

Acadêmicos: João Vítor Oliveira da Inhaia e Raquel Marafon

Disciplina: Programação Web I

Professor: Rafael Vivian

Data: 21/08/2025

Manipulação de Datas e Horas em Python utilizando o módulo datetime

1. Importando o módulo

```
from datetime import datetime, timedelta
```

- O **datetime** é um módulo da biblioteca padrão do Python (não precisa instalar nada).
- Dentro dele existem várias classes para lidar com **datas e horários**.
- Aqui usamos:
 - **datetime**: representa datas e horas completas (ano, mês, dia, hora, minuto, segundo).
 - **timedelta**: representa intervalos de tempo (diferença em dias, horas, segundos etc.).

2. Criar uma data/hora atual

```
agora = datetime.now()  
print("Data e hora atuais:", agora)
```

- **datetime.now()** pega a **data e hora exata do sistema**.
- Retorna algo como: **2025-08-21 08:15:32**

3. Criar uma data específica

```
data_especifica = datetime(2025, 12, 25, 20, 30, 0)
print("Data específica:", data_especifica)
```

- Aqui passamos: **ano, mês, dia, hora, minuto, segundo**.
- Exemplo: `2025-12-25 20:30:00` (Natal às 20h30).

4. Formatar datas para string (exibição)

```
formatada = agora.strftime("%d/%m/%Y %H:%M:%S")
print("Data formatada:", formatada)
```

- `strftime` (string format time) transforma uma data em **texto formatado**.
- `%d` → dia, `%m` → mês, `%Y` → ano, `%H:%M:%S` → hora:minuto:segundo.
- Exemplo: `21/08/2025 08:15:32`.

5. Converter string em data

```
data_str = "21/08/2025 14:30:00"
data_convertida = datetime.strptime(data_str, "%d/%m/%Y %H:%M:%S")
print("Data convertida de string:", data_convertida)
```

- `strptime` (string parse time) faz o **contrário do `strftime`**: pega um texto e transforma em data.
- Exemplo: `"21/08/2025 14:30:00"` → `datetime(2025, 8, 21, 14, 30, 0)`.

6. Calcular diferença entre duas datas

```
diferenca = data_especifica - agora
```

```
print("Diferença até o Natal:", diferenca)
print("Dias até o Natal:", diferenca.days)
```

- Subtraindo datas, obtemos um objeto `timedelta`.
 - Ele mostra quantos **dias, horas, minutos, segundos** faltam.
 - `.days` retorna apenas os dias inteiros.
-

7. Somar ou subtrair tempo (`timedelta`)

```
amanha = agora + timedelta(days=1)
ontem = agora - timedelta(days=1)
print("Amanhã:", amanhã)
print("Ontem:", ontem)
```

- `timedelta(days=1)` representa **1 dia**.
- Podemos usar também `hours`, `minutes`, `seconds`.
- Isso serve para calcular prazos, agendamentos, etc.

CÓDIGO

```
# =====
# Manipulação de Datas e Horas em Python
# Usando o módulo datetime
# =====

from datetime import datetime, timedelta

# Função auxiliar para garantir formato brasileiro sem
# milissegundos
def formatar(data):
    return data.strftime("%d/%m/%Y %H:%M:%S")

# 1. PEGAR DATA E HORA ATUAIS
agora = datetime.now()
print("Data e hora atuais:", formatar(agora), "\n")

# 2. CRIAR UMA DATA ESPECÍFICA
data_especifica = datetime(2025, 12, 25, 20, 30, 0)
print("Data específica (Natal):", formatar(data_especifica),
      "\n")

# 3. FORMATAR DATA PARA STRING
formatada = agora.strftime("%d/%m/%Y %H:%M:%S")
print("Data atual formatada:", formatada, "\n")

# 4. CONVERTER STRING PARA DATA
data_str = "21/08/2025 14:30:00"
data_convertida = datetime.strptime(data_str, "%d/%m/%Y
%H:%M:%S")
print("String original:", data_str)
print("Convertida para datetime:", formatar(data_convertida),
      "\n")

# 5. CALCULAR DIFERENÇA ENTRE DUAS DATAS
diferenca = data_especifica - agora
print("Diferença até o Natal:", diferenca)          # mostra
dias + horas + minutos
print("Dias até o Natal:", diferenca.days, "\n")

# 6. SOMAR E SUBTRAIR TEMPO COM timedelta
amanha = agora + timedelta(days=1)
ontem = agora - timedelta(days=1)
```

```
print("Amanhã será:", formatar(amanha))  
print("Ontem foi:", formatar(ontem), "\n")
```

7. EXEMPLO MAIS COMPLETO DE timedelta

```
mais_tempo = agora + timedelta(days=7, hours=3, minutes=15)  
print("Daqui a 7 dias, 3h e 15min será:",  
      formatar(mais_tempo))
```