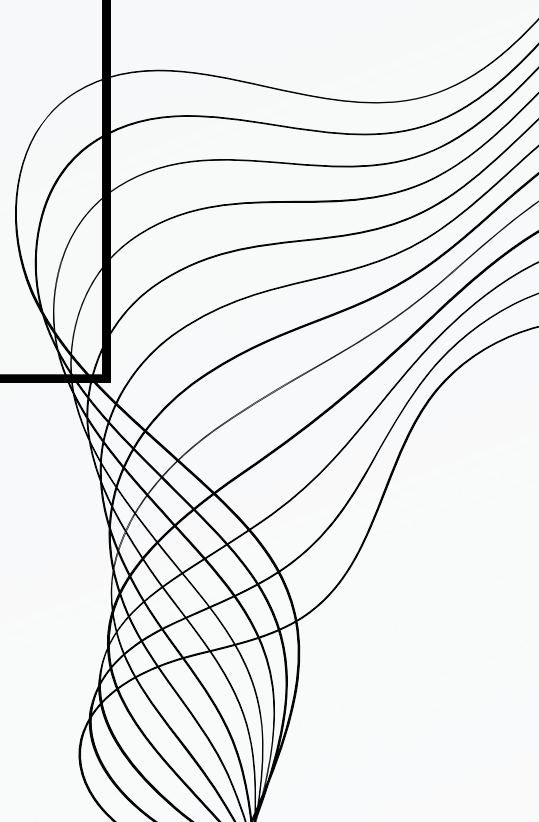




ВВЕДЕННЯ ДО ТЕОРІЇ ХМАРНИХ ОБЧИСЛЕНЬ

**ЛЕКЦІЯ 3. ФОРМАТУВАННЯ ДАНИХ ПРО ЧАС ТА ДАТУ,
ЧИТАННЯ ТА ЗАПИС ФАЙЛІВ В PYTHON. ВЗАЄМОДІЯ ІЗ
ЗОВНІШНІМИ ДОДАТКАМИ**



AGENDA

01

ФОРМАТУВАННЯ ДАНИХ ПРО ЧАС ТА ДАТУ У PYTHON

02

ЧИТАННЯ ТА ЗАПИС ФАЙЛІВ В PYTHON

03

ВЗАЄМОДІЯ ІЗ ЗОВНІШНІМИ ДОДАТКАМИ

3.1. ФОРМАТУВАННЯ ДАНИХ ПРО ЧАС ТА ДАТУ У PYTHON



Модуль `datetime` в мові програмування Python використовується для оброблення даних про час та дату



Модуль `datetime` включений до стандартної бібліотеки Python, але його необхідно імпортувати для використання в коді



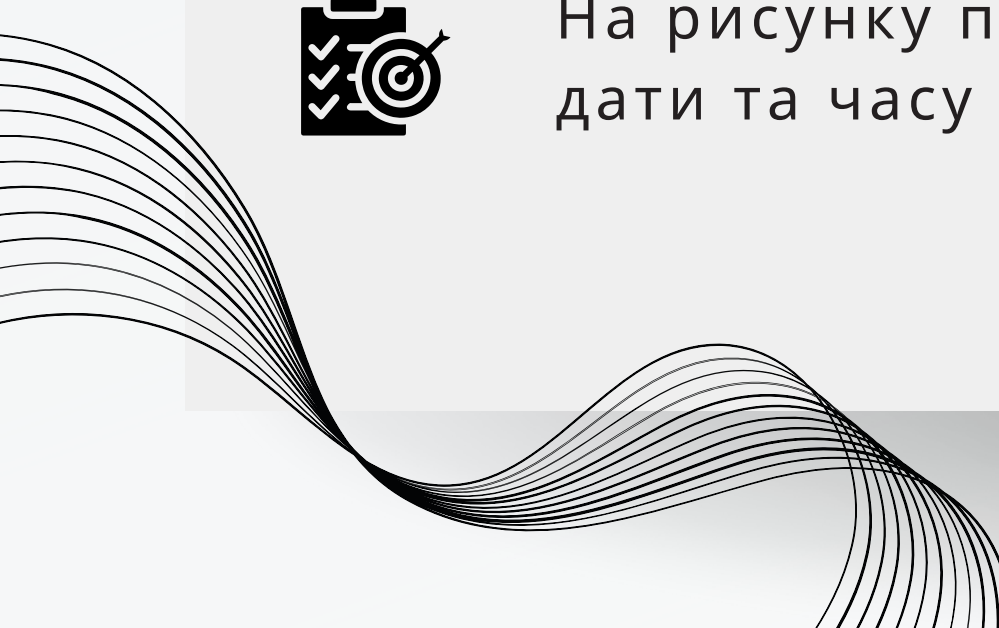
Модуль `datetime` орієнтований на об'єктно-орієнтовану парадигму програмування та містить класи для роботи з часом і датою



Метод стріпінгу використовує серію кодів форматування або директив як своїх параметрів



На рисунку показаний код Python, який використовує модуль `datetime` для подання дати та часу у форматі, який зазвичай використовується у США



ЗАСТОСУВАННЯ ДЕЯКИХ ОСНОВНИХ ОБ'ЄКТІВ І МЕТОДІВ, І ВКЛЮЧЕНИЙ В МОДУЛЬ DATETIME

Datetime Module Terminology

Term	Example	Use
module	<code>datetime</code>	<code>import datetime as dt</code>
class	<code>time</code> <code>date</code> <code>datetime, etc.</code>	<code>dt.time</code> <code>dt.date</code> <code>dt.datetime</code>
object	Variables <code>t</code> , <code>d</code> , and <code>dateAndTime</code>	<code>t = dt.time(12,31,00)</code> <code>d = dt.date(1972,12,31)</code> <code>dateAndTime = dt.datetime.now()</code>
method	<code>strftime()</code> <code>weekday()</code> <code>isoformat</code>	<code>t.strftime(" %H:%M:%S")</code> <code>d.weekday()</code> <code>dateAndTime.isoformat()</code>

СПИСОК КОДІВ ФОРМАТУВАННЯ МОДУЛЯ DATETIME

Directive	Meaning
%a	Locale's abbreviated weekday name
%A	Locale's full weekday name
%b	Locale's abbreviated month name
%B	Locale's full month name
%c	Locale's appropriate date and time representation
%d	Day of the month as a number [01,31]
%H	Hour (24-hour clock) as a number [00,23]
%I	Hour (12-hour clock) as a number [01,12]
%j	Day of the year as a number [001,366]
%m	Month as a number [01,12]
%M	Minute as a number [00,59]
%p	Locale's equivalent of either AM or PM
%S	Second as a number [00,61]
%w	Weekday as a number [00,61]
%W	Week number of the year (Monday as the first day of the week) as a number [00,53]. All days in a new year preceding the first Monday are considered to be in week 0.
%y	Year without century as a number [00,99]
%Y	Year with century as a number
%Z	Time zone name (no characters if no time exists)

ВИКОРИСТАННЯ МОДУЛЯ DATETIME

```
#load the datetime module as dt  
import datetime as dt  
  
#create a datetime object that contains the current time  
currentDT = dt.datetime.now()  
  
#view the value of currentDT  
print(currentDT)
```

```
#create a new string object that contains the reformatted date and time  
UDdt = currentDT.strftime('%b %d, %Y %I:%M %p')  
#display the result  
UDdt
```


3.2. ЧИТАННЯ ТА ЗАПИС ФАЙЛІВ В PYTHON



Модуль `csv` – це частина стандартної бібліотеки Python і призначений для роботи з `.csv` файлами. Він дозволяє зчитувати та записувати дані в файли цього формату. Для використання цього модуля в коді потрібно виконати імпорт.



У Python існують базові функції для роботи з файлами, такі як створення, відкриття та закриття. Функція `open()` використовується для створення нового файлу або відкриття наявного файлу, який буде використовуватися для збереження даних. Після завершення роботи з файлом, його потрібно закрити за допомогою функції `close()` для збереження системних ресурсів та захисту від можливих пошкоджень.



Після відкриття файлу для запису, дані можна записувати в нього за допомогою методу `write()`. У разі, якщо файл відкритий в режимі "а" (додавання), дані будуть додаватися до кінця файлу. Для коректного форматування даних, може знадобитися додавати символи форматування.

ВИКОРИСТАННЯ МЕТОДІВ OPEN () ТА CLOSE () МОДУЛЯ DATETIME

```
myFile = open("newText.txt", "w")  
myFile.close()  
myFile = open("newText.txt", "a")
```

```
print(myFile)
```

```
<open file 'newText.txt', mode 'a' at 0x729e2338>
```


ПАРАМЕТРИ МЕТОДУ OPEN ()

```
open('myFile.txt', "w" )
```

File open() Method Modes

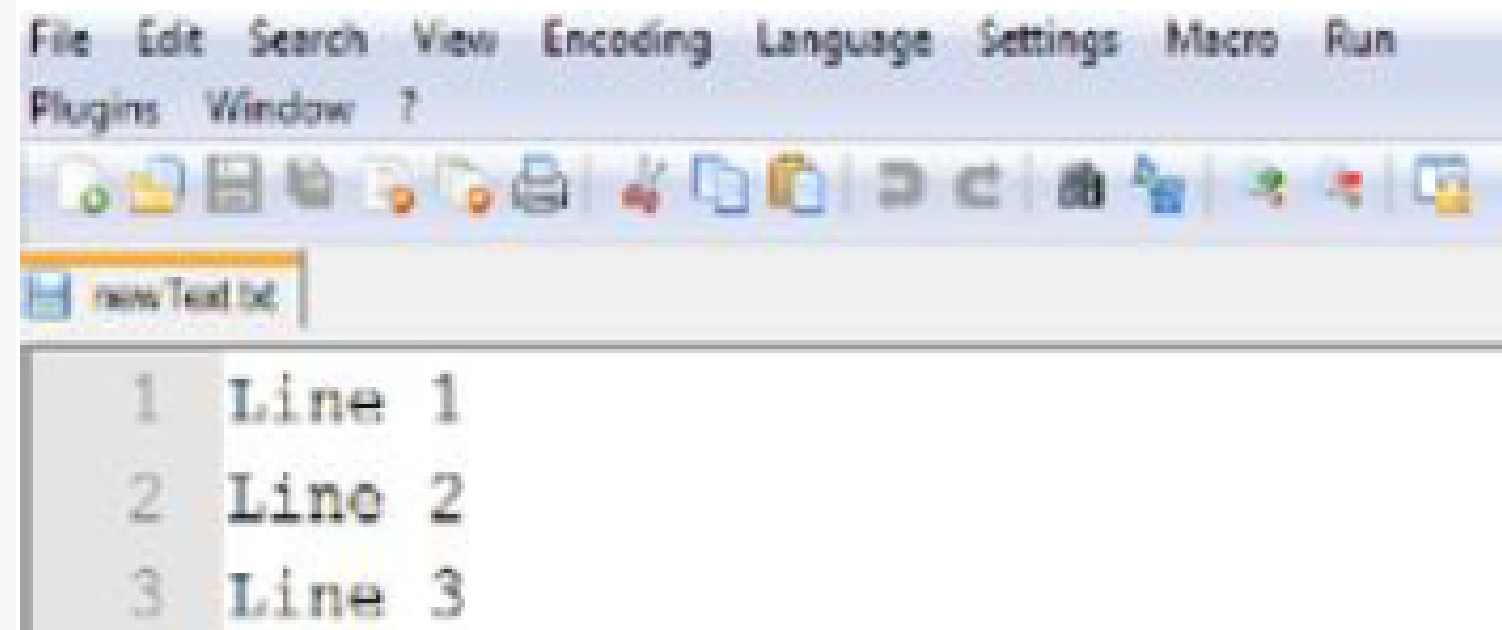
Modes	Description
w	Opens an existing file that has the file name specified. If the file does not exist, it opens a new file with that name.
r	Opens an existing file in read only mode.
a	Opens an existing file and will append new data to the end of the file.

ЧИТАННЯ ТА ЗАПИС ДАНИХ У ФАЙЛ

```
myFile = open('newText.txt', "w")  
myFile.close()
```

```
myFile = open('newText.txt', "a")  
myFile.write('Line 1\n')  
myFile.write('Line 2\n')  
myFile.write('Line 3\n')
```

```
myFile.close()  
myFile = open('newText.txt', "r")  
myFile.read()
```



The screenshot shows a text editor window with a menu bar (File, Edit, Search, View, Encoding, Language, Settings, Macro, Run, Plugins, Window, ?) and a toolbar. The file name 'newText.txt' is visible in the title bar. The editor content consists of three lines of text:

```
1 Line 1  
2 Line 2  
3 Line 3
```

3.3. ВЗАЄМОДІЯ ІЗ ЗОВНІШНІМИ ДОДАТКАМИ



Python надає можливість взаємодіяти з зовнішніми додатками та операційною системою, що відкриває широкі можливості для автоматизації та інтеграції різних завдань.



Jupyter Notebook є веб-програмою з відкритим вихідним кодом, яка дозволяє створювати документи, що містять діючий код, математичні рівняння, візуалізації та текст. Це інструмент, який знаходить застосування в областях очищення та перетворення даних, чисельного моделювання, статистичного аналізу, візуалізації, машинного навчання та інших видів обчислень.



У Jupyter Notebooks можна взаємодіяти з операційною системою за допомогою символу "!", що дозволяє виконувати зовнішні команди. Наприклад, можна взаємодіяти з оболонкою Linux, відправляти команди та отримувати їх вивід. Це демонструє можливість використання модуля `subprocess` для спілкування з зовнішніми програмами та обробки їхнього виводу в об'єкті Python.

КОМАНДИ LINUX, ВИКОНАНІ В JUPYTER NOTEBOOKS ДЛЯ ВЗАЄМОДІЇ З ОПЕРАЦІЙНОЮ СИСТЕМОЮ

```
!ls -al
```

```
total 80  
drwxr-xr-x 4 root root 16384 Feb 22 19:47 .  
drwxr-xr-x 5 root root 16384 Feb  6 22:34 ..  
drwxr-xr-x 4 root root 16384 Feb 19 21:01 chapter 3  
drwxr-xr-x 2 root root 16384 Feb 22 19:47 .ipynb_checkpoints  
-rw-r--r-- 1 root root    72 Feb 22 19:47 Untitled.ipynb
```

```
!head logins.csv
```

```
Student,Login_Day,Login_Time  
Rose,12/4/2016,20:29  
Joe,9/4/2016,22:37  
Jerry,2/20/2017,4:18  
Lawrence,4/7/2016,13:17  
Roy,6/24/2016,15:30  
Robin,9/6/2016,20:55  
Harry,8/12/2016,14:26  
Mary,1/26/2017,1:46  
Stephanie,12/31/2016,10:23  
Tammy,5/25/2016,7:08
```

ВИКОРИСТАННЯ МОДУЛЯ SUBPROCESS ДЛЯ ЗВ'ЯЗКУ З ЗОВНІШНЬОЮ ПРОГРАМОЮ

```
'''import the subprocess library which is necessary for communication with external
apps'''
import subprocess
#We now execute the ping process as of from the shell:

pingCmd = 'ping -c 127.0.0.1'
process = subprocess.Popen(pingCmd.split(), stdout=subprocess.PIPE)
'''creat an object to hold the output of the process and split the output elements
into a list'''

process_output = process.communicate()[0]
process_output = process_output.split()
#view the contents of the output object
print(process_output)
```

```
['PING', '127.0.0.1', '(127.0.0.1)', '56(84)', 'bytes', 'of', 'data.', '64', 'bytes',
'from', '127..0.0.1', 'icmp_seq=1', 'ttl=64', 'time=0.094', 'ms', '64', 'bytes',
'from', '127.0.0.1', 'icmp_seq=2', 'ttl=64', 'time=0.052', 'ms', '---', '127.0.0.1',
'ping', 'statistics', '---', '2', 'packets', 'transmitted', '2', 'received,', '0%',
'packet', 'loss,', 'time', '999ms', 'rtt', 'min/avg/max/mdev', '=',
'0.052/0.073/0.094/0.021', 'ms']
```

```
#view the first five elements of the list
process_output[0:5]
```

```
['PING', '127.0.0.1', '(127.0.0.1)', '56(84)', 'bytes']
```


ВИСНОВОК ДО ЛЕКЦІЇ 3

Для оброблення даних про час та дату, використовується модуль Python `datetime`. Модуль `csv` дозволяє читати та записувати у `.csv` файли. Python також має основні методи створення, відкриття та закриття зовнішніх файлів. Таблиці даних будуть існувати лише в оперативній пам'яті, поки вони не будуть збережені у файли. Метод `open ()` використовується для створення нового файлу або для відкриття наявного файлу, який буде містити дані, які потрібно зберегти. Функція `close ()` видаляє дані з буферів і закінчує функцію запису файлу для вказаного файлу. Python дозволяє взаємодіяти із зовнішніми додатками та операційною системою. Модуль `subprocess` використовується для зв'язку з зовнішньою програмою та збереження виводу команди, виданої цій програмі, в об'єкт Python.

