

31 Standardbibliothek – Modul datetime

31.1 Beschreibung

- das Modul datetime stellt folgende Klassen (Datentypen) zur Verfügung
 - `date` Datumsangabe mit Jahr, Monat und Tag
 - `time` Zeitangabe mit Stunde, Minute, Sekunde und Mikrosekunde
 - `datetime` Kombination aus Datums- und Zeitangabe
 - `timedelta` Zeitintervall / Zeitspanne
- das Modul datetime wird mit allen Klassen **`import datetime`** in das Programm eingebunden
- jede Klasse kann auch einzeln in das Programm eingebunden werden
 - `date` **`from datetime import date`**
 - `time` **`from datetime import time`**
 - `datetime` **`from datetime import datetime`**
- Datumsvariable definieren `datum = date(yyyy, m, d)`
`yyyy = Jahr` `m = Monatsnummer` `d = Tag`
- Variable für das aktuelle Datum definieren `heute = date.today()`
`yyyy = Jahr` `m = Monatsnummer` `d = Tag`
- Zeitvariable definieren `zeit = time(hh,, mm, ss)`
`hh = Stunde` `mm = Minute` `ss = Sekunde`
- Variable für Zeitstempel `zeitstempel = datetime(yyyy, m, d, hh,, mm, ss)`
- Variable für den aktuellen Zeitstempel definieren `jetzt = datetime.now()`
- Zeitintervalle / Zeitspannen entstehen bei der Datumsberechnung, z. B.
 - Differenz zwischen Datumsangaben `datum1 – datum2`
 - Zeitintervall hinzuzählen `datum + 90 tage`
- Hinweis zum Importieren des Modul datetime mit allen Klassen:
 - Methoden-Aufrufe mit datetime qualifizieren, z.B.
`heute = datetime.date.today()`

31.2 Modul datetime - date-Instanzen

31.2.1 Modul datetime - date-Instanzen erstellen

- Beispiele:

```
# vom Standardmodul datetime nur date einbinden
from datetime import date

# Datumsangabe (Datentyp date)
datum = date(1999,10,10)
print(datum)
```

```
1999-10-10
```

```
# ungültige Datumsangabe ergibt Fehlermeldung
# datum = date(1999, 2, 30)
# ValueError: day is out of range for month
```

```
# aktuelles Datum (Datentyp date)
heute = date.today()
print(heute)
```

```
2022-11-01
```

31.2.2 Modul datetime - Funktionen (Methoden) für date-Instanzen

- auf die einzelnen Teile des Datums (Attribute der date-Instanz) zugreifen
 - auf das Jahr **.year**
 - auf den Monat **.month**
 - auf den Tag **.day**
- Wochentag ermitteln (0 für Montag und 6 für Sonntag) **.weekday()**
- Datum in String umwandeln (im Format wtag mon DD HH:MM:SS YYYY) **.ctime()**
- Datum in einen String (ISO-Format) umwandeln (im Format YYYY-MM-DD) **.isoformat()**
- Datum in einen String gemäß Formatbeschreibung umwandeln **.strftime(format)**
 - Formatbeschreibung (Auszug):
 - Tag als Nummer **%d**
 - Monat als Nummer **%m**
 - Monat als Name **%b**
 - Jahr zweistellig **%y**
 - Jahr vierstellig **%Y**
 - Wochentag als Nummer **%w**
 - Wochentag als Name **%a**

- Beispiele:

```
# vom Standardmodul datetime nur date einbinden
from datetime import date
```

```
# aktuelles Datum (Datentyp date) - Tag, Monat und Jahr einzeln
print(heute.day, heute.month, heute.year)
print(f"Tag: {heute.day}")
print(f"Monat: {heute.month}")
print(f"Jahr: {heute.year}")
print(f"{heute.day}.{heute.month}.{heute.year}")
```

```
1 11 2022
Tag: 1
Monat: 11
Jahr: 2022
1.11.2022
```

```
# Wochentag zu einem Datum ermitteln (0: Montag ... 6: Sonntag)
print(f"Wochentag von {heute}: {heute.weekday()}")
```

```
Wochentag von 2022-11-01: 1
```

```
# Datum (Datentyp date) umwandeln in String aaa bbb DD HH:MM:SS YYYY
str_heute = heute.ctime()
print(str_heute)
```

```
Tue Nov 1 00:00:00 2022
```

```
# Datum (Datentyp date) umwandeln in String im ISO-Format YYYY-MM-DD
iso_heute = heute.isoformat()
print(iso_heute)
```

```
2022-11-01
```

```
# Datum (Datentyp date) druckaufbereiten
print(heute.strftime("%d.%m.%Y"))
```

```
01.11.2022
```

31.3 Modul datetime - time-Instanzen

31.3.1 Modul datetime - time-Instanzen erstellen

- Beispiele:

```
# vom Standardmodul datetime nur time einbinden
from datetime import time

# Uhrzeit (Datentyp time)
zeit = time(12, 30, 10)
print(zeit)
```

```
12:30:10
```

```
# ungültige Zeitangabe ergibt Fehlermeldung
# zeit = time(12, 75, 10)           ValueError: minute must be in 0..59
# print(zeit)
```

31.3.2 Modul datetime - Funktionen (Methoden) für time-Instanzen

- auf die einzelnen Teile der Uhrzeit (Attribute der time-Instanz) zugreifen
 - auf die Stunde **.hour**
 - auf die Minute **.minute**
 - auf die Sekunde **.second**
- Uhrzeit in einen String (ISO-Format) umwandeln (im Format HH:MM:SS) **.isoformat()**
- Uhrzeit in einen String gemäß Formatbeschreibung umwandeln **.strftime(format)**
 - Formatbeschreibung (Auszug):
 - Stunde **%H**
 - Minute **%M**
 - Sekunde **%S**

- Beispiele:

```
# vom Standardmodul datetime nur time einbinden
from datetime import time
```

```
# Datumsangabe (Datentyp time)
zeit = time(12, 30, 10)
print(zeit)
```

```
12 30 10
```

```
# ungültige Zeitangabe ergibt Fehlermeldung
# zeit = time(12, 75, 10)      ValueError: minute must be in 0..59
# print(zeit)
```

```
# Datum (Datentyp time) - Tag, Monat und Jahr einzeln
print(zeit.hour, zeit.minute, zeit.second)
print(f"Stunde: {zeit.hour}")
print(f"Minute: {zeit.minute}")
print(f"Sekunde: {zeit.second}")
print(f"{zeit.hour}:{zeit.minute}")
```

```
Stunde: 12
Minute: 30
Sekunde: 10
12:30
```

```
# Zeit (Datentyp time) umwandeln in String im ISO-Format HH:MM:SS.mmmmmmm
iso_zeit = zeit.isoformat()
print(iso_zeit)
```

```
12:30:10
```

```
# Uhrzeit (Datentyp time) druckaufbereiten
print(zeit.strftime("%H.%M.%S"))
```

```
12.30.10
```

31.4 Modul datetime - datetime-Instanzen

31.4.1 Modul datetime - datetime-Instanzen erstellen

- Beispiele:

```
# vom Standardmodul datetime nur datetime einbinden  
from datetime import datetime
```

```
# Zeitstempel (Datentyp datetime)  
zeitstempel = datetime(2022, 11, 1, 12, 30, 45)  
print(zeitstempel)
```

```
2022-11-01 12:30:45
```

```
# ungültige Angabe ergibt Fehlermeldung  
# zeit = datetime(12, 75, 10) ValueError: month must be in 1..12  
# print(zeit)
```

31.4.2 Modul datetime - Funktionen (Methoden) für datetime-Instanzen

- aktuellen Zeitstempel ermitteln **.now()**
- Wochentag ermitteln (0 für Montag und 6 für Sonntag) **.weekday()**
- auf die einzelnen Teile des Zeitstempels (Attribute der time-Instanz) zugreifen
 - auf das Jahr **.year**
 - auf den Monat **.month**
 - auf den Tag **.day**
 - auf die Stunde **.hour**
 - auf die Minute **.minute**
 - auf die Sekunde **.second**
- Zeitstempel in einen String (ISO-Format) umwandeln **.isoformat()**
(im Format YYYY-MM-DDTHH:MM:SS.mmmmmm)
- Zeitstempel in einen String gemäß Formatbeschreibung umwandeln **.strftime(format)**
 - Formatbeschreibung (Auszug):
 - Tag als Nummer **%d**
 - Monat als Nummer **%m**
 - Monat als Name **%b**
 - Jahr zweistellig **%y**
 - Jahr vierstellig **%Y**
 - Wochentag als Nummer **%w**
 - Wochentag als Name **%a**
 - Stunde **%H**
 - Minute **%M**
 - Sekunde **%S**

- Beispiele:

```
# vom Standardmodul datetime nur datetime einbinden
from datetime import datetime
```

```
# # aktueller Zeitstempel (Datentyp datetime)
jetzt = datetime.now()
print(jetzt)
```

```
2022-11-01 12:30:45.876543
```

```
# Wochentag zu einem Datum ermitteln (0: Montag ... 6: Sonntag)
print(f"Wochentag von {jetzt}: {jetzt.weekday()}")
```

```
Wochentag von 2022-11-01: 1
```



```

# Datum (Datentyp time) - Tag, Monat und Jahr einzeln
print(jetzt.hour, jetzt.minute, jetzt.second,
      jetzt.day, jetzt.month, jetzt.year)
print(f"Tag: {jetzt.day}")
print(f"Monat: {jetzt.month}")
print(f"Jahr: {jetzt.year}")
print(f"Stunde: {jetzt.hour}")
print(f"Minute: {jetzt.minute}")
print(f"Sekunde: {jetzt.second}")
print(f"{jetzt.hour}:{jetzt.minute} am "
      f"{jetzt.day}.{jetzt.month}.{jetzt.year}")

```

```
12 30 45 1 11 2022
```

```
Tag: 1
```

```
Monat: 11
```

```
Jahr: 2022
```

```
Stunde: 12
```

```
Minute: 30
```

```
Sekunde: 45
```

```

# jetzt (Datentyp datetime) umwandeln in String im
# ISO-Format YYYY-MM-DDTHH:MM:SS.mmmmmmm
iso_jetzt = jetzt.isoformat()
print(iso_jetzt)

```

```
2022-11-01T12:30:45.876543
```

```

# Zeitstempel (Datentyp datetime) druckaufbereiten
print(jetzt.strftime("%H.%M.%S"))

```

```
12.30.45
```

31.5 Modul datetime - timedelta-Instanzen

31.5.1 Modul datetime - timedelta-Instanzen erstellen

- entstehen durch die Bildung der Differenz von Zeitstempeln
- Beispiele:

```
# vom Standardmodul datetime nur timedelta einbinden
from datetime import datetime, timedelta

# Zeitstempel (Datentyp datetime)
d1 = timedelta(2022, 11, 1, 12, 30, 45)
d2 = timedelta(2022, 12, 1, 1, 0, 15)
# Zeitstempel (Datentyp datetime)
delta1 = d2 - d1
print(d1)
print(d2)
print(delta1)
```

```
2022-11-01 12:30:45
2022-12-01 01:00:15
29 days, 12:29:30
```

31.5.2 Modul datetime - Funktionen (Methoden) für timedelta-Instanzen

- erzeugen von Zeitintervallen / Zeitspannen **.timedelta(delta)**
 - Angaben in Tagen, Stunden, Minuten, usw.) möglich

- Beispiele:

```
# vom Standardmodul datetime nur timedelta einbinden
from datetime import timedelta
```

```
# Zeitspanne definieren und zu Zeitstempel addieren
print(d2)
woche = timedelta(days=7)
print(d2+woche)
```

```
2022-12-01 01:00:15
2022-12-08 01:00:15
```

```
# Zeitspanne definieren und zu Zeitstempel addieren
zeitspanne = timedelta(days=365, hours=12, minutes=10)
print(d2+zeitspanne)
```

```
2023-12-01 13:10:15
```

Aufgabe

1. Es sind Datumsdifferenzen zu berechnen:
 - 01.01.2020 bis heute
 - 01.07.2022 bis 30.09.2022
 - Beide Datumsangaben über die Konsole einlesen und die Differenz ermitteln
2. Die Laufzeit für die Ausführung einer Schleife ist zu ermitteln:
 - Es ist eine Schleife zu programmieren, die alle Zahlen von 1 bis 1000 addiert, deren Ausführungszeit ist zu ermitteln.
 - Eine weitere Schleife soll programmiert werden und die Zahlen von 1 bis 1.000.000 addiert werden, wieder ist die Ausführungszeit zu ermitteln.
3. Für die Erfassung von Artikeln (z.B. Banane, Orange, Zitrone usw.) und deren Einzelpreis ist eine Endlosschleife zu programmieren; als Abbruchkriterium wird die Eingabe von „q“ bei der Artikeleingabe vorgeschlagen.
Die Artikelbezeichnungen und die Einzelpreise sind zusammen mit einem Zeitstempel in einer Textdatei abzuspeichern.
4. Mit einem weiteren Programm ist diese Datei einzulesen: Für alle Artikel ist die gewünschte Menge über die Konsol-Eingabe anzufordern. Es ist Summe für jeden Artikel zu bilden (Einzelpreis * Menge) und der Gesamtbetrag zu errechnen und auf der Konsole auszugeben.