

Python en la vida cotidiana

Santiago Fernández

Introducción

El siguiente proyecto busca desarrollar algunas de las funciones que ofrece Python que se pueden aplicar a la vida cotidiana de cualquier persona que sepa usar este lenguaje de programación. Para esto, se explicarán los 3 ejercicios implementados y los principales resultados obtenidos, es necesario mencionar que, debido a la extensión que el documento solicita, no se pondrán todas las imágenes obtenidas, estos serán desarrollados en la presentación con todo detalle, mientras que los gráficos se encuentran en el código del GitHub, disponible al final de este documento.

Junto a todo lo anterior, hay que tener en cuenta que todas las referencias utilizadas se encuentran en el código de programación, esto debido a que su gran cantidad harían de este documento muy extenso, además, de esta forma se puede ver la referencia junto a la parte en la que fue utilizada esta. Con todo esto, se procede con la explicación de los ejercicios.

Explicación de los ejercicios

Envío de mensajes de WhatsApp

Para implementar este ejercicio se hizo uso de la librería *pywhatkit*, esta librería permite enviar mensajes de WhatsApp desde Python. Para usar esta librería se construyó un módulo de Python, el cual contiene la clase “MensajesWhatsapp”, la cual tiene 2 módulos, uno para enviar mensajes de manera instantánea y otro para dejar programado un mensaje en las próximas 24 horas.

El funcionamiento de esta librería es realmente simple, una vez que se ejecuta el código con alguno de los 2 métodos se verá un mensaje con el tiempo para que el mensaje se mande (en caso de que no sea de forma instantánea), cuando llegue el momento de enviar el mensaje se abrirá WhatsApp Web en el navegador predeterminado de la persona, se buscará el número correspondiente y se enviará el mensaje escrito desde Python.

Para que el proceso funcione correctamente es necesario poner el código del país del número al que se enviará el mensaje, su número y tener WhatsApp Web cargado (no necesariamente abierto) en la computadora, por ejemplo, con el módulo y la clase ya cargados:

```
MensajesWhatsapp('+50688888888')
```

Lo anterior guardará el número receptor, que en este caso es un número de Costa Rica, compuesto por 8888-8888, otros caracteres especiales fuera del '+' inicial pueden presentar problemas en la ejecución del código.

La principal utilidad que se encontró a esta clase fue la posibilidad de dejar mensajes programados para una persona en concreto, lo cual puede ser realmente útil cuando se tiene una agenda atareada y se desea organizar múltiples mensajes a una hora en concreto.

Podrá ver que el código de GitHub presentará un error al ejecutar esta parte, esto se debe a que no se dejó ningún número de celular escrito para mantener la privacidad de las personas receptoras, sin embargo, con tal de agregar un número de celular esta parte funcionará.

Automatización de procesos en Excel

Este ejercicio se realizó con la librería de *xlsxwriter*, el principal objetivo de este ejercicio es automatizar los distintos procesos que se pueden presentar en Excel, desde Python. Para la ejecución de este ejercicio se creó un módulo que contiene a la clase EditorExcel, el cual presenta 10 métodos con funciones comunes que una persona puede realizar en Excel.

La construcción de un objeto de la clase EditorExcel consiste en un DataFrame creado con Pandas y el nombre que tendrá el archivo de Excel, sin espacios, y finalizado con .xlsx, omitir alguno de estos pasos presentará errores al crear el objeto.

Una vez creado el objeto se creará automáticamente un Excel, este podrá ser editado con los métodos implementados en la clase y con las funciones que ofrece la librería *xlsxwriter*. Es importante mencionar que el archivo, mientras se esté editando, no podrá abrirse de forma convencional, para poder abrir el archivo hace falta cerrar el editor del archivo, lo cual es posible con el método cerrar() de la clase, luego, este archivo no podrá ser editado nuevamente, por lo

que para hacer más cambios desde Python se requiere eliminar el archivo de Excel y ejecutar nuevamente el código con los cambios a implementar.

Para mostrar el funcionamiento de esta clase se realizó un ejercicio en el cual se creó un DataFrame ficticio que simula algunos empleados de una empresa, este se puso en un objeto de EditorExcel y se le hicieron los siguientes cambios desde Python: se agregaron 3 observaciones más, se marcó en rojo las casillas de la columna horas_extra que son mayores a 0 con formato condicional, se pusieron fórmulas para calcular un bonus al salario según las horas extra, para calcular el salario total y el salario promedio, se hizo un gráfico pequeño (sparkline) para ver las horas extra de manera más sencilla, se protegió el archivo con una contraseña y se autoajustaron las columnas que tenían al menos una casilla con texto escrito.

El documento de Excel con los cambios enunciados anteriormente se puede obtener al ejecutar el código correspondiente del código, además, se le pueden hacer múltiples modificaciones con los métodos disponibles.

La utilidad de este ejercicio es para la presentación de documentos de Excel, lo cual puede requerirse comúnmente en cualquier trabajo en el cual se deba trabajar con bases de datos, además, se puede utilizar para uso personal en múltiples situaciones, por ejemplo, para manejar los gastos de un paseo en familia.

Web scraping

Para el proceso de *web scraping* se utilizó, en primer lugar, la librería *request* para mandar una solicitud a la página web para la obtención de su código HTML, luego, se usó la librería BeautifulSoup para leer y analizar el código HTML.

La construcción de este ejercicio se hizo con base en múltiples funciones que sirven para extraer algún elemento del HTML, en concreto, se hizo *web scraping* para comparar 2 páginas de automóviles, crautos y Mercado libre (en este último, la sección de automóviles usados, para comparar apropiadamente), con estas páginas, se extrajo la marca del carro, el modelo, el año en el salió su modelo, su precio y en el caso de crautos se obtuvo la transmisión, lo cual no se extrajo de Mercado libre pues no está disponible a simple vista como en mercado libre.

Junto a lo anterior, se realizó un DataFrame con las observaciones obtenidas, en el caso de Mercado libre, las primeras 3 páginas de la sección de automóviles usados, en el caso de crautos, los vehículos de la página principal de los días domingo 9 y lunes 10 de junio del 2024, luego, para asegurar la reproducibilidad del ejercicio, este DataFrame fue guardado en un archivo de Excel (disponible en el GitHub del proyecto), para luego generar resultados con respecto a estos.

Hay que mencionar que el proceso de *web scraping* es usualmente lento, en este caso, que se usaron 2 páginas web distintas y varias observaciones, el código tarda alrededor de 20-25 minutos en su ejecución, por lo que se adjunta el HTML (en el GitHub) como prueba de que efectivamente funciona.

Resultados

De la parte de WhatsApp se extrae la conclusión de que se pueden enviar mensajes de forma sencilla desde Python, aunque las restricciones de 24 podrían limitar su uso. No se adjuntan imágenes ya que simplemente se envía un mensaje de WhatsApp común.

Pasando a la parte de Excel, se tiene el documento inicial sin ningún cambio:

id	Empleado	Departamento	Salario_base	Horas extra	Bonus	Salario_total
1	Pedro	RRHH	750000	0	0	750000
2	Luis	RRHH	900000	0	0	900000
3	Carlos	Contaduría	1000000	5	0	1000000
4	Elena	Actuariado	1050000	0	0	1050000
5	Francisco	Actuariado	1000000	2	0	1000000
6	David	TI	900000	5	0	900000
7	Marta	RRHH	800000	0	0	800000
8	Ana	TI	500000	2	0	500000
9	Isabel	Proyectos	850000	12	0	850000
10	Verónica	Secretaría	650000	8	0	650000

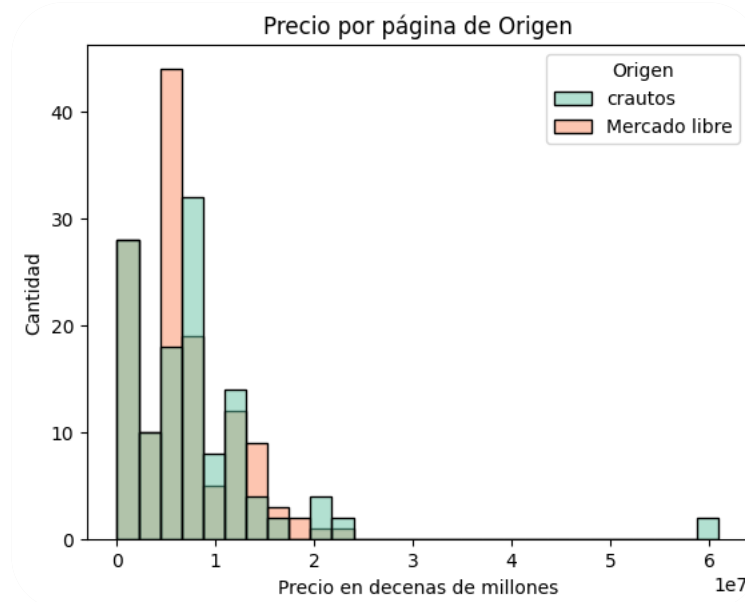
Luego, después de los cambios el Excel queda de la siguiente forma:

Id	Empleado	Departamento	Salario_base	Horas_extra	Bonus	Salario_total
1	Pedro	RRHH	750000	0	0	750000
2	Luis	RRHH	900000	0	0	900000
3	Carlos	Contaduría	1000000	5	0.05	1050000
4	Elena	Actuariado	1050000	0	0	1050000
5	Francisco	Actuariado	1000000	2	0.02	1020000
6	David	TI	900000	5	0.05	945000
7	Marta	RRHH	800000	0	0	800000
8	Ana	TI	850000	2	0.02	867000
9	Isabel	Proyectos	850000	12	0.12	952000
10	Verónica	Secretaría	650000	8	0.08	702000
11	José	Secretaría	600000	0	0	600000
12	Pablo	TI	800000	0	0	800000
13	Tatiana	Contaduría	900000	0	0	900000
						872000


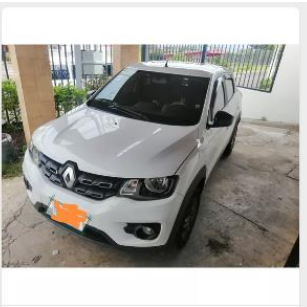

Con esto se puede decir que se logró automatizar múltiples cambios de forma satisfactoria, logrando un cambio significativo en el ejemplo, lo cual permite afirmar que los métodos implementados en la clase correspondiente permiten una gran flexibilidad de edición y una sintaxis simple.

Como una observación, todas las fórmulas del método formula, del módulo EditorExcel, deben escribirse en inglés, independientemente del idioma de la computadora o el Excel del usuario, esto debido a que la librería está implementada de esta manera.

Finalmente, del ejercicio de *web scraping* se extrae la conclusión de que, según las observaciones recolectadas, Mercado libre posee precios más bajos en general, esto se observa en el siguiente gráfico:



Sin embargo, haciendo un análisis más profundo se puede observar que, a pesar de estar en la sección de automóviles usados, Mercado libre no es capaz de filtrar correctamente algunos de los objetos en venta ni precios realistas, como se puede apreciar en la siguiente imagen:

		
Vw Amazonas 1984 Brasileña ₡ 500,000 1999 350 000 Km Central - Heredia	Renault 2021 Kwid ₡ 5,900,000 2021 123 000 Km Moravia - San José	Isuzu Dmax 2023 Dmax ₡ 26,000,000 2023 55 800 Km Naranjo - Alajuela

Por lo que es cierto que hay precios más bajos, pero con la información disponible no es posible hacer una comparación directa entre vehículos motorizados. En la presentación se agrega un análisis más exhaustivo de esta parte.

Recomendaciones

En la parte de mensajes de WhatsApp, se recomienda investigar más la librería, es posible enviar imágenes y mandar mensajes a grupos de WhatsApp con comandos simples, además, se podrían implementar métodos para enviar mensajes más de 24 horas después.

Posteriormente, en la parte de la automatización de Excel es similar, la librería ofrece una gran variedad de opciones que no fueron implementadas, desde la creación y personalización de gráficos para insertar en el documento final, inserción de imágenes, dejar un filtro en concreto dentro del documento e incluso la posibilidad de insertar botones y crear macros para trabajar con VBA.

Pasando a la parte del *web scraping*, es posible que la eficiencia del código sea uno de los principales aspectos a mejorar, asimismo, la comparación entre páginas exclusivamente de automóviles podría facilitar el llegar a una conclusión más clara. También hay que añadir que agregar una mayor cantidad de observaciones podría ayudar al análisis final.

Anexos

Código del GitHub:

https://github.com/Santifer0803/Proyecto_individual_herramientas_II