บทที่ 1

แนะนำภาษาโปรแกรม

ภาษาโปรแกรม (Programming Language) คือภาษาประดิษฐ์ (Formal Language) ที่ประกอบด้วยชุดคำสั่ง ซึ่งออกแบบมาเพื่อสื่อสารกับเครื่องจักร โดยเฉพาะอย่างยิ่งกับคอมพิวเตอร์ (Computer) บุคคลที่เขียนโปรแกรมจะถูกเรียกว่าโปรแกรมเมอร์ (Programmer) ในยุคแรกเริ่ม ภาษาโปรแกรมถูกสร้างขึ้นก่อนที่คอมพิวเตอร์ดิจิทัลจะถือกำเนิด เพื่อใช้ควบคุมเครื่องจักรกลอย่างเครื่องทอผ้าของแจ็กการ์ด (Jacquard loom) หรือเครื่องเล่นเปียโนอัตโนมัติ

ปัจจุบันมีภาษาโปรแกรมหลายพันภาษาที่ถูกสร้างขึ้น และมีภาษาใหม่เกิดขึ้นทุกปี ภาษาโปรแกรมส่วนใหญ่อธิบายการคำนวณในรูปแบบเชิงคำสั่ง (Imperative) ซึ่งระบุขั้นตอนการทำงานอย่างชัดเจน ในขณะที่ภาษาโปรแกรมรูปแบบอื่น ๆ ใช้วิธีการที่แตกต่างออกไป เช่น การเขียนโปรแกรมเชิงฟังก์ชัน (Functional Programming) หรือการเขียนโปรแกรมเชิงตรรกะ (Logic Programming)

สำหรับรายวิชาพื้นฐานการเขียนโปรแกรม (Programming Fundamentals) เราได้เลือกใช้ภาษาไพธอน (Python) ซึ่งเป็นภาษาโปรแกรมระดับสูง (High-level Language) ที่ได้รับความนิยมอย่างแพร่หลายในปัจจุบัน โดยเฉพาะในสาขาวิทยาศาสตร์ วิศวกรรมศาสตร์ วิทยาการข้อมูล (Data Science) และปัญญาประดิษฐ์ (Artificial Intelligence) ในบทนี้ เราจะแนะนำภาษาไพธอนตั้งแต่ประวัติความเป็นมา กระบวนการทำงาน ไปจนถึงข้อดีและความสามารถอันหลากหลายของภาษาโปรแกรมนี้

1.1 ประวัติภาษาโปรแกรมไพธอน

ภาษาไพธอนถูกพัฒนาขึ้นโดย Guido van Rossum โปรแกรมเมอร์ชาวดัตช์ ผู้สำเร็จการศึกษาระดับปริญญาโทสาขาคณิตศาสตร์และวิทยาการคอมพิวเตอร์จาก University of Amsterdam เขาได้เริ่มต้นโครงการไพธอนในช่วงปลายทศวรรษ 1980 ขณะทำงานอยู่ที่ Centrum Wiskunde & Informatica (CWI) ในประเทศเนเธอร์แลนด์ ต่อมาเขาได้ร่วมงานกับองค์กรชั้นนำในสหรัฐอเมริกาหลายแห่ง ไม่ว่าจะเป็น Google (2005-2012) และ Dropbox (2013-2019)

รูป 1.1: Guido van Rossum *(Photo of Guido van Rossum)*

หลังจากเกษียณอายุได้ไม่นาน ในปี 2020 Van Rossum ได้กลับเข้ามาร่วมงานอีกครั้งกับบริษัท Microsoft ในตำแหน่ง Distinguished Engineer โดยมีเป้าหมายหลักในโครงการ "Faster CPython" คือการเพิ่มความเร็วในการประมวลผลของภาษาไพธอนให้สูงขึ้นอย่างมีนัยสำคัญ

ที่มาของชื่อ "ไพธอน" นั้นไม่ได้มาจากงู แต่มาจากความชื่นชอบของ Van Rossum ที่มีต่อรายการโทรทัศน์ "Monty Python's Flying Circus" ซึ่งเป็นซีรีส์ตลกของ BBC ในยุค 1970 เขาต้องการชื่อที่สั้น มีเอกลักษณ์ และดูลึกลับน่าค้นหา จึงตัดสินใจเลือกใช้ชื่อ "Python"

ภาษาไพธอนเป็นภาษาโปรแกรมระดับสูงที่ได้รับอิทธิพลจากภาษาอื่น ๆ เช่น ABC, Modula-3 และ Lisp มีจุดเด่นด้านการจัดการหน่วยความจำอัตโนมัติ (Automatic Memory Management) และการกำหนดชนิดข้อมูลแบบพลวัต (Dynamic Typing) ซึ่งหมายความว่าโปรแกรมเมอร์ไม่จำเป็นต้องประกาศชนิดของตัวแปรก่อนใช้งาน

ไพธอนเวอร์ชันแรก (0.9.0) เปิดตัวสู่สาธารณะในปี 1991 และมีการพัฒนาอย่างต่อเนื่องจนถึงเวอร์ชัน 3 ในปี 2008 ซึ่งเป็นเวอร์ชันที่มีการเปลี่ยนแปลงโครงสร้างภาษาครั้งสำคัญ สำหรับภาษาไพธอนเวอร์ชัน 2 ได้สิ้นสุดการสนับสนุนอย่างเป็นทางการในวันที่ 1 มกราคม 2020 ทำให้ปัจจุบันเวอร์ชัน 3 ถือเป็นมาตรฐานหลักของภาษา ในปัจจุบัน (กันยายน 2025) เวอร์ชันล่าสุดคือ 3.13.0 และสามารถดาวน์โหลดได้โดยไม่มีค่าใช้จ่ายที่ `http://www.python.org` เนื่องจากเป็นซอฟต์แวร์โอเพนซอร์ส (Open Source) ประวัติการออกเวอร์ชันที่สำคัญแสดงดังตารางที่ 1.1

หนึ่งในปัจจัยสำคัญที่ทำให้ไพธอนเติบโตคือชุมชนนักพัฒนา (Community) ที่แข็งแกร่งและมีส่วนร่วมในการพัฒนาอย่างต่อเนื่อง ส่งผลให้มีไลบรารี (Library) คุณภาพสูงให้เลือกใช้มากมาย ครอบคลุมการใช้งานตั้งแต่การประมวลผลทางคณิตศาสตร์และวิทยาศาสตร์ ไปจนถึงการพัฒนาเว็บและปัญญาประดิษฐ์ นอกจากนี้ การสนับสนุนจากบริษัทเทคโนโลยีขนาดใหญ่อย่าง Microsoft, Google และ Meta (Facebook) ก็เป็นเครื่องยืนยันถึงอนาคตที่สดใสของภาษาโปรแกรมนี้

จากดัชนีความนิยมของภาษาโปรแกรม PYPL (Popularity of Programming Language Index)[1] ซึ่งวิเคราะห์จากความถี่ในการค้นหาคู่มือการใช้งาน (Tutorial) บน Google ในเดือนกันยายน 2025 พบว่าภาษาไพธอนยังคงครองอันดับ 1 ภาษาโปรแกรมที่ได้รับความนิยมสูงสุดทั่วโลก โดยมีส่วนแบ่งตลาดถึง 35.2% ทิ้งห่างภาษาอื่น ๆ เช่น Java, JavaScript และ C/C++

[1]The PYPL PopularitY of Programming Language Index, `https://pypl.github.io/PYPL.html`

เวอร์ชัน		วันเปิดตัว
Python 1.0	Python 1.0	January 1994
	Python 1.5	December 1997
	Python 1.6	September 2000
Python 2.0	Python 2.0	October 2000
	Python 2.7	July 2010
Python 3.0	Python 3.0	December 2008
	Python 3.5	September 2015
	Python 3.6	December 2016
	Python 3.7	June 2018
	Python 3.8	October 2019
	Python 3.9	October 2020
	Python 3.10	October 2021
	Python 3.11	October 2022
	Python 3.12	October 2023
	Python 3.13	October 2024

ตาราง 1.1: ประวัติเวอร์ชันของภาษาโปรแกรมไพธอน (แสดงเฉพาะเวอร์ชันที่สำคัญ) *(Selected Python Version History)*

1.2 การทำงานของภาษาโปรแกรมไพธอน

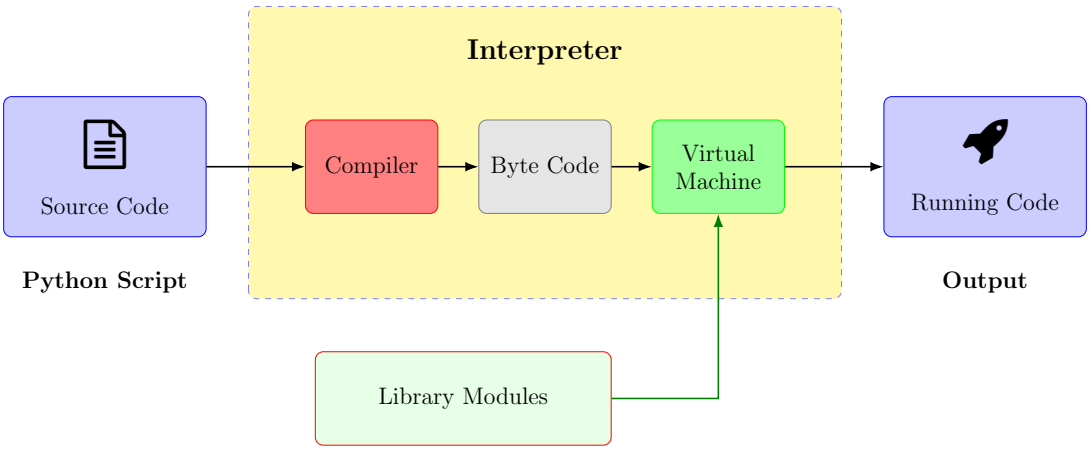
เพื่อให้เกิดความเข้าใจพื้นฐานในการพัฒนาโปรแกรม สิ่งสำคัญคือการเรียนรู้กระบวนการทำงานของภาษาไพธอน ซึ่งมีขั้นตอนสรุปได้ดังรูปที่ 1.2

การพัฒนาโปรแกรมเริ่มต้นจากการเขียนรหัสต้นฉบับ (Source Code) ซึ่งเป็นไฟล์ข้อความธรรมดา (Plain Text) ที่มีนามสกุล `.py` เรามักเรียกไฟล์นี้ว่า "ไพธอนสคริปต์" (Python Script) จากนั้น ตัวแปลภาษา (Interpreter) จะทำหน้าที่ประมวลผลสคริปต์นี้เพื่อให้คอมพิวเตอร์เข้าใจและทำงานตามคำสั่ง

กระบวนการภายในของตัวแปลภาษาไพธอนมีความซับซ้อนกว่าที่เห็น โดยเริ่มต้นจากขั้นตอนการคอมไพล์ (Compilation) ที่คอมไพเลอร์ (Compiler) จะแปลงรหัสต้นฉบับให้เป็นรูปแบบที่เรียกว่า "ไบต์โค้ด" (Bytecode) และบันทึกเป็นไฟล์นามสกุล `.pyc` ไบต์โค้ดนี้เป็นชุดคำสั่งระดับกลางที่ไม่ได้ผูกติดกับสถาปัตยกรรมของคอมพิวเตอร์โดยตรง

จากนั้น เครื่องจักรเสมือนของไพธอน (Python Virtual Machine: PVM) ซึ่งเป็นส่วนประกอบหลักของตัวแปลภาษา จะทำหน้าที่อ่านและประมวลผลไบต์โค้ดทีละคำสั่ง เพื่อให้โปรแกรมทำงานตามที่เขียนไว้ การทำงานผ่าน PVM นี้เองที่ทำให้โปรแกรมที่เขียนด้วยภาษาไพธอนสามารถทำงานได้บนระบบปฏิบัติการที่แตกต่างกัน (Cross-platform) ไม่ว่าจะเป็น Microsoft Windows, macOS หรือ Linux โดยไม่ต้องแก้ไขรหัสต้นฉบับ

ภาษาไพธอนใช้ตัวแปลภาษาแบบ Interpreter ซึ่งหมายถึงการแปลและประมวลผลคำสั่งทีละบรรทัด หากพบข้อผิดพลาดทางไวยากรณ์ (Syntax Error) ในบรรทัดใด โปรแกรมจะหยุดทำงานและแจ้งข้อผิดพลาดทันที โปรแกรมเมอร์จึงจำเป็นต้องแก้ไขข้อผิดพลาดให้ถูกต้องเพื่อให้โปรแกรมสามารถทำงานต่อไปได้



รูป 1.2: ขั้นตอนการทำงานของภาษาไพธอน *(Python Execution Process)*

1.3 ข้อดีของภาษาโปรแกรมไพธอน

ภาษาไพธอนถูกนำไปใช้งานอย่างกว้างขวางเนื่องด้วยข้อดีหลายประการ ดังนี้

1. **ไวยากรณ์เรียบง่ายและอ่านง่าย (Clean and Readable Syntax):** โครงสร้างภาษาไพธอนถูกออกแบบมาให้อ่านเข้าใจง่าย คล้ายกับภาษาอังกฤษ ทำให้เหมาะสำหรับผู้เริ่มต้นเขียนโปรแกรม

2. **การกำหนดชนิดข้อมูลแบบพลวัต (Dynamic Typing):** ไม่จำเป็นต้องประกาศชนิดของตัวแปรก่อนใช้งาน ตัวแปลภาษาจะกำหนดชนิดข้อมูลให้โดยอัตโนมัติตามค่าที่ถูกเก็บ ซึ่งช่วยลดความซับซ้อนและเพิ่มความเร็วในการพัฒนา

3. **ทำงานได้หลายแพลตฟอร์ม (Cross-platform):** โปรแกรมที่เขียนด้วยภาษาไพธอนสามารถนำไปใช้งานบนระบบปฏิบัติการที่แตกต่างกันได้โดยไม่ต้องแก้ไขรหัสต้นฉบับ

4. **ระบบนิเวศและไลบรารีที่แข็งแกร่ง (Rich Ecosystem and Libraries):** มีไลบรารีมาตรฐานและไลบรารีภายนอกให้เลือกใช้จำนวนมหาศาล ครอบคลุมการใช้งานหลากหลายด้าน เช่น `NumPy` และ `Pandas` สำหรับงานด้านข้อมูล, `TensorFlow` และ `PyTorch` สำหรับปัญญาประดิษฐ์ และ `Django` สำหรับการพัฒนาเว็บ

5. **เป็นซอฟต์แวร์โอเพนซอร์ส (Open Source):** สามารถใช้งานได้ฟรีทั้งเพื่อการศึกษาและเชิงพาณิชย์ และมีชุมชนนักพัฒนาทั่วโลกที่ช่วยกันพัฒนาและดูแลอย่างต่อเนื่อง

6. **รองรับกระบวนทัศน์การเขียนโปรแกรมที่หลากหลาย (Multi-paradigm):** สนับสนุนทั้งการเขียนโปรแกรมเชิงโครงสร้าง (Structured Programming), เชิงวัตถุ (Object-Oriented Programming: OOP) และเชิงฟังก์ชัน (Functional Programming)

7. **ทำงานร่วมกับภาษาอื่นได้ดี (Integration Capabilities):** สามารถเรียกใช้งานโค้ดที่เขียนด้วยภาษาอื่น ๆ เช่น C/C++ หรือ Java ได้ ทำให้สามารถนำไปต่อยอดกับระบบที่มีอยู่เดิมได้อย่างมีประสิทธิภาพ

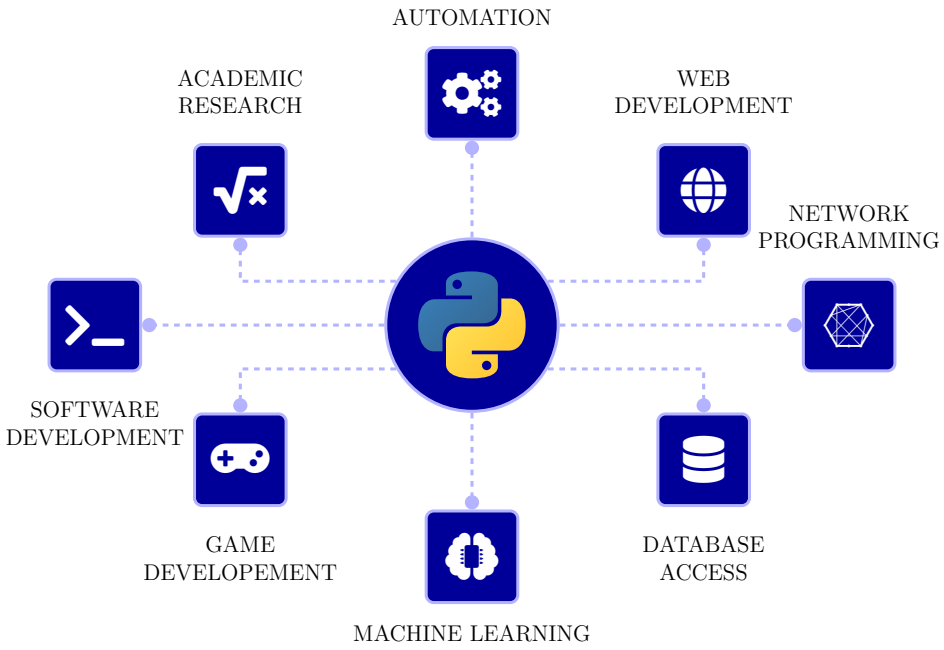
1.4 ความสามารถของภาษาโปรแกรมไพธอน

ด้วยความยืดหยุ่นและไลบรารีที่ครอบคลุม ภาษาไพธอนจึงถูกนำไปประยุกต์ใช้ในงานหลากหลายประเภท ตั้งแต่งานขนาดเล็กไปจนถึงระบบขนาดใหญ่ระดับโลก

1. **วิทยาการข้อมูลและปัญญาประดิษฐ์ (Data Science & AI):** เป็นแวดวงที่ไพธอนโดดเด่นที่สุด ด้วยไลบรารีอย่าง Scikit-learn, Pandas, TensorFlow และ PyTorch ทำให้นักวิทยาศาสตร์ข้อมูลสามารถวิเคราะห์ข้อมูล สร้างแบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) และพัฒนาระบบปัญญาประดิษฐ์ได้อย่างรวดเร็ว

2. **การพัฒนาเว็บ (Web Development):** เฟรมเวิร์ก (Framework) ที่ได้รับความนิยมสูงอย่าง Django และ Flask ช่วยให้นักพัฒนาสามารถสร้างเว็บแอปพลิเคชันที่มีประสิทธิภาพและปลอดภัยได้อย่างเป็นระบบ

3. **การทำงานอัตโนมัติและสคริปต์ (Automation and Scripting):** ไพธอนเป็นเครื่องมือที่ยอดเยี่ยมสำหรับการเขียนสคริปต์เพื่อทำงานซ้ำ ๆ โดยอัตโนมัติ เช่น การจัดการไฟล์ การดึงข้อมูลจากเว็บไซต์ (Web Scraping) หรือการดูแลรักษาระบบ



รูป 1.3: ตัวอย่างการประยุกต์ใช้งานภาษาไพธอน *(Examples of Python Applications)*

4. **การพัฒนาส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI Development):** สามารถสร้างแอปพลิเคชันบนเดสก์ท็อปที่มีหน้าตาสวยงามโดยใช้ไลบรารี เช่น Tkinter, PyQt หรือ Kivy

5. **การพัฒนาเกม (Game Development):** แม้จะไม่ใช่ผู้นำในด้านนี้ แต่ก็มีไลบรารีอย่าง PyGame ที่เหมาะสำหรับการสร้างเกมสองมิติและใช้เป็นเครื่องมือในการเรียนรู้การพัฒนาเกม

6. **Internet of Things (IoT):** ด้วยขนาดที่เล็กและประสิทธิภาพที่ดี ไพธอนจึงถูกนำไปใช้บนอุปกรณ์สมองกลฝังตัว (Embedded Systems) เช่น Raspberry Pi เพื่อสร้างอุปกรณ์ IoT สำหรับบ้านอัจฉริยะ (Smart Home) หรือฟาร์มอัจฉริยะ (Smart Farm)

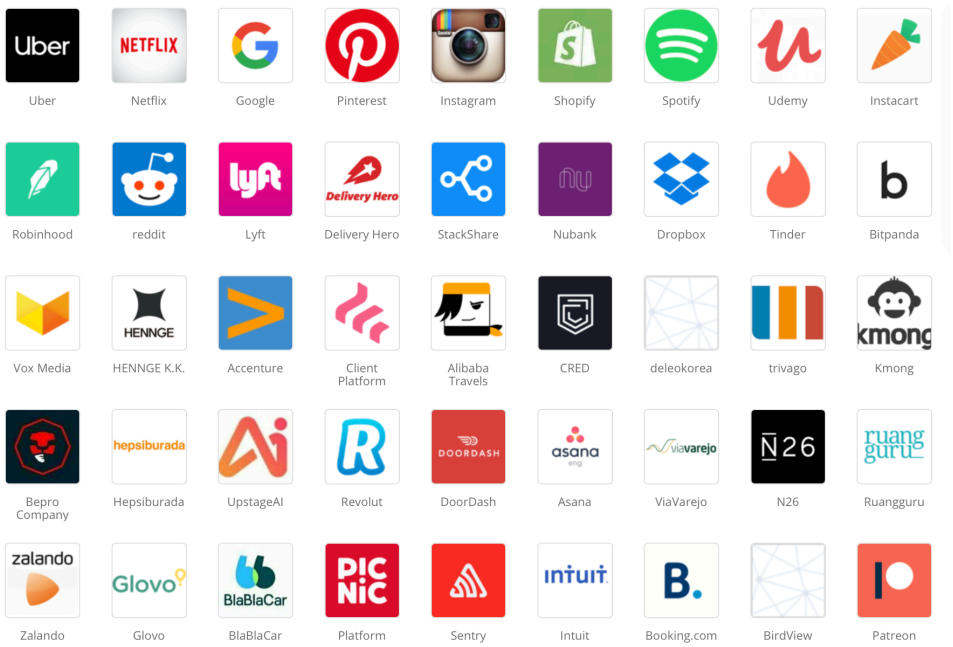
7. **การประมวลผลทางวิทยาศาสตร์และวิศวกรรม (Scientific and Engineering Computing):** ไลบรารีอย่าง SciPy และ Matplotlib เป็นเครื่องมือสำคัญสำหรับนักวิจัยและวิศวกรในการคำนวณที่ซับซ้อนและนำเสนอข้อมูลในรูปแบบกราฟ

1.5 องค์กรชั้นนำที่เลือกใช้ภาษาไพธอน

ความสามารถที่หลากหลายของภาษาไพธอนทำให้องค์กรเทคโนโลยีชั้นนำของโลกจำนวนมากเลือกใช้เป็นเครื่องมือหลักในการพัฒนาผลิตภัณฑ์และบริการ ตัวอย่างที่น่าสนใจมีดังนี้

- **Google:** ใช้ภาษาไพธอนอย่างกว้างขวางในระบบภายใน ตั้งแต่ส่วนของ Backend ของบริการต่าง ๆ เช่น YouTube ไปจนถึงสคริปต์สำหรับดูแลระบบ และเป็นภาษาหลักสำหรับเฟรมเวิร์กปัญญาประดิษฐ์อย่าง TensorFlow

- **Meta (Facebook):** ใช้ไพธอนในการจัดการโครงสร้างพื้นฐาน (Infrastructure Management) และการวิเคราะห์ข้อมูลจำนวนมหาศาล นอกจากนี้ ไลบรารีปัญญาประดิษฐ์ชื่อดังอย่าง PyTorch ก็ได้รับการพัฒนาโดย Meta



รูป 1.4: ตัวอย่างองค์กรที่เลือกใช้ภาษาโปรแกรมไพธอน *(Examples of Organizations Using Python)*

- **Netflix:** นักพัฒนาของ Netflix ใช้ไพธอนในการวิเคราะห์ข้อมูลเพื่อสร้างระบบแนะนำภาพยนตร์ (Recommendation System) และใช้ในการควบคุมกระบวนการผลิตสื่อ (Media Production Pipeline) รวมถึงระบบแจ้งเตือนและดูแลความเสถียรของบริการ
- **Dropbox:** บริการฝากไฟล์และซิงค์ข้อมูลชื่อดังได้พัฒนาแอปพลิเคชันสำหรับเดสก์ท็อป (Desktop Client) ด้วยภาษาไพธอน ซึ่งสามารถรองรับผู้ใช้งานที่ลงทะเบียนแล้วกว่า 700 ล้านคนทั่วโลก
- **Spotify:** ผู้ให้บริการสตรีมมิงเพลงชั้นนำ ใช้ไพธอนในส่วนของการวิเคราะห์ข้อมูล (Data Analytics) และระบบ Backend เพื่อจัดการกับข้อมูลเพลงและพฤติกรรมการฟังของผู้ใช้จำนวนมหาศาล
- **Instagram:** สร้างขึ้นบนพื้นฐานของ Django ซึ่งเป็นเว็บเฟรมเวิร์กที่เขียนด้วยภาษาไพธอน และสามารถรองรับการใช้งานจากผู้ใช้หลายร้อยล้านคนต่อวัน พิสูจน์ให้เห็นถึงความสามารถในการขยายระบบ (Scalability) ของไพธอน
- **NASA:** หน่วยงานอวกาศของสหรัฐอเมริกาใช้ไพธอนในโครงการคำนวณทางวิทยาศาสตร์และการประมวลผลข้อมูลจากดาวเทียมและยานสำรวจอวกาศ เนื่องจากความสามารถในการจัดการข้อมูลที่ซับซ้อนได้อย่างมีประสิทธิภาพ

ในประเทศไทย ภาษาไพธอนได้ถูกบรรจุเป็นส่วนหนึ่งของหลักสูตรการศึกษาในระดับมัธยมศึกษาตามแนวทางการเรียนรู้วิทยาการคำนวณ และเป็นภาษาหลักที่ใช้สอนในรายวิชาการเขียนโปรแกรมเบื้องต้นของมหาวิทยาลัยหลายแห่ง

สรุปท้ายบท

ในบทนี้ เราได้ทำความรู้จักกับภาษาไพธอน ตั้งแต่ประวัติความเป็นมาและปรัชญาการออกแบบ ไปจนถึงกระบวนการทำงานและความสามารถที่หลากหลาย ไพธอนเป็นภาษาโปรแกรมที่ทรงพลังแต่เรียนรู้ได้ง่าย มีชุมชนนักพัฒนาที่เข้มแข็งและมีไลบรารีสนับสนุนจำนวนมาก ทำให้กลายเป็นเครื่องมือสำคัญในยุคดิจิทัล โดยเฉพาะอย่างยิ่งในสาขาวิทยาการข้อมูล ปัญญาประดิษฐ์ และการพัฒนาเว็บ ความนิยมที่เพิ่มขึ้นอย่างต่อเนื่องทำให้การเรียนรู้ภาษาไพธอนเป็นการลงทุนที่คุ้มค่าสำหรับอนาคต

แบบฝึกหัด

1. จงอธิบายความแตกต่างระหว่างภาษาโปรแกรมระดับสูง (High-level) และภาษาโปรแกรมระดับต่ำ (Low-level) พร้อมทั้งยกตัวอย่างภาษาโปรแกรมในแต่ละประเภทมาอย่างน้อย 2 ภาษา

2. กระบวนการทำงานของ Interpreter และ Compiler แตกต่างกันอย่างไร จงอธิบายโดยละเอียด และระบุว่าภาษาไพธอนใช้กระบวนการทำงานแบบใดเป็นหลัก

3. จากที่ได้ศึกษาในบทนี้ เหตุใดภาษาไพธอนจึงได้รับความนิยมอย่างสูงในแวดวงวิทยาการข้อมูล (Data Science) และปัญญาประดิษฐ์ (Artificial Intelligence) จงอธิบายพร้อมยกตัวอย่างไลบรารีที่เกี่ยวข้องอย่างน้อย 3 ตัว

4. Guido van Rossum ผู้สร้างภาษาไพธอน ได้รับแรงบันดาลใจในการตั้งชื่อ 'Python' จากอะไร และเป้าหมายหลักของเขาในการกลับเข้าร่วมงานกับ Microsoft ในปี 2020 คืออะไร

5. จงยกตัวอย่างองค์กรหรือบริษัทที่มีชื่อเสียงอย่างน้อย 3 แห่งที่ใช้ภาษาไพธอนในการพัฒนาผลิตภัณฑ์หรือบริการของตนเอง พร้อมอธิบายโดยสังเขปว่านำไปใช้ในส่วนใด