

Estructura de Datos y Programación Orientada a Objetos

Cadenas y métodos de lista

Semana 3

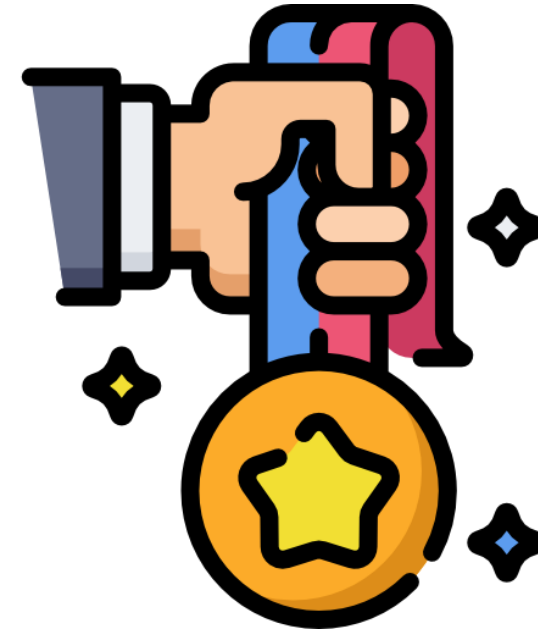


idat

Logro de aprendizaje



- Utiliza los diferentes métodos de cuerda a través de un lenguaje de programación.



Contenidos



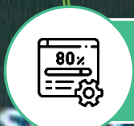
CADENAS
Y MÉTODOS
DE LISTA



Caracteres y cadenas



La naturaleza de las cadenas de Python



Métodos de cadenas



Cadenas en acción

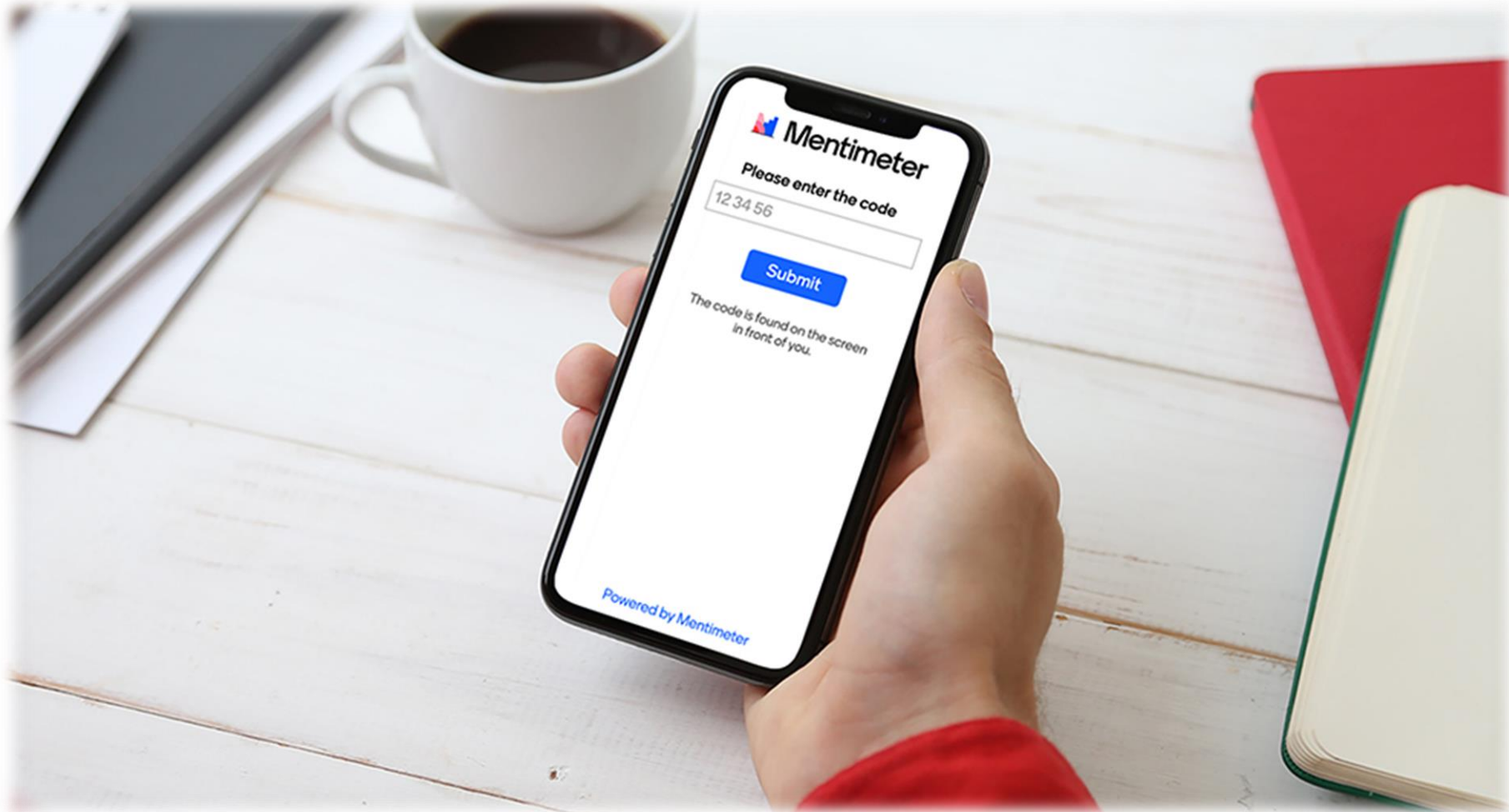


Cuatro programas simples

INICIO

www.menti.com

Ingresamos el siguiente código: 5677 5854



INICIO

30 métodos de las cadenas





[Inicio](#) [Códigos de fuente](#) [Guías y manuales](#) [Foro](#) [Micro](#) [Tutorial](#) [Newsletter](#) [Contacto](#) [Donar](#)

30 métodos de las cadenas

mayo 3, 2018 by Recursos Python [2 comentarios](#)

El tipo de dato `str` es una clase incorporada cuyas instancias incluyen variados métodos —más de treinta— para analizar, transformar, separar y unir el contenido de las cadenas de caracteres. En este pequeño artículo detallamos los que consideramos más relevantes para tener siempre a mano.

Métodos de análisis

El método `count()` retorna el número de veces que se repite un conjunto de caracteres especificado.

```
1. >>> s = "Hola mundo"
2. >>> s.count("Hola")
3. 1
```



Últimas entradas

[Ejecutar aplicación de Python al iniciar Windows](#)

[Ejecutar un código cada determinado tiempo](#)

[No se encontró Python; ejecuta sin argumentos para instalar desde Microsoft Store](#)

El tipo de dato `str` es una clase incorporada cuyas instancias incluyen variados métodos — más de treinta — para analizar, transformar, separar y unir el contenido de las cadenas de caracteres. En este pequeño artículo detallamos los que consideramos más relevantes para tener siempre a mano.



Fundamentos de Python 2: Módulo 2

Cadenas, Métodos de Listas y Excepciones

En este módulo, aprenderás sobre:

- Caracteres, cadenas y estándares de codificación.
- Cadenas frente a Listas: similitudes y diferencias.
- Métodos de listas.
- Métodos de cadenas.
- Cómo Python maneja los errores en tiempo de ejecución.
- Controlando el flujo de errores mediante `try` y `except`.
- La jerarquía de las excepciones.

Module 2:

Strings, string and list methods,
and exceptions

Python Essentials 2



Métodos de las cadenas



`upper()`

Devuelve la cadena con todos sus caracteres a mayúscula.

`lower()`

Devuelve la cadena con todos sus caracteres a minúscula.

`capitalize()`

Devuelve la cadena con su primer carácter en mayúscula.

`title()`

Devuelve la cadena con el primer carácter de cada palabra en mayúscula.

`count()`

Devuelve una cuenta de las veces que aparece una subcadena en la cadena.

`find()`

Devuelve el índice en el que aparece la subcadena (-1 si no aparece).

`rfind()`

Devuelve el índice en el que aparece la subcadena, empezando por el final.

`isdigit()`

Devuelve True si la cadena es todo números (False en caso contrario).

`isalnum()`

Devuelve True si la cadena es todo números o caracteres alfabéticos:

`isalpha()`

Devuelve True si la cadena es todo caracteres alfabéticos.



Métodos de las cadenas



`islower()`

Devuelve True si la cadena es todo minúsculas:

`isupper()`

Devuelve True si la cadena es todo mayúsculas.

`istitle()`

Devuelve True si la primera letra de cada palabra es mayúscula.

`isspace()`

Devuelve True si la cadena es todo espacios.

`startswith()`

Devuelve True si la cadena empieza con una subcadena.

`endswith()`

Devuelve True si la cadena acaba con una subcadena.

`split()`

Separa la cadena en subcadenas a partir de sus espacios y devuelve una lista.

`join()`

Une todos los caracteres de una cadena utilizando un caracter de unión.

`strip()`

Borra todos los espacios por delante y detrás de una cadena y la devuelve.

`replace()`

Reemplaza una subcadena de una cadena por otra y la devuelve.



Aplicación:



ACTIVIDAD 1:



Consigna: Desarrollo del Examen del Curso Python:

- Cada alumno tendrá que ingresar a Netacad con su cuenta de Cisco y desarrollar el Examen del Módulo 1.



Recursos: Cisco Networking Academy

<https://www.netacad.com/es>



Tiempo: 30 minutos



ACTIVIDAD 2:



Consigna: Trabajo Individual: Desarrollo de ejercicios

- Cada alumno deberá realizar un programa que de una calculadora utilizando el módulo tkinter..



Recursos: Pc o Laptop, Python, Visual Studio Code

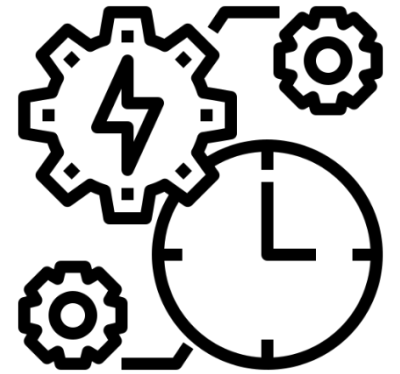


Tiempo: 30 minutos

Ejercicios



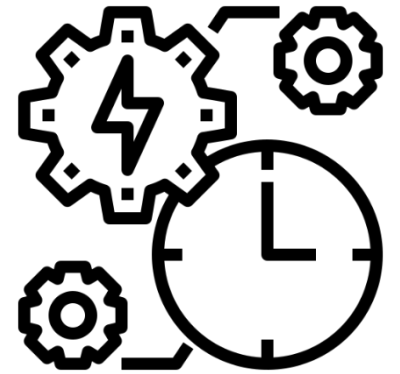
- Imprimir por consola cada carácter de una cadena introducida por teclado.
- Elabore un algoritmo que compruebe si una cadena leída por teclado comienza por una subcadena introducida por teclado.
- Elabore un programa que pida una cadena y un carácter por teclado (valida que sea un carácter) y muestra cuantas veces aparece el carácter en la cadena.
- Elabore un programa que pida una cadena por teclado que representa una frase (palabras separadas por espacios), finalmente que cuente cuantas palabras tiene.
- Elabore un programa que pida una cadena con un nombre y apellidos, y nos muestre las iniciales en mayúsculas.
- Elabore un programa que dada una cadena de caracteres, genere otra cadena que resulte de invertir la primera.



Ejercicios



- Elabore un programa que pida una cadena y dos caracteres por teclado (valida que sea un carácter), luego sustituya la aparición del primer carácter en la cadena por el segundo carácter.
- Elabore un programa que lea una cadena por teclado y convierta las mayúsculas a minúsculas y viceversa.
- Elabore un programa que compruebe si una cadena contiene una subcadena. Ambas cadenas se introducen por teclado.
- Elabore un programa que pida una cadena de caracteres e indique si es un palíndromo. Una palabra palíndroma es aquella que se lee igual adelante que atrás.



Aplicación:



ACTIVIDAD 3:



Consigna: Trabajo Individual: Conociendo el módulo tkinter

- Cada alumno deberá seguir los pasos indicados por el profesor.



Recursos: Pc o Laptop, Python, Visual Studio Code



Tiempo: 40 minutos

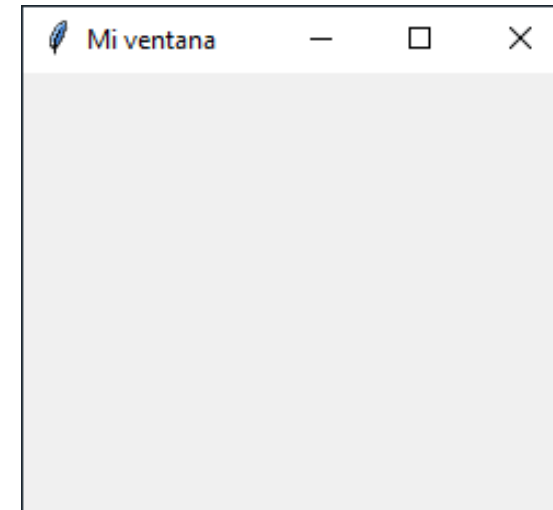
Conociendo el módulo tkinter



Crear tu primera aplicación GUI

```
from tkinter import *  
ventana = Tk()  
ventana.title("Mi ventana")  
ventana.mainloop()
```

El resultado debe verse como esto:

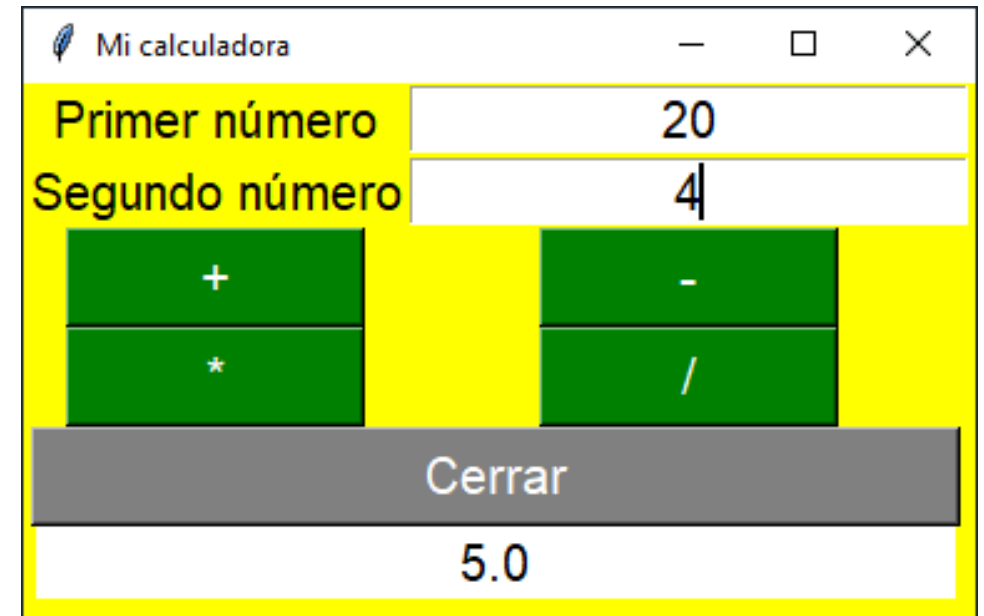


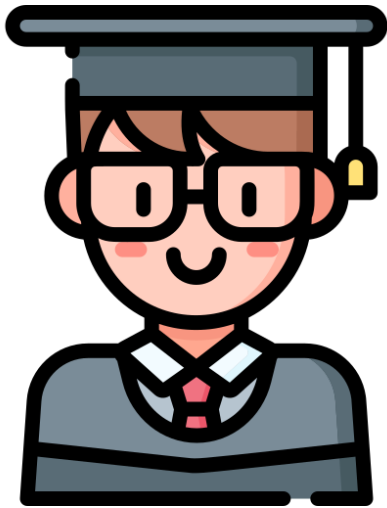
Creando mi primer proyecto (mi calculadora – forma 1)



Crearemos en primer lugar nuestra ventana:

```
from tkinter import *  
  
ventana = Tk()  
ventana.title("Mi calculadora")  
ventana.geometry("382x215")  
ventana.configure(background = "yellow")
```





1

¿Qué aprendimos hoy?

2

¿Por qué el tema tratado es importante en mi formación como programador?



Muchas Gracias
por su atención
¿Alguna pregunta?
¿No?
Excelente