

Datum a čas

PVA2 Programování a vývoj aplikací

Obsah

1. Obsah
2. Datum a čas
3. Modul `datetime`
4. Třída `date`
5. Třída `date` - příklad
6. Třída `time`
7. Třída `time` - příklad
8. Timestamp
9. Třída `datetime`
10. Třída `datetime` - příklad
11. Výpočty s datem a časem
12. Základní operace s datem a časem
13. Třída `timedelta`
14. Výpočty s třídou `timedelta`
15. Příklad
16. Formátování datumu a času
17. Konverze na text
18. Konverze z textu
19. Světový čas
20. Časové zóny

Datum a čas

- V pythonu nejsou datum a čas samostatným datovým typem.
- Pro práci s datem a časem používáme modul `datetime`
- Modul je nativně zabudován, není nutná externí instalace
- Třídy modulu poskytují řadu funkcí pro práci s daty, časy a časovými intervaly. Datum a časový údaj jsou v jazyce Python objektem, takže při manipulaci s nimi vlastně manipulujete s objekty, a nikoli s řetězci nebo timestampem.

Modul `datetime`

- Modul `datetime` obsahuje třídy pro manipulaci s datem a časem.
- Třídy jsou:
 - `date` - pro manipulaci s datem
 - `time` - pro manipulaci s časem
 - `datetime` - pro manipulaci s datem a časem
 - `timedelta` - pro manipulaci s časovými intervaly
 - `tzinfo` - pro manipulaci s časovými zónami

Třída `date`

- V informatice pracujeme standardně s datem ve formátu `YYYY-MM-DD` např. `2024-12-24`.
- Třída `date` reprezentuje datum.
- Datum je reprezentováno pomocí tří atributů:
 - `year` - rok
 - `month` - měsíc
 - `day` - den
- Datum je neměnný objekt, tzn. po vytvoření nelze změnit jeho hodnoty.

Třída `date` - příklad

```
1 # Nahrání modulu a import třídy
2 from datetime import date
3
4 # Vytvoření instance třídy date
5 d = date(2024, 12, 24)
6 print("Zadané datum: " + str(d)) # Zadané datum: 2024-12-24
7
8 # Aktuální datum
9 aktualniDatum = date.today()
10 print("Aktuální datum: " + str( aktualniDatum) )
11
12 # Parciální informace o datu
13 print("Rok: " + str(aktualniDatum.year))
14 print("Měsíc: " + str(aktualniDatum.month))
15 print("Den: " + str(aktualniDatum.day))
```

Třída `time`

- Třída `time` reprezentuje čas.
- Čas je reprezentován pomocí tří atributů:
 - `hour` - hodina
 - `minute` - minuta
 - `second` - sekunda
 - `microsecond` - mikrosekunda
- Čas je neměnný objekt, tzn. po vytvoření nelze změnit jeho hodnoty.

Třída `time` - příklad

```
1 # Nahrání modulu a import třídy
2 # datetime nahráváme kvůli získání aktuálního času
3 from datetime import time, datetime
4
5 # Vytvoření instance třídy time
6 t = time(12, 30, 45, 100000)
7 print("Zadaný čas: " + str(t)) # Zadaný čas: 12:30:45.100000
8
9 # Aktuální čas, pro získání aktuálního času používáme modul datetime
10 # Funkce datetime.now() vrací aktuální datum a čas a pomocí metody time() získáme pouze čas
11 aktualniDatumCas = datetime.now()
12 aktualniCas = aktualniDatumCas.time()
13 print("Aktuální čas: " + str(aktualniCas))
```

```
1 # Nebo lépe než pomocí aktualniDatumCas a aktualniCas
2 aktualniCas = datetime.now().time()
3 print("Aktuální čas: " + str(aktualniCas))
```


Timestamp

- Timestamp je časový údaj, který udává počet sekund, které uplynuly od 1. ledna 1970.
- V Pythonu můžeme získat timestamp pomocí funkce `time()`.
- Timestamp můžeme převést na datum a čas pomocí funkce `fromtimestamp()`.

```
1  from datetime import datetime
2
3  # Získání timestampu z aktuálního času
4  aktualniCas = datetime.now()
5  timestamp = aktualniCas.timestamp()
6
7  print("Timestamp: " + str(timestamp)) # Timestamp: 1740000000.0
8
9  # Převod timestampu na datum a čas
10 dt = datetime.fromtimestamp(timestamp)
11 print("Datum a čas: " + str(dt)) # Datum a čas: 2025-02-19 21:20:00
```

Třída `datetime`

- Třída `datetime` reprezentuje datum a čas.
- Datum a čas jsou reprezentovány pomocí tří atributů:
 - `year` - rok
 - `month` - měsíc
 - `day` - den
 - `hour` - hodina
 - `minute` - minuta
 - `second` - sekunda
 - `microsecond` - mikrosekunda

Třída `datetime` - příklad

```
1 # Nahrání modulu a import třídy
2 from datetime import datetime
3
4 # Vytvoření instance třídy datetime
5 dt = datetime(2024, 12, 24, 12, 30, 45, 100000)
6 print("Zadané datum a čas: " + str(dt)) # Zadané datum a čas: 2024-12-24 12:30:45.100000
7
8 # Aktuální datum a čas
9 aktualniDatumCas = datetime.now()
10 print("Aktuální datum a čas: " + str(aktualniDatumCas))
```

Výpočty s datem a časem



Základní operace s datem a časem

- Pro matematické operace s datem a časem můžeme použít operátory `+`, `-`, `*`, `/`.
- Podmínkou pro operace je, že oba objekty musí být stejného datového typu.

```
1  from datetime import datetime
2
3  dt = datetime(2024, 12, 24)
4
5  # Určení dne v týdnu pro zadané datum
6  dt.weekday() # 0-pondělí, 1-úterý, 2-středa, 3-čtvrtek, 4-pátek, 5-sobota, 6-neděle
7  dt.isoweekday() # 1-pondělí, 2-úterý, 3-středa, 4-čtvrtek, 5-pátek, 6-sobota, 7-neděle
8
9  # Rozdíl dvou dat
10 d1 = datetime(2024, 12, 24, 10, 00, 00)
11 d2 = datetime(2024, 12, 31, 22, 00, 00)
12
13 rozdil = d2 - d1
14 print("Rozdíl dvou dat: "+str(rozdil)) # 7 days, 12:00:00
15
16 # Porovnání dvou dat
17 d1 < d2 # True
18 d1 > d2 # False
```

Třída `timedelta`

- Třída `timedelta` reprezentuje časový interval.
- Časový interval je reprezentován pomocí tří atributů:
 - `days` - dny
 - `seconds` - sekundy
 - `microseconds` - mikrosekundy

```
1 from datetime import timedelta
2
3 td = timedelta(days=5, seconds=3600)
4
5 print("Zadaný časový interval: " + str(td)) # Zadaný časový interval: 5 days, 1:00:00
```

Výpočty s třídou `timedelta`

- Třída `timedelta` umožňuje provádět výpočty s časovými intervaly.
- Výpočty mohou být sčítání, odčítání, násobení a dělení.
- Výsledkem výpočtu je nový objekt třídy `timedelta`.
- Výsledek výpočtu je vždy nový objekt, původní objekt zůstává nezměněný.

```
1 from datetime import timedelta
2
3 td1 = timedelta(days=5, seconds=3600)
4 td2 = timedelta(days=2, seconds=7200)
5
6 print("Sčítání: " + str(td1 + td2)) # Sčítání: 7 days, 3:00:00
7 print("Odčítání: " + str(td1 - td2)) # Odčítání: 3 days, 1:00:00
8 print("Násobení: " + str(td1 * 2)) # Násobení: 10 days, 2:00:00
9 print("Dělení: " + str(td1 / 2)) # Dělení: 2 days, 12:30:00
```

Příklad

```
1 from datetime import datetime, timedelta
2
3 datum = datetime(2024, 12, 24, 10, 00, 00)
4 print("Zadané datum: ")
5 print(datum)
6
7 dnyDoBudoucnosti = 7
8 vysledek = datum + timedelta(days=dnyDoBudoucnosti, hours=12)
9 print("Nové budoucí datum: ")
10 print(vysledek) # 2024-12-31 22:00:00
11
12 dnyDoMinulosti = 10
13 vysledek = datum - timedelta(days=dnyDoMinulosti)
14 print("Nové datum z minulosti: ")
15 print(vysledek) # 2024-12-14 10:00:00
```


Formátování datumu a času



Konverze na text

- Pro konverzi objektů tříd `date`, `time` a `datetime` na textový řetězec používáme metodu `strftime()`.
- Metoda `strftime()` přijímá jako parametr formátovací řetězec, který určuje, jak bude datum a čas zobrazen.
- Formátovací řetězec může obsahovat speciální znaky, které se nahradí odpovídajícími hodnotami.
- Např. `%Y` - rok, `%m` - měsíc, `%d` - den, `%H` - hodina, `%M` - minuta, `%S` - sekunda.

```
1 from datetime import datetime
2
3 dt = datetime(2024, 12, 24, 10, 00, 00)
4 print(dt.strftime("%Y-%m-%d %H:%M:%S")) # 2024-12-24 10:00:00
5 print(dt.strftime("%d.%m.%Y %H:%M")) # 24.12.2024 10:00
```

Konverze z textu

- Pro konverzi textového řetězce na objekt tříd `date`, `time` a `datetime` používáme metodu `strptime()`.
- Metoda `strptime()` přijímá jako parametr textový řetězec a formátovací řetězec, který určuje, jak bude datum a čas interpretován.
- Formátovací řetězec musí odpovídat formátu textového řetězce.

```
1 from datetime import datetime
2
3 datum = "2024-12-24"
4 dt = datetime.strptime(datum, "%Y-%m-%d")
5 print(dt) # 2024-12-24 00:00:00
6
7 cas = "10:00:00"
8 t = datetime.strptime(cas, "%H:%M:%S")
9 print(t) # 1900-01-01 10:00:00
```

Světový čas



Časové zóny

- Pro práci s časovými zónami používáme modul `pytz`.
- Modul `pytz` obsahuje databázi časových zón, které můžeme použít pro konverzi času mezi různými zónami.
- Pro konverzi času mezi zónami používáme metodu `astimezone()`.

```
1 from datetime import datetime
2
3 # Nahrání modulu a import třídy
4 from pytz import timezone
5
6 # Vytvoření instance třídy datetime
7 dt = datetime(2024, 12, 24, 12, 30, 45, 100000)
8
9 # Převod na časovou zónu
10 dt = dt.astimezone(timezone('US/Pacific'))
11 print("V časové zóně US/Pacific je: " + str(dt))
```

```
1 from datetime import datetime
2 from pytz import timezone
3
4 format = "%d-%m-%Y %H:%M:%S %Z%z"
5
6 # Aktuální čas v UTC
7 aktualniCasUTC = datetime.now(timezone('UTC'))
8 print(now_utc.strftime(format))
9
10 casoveZony = [ 'Europe/Prague',
11               'Europe/Kiev',
12               'America/New_York',
13               'Australia/Sydney',]
14
15 for tzona in casoveZony:
16     # Převod na jednotlivé zóny
17     aktualniCasSvet = aktualniCasUTC.astimezone(timezone(tzona))
18     print(tzona, aktualniCasSvet.strftime(format))
```

Děkuji za pozornost

Otázky?

Repository / Prezentace