



# Tag 5

## Vorstellungen Module

17. März 2023

# datetime

- Wichtigste Klassen
  - Datum (***date***)
  - Zeit (***time***)
  - «Zeitstempel» (Zeit und Datum → ***datetime***)
  - Dauer (***timedelta***)
- Datentypen und Funktionen für Formatierung zu string: ***strptime***
- Rudimentäre Konversion string zu Datum / Zeit / Datum&Zeit

# Instanziierung

	Implizit	«Jetzt»	Aus String
date	<code>Date(2022, 12, 9)</code> <sup>1)</sup>	<code>date.today()</code>	<code>date.fromisostring("2022-12-09")</code>
time	<code>Time(13, 59, 21)</code>	- <sup>2)</sup>	<code>time.fromisostring("13:59:21")</code>
datetime	<code>datetime(2022, 12, 9, 13, 59)</code>	<code>datetime.now()</code>	<code>datetime.fromisostring("2022-12-09 13:59:21")</code>

<sup>1)</sup> 0 am Anfang von Integers ist nicht erlaubt, ausser für Hex- (0xAB) und Oktale (base8 , 0o77) Zahlenangaben

<sup>2)</sup> Ausser `datetime.now().time()`

# Formatierung

```
full = datetime(2022, 12, 9, 14, 59, 21, 1234)
datum = date(2022, 12, 9)
zeit = time(14, 59, 21, 1234)
```

## ISO8601 formatierte Ausgabe

```
print(full)    # 2022-12-09 14:59:21.1234
print(datum)   # 2022-12-09
print(zeit)    # 14:59:21.1234
```

## Benutzer-formatierte Ausgabe

```
full.strftime("%d. %b %Y %H:%M:%S") # 9. Dec 2022 14:59:21
datum.strftime("%d. %B %Y")         # 9. December 2022
zeit.strftime("%H:%M:%S")           # 14:59:21
```

<https://docs.python.org/3/library/datetime.html#strftime-and-strptime-format-codes>

# OS

- Betriebssystem-unabhängig
- Zugriff auf Dateisystem
- Starten von Prozessen

# Methoden & Funktionen

<code>os.getcwd()</code>	liefert das aktuelle Arbeitsverzeichnis (current working dir)
<code>os.chdir(path)</code>	wechselt das aktuelle Arbeitsverzeichnis
<code>os.listdir(path)</code>	listet die Dateien und Unterverzeichnisse auf
<code>os.rename(old, new)</code>	benennt eine Datei oder ein Verzeichnis um
<code>os.mkdir(path)</code>	erzeugt ein neues Verzeichnis
<code>os.rmdir(path)</code>	löscht ein Verzeichnis
<code>os.remove(path)</code>	löscht eine Datei
<code>os.path.exists(path)</code>	prüft, ob eine Datei oder ein Verzeichnis vorhanden ist
<code>os.system(path)</code>	startet ein neuer Prozess

→ path kann jeweils relativ oder absolut sein.

# platform

- Liefert Informationen zum zugrundeliegenden Betriebssystem, zu Prozessor und zur Maschine

# Methoden & Funktionen

```
import platform as pf
```

<code>pf.machine()</code>	Prozessorarchitektur, z. B. AMD64 oder x86
<code>pf.node()</code>	Computername, z. B. DESKTOP-QQNOK6D
<code>pf.platform()</code>	Betriebssystem-Information
<code>pf.python_version()</code>	Pythonversion z. B. 3.11.0
<code>pf.python_implementation()</code>	Pythonimplementation z. B. CPython