

Lección: Bibliotecas estándar de Python

Introducción

Las bibliotecas estándar de Python son un conjunto de herramientas y módulos que se incluyen en la instalación estándar de Python y que pueden ser utilizados en la mayoría de los programas. En esta lección, veremos algunos ejemplos de uso de algunas de las bibliotecas más comunes.

Ejemplos de uso de bibliotecas comunes

Biblioteca `os`

La biblioteca `os` se utiliza para trabajar con archivos y directorios en el sistema operativo. Por ejemplo, para cambiar el directorio actual a la carpeta "Documentos", se puede utilizar el siguiente código:

```
1 import os
2
3 os.chdir('Documentos')
```

Para listar los archivos y directorios en el directorio actual, se puede utilizar el siguiente código:

```
1 import os
2
3 contenido = os.listdir('.')
4 print("Contenido del directorio actual:", contenido)
```

Biblioteca `datetime`

La biblioteca `datetime` se utiliza para trabajar con fechas y tiempos. Por ejemplo, para obtener la fecha y hora actual, se puede utilizar el siguiente código:

```
1 from datetime import datetime
2
3 now = datetime.now()
4 print("Fecha y hora actual:", now)
```

Para convertir una cadena de texto en una fecha y hora, se puede utilizar el siguiente código:

```
1 from datetime import datetime
2
3 fecha_string = '01-01-2022 12:30:45'
4 fecha = datetime.strptime(fecha_string, '%d-%m-%Y %H:%M:%S')
5 print("Fecha y hora convertidas:", fecha)
```

Biblioteca random

La biblioteca `random` se utiliza para generar números aleatorios. Por ejemplo, para generar un número aleatorio entre 1 y 10, se puede utilizar el siguiente código:

```
1 import random
2
3 num_aleatorio = random.randint(1, 10)
4 print("Número aleatorio:", num_aleatorio)
```

Para obtener una lista aleatoria de elementos de una lista, se puede utilizar el siguiente código:

```
1 import random
2
3 lista = ['manzana', 'pera', 'naranja', 'kiwi']
4 lista_aleatoria = random.sample(lista, k=len(lista))
5 print("Lista aleatoria:", lista_aleatoria)
```

Biblioteca re

La biblioteca `re` se utiliza para trabajar con expresiones regulares. Por ejemplo, para buscar todas las palabras que empiezan con "a" en una cadena de texto, se puede utilizar el siguiente código:

```
1 import re
2
3 texto = "Hola amigos, ¿cómo están?"
4 palabras_con_a = re.findall(r'\ba\w+', texto)
5 print("Palabras que empiezan con 'a':", palabras_con_a)
```

Para sustituir todas las ocurrencias de una palabra en una cadena de texto, se puede utilizar el siguiente código:

```
1 import re
2
3 texto = "El perro marrón corre más rápido que el perro negro."
4 texto_modificado = re.sub(r'perro', 'gato', texto)
5 print("Texto modificado:", texto_modificado)
```

Biblioteca csv

La biblioteca `csv` se utiliza para leer y escribir archivos CSV. Por ejemplo, para leer un archivo CSV llamado "datos.csv" y mostrar el contenido en la consola, se puede utilizar el siguiente código:

```

1 import csv
2
3 with open('datos.csv', newline='') as archivo_csv:
4     lector_csv = csv.reader(archivo_csv)
5     for fila in lector:
6         print(fila)

```

Para escribir datos en un archivo CSV, se puede utilizar el siguiente código:

```

1 import csv
2
3 datos = [
4     ['Juan', 'Pérez', 25],
5     ['María', 'García', 30],
6     ['Pedro', 'Martínez', 40]
7 ]
8
9 with open('datos.csv', 'w', newline='') as archivo_csv:
10     escritor_csv = csv.writer(archivo_csv)
11     for fila in datos:
12         escritor_csv.writerow(fila)

```

Biblioteca json

La biblioteca `json` se utiliza para trabajar con archivos en formato JSON. Por ejemplo, para cargar datos de un archivo JSON llamado "datos.json" y mostrarlos en la consola, se puede utilizar el siguiente código:

```

1 import json
2
3 with open('datos.json', 'r') as archivo_json:
4     datos = json.load(archivo_json)
5     print("Datos cargados:", datos)

```

Para guardar datos en un archivo JSON, se puede utilizar el siguiente código:

```

1 import json
2
3 datos = {
4     'nombre': 'Juan Pérez',
5     'edad': 25,
6     'ciudad': 'Madrid'
7 }
8
9 with open('datos.json', 'w') as archivo_json:
10     json.dump(datos, archivo_json)

```

Conclusiones

Las bibliotecas estándar de Python son una herramienta poderosa y útil para cualquier programador de Python. Conocer algunas de las bibliotecas comunes y cómo utilizarlas correctamente puede ahorrar mucho tiempo y esfuerzo al escribir programas. En esta lección, hemos visto algunos ejemplos de uso de bibliotecas comunes, como `os`, `datetime`, `random`, `re`, `csv` y `json`, y esperamos que hayan sido útiles para ampliar tus conocimientos en el uso de las bibliotecas estándar de Python.