



Лекція 1. Програмне забезпечення для роботи з мовою Python. Основні ПОНЯТТЯ МОВИ

Лектор – доцент, кандидат фізико-математичних наук
Кириченко Віктор Вікторович



ПЛАН ЛЕКЦІЇ

1. Встановлення і налаштування інтерпретатора Python. Основні задачі, які розв'язуються за допомогою мови Python.
2. Огляд сучасних інтегрованих середовищ розробки (IDE) для Python.
3. Налаштування віртуального оточення. Створення проекту і простої програми на мові Python.



Основні задачі, які розв'язуються за допомогою мови Python

Основні задачі, які розв'язуються за допомогою Python:

- Автоматизація рутинних задач
- Веброзробка (Django, Flask)
- Аналіз даних, наука про дані (NumPy, Pandas, Matplotlib)
- Машинне навчання (scikit-learn, TensorFlow)
- Розробка ігор, GUI-додатків, скриптів для адміністрування та багато іншого



Встановлення і налаштування інтерпретатора Python

Python інтерпретована об'єктноорієнтована мова програмування високого рівня з простим синтаксисом.

Для виконання коду використовується програма інтерпретатор, який виконує код рядок за рядком. Для початку роботи з Python потрібно встановити інтерпретатор з офіційного сайту <https://www.python.org/>.

The screenshot shows the official Python website. The header includes the Python logo, a 'Donate' button, and a search bar. The navigation menu has links for 'About', 'Downloads', 'Documentation', 'Community', 'Success Stories', and 'News'. The 'Downloads' section is active, displaying a list of download options: 'All releases', 'Source code', 'Windows', 'macOS', 'Other Platforms', 'License', and 'Alternative Implementations'. The 'Windows' option is selected, leading to the 'Download for Windows' page. This page features a button for 'Python 3.12.5' and a note stating: 'Note that Python 3.9+ cannot be used on Windows 7 or earlier.' Below the note, it says: 'Not the OS you are looking for? Python can be used on many operating systems and environments. View the full list of downloads.' At the bottom of the page, a footer reads: 'Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)'.



Огляд сучасних інтегрованих середовищ розробки (IDE) для Python

Для написання коду на Python використовуються спеціальні середовища розробки:



PyCharm



VS Code



Jupyter



Огляд сучасних інтегрованих середовищ розробки (IDE) для Python

- PyCharm - середовище розробки з потужною підтримкою автодоповнення, налагодження, тестуванням та інтеграцією з фреймворками. Можна завантажити з сайту JetBrains. IDE дозволяє одразу створити віртуальне оточення для проекту та керувати бібліотеками

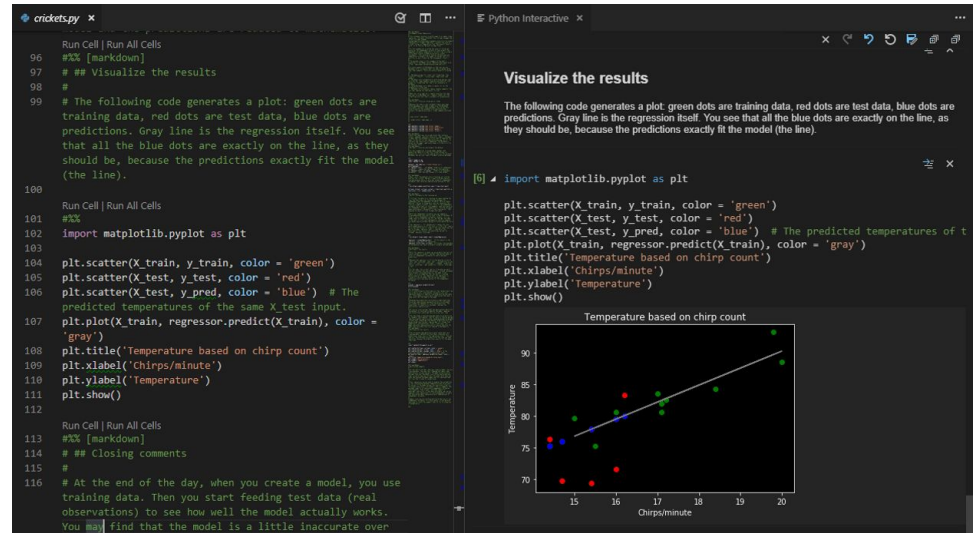
```
20
21
22 response = self.client.get(reverse('polls:index'))
23 self.assertEqual(response.status_code, 200)
24 self.assertContains(response, "No polls are available.")
25 self.assertQuerysetEqual(response.context['latest_question_list'], [])
26
27 @test_index_view_with_a_future_question(self)
28 def test_index_view_with_a_future_question(self):
29     """
30     Test index view with a future question and past question
31     """
32     self.assertEqual(response.status_code, 200)
33     self.assertQuerysetEqual(response.context['latest_question_list'], [])
34
35 @test_index_view_with_a_future_question_and_past_question(self)
36 def test_index_view_with_a_future_question_and_past_question(self):
37     """
38     Test index view with a future question and past question
39     """
40     self.assertEqual(response.status_code, 200)
41     self.assertQuerysetEqual(response.context['latest_question_list'], [])
42
43 @test_index_view_with_no_questions(self)
44 def test_index_view_with_no_questions(self):
45     """
46     Test index view with no questions
47     """
48     self.assertEqual(response.status_code, 200)
49     self.assertQuerysetEqual(response.context['latest_question_list'], [])
50
51 @test_index_view_with_two_past_questions(self)
52 def test_index_view_with_two_past_questions(self):
53     """
54     Test index view with two past questions
55     """
56     self.assertEqual(response.status_code, 200)
57     self.assertQuerysetEqual(response.context['latest_question_list'], [])
58
59 def test_index_view_with_a_future_question(self):
60     """
61     Questions with a pub_date in the future should not be displayed on
62     the index page.
63     """
64     create_question(question_text="Future question.", days=30)
65     response = self.client.get(reverse('polls:index'))
66     self.assertContains(response, "No polls are available.",
67                         status_code=200)
68     self.assertQuerysetEqual(response.context['latest_question_list'], [])
69
70 def test_index_view_with_future_question_and_past_question(self):
71     """
72     Even if both past and future questions exist, only past questions
73     should be displayed.
74     """
75     create_question(question_text="Past question.", days=-30)
76     create_question(question_text="Future question.", days=30)
77     response = self.client.get(reverse('polls:index'))
78     self.assertQuerysetEqual(
79         response.context['latest_question_list'],
80         ['<Question: Past question.>']
81     )
82
83 def test_index_view_with_two_past_questions(self):
84     """
85     Test index view with two past questions
86     """
87     self.assertEqual(response.status_code, 200)
88     self.assertQuerysetEqual(response.context['latest_question_list'], [])
```

<https://www.jetbrains.com/pycharm/>



Огляд сучасних інтегрованих середовищ розробки (IDE) для Python

- Visual Studio Code - безкоштовне і легке середовище розробки, розширюється плагінами, популярна серед розробників. Можна завантажити з офіційного сайту (<https://code.visualstudio.com/>). Після запуску IDE, потрібно зайти у розділ плагіни та встановити офіційний плагін для Python від Microsoft, можна додатково встановити Pylance.

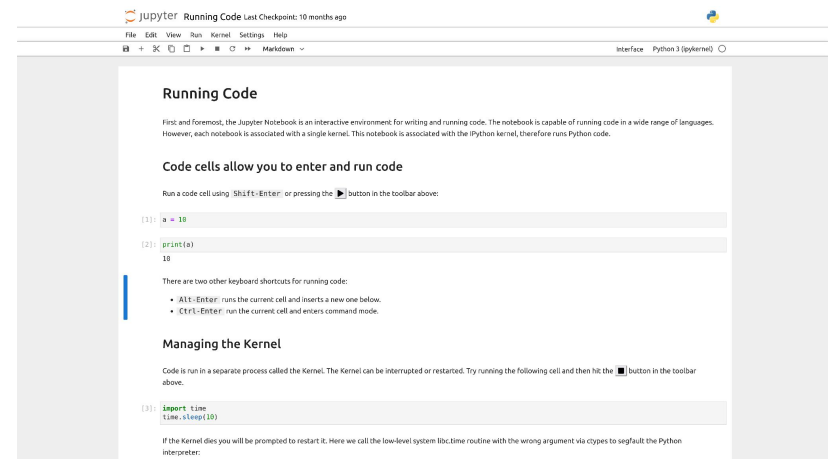


<https://code.visualstudio.com/>



Огляд сучасних інтегрованих середовищ розробки (IDE) для Python

- **Jupyter Notebook** - інтерактивне середовище для роботи з кодом і даними, зручно для експериментів та навчання. Встановлюється через **pip** (`pip install notebook`) або разом з дистрибутивом Anaconda. Запуск через команду `jupyter notebook` у терміналі відкриває вебінтерфейс для роботи з нотатками.



`pip install notebook`



Налаштування віртуального оточення

Команда `pip install` встановлює залежності глобально. Для ізолювання залежностей в рамках окремих проектів використовується віртуальне середовище. Це дозволяє для кожного проекту використовувати різні версії залежностей для уникнення конфліктів.

Для створення віртуального середовища у директорії використовується команда `python -m venv myenv`, де `myenv` назва середовища. Для використання віртуального середовища використовуються такі команди:

- Windows: `myenv\Scripts\activate`
- macOS/Linux: `source myenv/bin/activate`



Створення проекту і простої програми на мові Python.

Приклад програми hello world:

1. Створіть проект у PyCharm або інший IDE.
2. Додайте новий файл, наприклад, hello.py.
3. Введіть код:

```
print("Hello World!")
```

4. Запустіть програму через IDE або у терміналі командою:

```
python hello.py
```

5. Програма виведе "Hello World!".

```
test.py x
test.py
1 print('Hello World')
```



ДЯКУЮ ЗА УВАГУ