Реферат

Тема: Робота з датою і часом. Модуль datetime.

Основний функціонал для роботи з датами та часом зосереджений у модулі datetime у вигляді наступних класів:

* Date
* Time
* datetime

Клас date

Для роботи з датами скористаємося класом date, визначеним у модулі datetime. Для створення об'єкта date ми можемо використовувати конструктор date, який послідовно приймає три параметри: рік, місяць та день.

date(year, month, day)

Наприклад, створимо якусь дату:

import datetime

yesterday = datetime.date(2017,5, 2)

print(yesterday)     # 2017-05-02

Якщо необхідно отримати поточну дату, можна скористатися методом today():

from datetime import date

today = date.today()

print(today)     # 2017-05-03

print("{}.{}.{}".format(today.day, today.month, today.year))     # 2.5.2017

За допомогою властивостей day, month, year можна отримати відповідно день, місяць та рік  
Клас time  
  
За роботу з часом відповідає клас часу. Використовуючи його конструктор, можна створити об'єкт часу:  
1  
  
time([hour] [, min] [, sec] [, microsec])  
  
Конструктор послідовно приймає години, хвилини, секунди та мікросекунди. Усі параметри необов'язкові, і якщо ми якийсь параметр не передамо, то відповідне значення ініціалізуватиметься нулем.  
  
from datetime import time  
  
current\_time = time()  
print(current\_time) # 00:00:00  
  
current\_time = time(16, 25)  
print(current\_time) # 16:25:00  
  
current\_time = time(16, 25, 45)  
print(current\_time) # 16:25:45  
Клас datetime

Клас datetime з однойменного модуля поєднує можливості роботи з датою та часом. Для створення об'єкта datetime можна використовувати наступний конструктор:  
  
datetime(year, month, day [, hour] [, min] [, sec] [, microsec])  
  
Перші три параметри, що становлять рік, місяць і день, є обов'язковими. Інші необов'язкові, і якщо ми не вкажемо їм значення, то за замовчуванням вони ініціалізуються нулем.  
  
from datetime import datetime  
  
deadline = datetime(2017, 5, 10)  
print(deadline) # 2017-05-10 00:00:00  
  
deadline = datetime(2017, 5, 10, 4, 30)  
print(deadline) # 2017-05-10 04:30:00  
  
Для отримання поточної дати та часу можна викликати метод now():  
  
from datetime import datetime  
  
now = datetime.now()  
print(now) # 2017-05-03 11:18:56.239443  
  
print("{}.{}.{} {}:{}".format(now.day, now.month, now.year, now.hour, now.minute)) # 3.5.2017 11:21  
  
print(now.date())  
print(now.time())  
  
За допомогою властивостей day, month, year, hour, minute, second можна отримати окремі значення дати та часу. А через методи date() та time() можна отримати окремо дату та час відповідно.  
Перетворення з рядка на дату  
  
З функціональності класу datetime слід зазначити метод strptime(), який дозволяє розпарсувати рядок та перетворити його на дату. Цей метод приймає два параметри:  
  
strptime(str, format)  
  
Перший параметр str представляє рядкове визначення дати та часу, а другий параметр - формат, який визначає, як різні частини дати та часу розташовані у цьому рядку.  
  
Для визначення формату ми можемо використовувати такі коди:  
  
%d: день місяця у вигляді числа  
  
%m: порядковий номер місяця  
  
%y: рік у вигляді 2-х чисел  
  
%Y: рік у вигляді 4-х чисел  
  
%H: година у 24-годинному форматі  
  
%M: хвилина  
  
%S: секунда  
  
Застосуємо різні формати:  
  
from datetime import datetime  
deadline = datetime.strptime("22/05/2017", "%d/%m/%Y")  
print(deadline) # 2017-05-22 00:00:00  
  
deadline = datetime.strptime("22/05/2017 12:30", "%d/%m/%Y %H:%M")  
print(deadline) # 2017-05-22 12:30:00  
  
deadline = datetime.strptime("05-22-2017 12:30", "%m-%d-%Y %H:%M")  
print(deadline) # 2017-05-22 12:30:00

Фотографування дат та часу  
  
Для форматування об'єктів date і time в обох класах передбачений метод strftime(format). Цей метод приймає лише один параметр, який вказує на формат, який потрібно перетворити на дату або час.  
  
Для визначення формату ми можемо використовувати один із наступних кодів форматування:  
  
%a: абревіатура дня тижня. Наприклад, Wed – від слова Wednesday (за замовчуванням використовуються англійські найменування)  
  
%A: день тижня повністю, наприклад, Wednesday

%b: абревіатура назви місяця. Наприклад, Oct (скорочення від October)  
%B: назва місяця повністю, наприклад, October

%d: день місяця, доповнений нулем, наприклад, 01  
%m: номер місяця, доповнений нулем, наприклад, 05  
%y: рік у вигляді 2-х чисел  
%Y: рік у вигляді 4-х чисел  
%H: година у 24-годинному форматі, наприклад, 13  
%I: година у 12-годинному форматі, наприклад, 01  
%M: хвилина  
%S: секунда  
%f: мікросекунда  
%p: вказівник AM/PM  
%c: дата та час, відформатовані під поточну локаль  
%x: дата, відформатована під поточну локаль  
%X: час, форматований під поточну локаль  
Використовуємо різні формати:  
from datetime import datetime  
now = datetime.now()  
print(now.strftime("%Y-%m-%d")) # 2017-05-03  
print(now.strftime("%d/%m/%Y")) # 03/05/2017  
print(now.strftime("%d/%m/%y")) # 03/05/17  
print(now.strftime("%d %B %Y (%A)")) # 03 May 2017 (Wednesday)  
print(now.strftime("%d/%m/%y %I:%M")) # 03/05/17 01:36  
  
При виведенні назв місяців та днів тижня за промовчанням використовуються англійські найменування. Якщо ми хочемо використовувати поточну локаль, але ми можемо її заздалегідь встановити за допомогою модуля locale:  
  
from datetime import datetime  
import locale  
locale.setlocale(locale.LC\_ALL, "")  
  
now = datetime.now()  
print(now.strftime("%d %B %Y (%A)")) # 03 Травня 2017 (середа)  
Складання та віднімання дат і часу  
  
Нерідко при роботі з датами виникає необхідність додати до будь-якої дати певний проміжок часу або, навпаки, відняти певний період. І спеціально для таких операцій у модулі datetime визначено клас timedelta. Фактично цей клас визначає певний період.  
  
Для визначення проміжку часу можна використовувати конструктор timedelta:  
1  
  
timedelta([days] [, seconds] [, microseconds] [, milliseconds] [, minutes] [, hours] [, weeks])  
  
У конструктор ми послідовно передаємо дні, секунди, мікросекунди, мілісекунди, хвилини, години та тижні.  
  
Визначимо кілька періодів:  
  
from datetime import timedelta  
  
three\_hours = timedelta(hours=3)  
print(three\_hours) # 3:00:00  
three\_hours\_thirty\_minutes = timedelta(hours=3, minutes=30) # 3:30:00  
  
two\_days = timedelta(2) # 2 days, 0:00:00  
  
two\_days\_three\_hours\_thirty\_minutes = timedelta(days=2, hours=3, minutes=30) # 2 days, 3:30:00  
  
Використовуючи timedelta, ми можемо складати або віднімати дати. Наприклад, отримаємо дату, яка буде за два дні:

1. Які класи є в модулі datetime?
2. Які три параметра приймає клас date?
3. Що таке клас date?
4. Що таке клас datetime?
5. Як отримати теперішній місяць за допомогою класа date?
6. Які властивості є в класа date?
7. Як отримати теперішій день за допомогою класа date?
8. Як отримати теперішню дату за допомогою класа date?
9. Як отримати теперішній рік за допомогою класа date?
10. Які 4 параметра приймає клас time?
11. Як отримати теперішню годину за допомогою класа time?
12. Як отримати теперішній час за допомогою класа time?
13. Як отримати теперішн секунди за допомогою класа time?
14. Як отримати теперішні мікросекунди за допомогою класа time?
15. Які 10 параметрів приймає конструктор класа datetime?
16. Яким класом можна додавати і віднімати дати?
17. Яким методом з класа datetime можна відформатувати дату?
18. Як отримати теперішній час за допомогою класа datetime?
19. Як отримати теперішню дату за допомогою класа datetime­?
20. Яким методом с класа datetime можна перетворити строку в дату?

Джерела:

<https://metanit.com/python/tutorial/8.1.php>