

Misión Python – Tema 7 Módulos Y Paquetes

Adrian Mauricio Arandia Urrea

Centro de Desarrollo Agroempresarial y Turístico del Huila

2502636: Tecnólogo en Análisis y Desarrollo de Software

02 de noviembre del 2023

Apropiación punto 3

DateTime

El módulo **datetime** se utiliza para trabajar con fechas y horas en Python. Proporciona clases y funciones para manejar objetos de fecha y hora, calcular diferencias de tiempo, formatear fechas y horas, y mucho más.

Este módulo tiene algunas funciones principales:

- **datetime.datetime**: La clase principal que representa objetos de fecha y hora.
- **datetime.date**: Para trabajar con fechas.
- **datetime.time**: Para trabajar con horas.
- **datetime.timedelta**: Para calcular diferencias de tiempo.
- **datetime.timezone**: Para representar zonas horarias.
- **datetime.now()**: Devuelve la fecha y hora actual.
- **datetime.strptime()**: Convierte una cadena en un objeto **datetime**.
- **datetime.strftime()**: Formatea un objeto **datetime** en una cadena.

Ejemplo en código de su utilización

```
1  import datetime
2
3  now = datetime.datetime.now()
4  print("Fecha y hora actual:", now)
5
6  today = datetime.date.today()
7  print("Fecha actual:", today)
8
9  current_time = datetime.datetime.now().time()
10 print("Hora actual:", current_time)
11
12 delta = datetime.timedelta(days=7)
13 one_week_later = today + delta
14 print("Una semana después:", one_week_later)
15
16 formatted_date = today.strftime("%Y-%m-%d")
17 print("Fecha formateada:", formatted_date)
18
19 date_str = "2023-11-02"
20 parsed_date = datetime.datetime.strptime(date_str, "%Y-%m-%d")
21 print("Fecha parseada:", parsed_date)
22
```

El módulo **datetime** es esencial para trabajar con fechas y horas en Python y se utiliza en una amplia variedad de aplicaciones, desde programación de calendarios hasta registros de tiempo y más. Ofrece una amplia gama de funciones y métodos para facilitar el manejo de datos temporales en tus programas.