

## time – für Wartezeiten & aktuelle Zeit

Die Python-Bibliothek **time** ermöglicht den Umgang mit Zeitangaben innerhalb eines Programms. Damit kann man

- die **aktuelle Uhrzeit** auslesen,
- **Zeitspannen messen** oder
- den Programmablauf für eine bestimmte Dauer **pausieren**.

Viele Funktionen greifen direkt auf die **Systemzeit** zu, sodass Jahr, Monat, Tag, Stunde, Minute und Sekunde bequem ausgelesen werden können.

### Import der Bibliothek

```
1 import time
```

### Wichtige Befehle aus der Bibliothek **time**

#### 1. Aktuelle Zeit in Sekunden

```
1 import time
2
3 # Aktuelle Zeit in Sekunden
4 print(time.time())
```

Beispielausgabe am 23.11.2025

1763925327.5242565

#### Erklärung

Seit 1. Januar 1970, 0 Uhr, sind 1763925327 Sekunden vergangen.

#### 2. Formatiertes Datum und Uhrzeit

```
1 import time
2
3 # Formatiertes Datum und Uhrzeit
4 print(time.strftime("%Y-%m-%d %H:%M:%S"))
```

Ausgabe am 23.11.2025:

2025-11-23 19:20:39

**Erklärung der Formatzeichen:**

**Y** = Year  
**m** = month  
**d** = day  
**H** = hour  
**M** = minute  
**S** = second

Nur Jahr:

```
1 import time
2 print(time.strftime("%Y"))
```

Ausgabe: 2025

Nur Monat:

```
1 import time
2 print(time.strftime("%m"))
```

**3. Alternative Möglichkeit**

```
1 import time
2
3 # aktuelle Zeit holen
4 jetzt = time.localtime()
5 |
6 # jede Angabe untereinander ausgeben
7 print("Jahr:", jetzt.tm_year)
8 print("Monat:", jetzt.tm_mon)
9 print("Tag:", jetzt.tm_mday)
10 print("Stunde:", jetzt.tm_hour)
11 print("Minute:", jetzt.tm_min)
12 print("Sekunde:", jetzt.tm_sec)
```

Ausgabe am 23.11.2025:

```
📄 Jahr: 2025
📄 Monat: 11
📄 Tag: 23
📄 Stunde: 19
📄 Minute: 30
>_ Sekunde: 2
```

Erklärung:

tm steht für time

## 4. Warten

```
1 import time
2
3 # 2 Sekunden warten
4 print("Warte...")
5 time.sleep(2)
6 print("Weiter geht's!")
```

Ausgabe am 23.11.2025:

```
Warte...
Weiter geht's!
25
```