

Python Kurs 2019/2020

6: Datum, Zeit

Bernhard Mallinger

`b.mallinger [at] gmx.at`

<https://totycro.github.io/python-kurs>

Datetime

- `datetime` enthält Datentypen für ein Datum (`date`), Datum+Uhrzeit (`datetime`) und Zeitdauer (`timedelta`).
- Sinnvolle Operationen wie `<`, `+`, `-` werden unterstützt
- Kümmt sich bei Berechnungen um variable Monatslänge, auch Schalttage

```
>>> import datetime
>>> march_30 = datetime.date(2020, 3, 30)
>>> this_day = datetime.date.today()
>>> march_30 > this_day
True
>>> march_30 - datetime.date(2020, 2, 5)
datetime.timedelta(days=54)
>>> march_30.month # other properties: day, year
3
>>> march_30 + datetime.timedelta(days=4)
datetime.date(2020, 4, 3)
```

Datetime

Achtung: Das Modul und der Datentyp heißen `datetime`.



Datetime

Formatierte Ausgabe mittels [Codes](#) und `strftime`:

```
>>> now = datetime.datetime.now()
>>> now.strftime("%Y-%m-%d")
'2020-06-21'
```

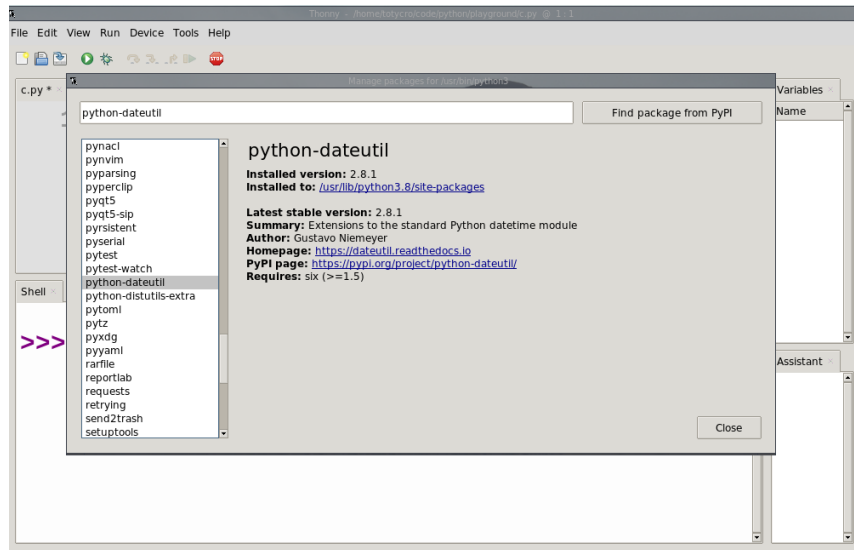
Mit timezone-aware Datum arbeite
(Sommer-/Winterzeitprobleme vermeiden)

```
>>> import pytz
>>> nown.astimezone(pytz.timezone('Europe/Vienna'))
datetime.datetime(2020, 6, 21, 16, 18, 29, 678644, tzinfo=<DstTzInfo 'E
>>> pytz.UTC.localize(datetime.datetime.utcnow())
datetime.datetime(2020, 6, 21, 14, 24, 30, 418818, tzinfo=<UTC>)
```

dateutil

Empfohlene `datetime`-Erweiterung: `python-dateutil`

Ggf. Installation über Thonny: Tools → Manage packages



dateutil

Ermöglicht erweiterte Datumsoperationen

```
>>> from dateutil.relativedelta import relativedelta, MO
>>> datetime.date(2020, 1, 31) + relativedelta(months=1)
datetime.date(2020, 2, 29)
>>> datetime.date(2020, 1, 31) + relativedelta(months=1, day=1)
datetime.date(2020, 2, 1) # der erste Tag am nächsten Monat
>>> datetime.date(2020, 1, 31) + relativedelta(weekday=MO)
datetime.date(2020, 2, 3) # nächster Montag
```

Alter berechnen:

```
>>> datetime.date.today() - datetime.date(1989, 6, 13)
datetime.timedelta(days=11218)
>>> relativedelta(datetime.date.today(), datetime.date(1989, 6, 13))
relativedelta(years=+30, months=+8, days=+16)
```

dateutil

dateutil stellt auch [Parser](#) zur Verfügung

```
>>> from dateutil.parser import parse
>>> parse("2020-01-05")
datetime.datetime(2020, 1, 5, 0, 0)
>>> parse("5.1.2020") # Achtung!
datetime.datetime(2020, 5, 1, 0, 0)
>>> parse("5.1.2020", dayfirst=True)
datetime.datetime(2020, 1, 5, 0, 0)
```

Kassensystem 2

AUFGABE

Erweitere das Kassensystem:

Am Dienstag ist Studierendentag. Alle Studierenden bekommen 15% zusätzlichen Rabatt. Das System hat eine Liste von Studierenden.

Am Donnerstag ist Seniorentag. Alle über 65 erhalten 65% zusätzlichen Rabatt. Im System gibt es ein Dictionary, das die Geburtstage aller Kunden gespeichert hat.