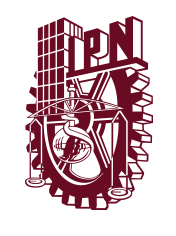
**Marco Teórico**



Instituto Politécnico Nacional

Escuela Superior de Cómputo

Teoría Computacional

Practica 1:Ejercicios de Cadenas en Python

Alumna: Ramírez Galindo Karina

Profesor:Rosas Trigueros Jorge Luis

Fecha de realización:26-08-2019

Fecha de entrega: 2-09-2019

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Python es un lenguaje de programación versátil, fácil de aprender, interesante y bastante útil. Su origen se remonta a finales de 1980 cuando Guido Van Rossum lo ideó. [1]

Este lenguaje multiparadigma que lleva por nombre Python ofrece herramientas para trabajar desde la programación orientada a objetos, así como de la programación funcional a partir de list y programación imperativa, se pueden programar distintos tipos de aplicaciones: scripts, aplicaciones web y aplicaciones de escritorio. Posee cientos de bibliotecas que hacen que cualquier tipo de proyecto sea posible, ya sea una aplicación móvil, aplicación web, ciencia de datos o inteligencia artificial. Por ejemplo, ‘Numpy’ para computación científica, ‘Pybrain’ para aprendizaje automático, ‘Scipy’ para computación avanzada y ‘AIMA’ para inteligencia artificial. [2]

Python es de tipado dinámico esto se debe a la capacidad que posee Python en no requerir la definición del tipo de datos en las variables, el lenguaje auto-asigna el tipo de datos según el valor declarado. [1]

Además este es un lenguaje de programación de alto nivel que fue diseñado con una sintaxis muy limpia que permitiese obtener códigos que fuesen fáciles de leer, es multiplataforma y soporta orientación a objetos, programación imperativa e, incluso, programación funcional. [3]

Características: [2]

* Un programa en python puede ser descompuesto en módulos, sentencias, expresiones y objetos.
* En Python todo está representado mediante objetos o relaciones entre objetos
* Cada objeto tiene una identidad, un tipo y un valor
* Python usa los espacios en blanco como separadores de bloques

Algunos de los entornos más conocidos de Python son: [4]

* Idle
* Spyder
* IPython+Jupyter
* Pydev
* Pycharm
* Emacs+Python-mode

**Material y equipo**

* Sistema Operativo Windows 7
* Python 3.7
* Phycharm

**Desarrollo de la práctica**

**Objetivo:**

Aprender el uso y manejo de cadenas a través de programas desarrollados en Python

**Cuestionario:**

* ¿Quién desarrollo Python?

Este lenguaje fue desarrollado por Guido Van Rossum a finales de los años ochenta en el Centro para las Matemáticas y la Informática (CWI, Centrum Wiskunde & Informática), en los Países Bajos. [5] [2]

* ¿Por qué este lenguaje fue llamado Python?

Python es un lenguaje de programación interpretado y su nombre proviene de la afición de Van Rossum por el grupo de comedia británico llamado Monty Python. [3]

* Explica el término “Pythónico”

Código Pythonico es un código escrito en Python, que no solo es sintácticamente correcto, sino que está escrito de forma tal que aglutina las mejores prácticas del lenguaje Python, cumpliendo con las convenciones establecidas por la propia comunidad de Pythonistas. [1]

* Describe la diferencia entre lista, tupla y diccionario

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. [6]

A diferencia de las listas, las tuplas son objetos de tipo secuencia, específicamente es un tipo de dato [lista](https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion3/tipo_listas.html#python-list) inmutable. Esta no puede modificarse de ningún modo después de su creación. [4] [2]

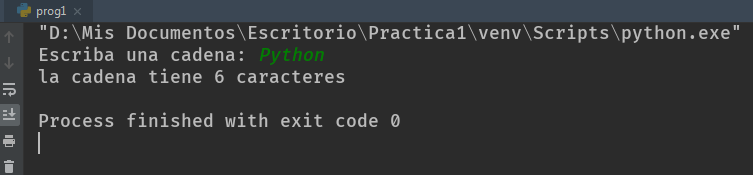
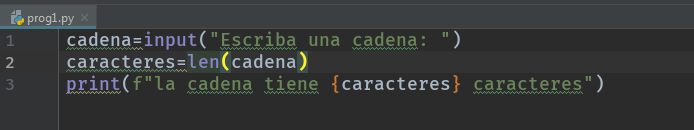
Un Diccionario es una estructura de datos y un tipo de dato en Python con características especiales que nos permite almacenar cualquier tipo de valor como enteros, cadenas, listas e incluso otras funciones. Estos  no tienen orden. [3]

**Ejercicios:**

1. Escriba un programa de Python para calcular la longitud de una cadena.

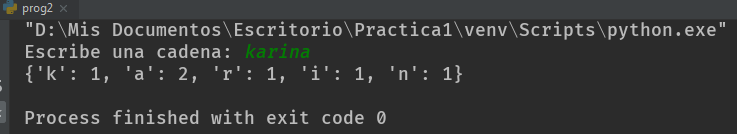
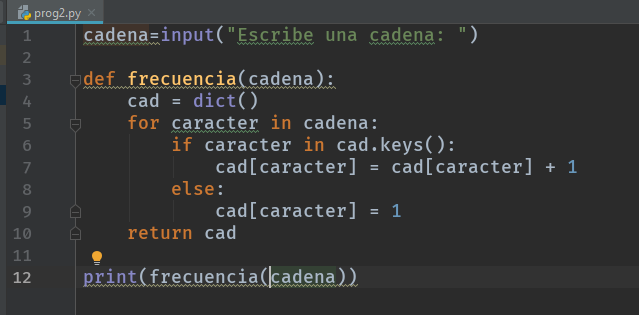
Utilizo la función **len** para contar los caracteres de la cadena ingresada por el usuario

Imagen 1: Código y ejecución del ejercicio 1



1. Escriba un programa de Python para contar el número de caracteres (frecuencia de caracteres) en una cadena.

Imagen 2: Código y ejecución del ejercicio 2



1. Escriba un programa de Python para obtener una cadena hecha de los 2 primeros y los 2 últimos caracteres de una determinada cadena. Si la longitud de cadena es menor que 2, devuelve la cadena vacía.

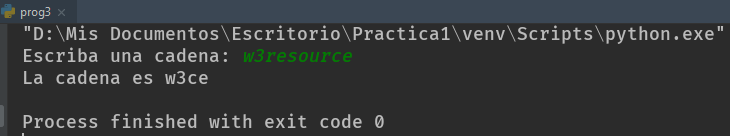
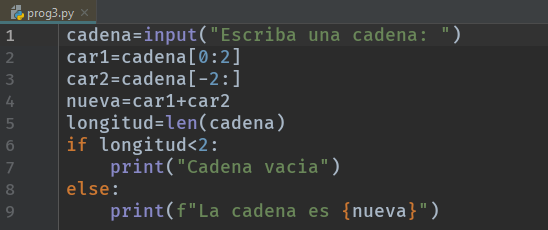
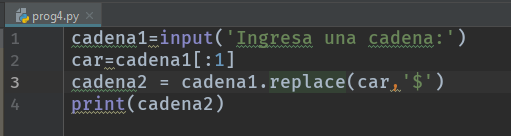


Imagen 3: Código y ejecución del ejercicio 3

1. Escriba un programa de Python para obtener una cadena de una cadena dada, donde todas las apariciones de su primer caracter se han cambiado a '$', excepto el propio primer carácter.

Imagen 4: Código y ejecución del ejercicio 4



1. Escriba un programa de Python para obtener una sola cadena de dos cadenas dadas, separadas por un espacio, además de intercambiar los dos primeros caracteres de cada cadena.

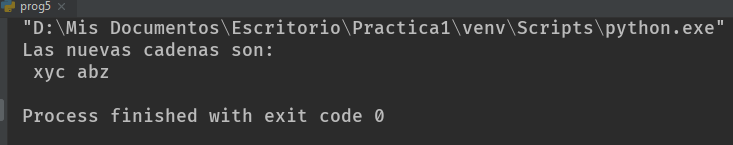
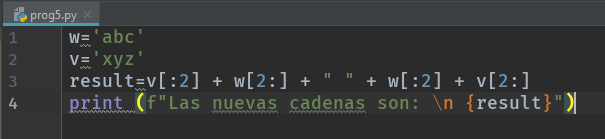
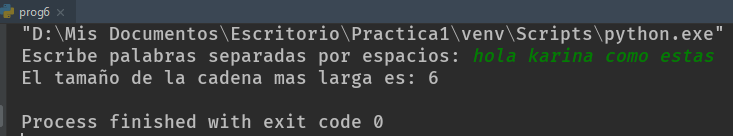
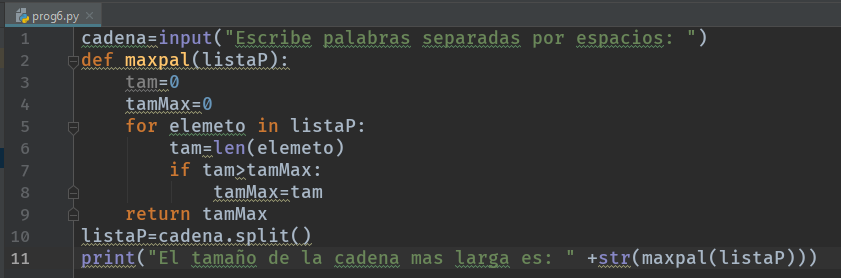


Imagen 5: Código y ejecución del ejercicio 5

1. Escribe una función de Python que tome una lista de palabras y devuelva la longitud de la más larga.

Imagen 6: Código y ejecución del ejercicio 6

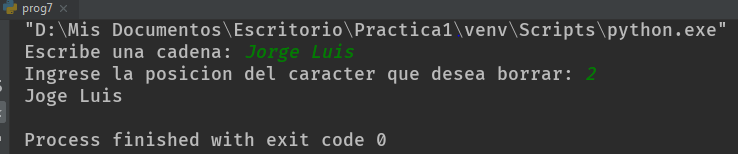
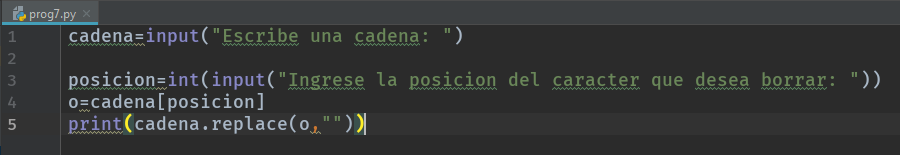
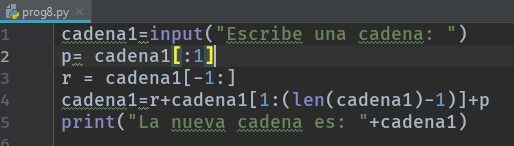


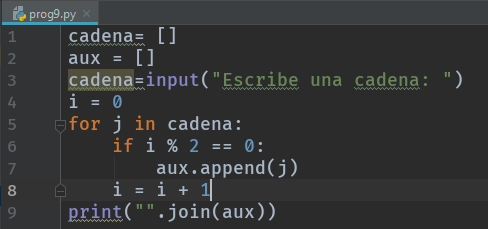
1. Escriba un programa de Python para quitar el carácter de una posición dada de una cadena no vacía.

Imagen 7: Código y ejecución del ejercicio 7

1. Escriba un programa de Python para cambiar una cadena dada a una nueva cadena donde se intercambiaron los caracteres primero y último.

Imagen 8: Código y ejecución del ejercicio 8



1. Escriba un programa de Python para eliminar los caracteres que tienen valores de índice impar de una cadena dada.

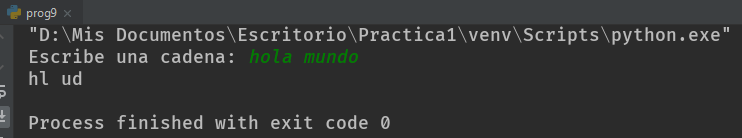
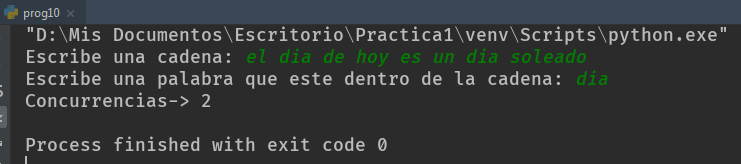
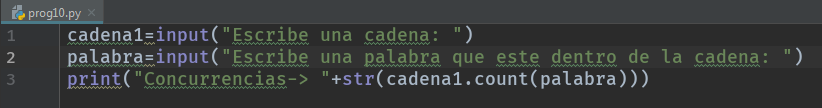


Imagen 9: Código y ejecución del ejercicio 9

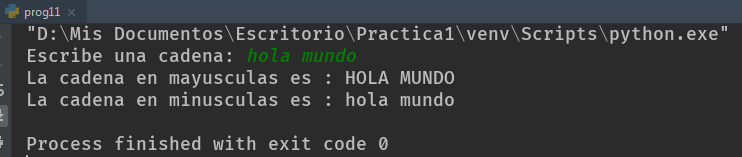
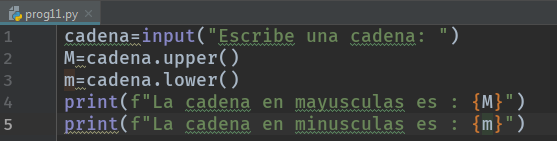
1. Escriba un programa de Python para contar las ocurrencias de cada palabra en una oración determinada.

Imagen 10: Código y ejecución del ejercicio 10



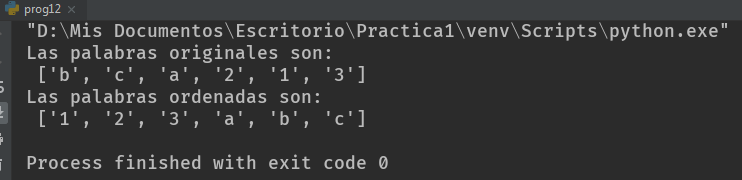
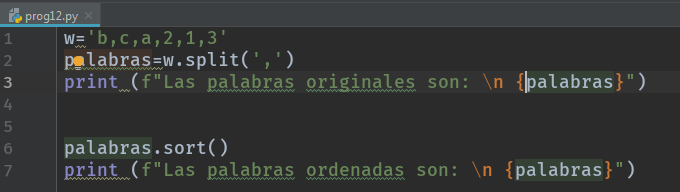
1. Escriba un script de Python que tome la entrada del usuario y muestre esa entrada de nuevo en solo mayúsculas y solo minúsculas.

Imagen 11: Código y ejecución del ejercicio 11



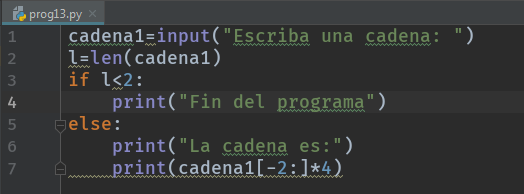
1. Escriba un programa de Python que acepte una secuencia de palabras separada por comas como entrada e imprima las palabras en forma ordenada (alfanuméricamente).

Imagen 12: Código y ejecución del ejercicio 12



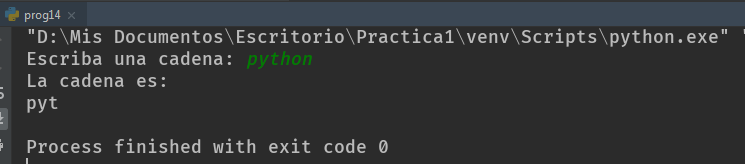
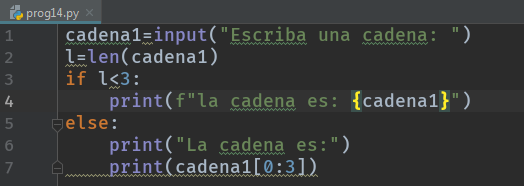
1. Escriba una función Python para obtener una cadena de 4 copias de los dos últimos caracteres de una cadena especificada (la longitud debe ser al menos 2).

Imagen 13: Código y ejecución del ejercicio 13



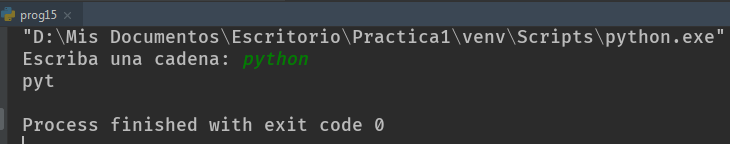
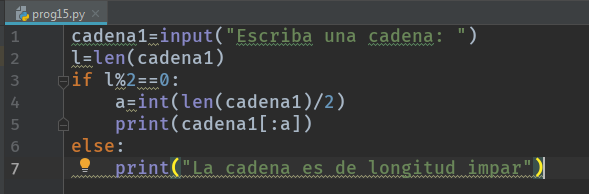
1. Escriba una función de Python para obtener una cadena hecha de los primeros tres caracteres de una cadena especificada. Si la longitud de la cadena es menor que 3, devuelva la cadena original.

Imagen 14: Código y ejecución del ejercicio 14



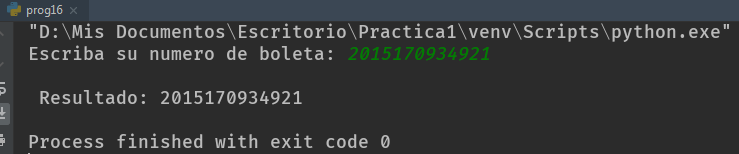
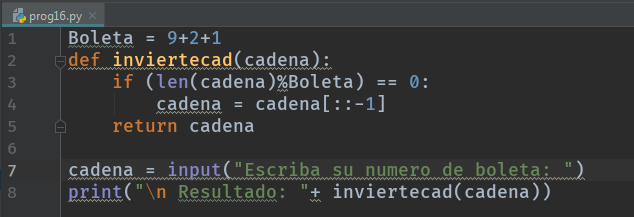
1. Escriba una función Python para obtener la primera mitad de una cadena especificada de longitud par.

Imagen 15: Código y ejecución del ejercicio 15



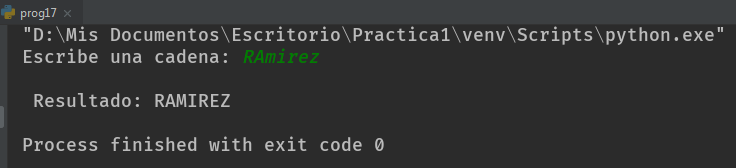
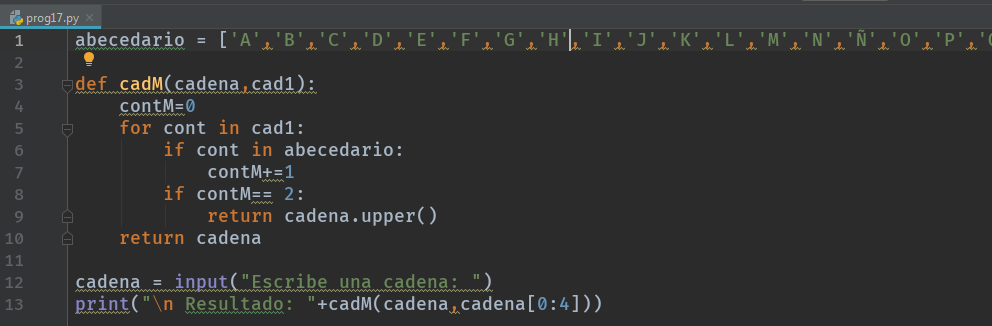
1. Escriba una función Python para invertir una cadena si su longitud es un múltiplo de la suma de los tres últimos dígitos de su No. de Boleta

Imagen 16: Código y ejecución del ejercicio 16



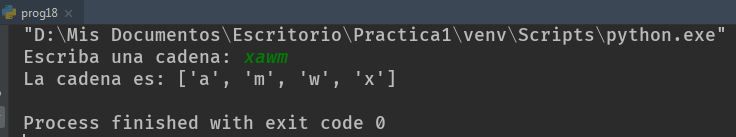
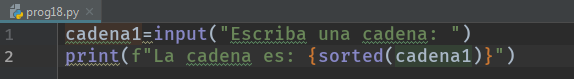
1. Escriba una función Python para convertir una cadena dada en mayúsculas si contiene al menos 2 caracteres en mayúsculas en los primeros 4 caracteres.

Imagen 17: Código y ejecución del ejercicio 17



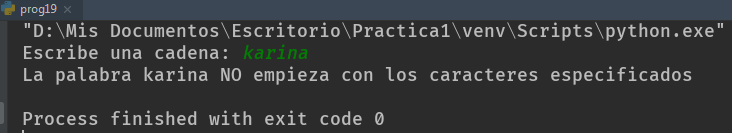
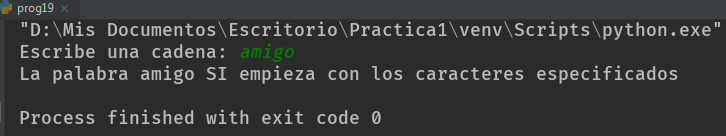
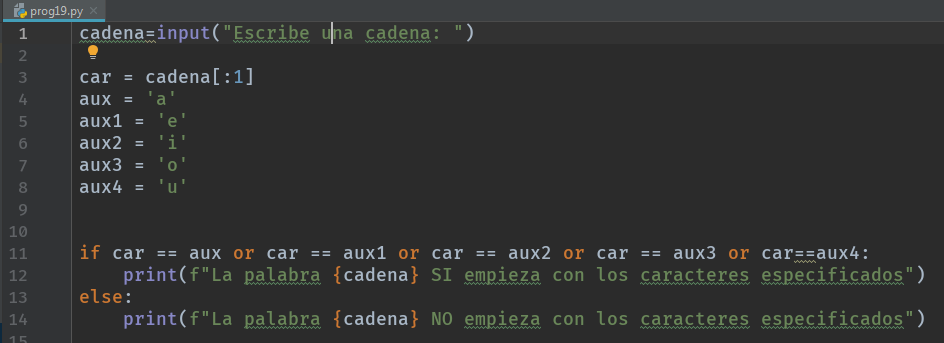
1. Escriba un programa de Python para ordenar una cadena lexicográficamente.

Imagen 18: Código y ejecución del ejercicio 18

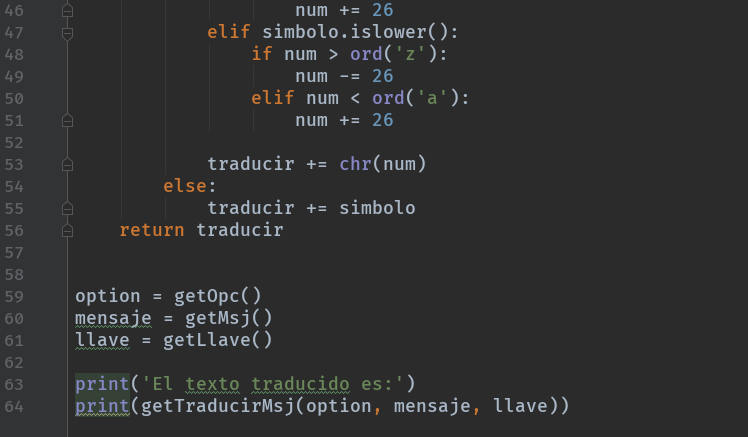
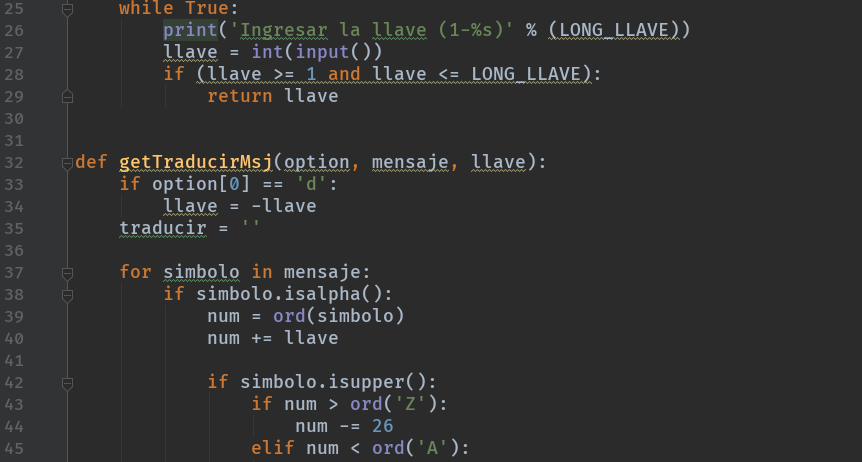


1. Escriba un programa de Python para comprobar si una cadena comienza con caracteres especificados.

Imagen 19: Código y ejecución del ejercicio 19



1. Escriba un programa de Python para crear un cifrado de César.



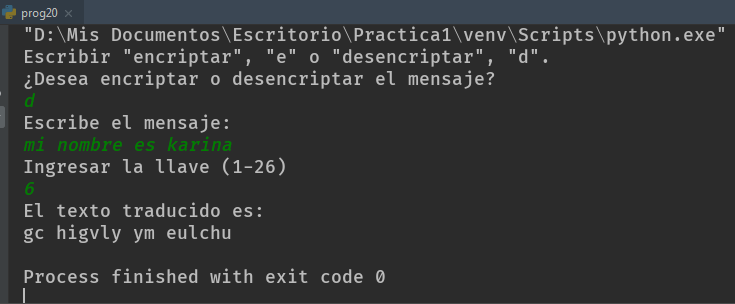
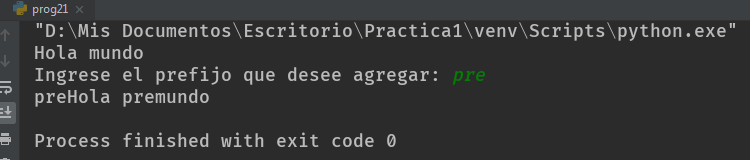
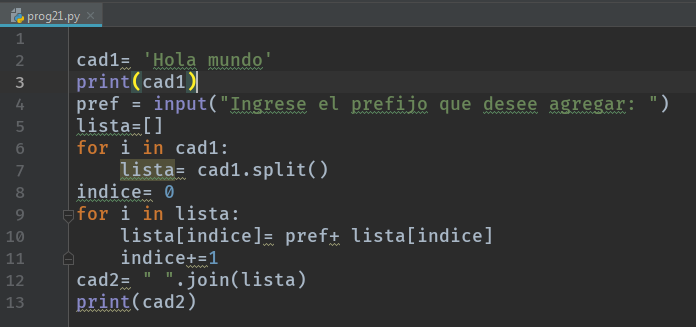


Imagen 20: Código y ejecución del ejercicio 20

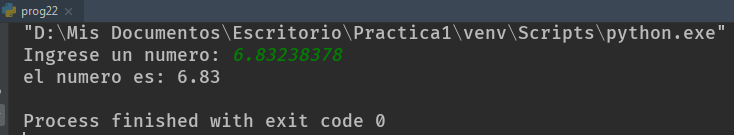
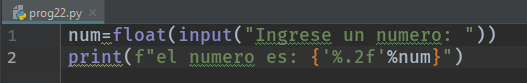
1. Escriba un programa Python para agregar un prefijo a todas las líneas de una cadena multilínea.

Imagen 21: Código y ejecución del ejercicio 21



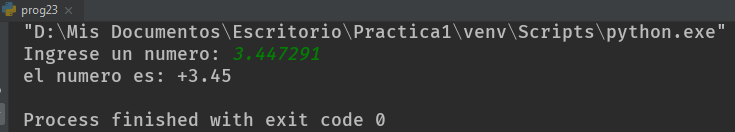
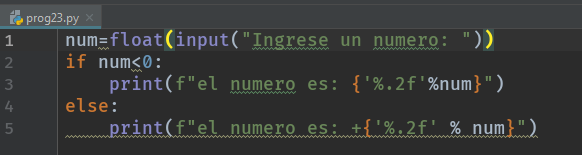
1. Escriba un programa Python para imprimir números flotantes hasta 2 decimales.

Imagen 22: Código y ejecución del ejercicio 22



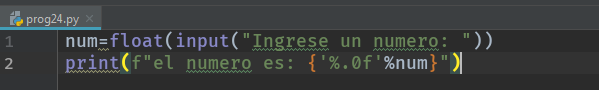
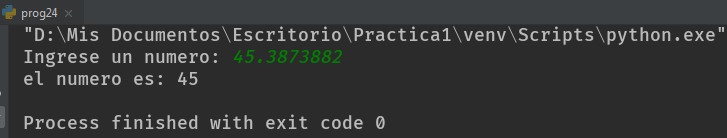
1. Escriba un programa Python para imprimir números flotantes hasta 2 decimales con signo.

Imagen 23: Código y ejecución del ejercicio 23



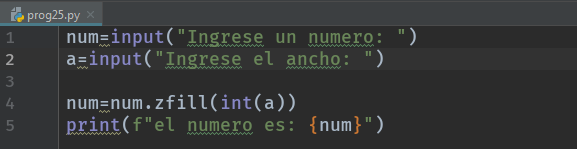
1. Escriba un programa Python para imprimir números flotantes sin decimales.

Imagen 24: Código y ejecución del ejercicio 24



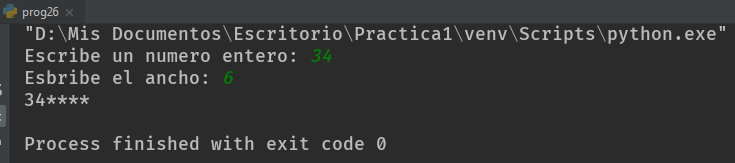
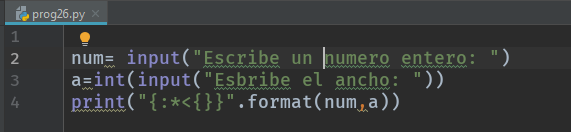
1. Escriba un programa Python para imprimir enteros con ceros a la izquierda hasta un ancho especificado.

Imagen 25: Código y ejecución del ejercicio 25



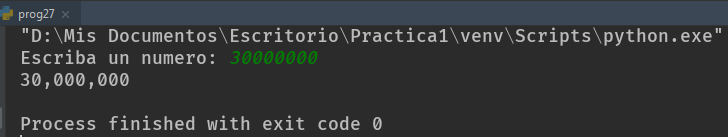
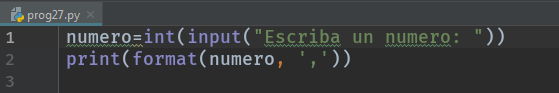
1. Escribe un programa Python para imprimir los siguientes números enteros con '\*' a la derecha del ancho especificado.

Imagen 26: Código y ejecución del ejercicio 26



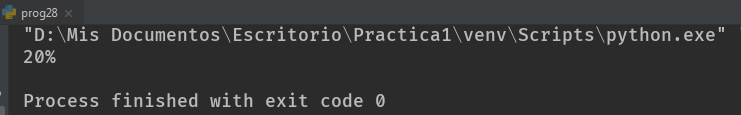
1. Escriba un programa de Python para mostrar un número con comas como separadores.

Imagen 27: Código y ejecución del ejercicio 27



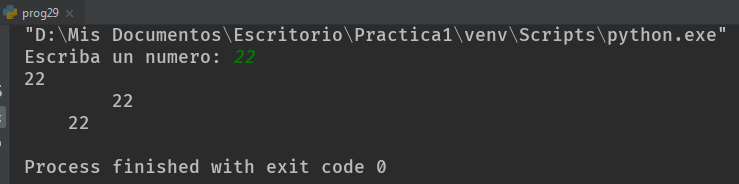
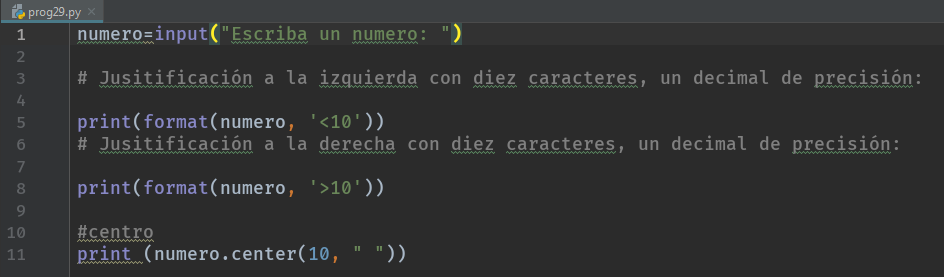
1. Escriba un programa de Python para formatear un número con formato de porcentaje.

Imagen 28: Código y ejecución del ejercicio 28



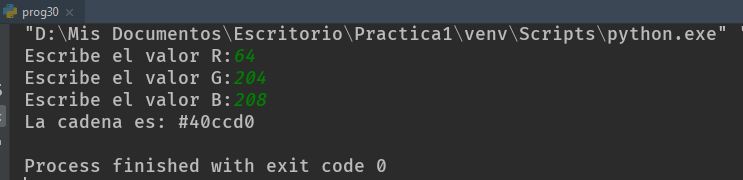
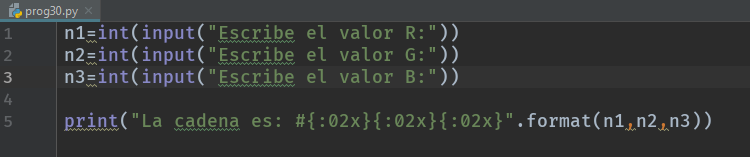
1. Escriba un programa de Python para mostrar un número en alineación izquierda, derecha y central alineados en ancho 10.

Imagen 29: Código y ejecución del ejercicio 29



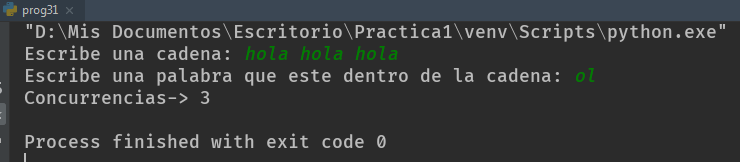
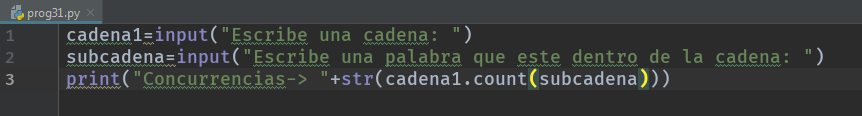
1. Escriba un programa en Python que coloque tres números en el rango (0,256) en una cadena hexadecimal que inicie con '#' (Formato RGB de Python TkInter).

Imagen 30: Código y ejecución del ejercicio 30



1. Escriba un programa Python para contar las ocurrencias de una subcadena en una cadena.

Imagen 31: Código y ejecución del ejercicio 31



1. Escriba un programa de Python para invertir palabras en una cadena.
2. Escriba un programa de Python para quitar un conjunto de caracteres de una cadena.
3. Escriba el programa de Python para contar los caracteres repetidos en una cadena.
4. Escriba un programa Python para imprimir el índice del carácter en una cadena.
5. Escriba un programa Python para comprobar si una cadena contiene todas las letras del alfabeto.
6. Escriba un programa Python para convertir una cadena en una lista.
7. Escriba un programa Python para poner en minúsculas los primeros n caracteres de una cadena.
8. Escriba un programa de Python para intercambiar coma y punto en una cadena.
9. Escriba un programa Python para contar las vocales de un texto dado.
10. Escriba un programa de Python para dividir una cadena en la última aparición del delimitador dado.
11. Escriba un programa Python para obtener la última parte de una cadena antes de un carácter especificado.
12. Escriba un programa en Python que imprima todas las subcadenas de una cadena y que identifique si se trata de un prefijo, un prefijo propio, un sufijo o un sufijo propio. Que también imprima el número total de subcadenas.

**Conclusiones**

De manera personal, esta práctica me ayudó mucho ya que yo nunca había programado en Python, empecé a practicar con estos ejercicios y a dame cuenta cómo funciona este lenguaje y en mi opinión es muy amigable con el usuario. Además de que estudie varias cosas sobre este lenguaje por mi cuenta para poder entender lo que me pedía hacer cada ejercicio.

Tuve varios problemas al entender algunos ejercicios por las razones que mencione anteriormente.

Pero en general python me parece un buen lenguaje de programación, fácil de entender con el cual podemos realizar diversas operaciones, entre otras cosas con cadenas, listas, tuplas y diccionarios.

# **Referencias**

|  |  |
| --- | --- |
| [1] | Gonzalo, «Cómo programar en Python de forma Pythonica.,» 10 Abril 2019. [En línea]. Available: http://pythonico.com/index.php/2019/04/10/que-se-entiende-por-codigo-pythonico/. |
| [2] | C. d. I. y. G. T. d. S. d. Cuba, «El lenguaje de programación Python,» Ivet Challenger-Pérez, Yanet Díaz-Ricardo, Roberto Antonio Becerra-García, 2014. [En línea]. Available: http://www.redalyc.org/articulo.oa?id=181531232001. |
| [3] | J. Pablo. [En línea]. Available: https://platzi.com/blog/historia-python/. |
| [4] | R. Python, «Listas y tuplas,» 9 Abril 2014. [En línea]. Available: https://recursospython.com/guias-y-manuales/listas-y-tuplas/. |
| [5] | L. E. Barrueto, «Maestros del Web,» 20 Enero 2010. [En línea]. Available: http://www.maestrosdelweb.com/introspectiva-guido-van-rossum-python/. |
| [6] | J. C. M. Urquiaga, «Listas en Python,» [En línea]. Available: https://devcode.la/tutoriales/listas-python/. |