1. Introducción

Para esta actividad hemos decidido hacer un proyecto general que recopile todas las nuevas tecnologías que hemos investigado.

Hemos creado un conversor de monedas ( euro, dólar, bitcoin y libra esterlina ) que actualiza sus datos en tiempo real y usando una interfaz gráfica en Tkinter

1. División del proyecto

Durante esta actividad hemos tenido el siguiente reparto:

Juan José López Vega → Tkinter, desarrollo de la interfaz gráfica, instalación de dependencias, manejo de clases y funciones matemáticas

Alexandru Blaga → Manejo de ficheros, instalación de módulos externos, uso de clases en Python, uso de Selenium.

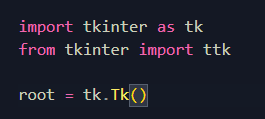
1. Tkinter, interfaces gráficas con Python 3
2. Ventanas en Tkinter:

Una ventana es un área visual rectangular, la cual contendrá todos nuestros iconos, botones, cuadros de entrada de texto, entre otros. Existen dos tipos: las ventanas de aplicación, que inician y finalizan las aplicaciones gráficas, y las ventanas de dialogo, que permiten la comunicación simple con el usuario, ambas formando la interfaz de usuario.

La ventana va a ser la unidad básica en la que nos vamos a basar para nuestro proyecto

1. Importación de módulos:

La importación y inicialización es realmente sencilla:



1. Propiedades de la ventana

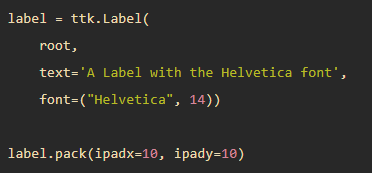
En Tkinter nada más declaremos nuestra ventana tenemos diferentes métodos para darle propiedades al que solo se lo tenemos que aplicar a nuestro objeto ventana

* title(): Cambiar el título de la ventana
* mainloop(): es el método principal sin la ventana no se mostrará ,no podemos olvidar ponerlo
* geometry() : configura el tamaño de la ventana
* iconbtimap(): cambia el icono, introduciendo como parámetro la ruta de la imagen relativa de la imagen
* resizable(boolean,boolean): nos permite redimensionar nuestra ventana. El primer parámetro es el eje x y el segundo y, si no queremos que se pueda redimensionar debemos scribir en ambos False.

1. Widgets

Son los diferentes elementos que irán alojados en nuestra ventana, algunos de los que hemos usado son : Label, Entry ,ComboBox y Button pero también hemos adjuntado ejemplos de RadioButtons y Text

* Label: Es una etiqueta de texto donde podremos mostrar información, similar a un <p> en html

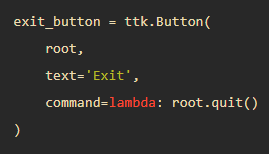


Para crear un Label nos tenemos que crear un objeto y paserle diferentes atributos

* + - * Contexto : donde lo valos a implentar, en este caso en la ventana root
      * Texto : el contenido que queremos incluir
      * Font : una tupla donde incluiremos la fuente y el tamaño

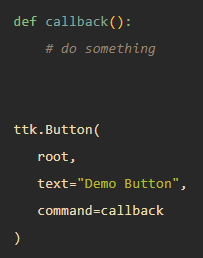
Por último tendremos que usar .pack() o .size() en el elemento para empaquetarlo y decirle las dimensiones que queremos para él , este paso se va a repetir en el resto de Widgets

* Button: Como su propio nombre indica es un elemento gráfico que va a recibir un evento de click y va a ejecutar una acción que le vamos a suministrar mediante una función.



En este sencillo ejemplo se ve como se declara un botón como en el label le pasamos el contexto y el texto que queremos añadir.

En particular tenemos el atributo “commnad” donde le tenemos que pasar la función que queremos que se ejecute cuando pulsamos el botón, en este caso usamos una lambda pero también podemos definirla en otro sitio y llamarla como se muestra ahora



\*\*\* No olvidar hacer .pack() a Button ya que si no no funcionará como se mencionó antes en el apartado de label \*\*\*

* Entry: Este Widget nos permitirá introducir texto y capturarlo para poder trabajar con el mediante código , similar a un <input> de html o un EditText en Android

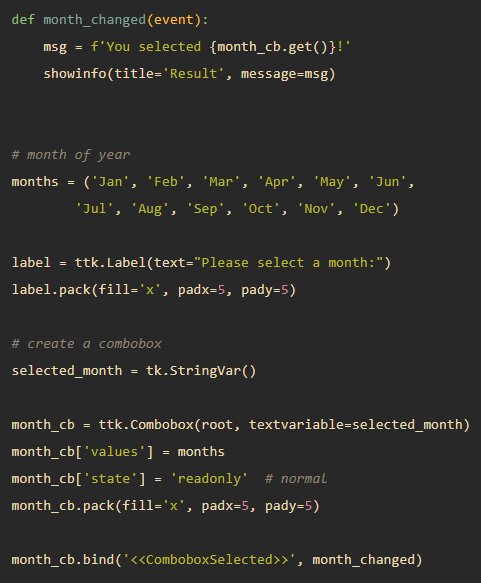
El primer paso que tenemos que hacer es declarar de manera especial las variables que van a contener información del Entry :



Como siguiente paso tenemos que crearnos nuestro Entry e asociar el valor que contiene al de nuestra variable



* ComboBox: Es el clásico desplegable que te permite elegir una opción entre varias
  + - * El primer paso es crear una colección con los valores que queremos incluir en el caso del ejemplo es una tupla
      * Nos creamos un Label (opcional ) para indicar al usuario que tiene que seleccionar uno
      * Creamos el combo box , una variable para guardar el dado como en Entry y las asociamos
      * Le pasamos nuestra tupla con los valores y lo empaquetamos
      * Además le hacemos un bind que es una propiedad que nos permite gestionar eventos, en el ejemplo le decimos que cuando sea seleccionado llame a la función month\_changed



4. Manejo de ficheros.  
  
 Tanto para la lectura como para la escritura he usado el módulo propio de Python llamado

OS para poder comprobar si el fichero al cual apunta nuestro programa existe o no.

A continuación pongo las capturas de lectura y escritura que explica el uso de este módulo y el de abrir y escribir ficheros.

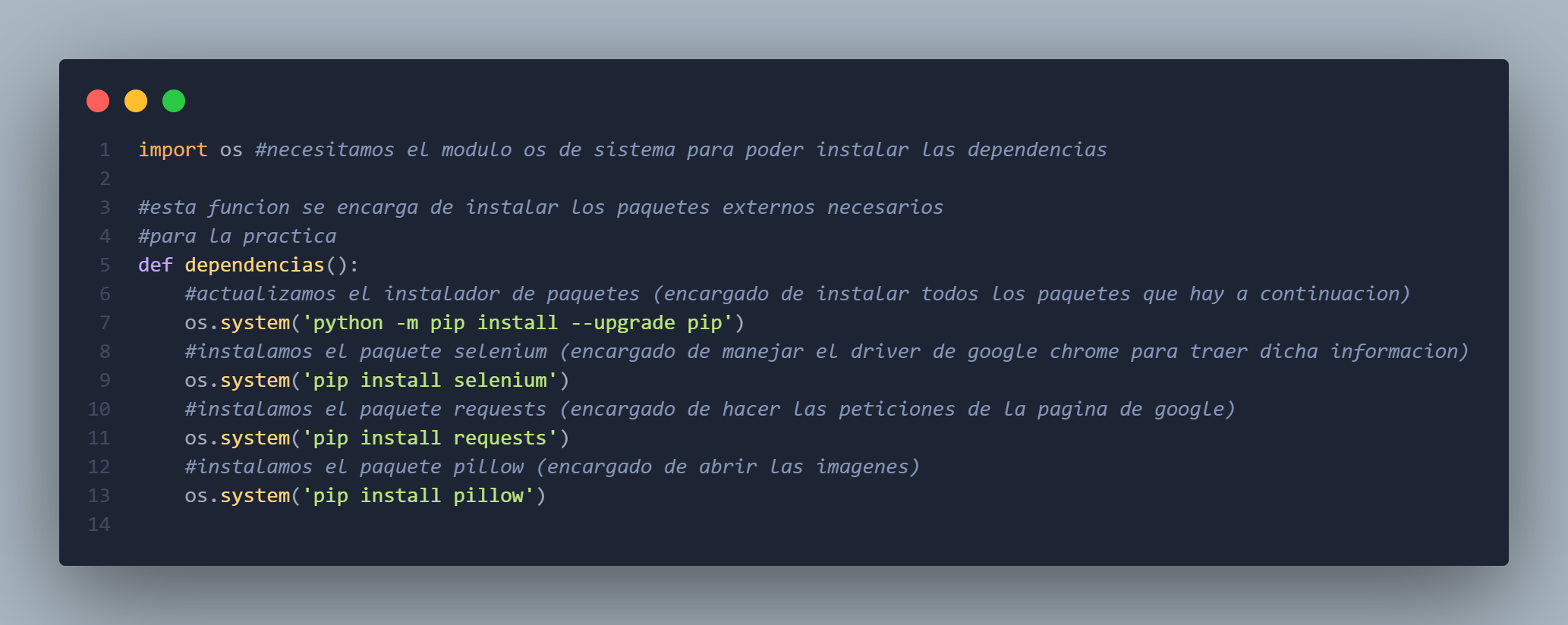




5. Instalación de dependencias (módulos externos).

En nuestra práctica hemos hecho uso de módulos externos que hemos tenido que instalar mediante el gestor de paquetes propio de Python (pip).

He desarrollado una función la cual se encarga de descargar todas las dependencias necesarias para que la aplicación funcione sin problemas además de que el primer paso que da es actualizar el gestor de paquetes para evitar problemas.



6. Clases en Python.

En nuestra práctica hemos tenido que utilizar una clase simple para poder mandar unos datos y solucionar un problema el cual pedía datos antes de abrir la interfaz gráfica.

A continuación he hecho un ejemplo básico de una clase en Python explicando cada parte de ella.



7. Configuración y uso de Selenium.



