

1 | Aktueller Zeitpunkt

```
from datetime import datetime
jetzt = datetime.now()
print(jetzt)
print(jetzt.date())
print(jetzt.time())
```

2 | Datum anlegen

```
from datetime import date, datetime
tag = date(2026, 9, 1)
zeitpunkt = datetime(2026, 9, 1, 14, 30)
```

3 | Formatieren

- strftime erzeugt Text.
- Formatcodes bestimmen Darstellung.

```
text = jetzt.strftime("%d.%m.%Y %H:%M")
print(text)
```

4 | Text einlesen

- .strptime parst Text nach festem Muster.
- Falsches Format löst ValueError aus.

```
from datetime import datetime
termin = datetime.strptime("01.09.2026 14:30", "%d.%m.%Y %H:%M")
```

5 | Wichtige Formatcodes

Code	Bedeutung
%d	Tag 01-31
%m	Monat 01-12
%Y	Jahr vierstellig
%H	Stunde 00-23
%M / %S	Minute / Sekunde

6 | Zeitdifferenz

```
from datetime import datetime
start = datetime(2026, 9, 1, 10, 0)
ende = datetime(2026, 9, 1, 12, 30)
dauer = ende - start
print(dauer.total_seconds())
```

7 | timedelta

```
from datetime import datetime, timedelta
jetzt = datetime.now()
morgen = jetzt + timedelta(days=1)
spaeter = jetzt + timedelta(minutes=90)
```

8 | Laufzeit messen

- perf_counter für Zeitmessung von Code.
- Differenz in Sekunden.

```
from time import perf_counter
start = perf_counter()
# zu messender Code
dauer = perf_counter() - start
print(f"{dauer:.3f} s")
```

9 | Warten

- sleep blockiert aktuellen Thread.
- Nicht als präzisen Timer oder GUI-Lösung missverstehen.

```
from time import sleep
print("Start")
sleep(1.5)
print("Weiter")
```

10 | Zeitzonen

- Naive datetime hat keine Zeitzone.
- Für Austausch timezone-aware Werte nutzen.
- Intern häufig UTC, Anzeige lokal.

```
from datetime import datetime, timezone
jetzt_utc = datetime.now(timezone.utc)
print(jetzt_utc.isoformat())
```

11 | Typische Fehler

Problem	Prüfen
Tag/Monat vertauscht	Formatmuster
Sekunden statt ms	Einheit
Sommerzeit falsch	Zeitzone
Laufzeit mit now	perf_counter
sleep ungenau	Scheduler/Timer

12 | Mini-Übung: Dauer

Aufgabe

Miss die Laufzeit einer Schleife und gib Millisekunden aus.

Musterlösung

```
from time import perf_counter
start = perf_counter()
summe = sum(range(100000))
dauer_ms = (perf_counter() - start) * 1000
print(f"{dauer_ms:.2f} ms")
```