

Урок №49 Робота з додатковими бібліотеками мови програмування.

Модуль `datetime`

Онлайн середовище: [Trinket](#)

Завдання 1. Отримання поточної місцевої дати та створення об'єктів дати

Один із класів, що надаються модулем **`datetime`**, - це клас з ім'ям **`date`**. Об'єкти цього класу є датою, що складається з року, місяця і дня.

1. Введіть код у редакторі, щоб побачити, як він виглядає на практиці, і отримайте поточну місцеву дату за допомогою методу **`today`**.

```
1 from datetime import date
2
3 today = date.today()
4
5 print("Today:", today)
6 print("Year:", today.year)
7 print("Month:", today.month)
8 print("Day:", today.day)
```

2. Запустіть код, щоб дізнатися, що відбувається.

Метод **`today`** повертає об'єкт **`date`**, що представляє поточну місцеву дату. Зверніть увагу, що об'єкт **`date`** має три атрибути: рік, місяць та день.

Будьте обережні, тому що ці атрибути доступні лише для читання.

3. Щоб створити об'єкт **`date`**, Ви повинні передати параметри *year*, *month* і *day* наступним чином:

```
1 from datetime import date
2
3 my_date = date(2025, 11, 14)
4 print(my_date)
```

Запустіть код, щоб побачити, що вийде.

Під час створення об'єкта **`date`** пам'ятайте про такі обмеження:

Параметр	Обмеження
year	Параметр year повинен бути більшим або дорівнює 1 (константа MINYEAR) і менше або дорівнює 9999 (константа MAXYEAR).
month	Параметр month повинен бути більшим або дорівнює 1 і менше

	або дорівнює 12.
day	Параметр day повинен бути більшим або дорівнює 1 і менше або дорівнює останньому дню цього місяця та року.

Завдання 2. Створення об'єкта дати з позначки часу

Клас **date** дає нам можливість створити об'єкт **date** із позначки часу.

У Unix позначка часу виражає кількість секунд з 1 січня 1970, 00:00:00 (UTC). Ця дата називається епохою Unix, тому що саме з неї розпочався відлік часу у системах Unix.

Тимчасова позначка – це різниця між певною датою (включаючи час) та 1 січня 1970 року, 00:00:00 (світовий координований час), виражена в секундах.

Щоб створити об'єкт дати з позначки часу, ми повинні передати позначку часу Unix методом від **timestamp**.

Для цього ми можемо використовувати модуль **time**, який надає функції, пов'язані з часом. Одна з них - функція під назвою **time()**, яка повертає кількість секунд з 1 січня 1970 до поточного моменту у вигляді числа з плаваючою комою.

```

1 from datetime import date
2 import time
3
4 timestamp = time.time()
5 print("Timestamp:", timestamp)
6
7 d = date.fromtimestamp(timestamp)
8 print("Date:", d)
```

Запустіть код, щоб побачити результат.

Якщо Ви запустите код кілька разів, побачите, як збільшується сама позначка часу. Варто додати, що результат функції часу залежить від платформи, оскільки в системах Unix і Windows високосні секунди не враховуються.

Завдання 3. Метод **replace()**

Іноді може знадобитися замінити рік, місяць або день іншим значенням. Ви не можете зробити це з атрибутами року, місяця та дня, тому що вони доступні лише для читання. У цьому випадку Ви можете використовувати метод із ім'ям **replace**.

Запустіть код у редакторі:

```
1 from datetime import date
2
3 d = date(1991, 2, 5)
4 print(d)
5
6 d = d.replace(year=1992, month=1, day=16)
7 print(d)
```

Результат:

1991-02-05

1992-01-16

Параметри рік, місяць та день є не обов'язковими. Ви можете передати лише один параметр методу **replace**, наприклад, **year**, або всі три, як у прикладі.

Метод **replace** повертає змінений об'єкт **date**, тому Ви не повинні забути присвоїти його будь-якій змінній.

Завдання 4. Який день тижня?

Одним із найбільш корисних методів, що спрощують роботу з датами, є метод **weekday**. Він повертає день тижня як ціле число, де 0 – понеділок, а 6 – неділя. Запустіть код у редакторі:

```
1 from datetime import date
2
3 d = date(2025, 11, 14)
4 print(d.weekday())
```

Результат: 4

Клас **date** має аналогічний метод під назвою **isoweekday**, який також повертає день тижня у вигляді цілого числа, але 1 – це понеділок, а 7 – неділя:

```
1 from datetime import date
2
3 d = date(2025, 11, 14)
4 print(d.isoweekday())
```

Результат: 5

Завдання 5. Створення об'єктів **time**

Ви вже знаєте, як уявити дату за допомогою об'єкта **date**. У модулі **datetime** також є клас, що дозволяє відображати час. Чи можете ви вгадати його назву? Так, він називається **time**:

time(hour, minute, second, microsecond, tzinfo, fold)

Конструктор класу **time** приймає такі необов'язкові параметри:

Параметр	Обмеження
hour	Параметр hour повинен бути більшим або дорівнює 0 і менше 23.
minute	Параметр minute повинен бути більшим або дорівнює 0 і менше 59.
second	Параметр second повинен бути більшим або дорівнює 0 і менше 59.
microsecond	Параметр microsecond повинен бути більшим або дорівнює 0 і менше 1000000.
tzinfo	Параметр tzinfo має бути об'єктом підкласу tzinfo або None (за замовчуванням).
fold	Параметр fold повинен бути 0 або 1 (за замовчуванням 0).

Параметр **tzinfo** пов'язаний із часовими поясами, а параметр **fold** пов'язаний із реальним часом.

Давайте подивимося, як практично створити об'єкт **time**.

Запустіть код у редакторі.

```

1  from datetime import time
2
3  t = time(14, 53, 20, 1)
4
5  print("Time:", t)
6  print("Hour:", t.hour)
7  print("Minute:", t.minute)
8  print("Second:", t.second)
9  print("Microsecond:", t.microsecond)
```

Результат:

Time: 14:53:20.000001

Hour: 14

Minute: 53

Second: 20

Microsecond: 1

У цьому прикладі ми передали конструктору класу чотири параметри: hour, minute, second та microsecond. До кожного можна отримати доступ за допомогою атрибутів класу.

Модуль calendar

Онлайн середовище: <https://trinket.io/library/trinkets/create?lang=python3>

Завдання 6. Ваш перший календар

Ви розпочнете свою пригоду з модуля `calendar` із простою функцією під назвою `calendar`, яка дозволяє відображати календар на весь рік. Давайте подивимося, як використовувати його для відображення календаря на 2025 рік.

Запустіть код у редакторі та подивіться, що станеться.

```
1 import calendar
2 print(calendar.calendar(2025))
```

Завдання 7. Календар на певний місяць

У модулі `calendar` є функція `month`, що дозволяє відображати календар на певний місяць. Його використання дійсно просте, Вам просто потрібно вказати рік і місяць – подивіться код у редакторі.

```
1 import calendar
2 print(calendar.month(2025, 11))
```

На прикладі відображається календар на листопад 2025 року.

Завдання 8. Функція weekday()

Ще одна корисна функція, що надається модулем `calendar` - це функція з ім'ям `weekday`, яка повертає день тижня у вигляді цілого числа для заданого року, місяця і дня. Подивимося практично.

Запустіть код у редакторі, щоб перевірити день тижня, що випадає на 25 грудня 2025 року.

```
1 import calendar
2 print(calendar.weekday(2025, 12, 25))
```

Результат: 3

Функція `weekday` повертає 3, що означає, що 25 грудня 2025 р. – четвер.

Завдання 9. Як перевірити, чи рік високосний?

Модуль `calendar` надає дві корисні функції для перевірки того, чи є роки високосними.

Перший, званий `isleap`, повертає `True`, якщо пройшов високосний рік, або `False` інакше. Другий, званий `leapdays`, повертає кількість високосних років у заданому діапазоні.

Запустіть код у редакторі.

```
1 import calendar
2
3 print(calendar.isleap(2025))
4 print(calendar.leapdays(2015, 2025))
```

Результат:

False

3

У прикладі ми отримуємо результат 3, тому що в період з 2015 по 2025 рік лише три високосні роки (примітка: 2025 рік не включений). Це 2016, 2020 та 2024 роки.

Модуль Turtle

Онлайн середовище: [Trinket](https://trinket.io)

Для того, щоб програми малювання працювали завжди повинні бути рядок:

from turtle import *

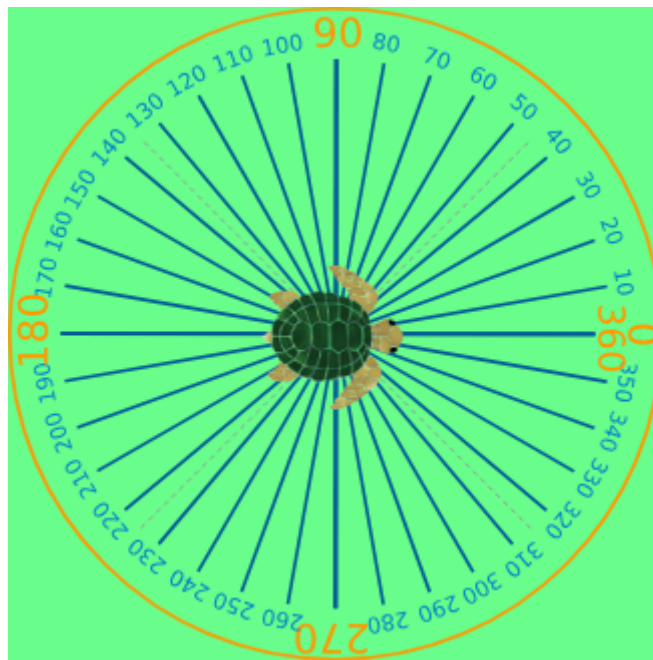
Перший рядок підключає бібліотеку малювання. Ось основні команди малювання:

- **forward (distance)** Проповзти вперед на distance пікселів;
- **backward (distance)** Проповзти назад на distance пікселів;
- **right (angle)** Повернутися направо на angle градусів;
- **left (angle)** Повернутися наліво на angle градусів;

Кути обчислюються в градусах. Ось це коло допоможе з цим розібратися у поворотах. В центрі кола стоїть наша пайтон-черепашка. Щоб їй повернутися в інший бік – треба прописати поворот на 180 градусів (не важливо вправо чи вліво).

А щоб піти вгору – повертаємо вліво на 90 градусів, або вправо на 270 (другий варіант складніший)

Вниз – повертаємося вправо на 90 або вліво на 270.



1: Почнемо малювання з блискавки. :

```
1  from turtle import *
2
3  left(45)
4  forward(100)
5  left(45)
6  back(100)
7  right(45)
8  forward(100)
```



2: Спробуйте намалювати квадрат.

3: А тепер прямокутник.

Щоб курсор виглядав веселіше, можна зробити його у вигляді черепашки
Для цього відразу після підключення бібліотеки треба написати:

```
shape ("turtle")
```

Ще курсори можуть бути: "arrow", "turtle", "circle", "square", "triangle", "classic". Можете вибрати той, який подобається.

4: Змінити вигляд курсора на той, який більше до душі

```
width (3) - товщина лінії 3 пікселя.
```

Задати колір лінії можна командою color, в дужках вказується назва кольору англійською.

```
color('red')
```

```
color("blue")
```

Важливо! Назва кольору **ОБОВ'ЯЗКОВО** пишеться в одинарних або подвійних лапках.

5: Додайте кольорів вашим рисункам

6: Що намалює черепашка, якщо запустити цей код?

```
from turtle import *  
color ("green")  
width (4)  
forward(60)  
up()  
forward(30)  
down()  
  
forward(60)  
up()  
forward(30)  
down()
```