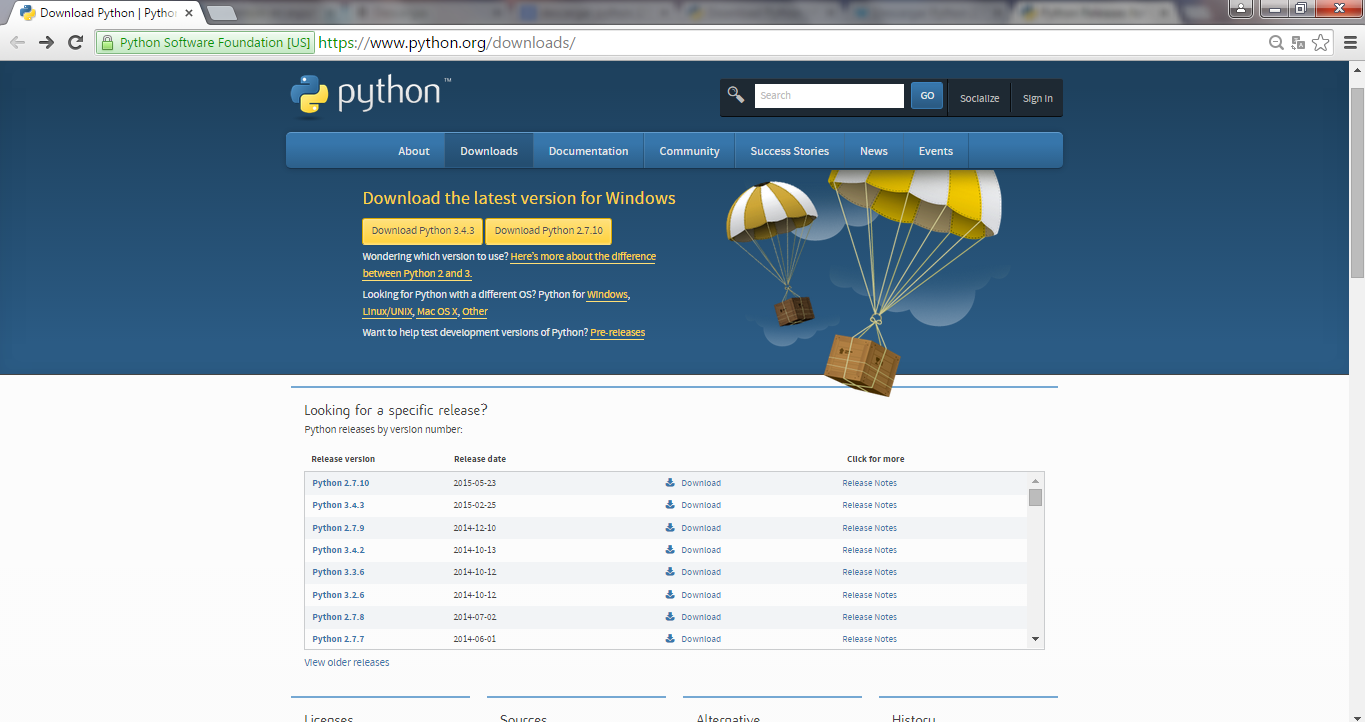
Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. Es administrado por la Python Software Posee una licencia de código abierto, denominada Python Software que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores

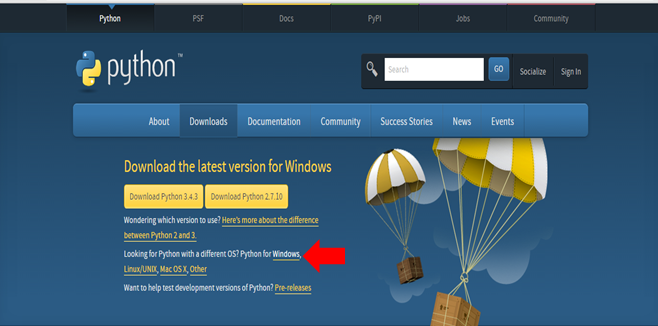
**Instalaciones de Python**

El traductor del lenguaje de programación Python es público y de acceso libre. Se puede descargar e instalar para WINDOWS y para otros sistemas operativos de 32 y 64 bits desde la página oficial de Python en la dirección de la red internet:

Primero descargaremos Python versión 2.7.9 desde la página principal <https://www.python.org/downloads/> la versión del Python que vayas a instalar es muy importante ya que la versión 3 adelante las librerías y sintaxis cambian

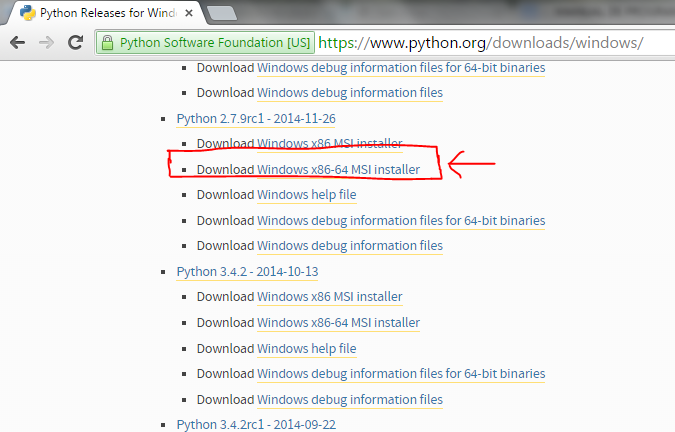
*Imagen 01 fuente www.python.org*

Tendrás la opción de descarga para distintos tipos de sistemas operativos como por ejemplo en Linux/UNIX, Mac OS X, en nuestro caso será en Windows de 64 bits es por eso que daremos clic en:



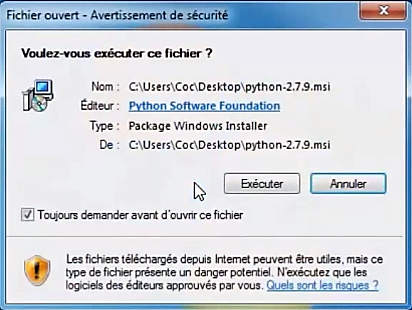
*Imagen 02 fuente www.python.org*

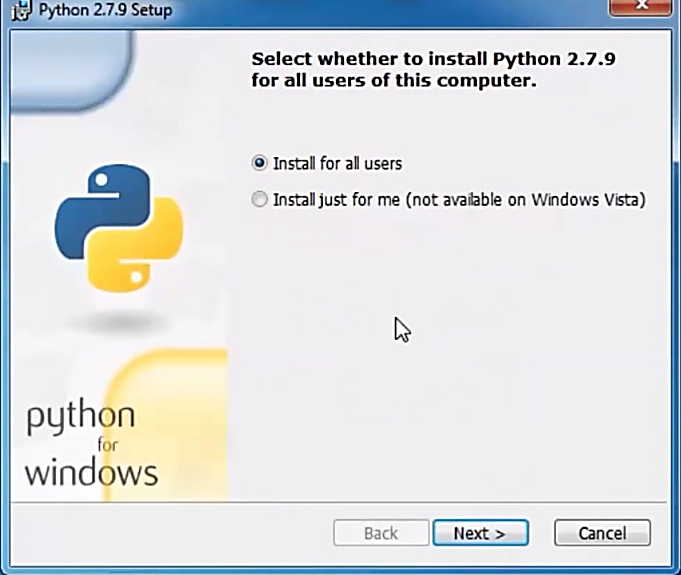
Nos re-direccionara a otra página donde tendremos que buscar Python 2.7.9

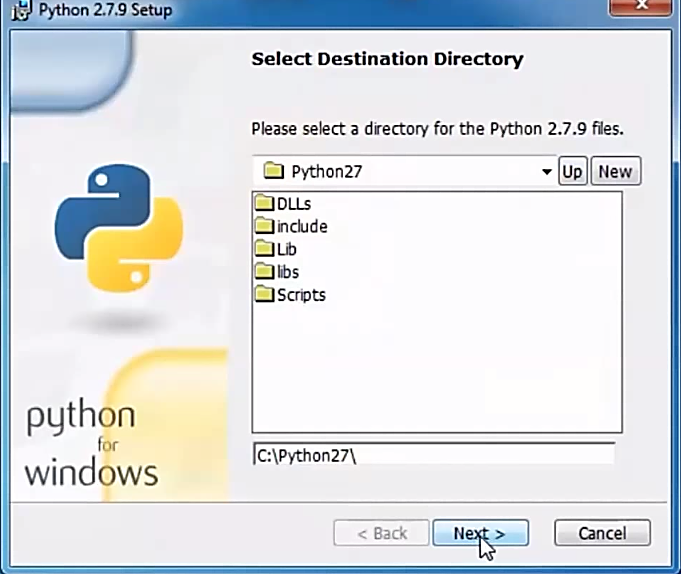
*Imagen 03 fuente www.python.org*

Una vez descargada empezamos con la instalación:

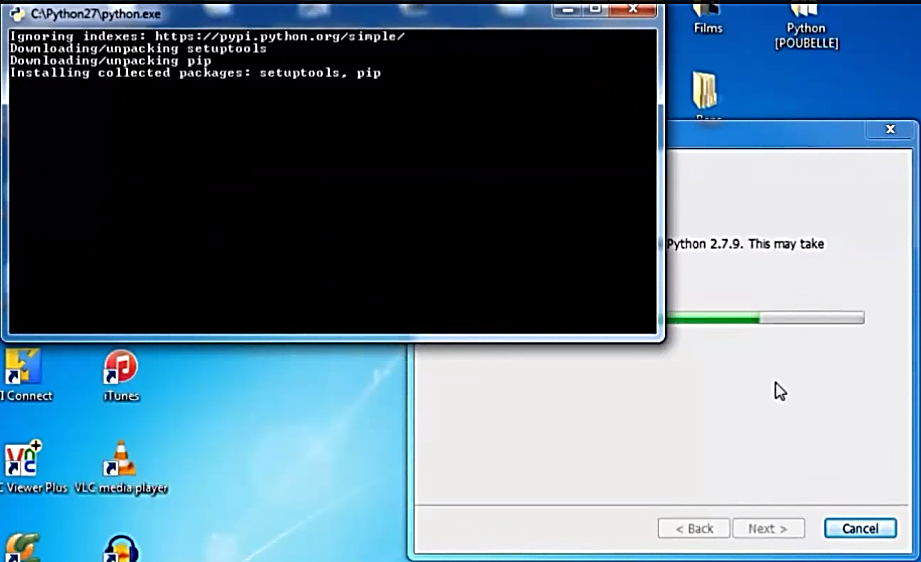








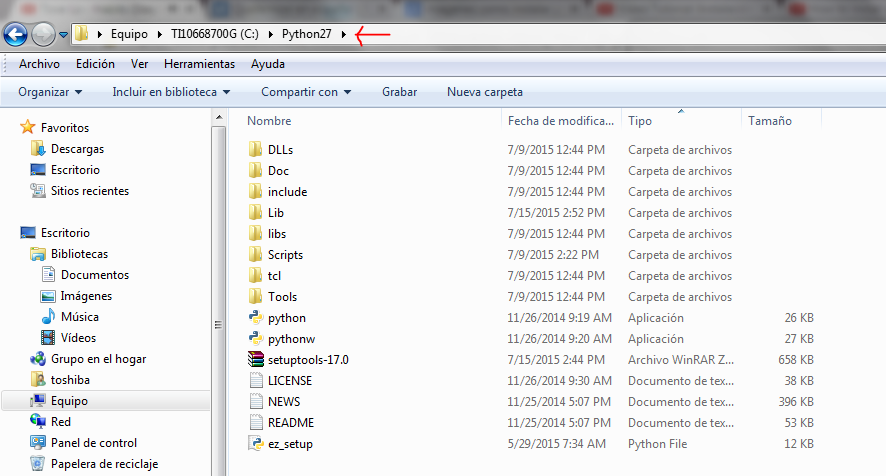




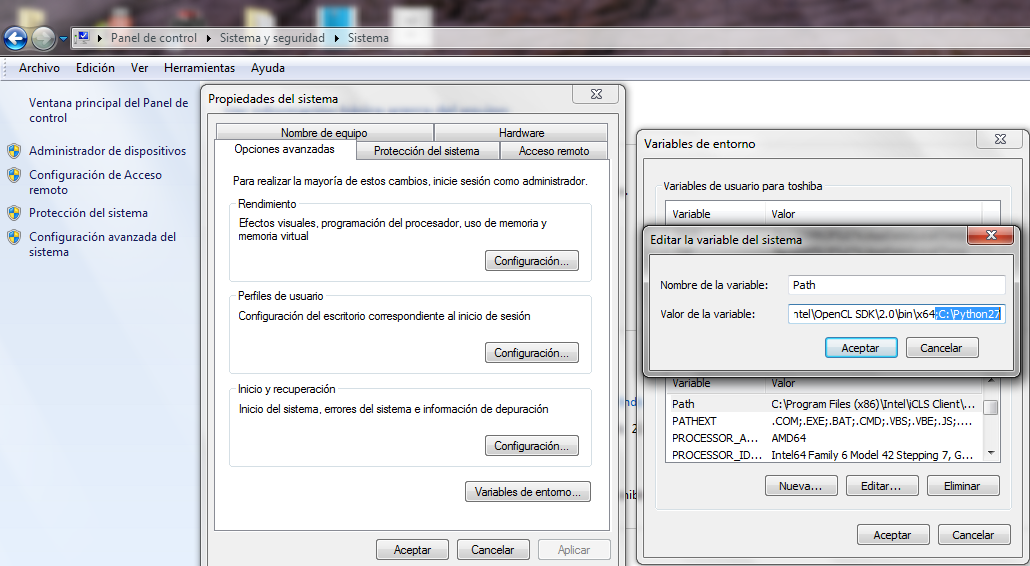




Una vez terminada la instalación agregamos la ruta de Python donde fue instalado C:\Python27 al pacht



Copiar al patch



Iniciamos Python y eso es todo, ahora pasaremos a instalar las librerías que utilizaremos para nuestro proyecto:

**Extensiones del lenguaje**

Para enriquecer el lenguaje, los usuarios de la comunidad Python han desarrollado muchas extensiones organizadas en paquetes o librerías las cuales están disponibles de manera gratuita en la red internet para integrarlas al entorno del lenguaje.

Un sitio de internet que contiene una gran cantidad de paquetes con extensiones para Python ha sido compilado por Christoph Gohlke de la Universidad de California en Irvine. De este sitio se pueden descargar e instalar para WIINDOWS, en la dirección:

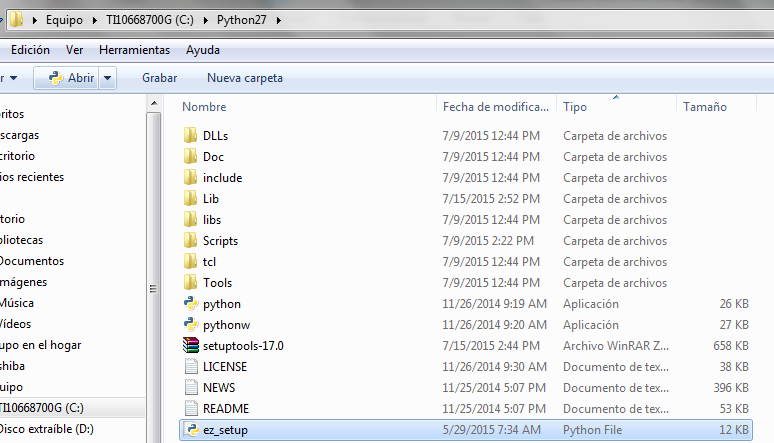
<http://www.lfd.uci.edu/~gohlke/Pythonlibs/>

Usaremos las librerías Numpy, SymPy,matplotlib.pyplot, matplotlib widgets describiremos la importancia de cada uno:

* **Numpy:** es una extensión de Python, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices. El ancestro de NumPy, Numeric, fue creado originalmente por Jim Hugunin con algunas contribuciones de otros desarrolladores.
* **Matplotlib.pyplot:** es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática NumPy.
* **Matplotlib widgets:** matplotlip posee dentro de sí la librería WIDGETS que nos permite crear controles de usuario básico controles de interfaz de usuario como por ejemplo botones ,checkbuttons, menus
* **SymPy:** Librería para aplicaciones con matemáticas simbólicas

Ahora para instalar cada una de las librerías abrimos el cd y ejecutamos el siguiente código:

1. copiar el ez\_setup.py a la ruta de python27(C:\python27)



Ejecutar en cmd para la instalación de las librerías importantes que nos servirá para el correcto funcionamiento de nuestro lenguaje Python hacer los siguientes pasos.

1. Cd...
2. Cd...
3. hasta llegar al disco local “C”
4. cd python27
5. python27 ez\_setup.py
6. cd scripts
7. Easy\_install pip
8. pip install numpy
9. pip install matplotlib
10. pip install click
11. pip install pyplot

* **EL ARCHIVO “Rubik.py”**

Una vez instalado las librerías necesarias definiremos las clases a usar:



Esta es la clase principal que define el punto de la matriz



Esta función define los colores por defecto vacío y al cubo le damos 2x2x2



Inicializando arrays para luego utilizarlo en la matriz como.

En asta parte del código lo estamos definiendo el giro de los todas la caras del cubo



Es la definición del tamaño de la ventana que se va ha mostrar en la pantalla de la computadora.

* **EL ARCHIVO “Juego.py”**



En la clase GRAFICO\_CUBO: creamos todos los giros que va hacer el cubo de 90º



Esta función te devuelve el Angulo que se va girar el cubo.



La función as *rotation\_matrix(self):* define la rotación de la matriz por los tres ejes con el Angulo que se está definiendo en la función anterior .

Aquí en esta parte estamos definiendo los puntos del grafico para en el eje x,y,z en un espacio para hacer giros con las teclas de dirección.

* **EL ARCHIVO “F\_Teclas.py”**

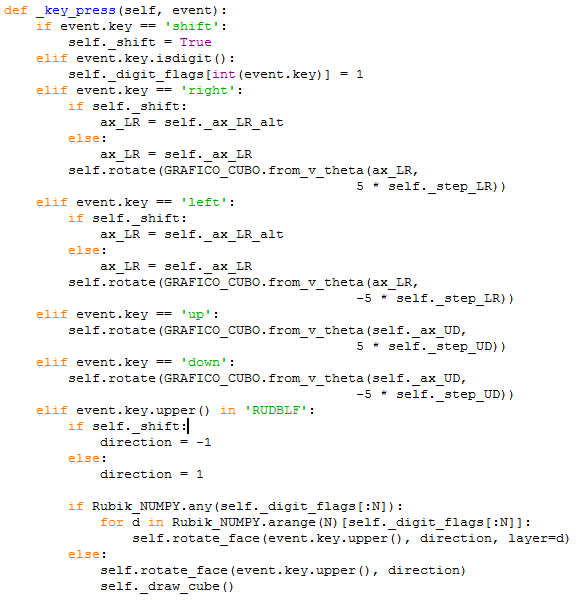


Esta clase lo estamos definiendo para las funciones de todas las teclas y hacer girar en los ejes x, y, z respectivamente,



Esta función lo que hace son las combinaciones de los colores del cubo





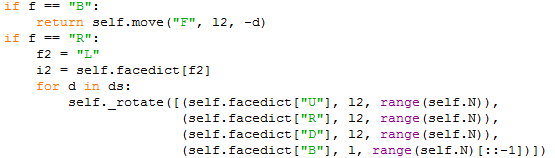
La función anterior es para los movimiento pero al contrario de los de la función *\_draw\_cube(self):* con las teclas shift .

* **EL ARCHIVO “Cubo.py”**

****

En esta parte le asignamos las teclas "U":0, "D":1, "B":2, "R":3,"L":4, "F":5 para las soluciones del cubo que va estar en desorden.



Esta función su principal labor es para armar los colores del cubo haciendo comparaciones.

**Referencias**

<https://github.com/jakevdp/MagicCube/blob/master/code/cube.py>

<http://stackoverflow.com/questions/11140163/python-matplotlib-plotting-a-3d-cube-a-sphere-and-a-vector>

<http://matplotlip.org/gallery.html>

http://github.com/bluquar/cubr