СУЧАСНІ ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ

122 «Комп'ютерні науки» КН-18 2019 / 2020 навчальний рік

PYTHON #8

- 1. Інтерпретація скриптів Python. Байт-код.
- 2. Модулі Python.
- 3. Пакети Python.

https://github.com/eabshkvprof/2020_Mod_Prog_Techn

ІНТРПРЕТАЦІЯ СКРИПТІВ

Інтерпретатор (interpreter) – програма, необхідна для виконання інших програм, вид транслятору, який здійснює пооператорну (покомандну, построкову) обробку, **перетворення** у машинний код та **виконання** програми.

Компілятор (compiler) — програма (набір програм), що перетворює (компілює) деякий вхідний код, написаний певною мовою програмування (source language), на семантично еквівалентний код в іншій мові програмування (цільова мова, target language), з подальшою можливістю виконання програми комп'ютером.

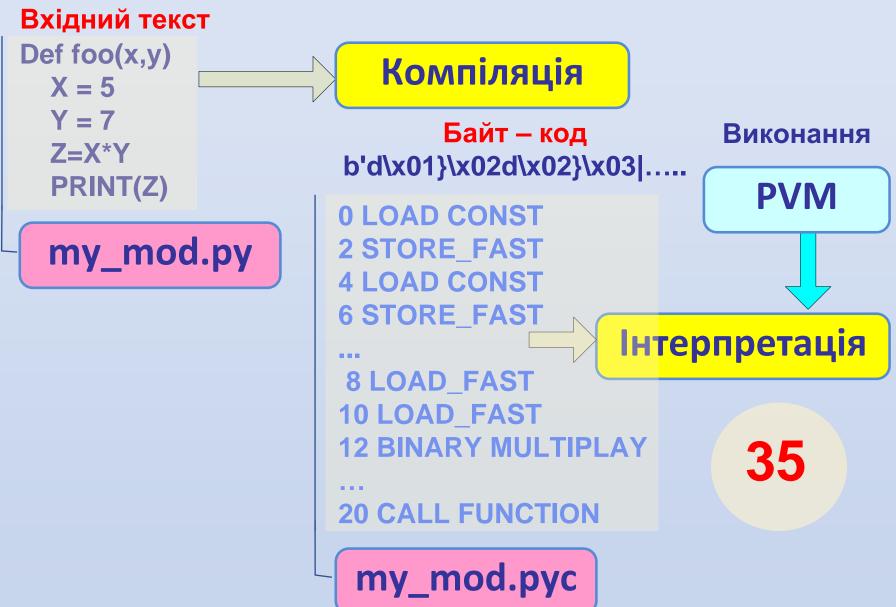
ІНТРПРЕТАЦІЯ СКРИПТІВ

Простий інтерпретатор – аналізує і **відразу виконує** (власне інтерпретація) програму покомандно (або порядково), по мірі надходження тексту програми на вхід інтерпретатора.

Інтерпретатор компілюючого типу — це система з компілятора, який перекладає текст програми в деяке проміжне представлення (найчастіше в байт-код), і власне інтерпретатора, який виконує отриманий проміжний код - так звана віртуальна машина.

Загальна думка: машинний код набагато швидше, але байт-код краще захищений та є переносним.

ВИКОНАННЯ РҮТНОМ ПРОГРАМИ



РЕАЛІЗАЦІЇ РҮТНОМ

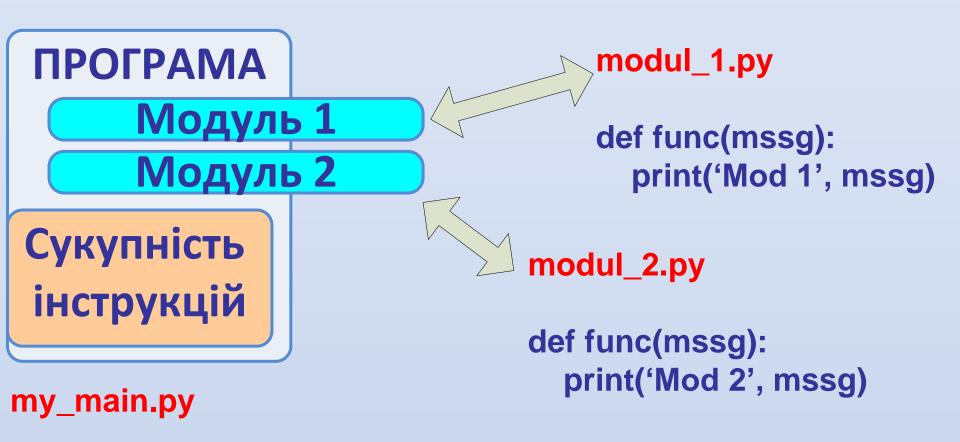
Implementation	Virtual Machine	Compatible Lang
CPython	CPython VM	C
Jython	JVM	Juva
IronPython	CLR	C #
Brython	Javascript Eng.	JavaScript
RubyPython	RubyVM	Ruby
PyPy		Python
JupyterLab Spyder		

An enhanced Interactive Python.

7.13.0 Released on Fab 29, 2020.

IPython / **IP[y]**

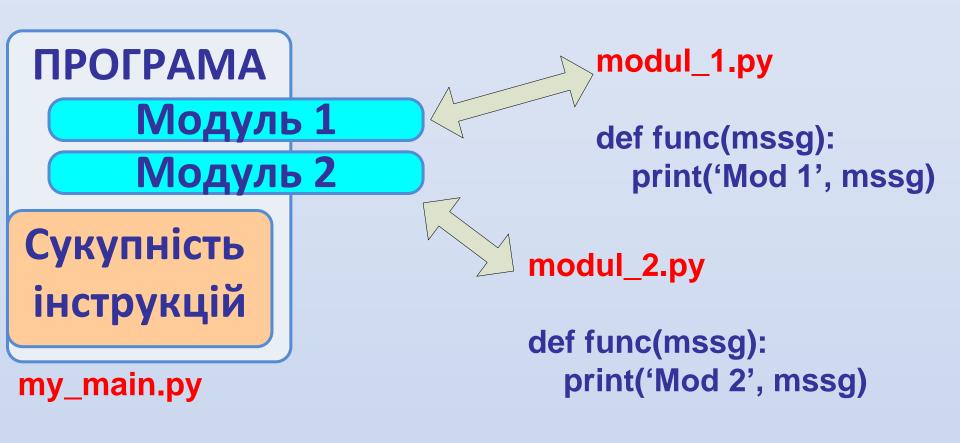
модулі



import modul_1

modul_1.func('OhOhOh')

МОДУЛІ



import modul_1

import modul_2

modul_1.func('OhOhOh')

modul_1.func('HaHaHa')

ПРОСТІР ІМЕН МОДУЛЯ

$MOДVЛЬ \rightarrow \Pi AKET IMEH$

- **□Інструкції модуля виконуються** під час першої спроби імпорту. Перший імпорт інтерпретатор Python створює порожній об'єкт модуля і виконує інструкції в модулі одну за одною, від початку файлу до кінця.
- **□Операції присвоювання** (не вкладені в def () або class()), що виконуються на верхньому рівні, створюють атрибути об'єкта модуля і зберігаються в просторі імен модуля.
- □Область видимості модуля (простір імен) після завантаження модуля перетворюється в атрибутсловник об'єкта модуля. Доступ до простору імен модуля можна отримати через атрибут dict_ aбo dir (M).

ІНСТРУКЦІЇ ІМПОРТУ

	Дія
import < module >	Файл module завантажується та перетворюється в ім'я змінної , яка посилається на повний об'єкт модуля із усіма його атрибутами
import < module > as < mod_name >	ім'я змінної := mod_name (нові, короткі імена)
from < module > import *	Завантажуються всі об'єкти-функції модуля, перетворюються в імена змінних та копіюються в область видимості програми
4. from < module > import < fun1, fun2, >	Імпортуються окремі функції
5 from <module> Import <fun1, fun2,=""> as F1,F2,</fun1,></module>	Імпортуються окремі функції з новими іменами

ІНСТРУКЦІЇ ІМПОРТУ

import , **from** → !!! інструкції (як і всі інші)

from ..!.. вставляє простір імен модуля в простір імен програми.

! ПЕРЕВАГУ incrpyкції import

import ..!.. створює **єдине** ім'я в просторі імен програми що посилається на об'єкт-модуль (на простір імен модуля).

Функція	Дія
from imp import	reload примусово виконує
reload	повторну завантаження вже
	завантаженого модуля і
	запускає його програмний код.
reload <module></module>	Інструкції присвоювання, що виконуються при повторному запуску, будуть змінювати існуючий об'єкт модуля.

ІМПОРТ МОДУЛЯ

Крок	Дія
1.Пошук	Пошук модуля в робочому середовищі
2.Компіляція	Перетворення в байт-код (*.рус), якщо це необхідно
3.3апуск	Виконується байт-код, створюються всі об'єкти модуля та, відповідно, простір імен модуля (атрибути всіх об'єктів модуля).

пошук модуля

Де шукати модуль ?

	Напрямки пошуку
1.	Домашня директорія програми
2.	Змінна оточення РҮТНО Р
3.	Каталоги стандартних бібліотек
4.	Зміст будь-яких файлів $*.pht$, якщо вони ϵ

import sys

sys.path – перегляд шляхів пошуку

СИНТАКСИС КВАЛІФІКАЦІЇ ІМЕН

Для доступу до атрибутів будь-якого об'єкта використовується синтаксис кваліфікації імені *object.attribute ->* це вираз, що повертає значення, яке присвоєно імені атрибуту.

Звернення до імен, кваліфікуючи їх, явно вказує інтерпретатору об'єкт, атрибут якого потрібно використати.

- □Прості змінні (правило **LEGB**), наприклад **X** пошук імені **X** в поточних областях видимості.
- □Кваліфіковані імена **X.Y** пошук імені **X** спочатку в поточних областях видимості, а потім буде виконаний пошук атрибута **Y** в об'єкті **X** (не в областях видимості).
- □Кваліфіковані шляхи **X.Y.Z** буде проведений пошук імені **Y** в об'єкті **X**, а потім пошук імені **Z** в об'єкті **X.Y**.

ПАКЕТ МОДУЛІВ

ПАКЕТ → директорія (каталог), в якому розташовані модулі

```
.../work_dir/dir import dir.modul_1
__init__.py
modul_1
modul_2
```

__init__.py призначений для виконання дій по першій ініціалізації пакету, створення простору імен для каталогу і реалізації поведінки інструкцій імпорту, наприклад створення файлів, підключення до БД та інше.

МОДУЛЬ як СКРИПТ

Кожен модуль має вбудований атрибут ______, який встановлюється

→ <modul>, якщо файл модуля імпортується

```
→ "__main__", якщо файл модуля запускається як головний файл програми.
```

Тобто модуль може перевірити власний атрибут **__name__** і визначити, чи був він запущений як самостійна програма або імпортовано іншим модулем.

Рекомендована ЛІТЕРАТУРА

- Програмування числових методів мовою Python: підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий; за ред. А. В. Анісімова. К.: Видавничо-поліграфічний центр "Київський університет", 2014. 640 с.
- Програмування числових методів мовою Python: навч. посіб. / А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий, Є. В. Глушко; за ред. А. В. Анісімова. К.: Видавничо-поліграфічний центр "Київський університет", 2013. 463 с.
- Основи програмування Python: Підручник для студ. спеціальності 122 «Компютерні науки» / А.В.Яковенко; КПІ.- Київ: КПІ, 2018. 195 с.
- **Лутц М.** Изучаем Python, 4-е издание. СПб.: Символ-Плюс. 2011.- 1280 с.: ил.

Контрольні запитання

- Надайте визначення компілятора та інтерпретатора. Наведіть порядок виконання Python скриптів.
- Обгрунтуйте необхідність та вкажіть переваги використання модулів.
- Надайте перелік інструкцій імпорту модулів, поясніть їх відмінності та наведіть відповідні приклади.
- Наведіть способи завдання шляхів пошуку модуля.
- Поясніть сутність поняття кваліфіковані імена та наведіть відповідні приклади.
- Надайте призначення пакетів модулів та поясніть порядок створення пакетів.

The END Mod 1. Lec 9.