

ВСТУП

В основу пропонованого навчального посібника покладено цикл занять, проведених автором для студентів природничо-математичного факультету Національного університету «Чернігівський колегіум» імені Т.Г. Шевченка.

Навчання програмування займає досить вагоме місце в освітньому процесі. Досвід багатьох країн показує, що вивчення правил кодування і мов програмування сприяє розвитку логічного та креативного мислення. Також варто відмітити те, що програмування навчає творчості, шляхом набуття навичок шукати рішення, оскільки для розв'язання практичних задач можна підійти з різних сторін.

Взагалі, мов програмування досить багато. Більш того, час від часу з'являються нові мови. Тому природним чином виникає питання: чому саме Python? Тут можна видалити кілька пунктів.

Мова програмування Python - це мова високого рівня, досить "молода", але дуже популярна, яка вже зараз широко використовується на практиці і сфера застосування Python постійно розширюється.

Мова Python бурхливо розвивається. Цьому сприяє не тільки досить вдала концепція мови, але також сформоване згуртоване співтовариство розробників і популяризаторів мови.

Синтаксис мови Python мінімалістичний і гнучкий. На цій мові можна складати прості та ефективні програми.

Стандартна бібліотека для цієї мови містить безліч корисних функцій, що значно полегшує процес створення програмних кодів.

Мова Python підтримує кілька парадигм програмування, включаючи структурну, об'єктно-орієнтовану і функціональну.

Мова Python цілком вдалий вибір для першої мови при навчанні програмуванню

Важливий і той факт, що необхідне програмне забезпечення, включаючи середовища розробки, в основній своїй масі безкоштовні.

Все це дає підстави розглядати Python в якості одного з найбільш перспективних мов програмування.

1. Знайомство з мовою програмування Python

1.1. Історія мови

Автором мови Python є нідерландський програміст Гвідо ван Россум. На той час він був співробітником голландського інституту CWI і працював над мовою ABC[en], яка мала стати максимально простою і зрозумілою декларативною мовою. Мова ABC була достатньо цікавою, але вона мала значний недолік: нею ніхто не користувався. Тому в грудні 1989 року Гвідо розпочав свій «хобі проект» Python, в основу якого лягли окремі елементи мови ABC (фактично, Python створювався як спроба виправити помилки, допущені при проектуванні ABC). Датою появи мови Python вважається 20 лютого 1991 року.

До недавнього часу Гвідо ван Россум був не лише автором мови, а і «доброчинним пожиттєвим диктатором» («Benevolent Dictator For Life») тобто мав право приймати остаточні рішення щодо мови. Проте 12 липня 2018 року він вирішив залишити цей «пост».

Як зазначає сам Гвідо ван Россум, мова названа на честь британського телешоу «Літаючий цирк Монти Пайтона» (англ. *Monty Python's Flying Circus*), хоча інколи мову називають «Пітон».

З'явившись порівняно пізно, Python створювався під впливом багатьох мов програмування [1, 4, 9]:

- ABC – на відміну від значної кількості мов програмування, в мові ABC відразу використовувалися відступи (поля) замість операторних дужок для групування операторів. Такий же принцип був обраний і для Python;
- Modula-3 – популярна в академічних кругах, але майже без практичного програмування мова була однією з перших, де були винятки (try ... except), які і були взяті для Python;
- C, C++ – для Python були взяті деякі синтаксичні конструкції (як пише сам Гвідо ван Россум - він використовував найбільш несуперечливі конструкції з C, щоб не викликати неприязнь у C-програмістів до Python);
- Smalltalk – елементи об'єктно-орієнтованого програмування;

- Lisp – окремі риси функціонального програмування;
- Fortran – зрізи масивів, комплексна арифметика;
- Miranda – спискові вирази;
- Java – модулі logging, unittest, threading (частина можливостей оригінального модуля не реалізована), xml.sax стандартної бібліотеки, спільне використання finally та except при обробці винятків, використання @ для декораторів;
- Icon – генератори.

На даний момент існують дві актуальні версії мови: Python 2.x (остання версія на момент написання – 2.7) і Python 3.x (остання версія на момент написання – 3.8). Це пов'язане з тим, що за свій час «життя» мова Python 2.x акумулювала певну кількість недоліків, які не можна було виправити, не зламавши чийсь код. Основний недолік пов'язаний з рядками (спочатку рядки були байтовими, потім було додано підтримку Юнікоду (Unicode) з неявним приведенням).

Тому в більшості випадків Python 2.x використовується для забезпечення сумісності з раніше написаними програмами. Проте підтримка гілки 2.x буде завершена в січні 2020 року.

Python 3.0 обернено не сумісний з попередньою серією 2.x. Код Python 2.x швидше за все буде видавати помилки при виконанні в Python 3.x. Динамічна типізація Python, разом з планами зміни декількох методів словників, робить механічне портування з Python 2.x в Python 3.0 дуже складним. [1, 2, 6]

Тому нові проекти варто створювати на Python 3.x, якщо немає явних причин використовувати Python 2.x.

1.2. Особливості мови

Мову Python можна охарактеризувати як: інтерпретовану високорівневу мову програмування загального призначення. В різній літературі можна зустріти набагато більше складових мови, які вказують при її описі, тому просто сформулюємо її особливості.

Проста і мінімалістична мова

Читання хорошої програми на Python дуже нагадує читання англійського тексту, хоча і досить строгого. Така псевдокодова природа Python є однією з його найсильніших сторін. Python надає можливість програмісту зосередитися на розв'язуванні задачі, а не на самій мові. Ядро мови без додаткових модулів досить просте і мінімальне - це надає мові простоти та логічності, що відповідно спрощує її вивчення.

Мова легка в освоєнні та простота для вивчення

Мовою Python надзвичайно легко почати програмувати. Python має виключно простий синтаксис. Досить часто в закладах освіти обирають Python як першу мову програмування для своїх студентів.

Вільна і відкрита мова

Python - це приклад вільного і відкритого програмного забезпечення – FLOSS (Free / Libre and Open Source Software). Простіше кажучи, ви маєте право вільно використовувати копії цього програмного забезпечення, переглядати його вихідні тексти, вносити зміни, а також використовувати його частини в своїх програмах. Python був створений і постійно поліпшується співтовариством, яке просто хоче зробити його краще.

Високорівнева мова

Машина (комп'ютер, процесор) розуміє лише певний обмежений та примітивний набір спеціальних команд. Коли пишеться код деякою мовою, по суті, вказуються інструкції, які повинна виконати машина. В залежності від того, які ці інструкції, мови і поділяються на високорівневі та низькорівневі. Якщо код пишеться в машинних інструкціях (командах), то це низькорівневі мови. Якщо ж додатково необхідне певне програмне забезпечення, яке буде переводити код, написаний мовою програмування, в машинні інструкції (команди) то це високорівнева мова програмування. Програмне забезпечення, за яким відбувається вказане переведення, називається компіляторами або інтерпретаторами.

Мова загального призначення

Мова призначена для розв'язування досить широкого спектру задач.

Кросплатформенна мова

Завдяки своїй відкритій природі, Python був портований на багато платформ (змінений таким чином, щоб працювати на різних пристроях та різних операційних системах). Один і той самий код буде запускатися на різних платформах без будь-яких змін за умови, що в коді відсутні системно-залежні функції.

Python можна використовувати в GNU / Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE і навіть на PocketPC.[1, 8, 10]

Інтерпретована мова

Як зазначалося раніше, для виконання програми на комп'ютері необхідно команди мови програмування перетворити у команди, які здатен виконувати процесор комп'ютера. Існує два типи таких перетворень – інтерпретація і компіляція, які здійснюються спеціальними програмами-трансляторами.

Програма, написана компільованою мовою програмування, як наприклад, Pascal або C++, аналізується і при відсутності помилок перетворюється, за допомогою компілятора, в мову, зрозумілу комп'ютеру (бінарний код), після чого записується в оперативну пам'ять комп'ютера або на диск. Утворений на диску файл є виконуваним файлом і може бути виконаний засобами операційної системи без повторного використання транслятора.

При інтерпретації транслятор аналізує кожен команду програми окремо і, якщо команда не містить синтаксичних помилок, перетворює її в машинний код і відразу виконує. [7] Проте в Python цей процес дещо складніший. Після запуску програми Python аналізує кожен команду програми і, якщо команда не містить синтаксичних помилок, перетворює її в байт-код (машино-незалежний код низького рівня). Далі байт-код передається віртуальній машині Python, яка і виконує інструкції байт-коду, перетворюючи їх в машинний код.

Коли байт-код сформовано, його можна виконати на різних інтерпретаторах, що підтримують цей байт-код. Так, для Python можна виділити такі інтерпретатори:

- CPython – інтерпретатор, реалізований на C. Вважається еталонною реалізацією Python і підтримує більшість активно використовуваних платформ;
- IronPython – Python для .NET Framework і Mono. Компілює Python програми в MSIL (Microsoft Intermediate Language), таким чином надаючи повну інтеграцію з .NET-системою;
- Jython – реалізація Python, що використовує JVM (Java Virtual Machine);
- Stackless – патчі до CPython, який надає розширені можливості багатопотокового програмування і значно більшу глибину рекурсії;
- PyPy – реалізація Python, написана на RPython. В PyPy вбудований трасуючий JIT-компілятор (Just-in-time compilation), який може перетворювати код Python в машинний код під час виконання програми.

Мультипарадигменна мова

В основі Python лежить декілька парадигм програмування: об'єктно-орієнтовна, імперативна, функціональна (функційна), структурна. Проте досить часто про Python говорять зокрема як про об'єктно-орієнтовну мову програмування. В об'єктно-орієнтованих мовах програмування програми будуються на основі об'єктів, які об'єднують в собі дані і функціонал. Об'єктів в мові Python багато, можна навіть сказати, що все є об'єкт (це не дуже коректно, але допустимо).

Розширювана мова

Якщо потрібно, щоб деяка критична частина програми працювала дуже швидко, або є необхідність приховати частину алгоритму, можна написати цю частину програми мовою C або C++, а потім викликати її з програми, написаною мовою Python.

Вбудовувана мова

Код Python можна легко вбудувати в програми на Java, C/C++, щоб надати користувачам можливість написання власних сценаріїв. Взаємодія Python-додатків з іншими системами можлива також за допомогою CORBA (Common Object Request Broker Architecture), XML-RPC (Extensible Markup Language Remote Procedure Call), SOAP (Simple Object Access Protocol), COM (Component Object Model).

Мова з "широкою" стандартною бібліотекою

Стандартна бібліотека Python досить велика і її використання надає можливість програмістам розв'язувати досить різноманітні задачі: з використанням регулярних виразів, перевіркою блоків коду, розпаралелюванням процесів, базами даних, веб-браузерами, CGI, FTP, електронною поштою, XML, XML-RPC, HTML, WAV файлами, криптографією, GUI (графічний інтерфейс користувача) та іншими системно-залежними речами. Крім стандартної бібліотеки, існує безліч інших високоякісних бібліотек, які можна знайти в каталозі пакетів Python

Таким чином, Python – потужна мова, в якій поєднано продуктивність та значні можливості, що робить написання програм одночасно цікавим і легким.

Незважаючи на досить багато переваг, Python має і ряд недоліків, а саме:

- Низька швидкодія в порівнянні з компільованими мовами. Це пов'язане з інтерпретованістю мови, тобто переведенням коду програми в машинний код в процесі роботи програми. Цей недолік частково знімається збереженням модулів у вигляді байт-коду (файли .рус і .pyo).
- В Python відсутня можливість модифікувати вбудовані класи, такі, як int, str, float, list та інші. Однак це дозволяє Python споживати менше оперативної пам'яті і швидше працювати.
- Глобальне блокування інтерпретатора GIL (Global Interpreter Lock) або проблеми з багатопоточністю. В кожен момент часу може виконуватися тільки одна нитка (потік) Python коду, навіть якщо на комп'ютері є кілька процесорів або процесорних ядер. Це зроблено для

унеможливлення руйнування даних при спільній модифікації з різних потоків, оскільки в своїй роботі інтерпретатор Python постійно використовує велику кількість нитко-небезпечних даних. Проте GIL-проблема притаманна CPython, Stackless та PyPy, але відсутня в Jython та IronPython.

1.3. Застосування мови

Python є не лише академічною мовою програмування, на прикладі якої можна тільки навчати основам програмування і далі про неї забути. Серед областей, де активно застосовується мова Python, можна виділити: системне програмування, розробку динамічних веб-сайтів, інтеграцію компонентів, розробку програм для роботи з базами даних, швидке створення прототипів, розробку програм для наукових обчислень, навчання, розробку ігор.

Розробка прикладного програмного забезпечення та ігор.

- BitTorrent – програми для файлообміну. Всі версії до 6-ї були написані мовою Python. Версія 6 була переписана мовою C++.
- Blender – вільний, професійний пакет для створення тривимірної комп'ютерної графіки. Python використовується як засіб створення інструментів і прототипів, системи логіки в іграх, як засіб імпорту / експорту файлів, автоматизації завдань.
- GIMP – растровий графічний редактор. Python використовується для написання додаткових модулів (фільтрів).
- Maya – система тривимірного моделювання та створення мультиплікації підтримує інтерфейс для управління з сценаріїв мовою Python
- Civilization IV. Велика частина гри написана мовою Python.
- World of Tanks. Як скриптова мова в проєкті використовується Python.
- LibreOffice. Написання макросів.

Web-розробка.

- Яндекс. Python використовується в ядрі Яндекс диску.
- DropBox. Сервіс розроблений мовою Python.
- Instagram. Розроблений мовами Python і Django.

- Компанія Google широко використовує Python в своїй пошуковій системі.
- Служба колективного використання відеоматеріалів YouTube в значному ступені реалізована мовою Python.
- Реалізація поштового сервера IronProt використовує понад 1 мільйон рядків програмного коду мовою Python.
- Мовою Python написані багато веб-фреймворків: App Engine, Pylons, TurboGears, CherryPy, Flask, Pyramid та інші.

Компанії, які використовують Python.

- iRobot використовує Python в розробці комерційних роботизованих пристроїв.
- ESRI використовує Python в якості інструменту налаштування своїх популярних геоінформаційних програмних продуктів під потреби кінцевого користувача.
- NSA використовує Python для шифрування і аналізу розвідданих.
- EVE Online і Massively Multiplayer Online Game (MMOG) широко використовують Python в своїх розробках.
- Intel, Cisco, Hewlett-Packard, Seagate, Qualcomm і IBM, використовують Python для тестування апаратного забезпечення.
- Industrial Light & Magic, Pixar і інші використовують Python у виробництві анімаційних фільмів.
- JPMorgan Chase, UBS, Getco і Citadel застосовують Python для прогнозування фінансового ринку.
- NASA, Los Alamos, Fermilab, JPL і інші використовують Python для наукових обчислень.

Системне адміністрування. Python встановлюється за замовчуванням на всі Linux-сервери і використовується в автоматизації роботи системного адміністратора.

Вбудовані системи. Дуже часто Python використовується для програмування вбудованих систем. Найвідоміший проект, який використовує Python - це Raspberry Pi.