



# Data e hora

**Guilherme Arthur de Carvalho**

Analista de sistemas

**<https://linktr.ee/decarvalhogui>**

# Objetivo Geral

Aprender a trabalhar com datas, horas e fusos horários em Python, dominando o módulo 'datetime' para manipulações precisas.



# Pré-requisitos

✓ Python e VSCode



# Conteúdo

- ☐ Introdução ao módulo datetime
- ☐ Manipulação de datas e horas
- ☐ Conversão e formatação de datas e horas
- ☐ Trabalhando com timezones


# Introdução ao módulo datetime

Data e hora

# O que é o módulo datetime?

O módulo 'datetime' em Python é usado para lidar com datas e horas. Ele possui várias classes úteis como date, time e timedelta.

# Exemplo de código:



```
1 import datetime
2
3 d = datetime.date(2023, 7, 19)
4 print(d) # 2023-07-19
```

# Manipulação de datas e horas

Data e hora



# Introdução

Podemos criar e manipular objetos date, time e datetime de várias maneiras. Por exemplo, podemos adicionar e subtrair datas, verificar a diferença entre datas e muito mais.

# Exemplo de código:



```
1 import datetime
2
3 # Criando data e hora
4 d = datetime.datetime(2023, 7, 19, 13, 45)
5 print(d) # 2023-07-19 13:45:00
6
7 # Adicionando uma semana
8 d = d + datetime.timedelta(weeks=1)
9 print(d) # 2023-07-26 13:45:00
```

# Conversão e formatação de datas e horas

Data e hora

# Introdução

Python também permite converter e formatar datas e horas. Para isso, usamos os métodos 'strftime' (string format time) e 'strptime' (string parse time).

# Exemplo de código:

```
1 import datetime
2
3 d = datetime.datetime.now()
4
5 # Formatando data e hora
6 print(d.strftime("%d/%m/%Y %H:%M")) # 19/07/2023 14:20
7
8 # Convertendo string para datetime
9 date_string = "20/07/2023 15:30"
10 d = datetime.datetime.strptime(date_string, "%d/%m/%Y %H:%M")
11 print(d) # 2023-07-20 15:30:00
```

# Trabalhando com timezones

Data e hora

# Introdução

Quando trabalhamos com data e hora, lidar com fusos horários é uma necessidade comum. Python facilita isso através do módulo 'pytz'.

# Exemplo de código:



```
1 # pip install pytz
2
3 import datetime
4 import pytz
5
6 # Criando datetime com timezone
7 d = datetime.datetime.now(pytz.timezone("America/Sao_Paulo"))
8 print(d) # 2023-07-19 14:25:00-03:00
```



# Trabalhando com tz sem bibliotecas externas

O Python permite fazer isso com o módulo datetime padrão, embora seja um pouco mais complexo do que usando bibliotecas como 'pytz'.

# Exemplo de código:

```
1 import datetime
2
3 # Criando datetime com timezone
4 d = datetime.datetime.now(datetime.timezone(datetime.timedelta(hours=-3), "BRT"))
5 print(d) # 2023-07-19 14:25:00-03:00
6
7 # Convertendo para outro timezone
8 d_utc = d.astimezone(datetime.timezone.utc)
9 print(d_utc) # 2023-07-19 17:25:00+00:00
```

# Dúvidas?

> Fórum/Artigos - <https://web.dio.me/articles>

# Exemplo de código:

Insira sua imagem dentro deste espaço  
(retire o retângulo azul, ele deverá ser utilizado  
somente para referência)

