Курс з Python для початківців

Цей курс розроблений для тих, хто робить перші кроки у програмуванні та хоче опанувати Python. Ми пройдемо шлях від встановлення Python до написання перших програм, розглядаючи основні концепції та інструменти.

Модуль 1: Вступ до Python та основи

1.1. Що таке Python і чому його варто вивчати?

* Огляд Python: Історія, філософія, основні сфери застосування (веб-розробка, аналіз даних, штучний інтелект, автоматизація).
* Переваги Python: Читабельність, велика спільнота, безліч бібліотек.

1.2. Встановлення Python

* Завантаження та інсталяція: Де завантажити Python (офіційний сайт python.org), процес встановлення для різних операційних систем (Windows, macOS, Linux).
* Перевірка встановлення: Використання командного рядка/терміналу для перевірки версії Python.
* Інтегровані середовища розробки (IDE): Короткий огляд та рекомендації (PyCharm Community Edition, VS Code, Thonny).

1.3. Перша програма: "Hello, World!"

* Інтерактивний режим Python (REPL): Як використовувати.
* Написання та запуск першого скрипту: print("Hello, World!").
* Коментарі: Однорядкові та багаторядкові коментарі.

1.4. Змінні та типи даних

* Поняття змінної: Що це і як їх створювати.
* Основні типи даних:
  * Цілі числа (int): Приклади, операції.
  * Числа з плаваючою комою (float): Приклади, операції.
  * Рядки (str): Створення, конкатенація, форматування рядків (f-strings).
  * Булеві значення (bool): True і False.
* Функція type(): Визначення типу змінної.

1.5. Оператори

* Арифметичні оператори: +, -, *, /, // (цілочисельне ділення), % (остача від ділення), ** (піднесення до степеня).
* Оператори присвоєння: =, +=, -=, *=, /=, etc.
* Оператори порівняння: ==, !=, <, >, <=, >=.
* Логічні оператори: and, or, not.

Модуль 2: Структури управління потоком та функції

2.1. Умовні оператори: if, elif, else

* Логіка умовних конструкцій: Як програма приймає рішення.
* Приклади використання: Перевірка віку, оцінок, тощо.
* Вкладені if: Коли і як використовувати.

2.2. Цикли

* Цикл for: Ітерація по послідовностях (рядки, списки, діапазони).
  * range() функція: Генерація послідовностей чисел.
* Цикл while: Повторення доки умова істинна.
  * Безкінечні цикли: Запобігання та переривання.
* break та continue: Управління потоком циклу.

2.3. Функції

* Що таке функція: Переваги використання (повторне використання коду, модульність).
* Визначення функції: Ключове слово def, параметри, тіло функції.

* Виклик функції: Передача аргументів.
* Повернення значень: Ключове слово return.
* Параметри за замовчуванням: Як їх задавати.
* Область видимості змінних (Scope): Локальні та глобальні змінні.

Модуль 3: Колекції даних

3.1. Списки (list)
* Створення списків: Синтаксис [].
* Доступ до елементів: Індекси, зрізи (slices).
* Операції зі списками: Додавання (append, extend, insert), видалення (remove, pop, del), сортування (sort, sorted), копіювання.
* Ітерація по списках: Використання циклу for.

3.2. Кортежі (tuple)
* Створення кортежів: Синтаксис ().
* Відмінності від списків: Немінюваність (immutable).
* Коли використовувати кортежі: Наприклад, для фіксованих наборів даних.

3.3. Словники (dict)
* Створення словників: Синтаксис {ключ: значення}.
* Доступ до елементів: За ключем.
* Операції зі словниками: Додавання, зміна, видалення елементів.
* Методи словників: keys(), values(), items().
* Ітерація по словниках.

3.4. Множини (set)
* Створення множин: Синтаксис {}, функція set().
* Унікальність елементів: Автоматичне видалення дублікатів.
* Операції з множинами: Об'єднання, перетин, різниця.

Модуль 4: Робота з файлами та обробка винятків

4.1. Робота з файлами
* Відкриття та закриття файлів: Функція open(), режими роботи ('r', 'w', 'a').
* Читання файлів: read(), readline(), readlines().
* Запис у файли: write(), writelines().
* Конструкція with open(...) as f:: Безпечна робота з файлами.

4.2. Обробка винятків (Errors and Exceptions)
* Що таке винятки: Помилки, які виникають під час виконання програми.
* Блок try-except: Перехоплення та обробка помилок.
* Типи винятків: ValueError, TypeError, FileNotFoundError, ZeroDivisionError, etc.
* Блок finally: Виконання коду незалежно від наявності винятків.

Модуль 5: Об'єктно-орієнтоване програмування (ООП) - основи

5.1. Концепції ООП
* Що таке ООП: Основні принципи (інтуїтивне розуміння).
* Об'єкти та класи: Відносини між ними (клас як шаблон, об'єкт як екземпляр класу).

5.2. Створення класів
* Визначення класу: Ключове слово class.
* Атрибути: Змінні, що належать класу або об'єкту.
* Методи: Функції, що належать класу.
* __init__ метод (конструктор): Ініціалізація об'єктів.
* self параметр: Посилання на поточний об'єкт.

5.3. Створення об'єктів та робота з ними
* Створення екземплярів класу: Виклик класу як функції.

 * Доступ до атрибутів та методів: Використання крапкової нотації.

Модуль 6: Модулі та пакети

6.1. Модулі

 * Що таке модуль: Файл .py, що містить Python-код.
 * Імпорт модулів: Ключове слово import.
 * Імпорт конкретних об'єктів: from module import function_name.
 * Псевдоніми: import module as alias.

6.2. Вбудовані модулі Python

 * math: Математичні функції.
 * random: Генерація випадкових чисел.
 * datetime: Робота з датами та часом.
 * os: Взаємодія з операційною системою.

6.3. Пакети

 * Що таке пакет: Колекція модулів.
 * Структура пакету: Каталоги з файлом __init__.py.
 * Імпорт з пакетів.

Наступні кроки та куди рухатися далі:

 * Системи контролю версій: Git та GitHub (для спільної розробки та зберігання коду).
 * Віртуальні середовища: venv (для ізоляції залежностей проектів).
 * Розробка власного проекту: Застосування отриманих знань на практиці.
 * Вивчення фреймворків та бібліотек:
   * Веб-розробка: Django, Flask.
   * Аналіз даних: NumPy, Pandas, Matplotlib.
   * Машинне навчання: Scikit-learn, TensorFlow, PyTorch.

Цей курс надає міцну основу для подальшого вивчення Python. Головне — це практика! Пишіть код щодня, експериментуйте та не бійтеся помилятися. Успіхів у навчанні!