

Comparar dos fechas en Python

Syed Moiz Haider

- 1.
2. [HowTo](#)
3. [How-To's de Python](#)
4. [Comparar dos fechas en Python](#)

Creado: December-28, 2020 | Actualizado: July-24, 2021

1. [Utiliza el módulo `datetime` y el operador `</>` para comparar dos fechas en Python](#)
2. [Use el método `datetime.date\(\)` para comparar dos fechas en Python](#)
3. [Use el módulo `time` para comparar dos fechas en Python](#)
4. [Usa el módulo `datetime` para comparar dos fechas](#)

El tutorial explica cómo comparar dos fechas en Python. Tiene múltiples métodos para identificar qué fecha es mayor, por lo que el tutorial también enumera diferentes códigos de ejemplo para elaborar diferentes enfoques.

[Utiliza el módulo `datetime` y el operador `</>` para comparar dos fechas en Python](#)

La `datetime` y los simples operadores de comparación `<` o `>` pueden ser usados para comparar dos fechas. El módulo [datetime](#) proporciona el método `timedelta` para manipular fechas y horas. El método `timedelta()` toma el número de días como entrada y puede realizar aritmética sobre él.

A continuación se muestra un código de ejemplo:

pythonCopy

```
from datetime import datetime, timedelta

previous_date = datetime.now() - timedelta(days=1)
current_date = datetime.now()
print(previous > past)
```

Resultado:

textCopy

```
True
```

[Use el método `datetime.date\(\)` para comparar dos fechas en Python](#)

`datetime.date()` también se puede usar para comparar dos fechas. El método `datetime.date()` toma como entrada `year`, `month`, `day`. Crea dos fechas para ser comparadas y usa un simple operador de comparación para comparar dos fechas.

A continuación se muestra un código de ejemplo.

pythonCopy

```
import datetime

first_date = datetime.date(2020, 12, 16)
second_date = datetime.date(2015, 12, 16)

result = first_date < second_date
print(result)
```

Resultado:

textCopy

False

[Use el módulo time para comparar dos fechas en Python](#)

El módulo `time` proporciona el método `strptime` para manipular las fechas. Toma la fecha en formato de cadena como entrada y la convierte en el formato de fecha de Python. Más tarde, una simple comparación puede ser realizada para comparar dos fechas.

Un ejemplo de código se da como sigue.

pythonCopy

```
import time

first_date = "30/11/2020"
second_date = "12/10/2019"

formatted_date1 = time.strptime(first_date, "%d/%m/%Y")
formatted_date2 = time.strptime(second_date, "%d/%m/%Y")
print(formatted_date1 > formatted_date2)
```

Resultado:

pythonCopy

True

[Usa el módulo datetime para comparar dos fechas](#)

El módulo `datetime` proporciona el método `datetime()` que toma tres parámetros para crear una fecha a partir del año, el mes y el día. Después de obtener las fechas, pueden ser comparadas usando los operadores de comparación.

A continuación se da un código de ejemplo:

pythonCopy

```
import datetime

# date in yy/mm/dd format
first_date = datetime.datetime(2020, 5, 11)
second_date = datetime.datetime(2020, 6, 10)

print("first date is greater than second_date: ", first_date > second_date)
print("first date is smaller than second_date: ", first_date < second_date)
print("first date is not equal to second_date: ", first_date != second_date)
```

Resultado:

textCopy

```
first date is greater than second_date: False
first date is smaller than second_date: True
first date is not equal to second_date: True
```

Artículo relacionado - Python DateTime

- [Cómo convertir la columna del DataFrame a Datetime en Pandas](#)
- [Cómo obtener la hora actual en Python](#)
- [Consigue el día de la semana en Python](#)
- [Cómo convertir una cadena a datetime en Python](#)

