

ampliación de conocimientos python 3

DA2D1E





JUAN JOSÉ LÓPEZ VEGA

ALEXANDRU BLAGA

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, instalación de dependencias, funciones matemáticas, ejemplos y documentación

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

Nuestro trabajo consta de un proyecto grande (conversor de divisas) y una recopilación de ejemplos sobre las tecnologías utilizadas

¡ Importante !

* Para probar el proyecto hay que acceder al archivo App.py y ejecutarlo. No se necesita hacer nada más, el resto es automático
* Es imprescindible tener acceso a internet ya que la información se saca de Google.com
* La aplicación tarda un poco en extraer los datos e instalar las dependencias si es la primera vez que se ejecuta, rogamos paciencia, la espera no será nunca mayor a 1 min

1. Tkinter, interfaces gráficas con Python 3

¿Qué es Tkinter?

* Tkinter es una librería de Python 3 que nos permite desarrollar programas con interfaces gráficas.

¿Qué aporta Tkinter?

* Permite crear aplicaciones de escritorio, con él damos un gran salto a la hora de programar, se acabó ejecutar en consola cada uno de nuestros programas
* Permite la interacción programa-usuario

Documentación Tkinter

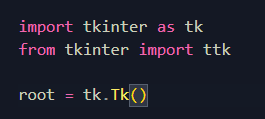
1. Ventanas en Tkinter:

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

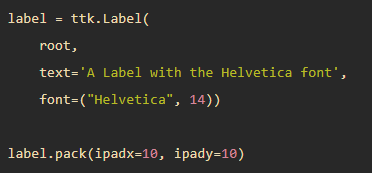
En Tkinter nada más declaremos nuestra ventana tenemos diferentes métodos para darle propiedades al que solo se lo tenemos que aplicar a nuestro objeto ventana

* 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

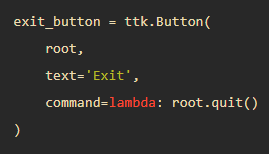


Para crear un Label nos tenemos que crear un objeto y paserle diferentes atributos

* + - * 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

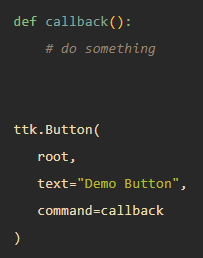
Por último tendremos que usar .pack() o .size() en el elemento para empaquetarlo y decirle las dimensiones que queremos para él , este paso se va a repetir en el resto de Widgets

* 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

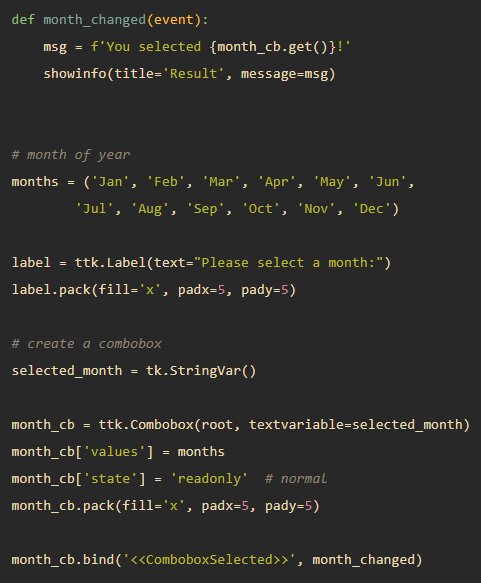
El primer paso que tenemos que hacer es declarar de manera especial las variables que van a contener información del Entry :



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



1. Manejo de ficheros con JSON

¿Qué es JSON?

JavaScript Object Model , es una notación en JavaScript ampliamente utilizada en el mundo web y en base de datos para transportar información usando objetos en JavaScript

¿Qué nos aporta?

Nos permite guardar información sobre estados, variables en ficheros creados con esta notación para luego usarlo en el momento que nos convenga

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.





1. Instalación de dependencias (módulos externos).

¿Qué son las dependencias?

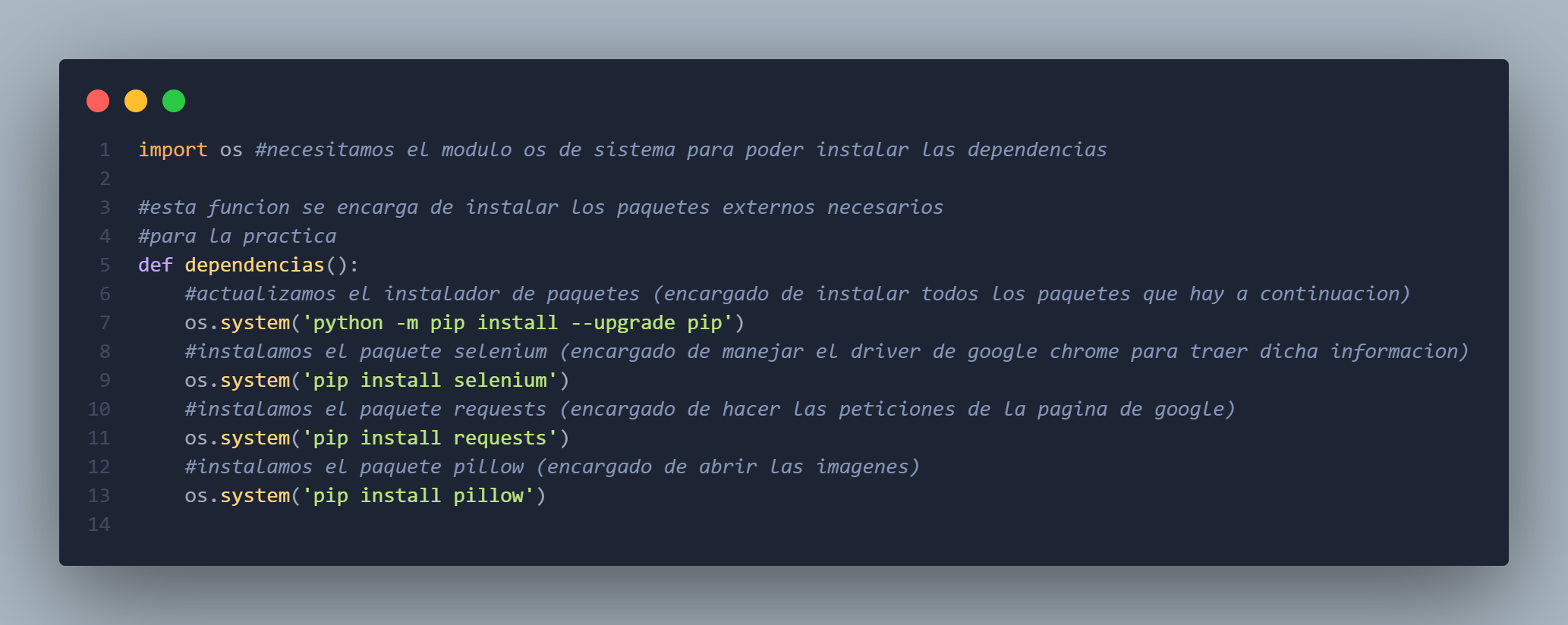
Son pequeñas automatizaciones de instalación de paquetes

¿Qué nos aportan?

Nos permite instalar paquetes al iniciar un programa sin la necesidad de utilizar la consola, nos ofrece sobre todo capacidad de portabilidad de un programa ya que el receptor no necesitará instalar nada

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

Hemos 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.



1. Clases en Python.

¿Qué son las clases?

Una clase es un tipo de dato definido por el usuario, y al crear instancias de una clase hace relación a la creación de objetos de ese tipo. Las clases y los objetos son considerados los principales bloques de desarrollo para Python, el cual es un lenguaje de programación orientado a objetos

¿Qué nos aportan?

Con esta implementación nos permite crear objetos con los atributos y funciones que hemos predefinido, es de extrema utilidad en cualquier lenguaje y en Python no iba a ser menos.

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.

¿Qué es Selenium?

Selenium es un proyecto que alberga un abanico de herramientas y librerías que permiten y apoyan la automatización de navegadores web

¿Qué nos aporta?

Selenium nos permite obtener datos a tiempo real de la web como puede ser el valor de las divisas en Google.com, para ello debemos configurarlo y capturar esa información como se muestra a continuación y en los ejemplos





8. Bibliografía utilizada

* <https://www.w3schools.com/python/>
* <https://docs.python.org/es/3/>
* <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/tutorial-de-selenium-webdriver/>
* <https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>