

Calculadora de Monedas, Prestamos e Inversiones

Sebastián Naranjo, David Porras, Raúl Ortega

Universidad Castro Carazo

MSc. Ronald Steven Arias Fallas

3 de mayo de 2022

Tabla de contenido

Resumen.....	4
Objetivo General	5
Bases Teóricas	5
Python	5
Visual Studio Code	5
Módulos	6
Librerías	6
Tkinter.....	7
Requests	7
Json	7
Funciones	7
Argumentos y Parámetros.....	8
Variables	8
Listas	8
Diccionarios	9
Métodos.....	9
Tasa de cambio	9
ISO 4217	9
Capital	10
Tasa de Interés	10

Calculadora de Monedas, Prestamos e Inversiones	3
Tasa Efectiva Mensual	10
Tasa Efectiva Anual	10
Análisis de la Interfaz Grafica	11
Análisis del Código.....	13
Conclusiones	17

Resumen

El programa es la respuesta a una asignación en forma de proyecto final, en donde se asignan requerimientos para dicho programa. El software está escrito en lenguaje Python y se usa como entorno de desarrollo el Visual Studio Code. Entre sus funciones esta la de conversiones de monedas, cuenta con 170 monedas mundiales, además puede realizar el cálculo de los pagos de un préstamo con interés mensual o anual y por ultimo una calculadora de inversiones para proyecciones mensuales, anuales y totales, todo desde una simple interfaz gráfica fácil e intuitiva donde se puede ver un historial de las operaciones hechas por el usuario, una pantalla de login para diferentes roles como administrador e invitados, y finalmente se puede cambiar el tema de los colores de la aplicación.

Objetivo General

Desarrollar y aplicar los conocimientos vistos en clase acerca del lenguaje Python y sus múltiples aplicaciones, así como en el entorno grafico Tkinter, aplicar widgets, funciones, módulos, y toda una serie de habilidades vistas en clase Durante el último cuatrimestre.

Bases Teóricas

En esta sección se verán superficialmente algunos conceptos aplicados en el proyecto final

Python¹

Python es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo, ejemplos: Instagram, Netflix, Panda 3D, entre otros.² Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma. Administrado por Python Software Foundation, posee una licencia de código abierto se clasifica constantemente como uno de los lenguajes de programación más populares.

Visual Studio Code²

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor,

¹ «Python Documentación by Versión». python.org. Consultado el 3 mayo del 2022.

² Microsoft (ed.). «Microsoft Software License Terms». code.visualstudio.com (en inglés). Consultado el 3 mayo del 2022.

los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

Módulos³

Un módulo le permite a usted organizar lógicamente su código Python. Agrupando código relacionado dentro de un módulo hace el código más fácil de entender y usar. Un módulo es un objeto de Python con atributos con nombres arbitrarios que puede enlazar y hacer referencia. Simplemente, un módulo es no es otra cosa sino un archivo con extensión .py. Un módulo puede definir funciones, clases y variables, también puede incluir código ejecutable.

Librerías⁴

Según David Zarruk, profesor del curso de analítica predictiva y modelos de regresión en Python, una librería de programación es un conjunto de funciones que alguien escribió en alguna parte del mundo y ha dispuesto para que cualquiera pueda utilizarlo de forma gratuita. Entonces, ¿qué es una librería en Python? Este concepto hace referencia al conjunto de implementos funcionales que te ayudarán a codificar todo este lenguaje de programación para crear una interfaz independiente. Las librerías de Python son amplias y cuentan con gran cantidad de producciones en contenidos. Constan de diversos módulos que permiten el acceso de funcionalidades específicas del sistema como entrada y salida de archivos, soluciones estandarizadas a problemas de programación, etc.

³ Módulos Python. Covantec R.L.. entrenamiento-python-basico.readthedocs.io. Consultado el 3 mayo del 2022.

⁴ +15 librerías de Python que te ayudarán a transformar el mundo digital sin complicarte. Andrea Flores. crehana.com. Consultado el 3 mayo del 2022.

Tkinter⁵

Tkinter es el paquete más utilizado para crear interfaces gráficas en Python. Es una capa orientada a objetos basada en Tcl (sencillo y versátil lenguaje de programación open-source) y Tk (la herramienta GUI estándar para Tcl).

Requests⁶

Requests es una librería Python que facilita enormemente el trabajo con peticiones HTTP. Antes o después, en algún proyecto, es posible que tengas que hacer peticiones web, ya sea para consumir un API, extraer información de una página o enviar el contenido de un formulario de manera automatizada

Json⁷

Esta librería principalmente parsea el JSON de archivos o strings. También parsea el JSON en un diccionario o en una lista en Python y viceversa, es decir, convierte los diccionarios y las listas de Python en cadenas JSON.

Funciones⁸

Una función te permite definir un bloque de código reutilizable que se puede ejecutar muchas veces dentro de tu programa. Las funciones te permiten crear soluciones más modulares y DRY para problemas complejos. Si bien Python ya proporciona muchas funciones integradas como `print()` y `len()`, también puedes definir tus propias funciones para usar en tus proyectos. Una de las grandes ventajas de usar funciones en tu código es que reduce el número total de líneas de código en tu proyecto.

⁵ Interfaces gráficas en Python con Tkinter. Gwydion Martín. adictosaltrabajo.com. Consultado el 3 mayo del 2022.

⁶ Python requests. La librería para hacer peticiones http en Python. Juan José Lozano Gómez. j2logo.com. Consultado el 3 mayo del 2022.

⁷ Cómo trabajar con datos JSON utilizando Python. A. Ahmed. programacion.net. Consultado el 3 mayo del 2022.

⁸ Guía de funciones de Python con ejemplos. Sebastián J. Bustamante. freecodecamp.org. Consultado el 3 mayo del 2022.

Argumentos y Parámetros⁹

En la definición de una función los valores que se reciben se denominan parámetros, pero durante la llamada los valores que se envían se denominan argumentos.

Variables¹⁰

En Programación, las variables están asociadas a variables concretos. Además, cada lenguaje de programación tiene su forma de implementar el concepto de variable, por lo que lo que se explica a continuación es válido para muchos lenguajes de programación, aunque otros lenguajes de programación permiten otras posibilidades. En muchos lenguajes de programación, una variable se puede entender como una especie de caja en la que se puede guardar un valor (por ejemplo, un valor numérico). Esa caja suele corresponder a una posición de memoria en la memoria del ordenador. Las variables se representan también mediante letras o palabras completas: x, y, a, b, nombre, apellidos, edad, etc.

Listas¹¹

En algunos lenguajes de programación se las conocen como arreglos o matrices; y se caracterizan porque los elementos están entre corchetes y separados por una coma. Una lista es una estructura de datos y un tipo de dato en python con características especiales. Lo especial de las listas en Python es que nos permiten almacenar cualquier tipo de valor como enteros, cadenas y hasta otras funciones.

⁹ Argumentos y parámetros. Héctor Costa Guzmán. docs.hektorprofe.net. Consultado el 3 mayo del 2022.

¹⁰ Introducción a la programación con Python. Bartolomé Sintet Marco. mclibre.org. Consultado el 3 mayo del 2022.

¹¹ Listas en Python. Jean Carlos Mariños Urquiaga. devcode.la. Consultado el 3 mayo del 2022.

Diccionarios¹²

Un Diccionario es una estructura de datos y un tipo de dato en Python con características especiales que nos permite almacenar cualquier tipo de valor como enteros, cadenas, listas e incluso otras funciones. Estos diccionarios nos permiten además identificar cada elemento por una clave (Key).

Métodos¹³

Un método es una función que «pertenece a» un objeto. En Python, el término método no está limitado a instancias de clase: otros tipos de objetos pueden tener métodos también. Por ejemplo, los objetos lista tienen métodos llamados append, insert, remove, sort, y así sucesivamente.

Tasa de cambio¹⁴

La tasa o tipo de cambio entre dos divisas es la tasa o relación de proporción que existe entre el valor de una y la otra. Dicha tasa es un indicador que expresa cuántas unidades de una divisa se necesitan para obtener una unidad de la otra. Por ejemplo, si la tasa de cambio entre el euro y el dólar estadounidense EUR/USD fuera de 1.12, ello significa que el euro equivale a 1.12 dólares. Del mismo modo, si la tasa se calcula a la inversa (USD/EUR), ello resulta en una tasa de 0.89, lo cual significa que el dólar equivale a 0.89 euros.

ISO 4217¹⁵

El ISO 4217 es un estándar internacional publicado por la ISO con el objetivo de definir códigos de tres letras para todas las divisas del mundo. Esto elimina las confusiones

¹² Diccionarios en Python. Carlos Eduardo Plasencia Prado. devcode.la. Consultado el 3 mayo del 2022.

¹³ 9.3.3. Objetos instancia. Python Software Foundation. docs.python.org. Consultado el 3 mayo del 2022.

¹⁴ «Le taux de change : définition et fonctionnement» [El tipo de cambio: definición y funcionamiento] (html). In Finance (en inglés). Consultado el 3 mayo del 2022.

¹⁵ «Currency codes -; ISO 4217;». Organización Internacional de Normalización (en inglés). Consultado el 3 mayo del 2022.

causadas por algunos nombres de divisas como dólar, franco, peso o libra, que son utilizados en numerosos países, pero tienen tipos de cambio muy diferentes. Las dos primeras letras del código son las dos letras del código del país de la divisa según el estándar ISO 3166-1 y la tercera es normalmente la inicial de la divisa en sí.

Capital¹⁶

El capital es el total de recursos físicos y financieros que posee una entidad económica, obtenidos mediante aportaciones de los socios o accionistas, para generar beneficios o ganancias. Esta cantidad siempre es distinguida de los intereses cobrados. Básicamente, se define como la cantidad de dinero que se presta o impone.

Tasa de Interés¹⁷

La tasa de interés es la cantidad de dinero que por lo regular representa un porcentaje del crédito o préstamo que se ha requerido y que el deudor deberá pagar a quien le presta. En términos simples; es el precio del uso del dinero.

Tasa Efectiva Mensual¹⁸

Es la tasa de interés expresada de forma efectiva mensual y que se aplica sobre una suma de dinero en función de un mes (30 días).

Tasa Efectiva Anual¹⁹

La tasa efectiva anual (TEA) es un indicador expresado como tanto por ciento anual que muestra el costo o rendimiento efectivo de un producto financiero. El cálculo de la TEA

¹⁶ Capital. Red Amigo DAL. konfio.mx. Consultado el 3 mayo del 2022.

¹⁷ ¿Qué es la tasa de interés y por qué se cobra?. Red Amigo DAL. konfio.mx. Consultado el 3 mayo del 2022.

¹⁸ FORMULAS Y EJEMPLOS EXPLICATIVOS. R. Pacheco. credivisionperu.com.pe. Consultado el 3 mayo del 2022.

¹⁹ Tasa de interés efectiva. Banco Central del Uruguay. usuariofinanciero.bcu.gub.uy. Consultado el 3 mayo del 2022.

está basado en el tipo de interés compuesto y parte del supuesto que los intereses obtenidos se vuelven a invertir a la misma tasa de interés.

Análisis de la Interfaz Grafica

En primera instancia tenemos la pantalla de login o roles, aquí se colocaron dos botones, uno para entrar a la aplicación como administrador o invitado, el rol de administrador permite la visualización y el acceso a algunos elementos que como invitado estarán ocultos.

El botón de administrador te permite entrar en una nueva ventana en la cual tenemos un botón para volver atrás, una entrada de texto en la cual se requiere la contraseña para entrar con privilegios a la aplicación, y un botón final para comprobar que la contraseña sea correcta, además de un espacio final en donde se mostrara un mensaje en caso de que la contraseña sea incorrecta.

Una vez se ingresa al sistema se mostrará la primera pestaña de la ventana principal, el conversor de moneda es una calculadora que convierte el valor o la cantidad de una moneda a los valores relativos o las cantidades de otras monedas. Lo primero que se visualiza en esta instancia es la clave de api, la cual se conforma por un texto como título “Clave de API”, una entrada de texto en la cual se debe ingresar el código de API querido para actualizar la base de datos, y un botón para verificar y realizar la actualización de las tasas de cambio, una fila mas abajo se encuentra un espacio para el mensaje de conexión, este mostrara si la actualización fue exitosa, fallida o existe un error de conexión con el servidor, seguidamente un botón de información para la nomenclaturas (ISO 4217) de las monedas disponibles en el programa la cual mostrara al usuario una ventana con la información disponible. En la parte de la conversión se seleccionan las monedas en dos cuadros desplegables en los cuales se puede seleccionar la moneda principal y la moneda destino, además de una zona central en la cual se puede visualizar la última fecha de actualización de

la tasa de cambio, además de una entrada de texto en la cual solo permitirá ingresar caracteres de tipo numérico y por ultimo un botón que dispara el cálculo de las variables seleccionadas por el usuario y anteriormente mencionadas mencionadas el cual se mostrara del lado inferior derecho en forma de respuesta al cálculo.

En la pestaña de préstamos se presenta la calculadora de préstamos, la cual permite calcular de manera sencilla la cuota mensual a pagar por el préstamo, así como el total de intereses. Así, este ayudará a realizar un comparativo de las diferentes opciones que otorgan las entidades financieras. En esta tenemos 3 entradas de texto, la primera para ingresar el capital esperado, la segunda para el periodo total del préstamo y en la ultima se ingresa el porcentaje de interés que la entidad bancaria espera por el préstamo, además como cuarta variable la selección del tipo de interés que este se impondrá al capital rentado, este puede ser efectivo mensual o efectivo anual, además de una línea de información que mostrara los cálculos realizados por el programa una vez se oprima el botón de calcular.

Como tercera opción y parecida a la anterior tenemos la calculadora de inversiones, donde se debe incluir el monto a invertir y el rendimiento estimado, por periodo de tiempo según corresponda a cada fondo, estos 3 datos ingresados en los 3 apartados para texto serán capturados por el algoritmo y una vez oprimido el botón calcular arrojará las ganancias proyectadas por mes, por año y las ganancias totales.

En la sesión historial tenemos una leyenda con el significado de todas las nomenclaturas propuestas, además de una tabla donde se guardan y visualizan las ultimas acciones hechas por el usuario.

Por último, en la pestaña de acerca de están los créditos respectivos, la información sobre la API y las respectivas claves accesibles solo al administrador, también como opción interesante 2 botones que pueden cambiar el tema de la aplicación.

Análisis del Código

En las primeras líneas tenemos varias librerías que utilizamos durante todo el código. la librería Tkinter Se utiliza para el entorno gráfico de la aplicación, la librería **ttk** es utilizada para otras funciones gráficas, la librería **requests** se utiliza para la comunicación con el servidor web, la librería **json** es importante para guardar y extraer datos de un archivo con el mismo nombre, además tenemos la librería **os** que borra las líneas que produce la terminal al ejecutarse la aplicación. como primera variable tenemos **roles** la cual es de tipo booleano y se utiliza para verificar los privilegios del usuario en cuestión, seguidamente la variable **count** que es de tipo lista **int** la cual es un contador para diferentes iteraciones que tenemos en el código, también tenemos la variable **theme** que cambiará el tema de la aplicación.

El código se empieza a ejecutar en la última línea donde llamamos a la función **rol ()** la cual se encarga de abrir una ventana llamada login del tipo **tk** y ésta va a ser global, en ésta se encuentra los 2 botones necesarios para seleccionar el tipo de usuario, dependiendo del usuario que se escoja, si se presiona el botón **admin** éste dispara la función **credentials()** y si por el contrario se presiona el botón de invitado este dispara la función de invitado().

La función **credentials()** Es una ventana aparte y se utiliza para verificar la contraseña y los privilegios de administrador, una vez introducida la contraseña correcta esta cambiará la variable **roles** a un valor verdadero, de lo contrario siempre será falso.

Además, hay una función llamada **atrás ()** la cual destruye la ventana anterior y vuelve a abrir la función **rol ()**.

Una vez que se prime el botón de comprobar en la ventana de credenciales está disparara la función **comprobar ()** que se encarga de comprobar si la contraseña introducida es igual a la que está guardada en el sistema, si esta es correcta el usuario podrá visualizar algunos componentes ocultos en el programa, De lo contrario se mostrará un mensaje con el texto contraseña incorrecta. Para ello se utiliza una condicional **if** y **else**.

La función **invitado** () no compara ninguna condición sino simplemente destruye la ventana login y dispara la función **main** ().

La función **main** () contiene a nuestra ventana principal y todos sus widgets y componentes, para ésta se utiliza la librería técnica y su clase notebook la cual nos permite crear pestañas en la parte superior de la ventana, además algunas pestañas están dentro de una condicional que se mostrará solo y únicamente cuando la contraseña del administrador es ingresada correctamente.

La primera pestaña de nuestra ventana principal es la de convertidor de monedas, ésta contiene varias variables de tipo global ya que serán utilizadas en otras funciones, además aquí se abre el archivo local Jason que contiene los datos de las tasas de cambio de las monedas y éstas se guardan en una variable llamada local.

En **fechaDeActualizacion** se guarda la fecha de actualización de la base de datos, en **dicAbrebiaturas** se extraen todas las tasas de cambio de todas las monedas disponibles y en **listAbrebiauras** se guardan las abreviaturas de las monedas para que estas puedan ser utilizadas próximamente en los **Combobox**, para ello se utiliza un ciclo **for** en el cual por cada ciclo se agrega una abreviatura a la lista anterior.

Además, tenemos un widget de tipo **Entry** dónde podremos ingresar la clave de API necesaria para la actualización de la base de datos y un botón que dispara la función **actualizar** (). Esta función captura los valores ingresados en el **Entry** y los guarda en una variable llamada API, seguidamente tenemos un manejo de excepciones en dónde en el Try se encuentra la variable **res_a** la cual mediante la librería **request** envía una petición a la página web dando como parámetro la clave de API anteriormente captura y ésta devuelve una respuesta, esta respuesta es convertida **json** y guardada en la variable **dato_a**, además, dentro de los datos capturados existe uno llamado **sucess**, el cual permite verificar si la conexión con el servidor fue correcta o fallida, esta se guarda en una variable llamada prueba y se utilizará

como condición para qué el algoritmo decida si guardar los datos obtenidos en la respuesta en el archivo **Json** local o utilizar los que ya existen, además existen 3 acciones disponibles, una de ellas es que la comunicación con el servidor sea correcta y los archivos sean actualizados correctamente, la segunda manera es que la comunicación con el servidor sea errónea por problemas con el servidor o la petición, y la tercera y última acción es la que la conexiones no haya permitido la comunicación con el servidor y por lo tanto está falla, para cualquiera de las 3 se mostrará un mensaje dependiendo de cuál de ellas sucedió.

Regresando a nuestra pantalla principal tenemos un botón que dispara la función **infoDeMonedas ()**, esta se encarga simplemente de abrir una ventana donde tendremos toda la información de las abreviaturas y los países disponibles en nuestra base de datos.

Seguidamente en nuestra pantalla principal tenemos 2 **Combobox** en los cuales se puede escoger el tipo de moneda base y el tipo de moneda de destino, también tenemos una entrada de texto en forma de **Entry** en la cual sólo se podrán ingresar caracteres de tipo numérico (para esto se utiliza la función **validate_entry ()**), además un botón que dispara la función **calculoDeMoneda ()**.

Esta función tiene varias utilidades entre ellas captura los valores que se han asignado a los **Combobox** y la cantidad que se ha digitado en el **Entry**, seguidamente convierte esta cantidad en un valor flotante, Se busca en el diccionario **local** las tasas de cambio en base al índice que dan los valores de los **Combobox**, En síntesis tendremos 2 variables con 2 índices diferentes (**rate_a** y **rate_b**) con los cuales se hará una operación matemática dividiendo el primer valor entre el segundo y a todo esto se le aplicará un round para que tenga solo cuatro decimales, esta cantidad se guarda en una variable llamada **rate**, seguidamente ese valor es mostrado en pantalla y guardados en un archivo de texto para posteriormente ser mostrado en el historial como una operación más, además esta operación se agrega a la tabla de historial con una función llamada **insert**.

Ahora continuamos con nuestra pestaña para calcular préstamos, es pestaña cuenta con 3 **Labels**, el primero capital, el segundo periodo y el tercero la tasa de interés y estos a su vez cuentan con 3 **Entry** en los cuales se asignará un valor numérico el cual se verifica mediante la función **validate_entry** (), además cuentan con 2 **Radiobutton** para seleccionar el tipo de tasa de interés preferido, a la hora de que el usuario digita los 3 valores numéricos y presiona el botón de calcular se dispara la función **calculoDePrestamo** ()

Esta función lo primero que hace es capturar los valores de los widgets antes mencionados y guardarlos en variables independientes, Los cuales mediante una condicional (que será el tipo de interés que el usuario a escogido) procederá a realizar el cálculo mediante operaciones matemáticas, el cálculo con un tipo de interés o con otro son diferentes por tanto esta condicional es muy importante para una operación correcta, sin embargo ambas comparten el guardar la operación en el archivo **historial.txt**, agregarlo a la tabla de historiales y por último mostrar el mensaje de la operación deseada mediante un **Label** en la interfaz gráfica, Si alguno de los valores está vacío se mostrará un aviso que dice datos faltantes y ninguna de las operaciones anteriores se efectuará.

Se debe mencionar que la variable **count** es muy importante en este código ya que va aumentando conforme los valores se van agregando a la tabla de historiales, así la tabla sabrá en qué posición debe poner la última operación

La calculadora de inversión es nuestra tercera pestaña y por tanto nuestra última funcionalidad productiva. en ésta al igual que en la anterior ventana se muestran 3 entradas de texto en forma de **Entry**, los cuales el usuario debe llenar con caracteres numéricos para que la siguiente función pueda capturarlos y resolver la operación.

La función **calculodeinversion** () captura todos los datos ingresados por el usuario en la pestaña anterior, los guarda en variables independientes, y hacer los cálculos correspondientes para poder mostrarlos en forma de texto en la interfaz gráfica, además se

procede al guardar la operación en el archivo **historial.txt** y a agregarla a la última línea de la tabla de historiales, si alguno de los datos anteriores estuviese vacío se procede a dar un aviso en forma de texto indicando que hay datos faltantes.

Como penúltima ventana tenemos el historial el cual se conforma de un **Frame** el cual contiene texto indicando o explicando la leyenda o las nomenclaturas usadas para las operaciones, además de una tabla hecha con la clase **Treview** de **Ttk**, esta contiene una columna única en la que se muestran las últimas operaciones realizadas por el usuario, la tabla se actualiza con base en el **historial.txt** Con la utilización de una iteración en la que utilizando el contador se agrega el valor del primer índice del historial hasta el último, Así pues cada vez que alguna de las funciones agrega una operación nueva a la tabla esta se agrega a la última línea y la tabla puede mostrar todos los valores correctamente.

Por último, tenemos la ventana “**acerca de**” en la que se muestran los créditos y los agradecimientos respectivos por el proyecto, Además de una pequeña funcionalidad añadida como parte de la curiosidad del equipo por profundizar en el tema de la estética, éste contiene 2 botones que simplemente disparan una función, esta función puede ser **light** () o **dark** () respectivamente, cualquiera de las 2 funciones hacen exactamente lo mismo, primero, cambia el valor de la variable **theme**, destruye la ventana actual y la vuelve a abrir para sí poder hacer visible el cambio de tema.

Conclusiones

Para finalizar en el proyecto se incluyeron muchas funcionalidades o características experimentales que al principio no se sabía a ciencia cierta cómo iban a funcionar, pero conforme pasó el tiempo y el código se fue desarrollando entre todo el equipo se fueron aprendiendo cosas nuevas, nuevos conocimientos, Nuevas formas de trabajo y de implementación del código.

Ciertamente para todos los miembros del equipo fue una experiencia muy satisfactoria haber concluido con un proyecto de estos requerimientos ya que es nuestro primer proyecto universitario y gracias a las enseñanzas del profesor, las experiencias compartidas de otros compañeros o amigos este proyecto fue concluido exitosamente

así concluye este informe no sin antes proponer una continuación de la aplicación que se hizo para futuras mejoras y una ampliación de nuestra experiencia en el lenguaje Python.