

บทที่ 1

การติดตั้งและพื้นฐานของภาษาไพธอน

ขั้นตอนการวิเคราะห์และออกแบบโปรแกรม สิ่งที่ได้จากกระบวนการนี้คือ อัลกอริทึม ข้อมูล และแผนผังโปรแกรม จากผลลัพธ์ที่ได้ดังกล่าวจะนำไปสู่กระบวนการของการเขียนโปรแกรมต่อไป ซึ่งการนำไปเขียนโปรแกรมนั้นสามารถนำไปเขียนได้ทุกภาษา ตามที่ได้ออกแบบไว้ การเลือกภาษาคอมพิวเตอร์มาพัฒนาโปรแกรมเป็นสิ่งสำคัญที่จะต้องคำนึงถึง ทั้งในด้านของคุณสมบัติของภาษาคอมพิวเตอร์ ความเหมาะสม ประสิทธิภาพ งบประมาณ ข้อดีและข้อเสีย ในการนำมาใช้ในการพัฒนาโปรแกรม

ไพธอน (Python) คือ ภาษาโปรแกรม (Programming Language) (ราชบัณฑิตยสภา, 2564) ลักษณะของภาษาเรียนรู้การใช้งานได้ง่าย มีโครงสร้างและส่วนประกอบที่สนับสนุนการในการพัฒนาโปรแกรมมากมาย ดังนั้นเหมาะกับการนำไปใช้งานเขียนโปรแกรมตั้งแต่ขั้นพื้นฐาน ไปจนถึงพัฒนาโปรแกรมที่มีความซับซ้อนสูงได้ นอกจากนี้ภาษาไพธอนยังสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกันได้ (Rossum, 2013) ด้วยคุณลักษณะนี้ จึงนิยมนำมาพัฒนาโปรแกรมในปัจจุบัน การใช้งานภาษาไพธอนมีรายละเอียดดังต่อไปนี้

ประวัติความเป็นมาของภาษาไพธอน

Guido Van Rossum เป็นบุคคลเริ่มต้นพัฒนาภาษาไพธอน เริ่มใช้งานครั้งแรกเมื่อปี พ.ศ. 2534 ในลักษณะของการแจกจ่ายให้นักพัฒนาโปรแกรมสามารถใช้งานได้ฟรี นอกเหนือจากนี้ยังได้รับทุนสนับสนุนจากบริษัทขนาดใหญ่ทั้ง Google และ eBay เพื่อให้กลุ่มสมาคมที่มีความสนใจได้พัฒนาภาษาไพธอนให้มีประสิทธิภาพเพิ่มมากขึ้น ปัจจุบัน Guido Van Rossum ได้ร่วมทำงานกับบริษัท Google และได้ประกาศใช้ภาษาไพธอนพัฒนาร่วมกับระบบปฏิบัติการ “แอนดรอยด์ (Android)” พร้อมทั้งรองรับการเติบโตทางด้านตลาดอุตสาหกรรม เช่น การจัดการทรัพยากรองค์กรขนาดใหญ่ (Enterprise Resource Management) การจัดการความสัมพันธ์กับลูกค้า (Customer Relationship Management) ระบบจัดการเนื้อหา (Content Management System) มือถือ เกม เป็นต้น ซึ่งนอกจากนี้ ยังมีการใช้งานโปรแกรมภาษาไพธอนร่วมกับโปรแกรมภาษาอื่น เช่น อุตสาหกรรมบันเทิง ได้ใช้ภาษาไพธอนร่วมกับภาษาซี (C) และซีพลัสพลัส (C++) ในการพัฒนาคอมพิวเตอร์กราฟิก เสียง เป็นต้น (โชติพันธุ์ หล่อเลิศสุนทร และ จิตะพันธุ์ หล่อเลิศสุนทร, 2559)

ภาษาไพธอน มีลักษณะที่เป็นภาษาโปรแกรมในการพัฒนาโปรแกรมอย่างแท้จริง การเขียนโปรแกรมกระทัดรัดและอ่านง่าย เช่น การเขียนโปรแกรมการทำงานเพื่อให้ได้ผลลัพธ์ของโปรแกรม

เหมือนกัน การเขียนด้วยภาษาไพธอนจะมีขนาดสั้นกว่าภาษาโปรแกรม C C++ และ Java ซึ่งเป็นลักษณะพิเศษของภาษาไพธอน นอกจากนี้ยังสนับสนุนการใช้งาน โดยมีเครื่องมือสนับสนุนการใช้งานมากมาย เหมาะสมกับการนำไปใช้พัฒนาโปรแกรมที่มีขนาดใหญ่มากกว่า Shell Script ของ Unix และ Batch File ของ Windows นอกจากนี้ยังมีส่วนที่ช่วยในการตรวจสอบข้อผิดพลาดมากกว่าภาษา C และปัจจุบันกำลังเป็นภาษาที่ระดับสูงมาก (Very High Level Language) เช่น การกำหนดชนิดตัวแปรให้โดยอัตโนมัติ โดยโปรแกรมจะกำหนดชนิดตัวแปรให้เหมาะสมกับตัวแปรโดยอัตโนมัติ โดยไม่ต้องประกาศชนิดของตัวแปร ป้องกันการผิดพลาดจากการกำหนดตัวแปร ซึ่งส่วนนี้เป็นปัญหาใหญ่ของบางภาษา เช่น Awk และ Perl การสร้างตัวแปรด้วยวิธีการง่าย การกำหนดเงื่อนไขหรือวงซ้ำ ที่มีคำสั่งไม่มากเกินไปและมีประสิทธิภาพในการทำงานสูง (Rossum, 2013)

นอกจากนี้ภาษาไพธอนยังใช้ตัวแปลภาษาอินเทอร์พรีเตอร์ ที่จะช่วยลดระยะเวลาในช่วงของการพัฒนาโปรแกรม เพราะจะไม่ประมวลผลสิ่งที่ไม่จำเป็น อินเทอร์พรีเตอร์จะเป็นลักษณะของการโต้ตอบ ซึ่งจะช่วยให้ง่ายต่อการทดลองส่วนต่าง ๆ ของคำสั่ง เพื่อตรวจสอบฟังก์ชัน ชุดคำสั่งทั้งหมดได้ การใช้งานที่มีอยู่ในภาษาโปรแกรม คุณสมบัติที่ดีของภาษาไพธอน คือ ชุดคำสั่งที่พัฒนาขึ้น สามารถนำไปใช้งานกับระบบปฏิบัติการที่มีความแตกต่างกันได้

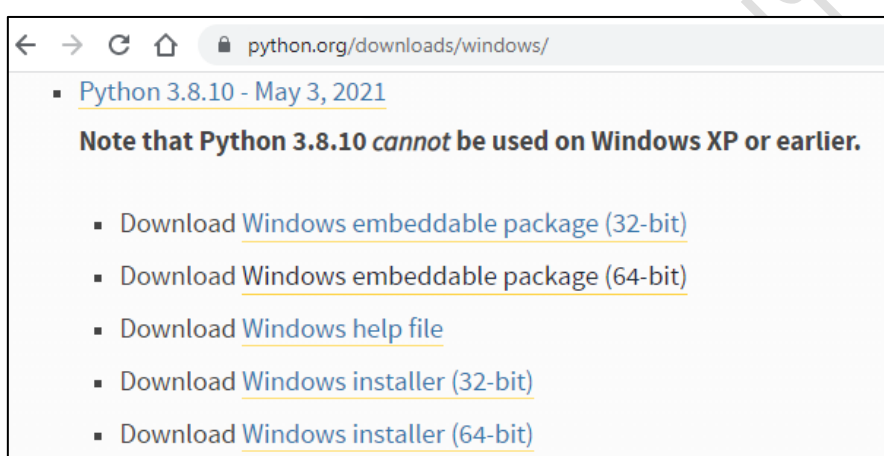
สถาปัตยกรรมของภาษาไพธอน สามารถนำไปใช้ในการพัฒนาโปรแกรมได้ 2 ประเภท คือ โปรแกรมประยุกต์ทั่วไป (Application) และโปรแกรมประยุกต์บนเว็บ (Web Application) โครงสร้างการเขียนโปรแกรมไม่ได้แตกต่างกัน แต่แตกต่างกันที่รูปแบบของการนำไปใช้ เช่น ถ้าใช้งานประยุกต์บนเว็บ ก็จะต้องมีการใช้งานร่วมกับภาษาอื่น เพื่อเพิ่มประสิทธิภาพในการทำงาน เช่น HTML CSS JavaScript และฟังก์ชันการใช้งานร่วมกับฐานข้อมูล นอกเหนือจากนี้ขีดความสามารถและขอบเขตการใช้งานในเชิงลึก สามารถใช้งานโมดูลพื้นฐาน (Standard Module) หรือคำสั่งมาตรฐาน (ราชบัณฑิตยสภา, 2564) ซึ่งเป็นที่เก็บรวบรวมคำสั่งใช้งานในภาษาไพธอน แต่ละโมดูลจะทำหน้าที่แตกต่างกันออกไป เช่น โมดูลที่เกี่ยวข้องกับระบบปฏิบัติการ จะทำหน้าที่ในการคัดลอก (Copy) การตัด (Cut) การจัดการเกี่ยวกับไฟล์หรือโฟลเดอร์ เป็นต้น นอกเหนือจากนี้ยังขึ้นอยู่กับจุดประสงค์ของการนำไปพัฒนา ตัวอย่างเช่น ถ้าต้องการอ่านข้อมูลในเครื่องและส่งผ่านไปทางอีเมล ภาษาไพธอนจำเป็นจะต้องเพิ่มเครื่องมือในการจัดการระบบปฏิบัติการและเพิ่มโมดูลเกี่ยวกับอีเมล (SMTP) เพื่อให้สามารถทำงานได้ตามที่ต้องการครบถ้วน นอกเหนือจากโมดูลที่มีอยู่ในภาษาไพธอน ยังได้เตรียมโมดูลที่ได้รับการพัฒนาจากภาษาโปรแกรมอื่น เช่น C C++ Java .Net และเครื่องมือที่ใช้พัฒนาด้านอื่น เช่น การใช้งานระบบฐานข้อมูล MySQL ซึ่งโมดูลภาษาโปรแกรมไพธอนไม่ได้สร้างไว้ จำเป็นต้องนำคำสั่ง (Library) (ราชบัณฑิตยสภา, 2564) พิเศษที่ใช้งานร่วมกับฐานข้อมูล MySQL นำเข้ามาใช้งานให้สามารถติดต่อกับฐานข้อมูลได้ (โชติพันธุ์ หล่อเลิศสุนทร และ จิตะพันธุ์ หล่อเลิศสุนทร, 2559)

การติดตั้งโปรแกรม

ภาษาไพธอน เป็นภาษาโปรแกรมที่สามารถนำมาใช้งานได้โดยไม่มีค่าใช้จ่าย ดังนั้นผู้พัฒนาได้เปิดช่องทางในการนำไปใช้และการติดตั้งโปรแกรม โดยมีรายละเอียดดังต่อไปนี้

1. การนำชุดภาษาไพธอนมาใช้งาน

การนำภาษาไพธอนมาใช้งาน สามารถนำชุดภาษาไพธอนมาใช้งานได้ผ่านเว็บไซต์ <https://www.python.org/downloads/windows/> โดยมีรายละเอียดดังภาพที่ 1.1



ภาพที่ 1.1 การนำชุดภาษาไพธอนมาใช้งาน

ที่มา: Python (2021)

ภาษาไพธอนในปัจจุบันได้มีการพัฒนามาถึงรุ่นที่ 3 ซึ่งสามารถนำใช้งานได้ในแพลตฟอร์ม (Platform) (ราชบัณฑิตยสภา, 2564) ที่แตกต่างกันและสามารถนำไปใช้งานได้ทั้งระบบปฏิบัติการที่เป็นแบบ 32 บิต และ 64 บิต

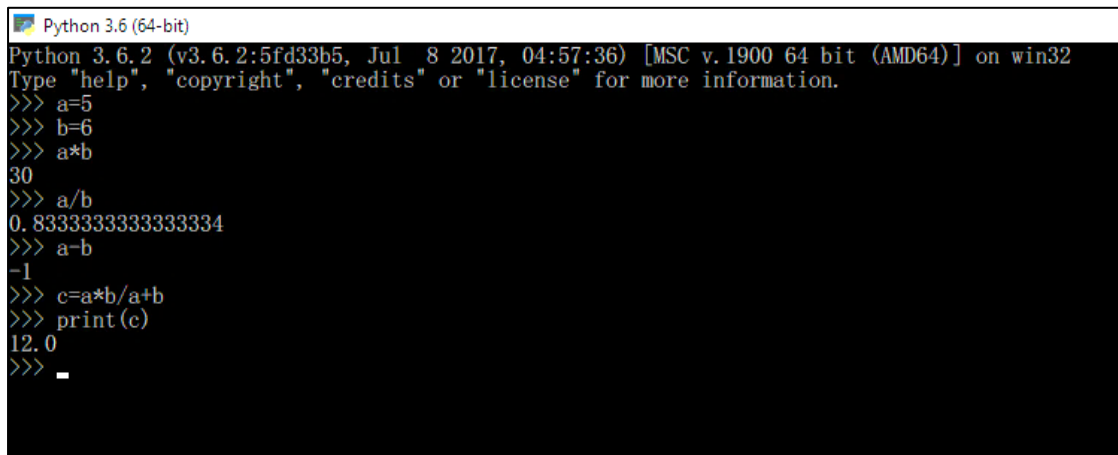
2. การใช้งานภาษาไพธอน

การติดตั้งของภาษาไพธอนประกอบการใช้เอกสารประกอบการสอนเล่มนี้ จะใช้ภาษาไพธอน รุ่น 3.6.2 เมื่อผ่านกระบวนการของการติดตั้งภาษาไพธอน ส่วนประกอบของการใช้งานโปรแกรม ประกอบไปด้วยรายละเอียดดังต่อไปนี้

2.1 Python Command Line

การเขียนโปรแกรมในส่วน Command Line คือ การสร้างคำสั่งบนหน้าต่างพื้นฐานของโปรแกรม เป็นการเขียนทีละบรรทัด เพื่อตรวจสอบการทำงานของแต่ละคำสั่ง ไม่มีฟังก์ชันการใช้งาน

อื่นเพิ่มเติมที่จะอำนวยความสะดวกต่อการใช้งาน (Fangohr, 2015) ส่วนนี้จะมีความสำคัญ ควบคุมการทำงานของภาษาไพธอนทั้งหมด เช่น การนำเข้าโปรแกรมที่สนับสนุนการทำงานร่วมกับภาษาไพธอน การเรียกใช้ไฟล์คำสั่งที่มีอยู่ในภาษาไพธอน เป็นต้น มีรายละเอียดการใช้งานดังภาพที่ 1.2



```
Python 3.6 (64-bit)
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:57:36) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a=5
>>> b=6
>>> a*b
30
>>> a/b
0.8333333333333334
>>> a-b
-1
>>> c=a*b/a+b
>>> print(c)
12.0
>>> _
```

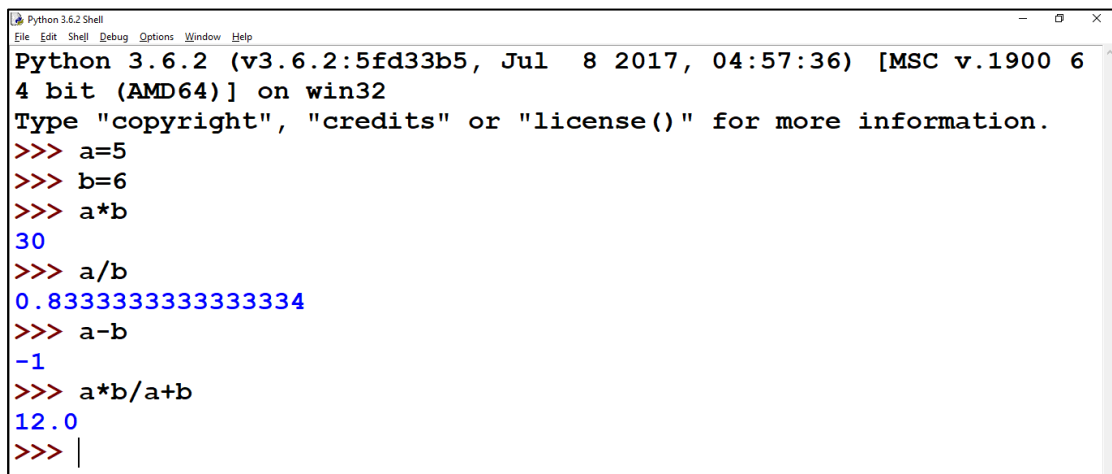
ภาพที่ 1.2 การใช้งานภาษาไพธอน Command Line

การใช้งานภาษาไพธอน Command Line สามารถให้ข้อมูลลงไปโปรแกรม โดยการประมวลผลจะประมวลผลทีละบรรทัดและแสดงผลลัพธ์ที่ได้จากการประมวลแต่ละบรรทัด ดังผลลัพธ์ที่ได้จากภาพที่ 1.2

2.2 Python IDLE

ภาษาไพธอนได้เตรียมเครื่องมือที่มีฟังก์ชันการใช้งานในรูปแบบของส่วนต่อประสานกราฟิกกับผู้ใช้ ที่ช่วยอำนวยความสะดวกต่อการเขียนโปรแกรม เพื่อให้เกิดประสิทธิภาพในการเขียนโปรแกรม เรียกว่า “IDLE (Integrated Development Environment)” ซึ่งสามารถใช้งานสร้างชุดคำสั่งแบ่งออกเป็น 2 ส่วนดังนี้คือ

2.2.1 IDLE Shell คือ การทำงานส่วนของ Shell มีลักษณะการทำงานเหมือนกับ Python Dos Mode คือ การทดสอบการทำงานของคำสั่งทีละบรรทัด จะมีความแตกต่างที่ Python Shell สามารถปรับแต่งส่วนประสานกราฟิกกับผู้ใช้ได้ตามรูปแบบที่ต้องการ ตามรายการเมนูของโปรแกรม เช่น การบันทึกไฟล์ การคัดลอกคำสั่ง การปรับรูปแบบของตัวอักษร เป็นต้น ซึ่งสิ่งเหล่านี้จะอำนวยความสะดวกต่อการใช้งานโปรแกรม ดังภาพที่ 1.3



```

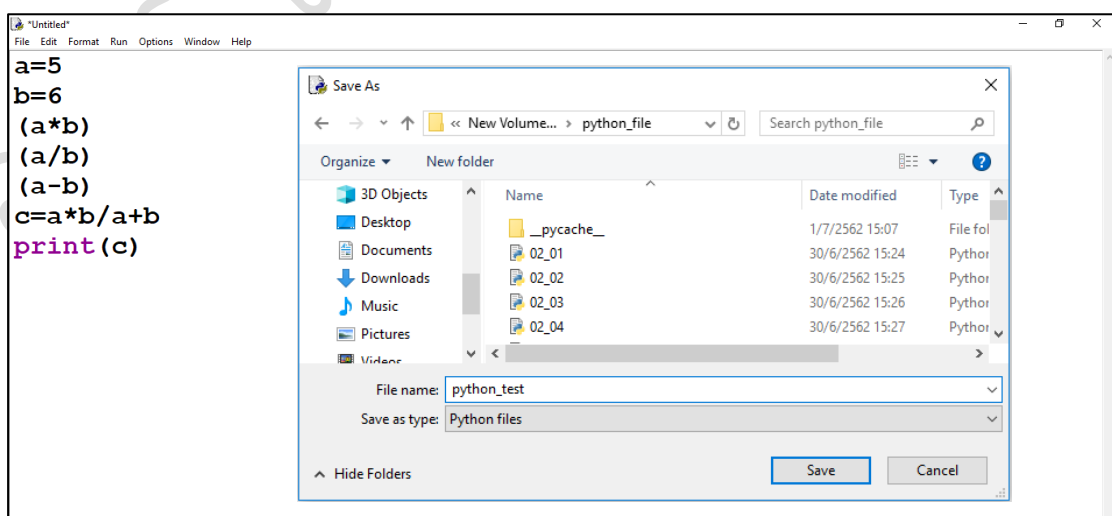
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:57:36) [MSC v.1900 6
4 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> a=5
>>> b=6
>>> a*b
30
>>> a/b
0.8333333333333334
>>> a-b
-1
>>> a*b/a+b
12.0
>>> |

```

ภาพที่ 1.3 การใช้งาน IDLE Shell

การใช้งาน IDLE Shell ดังภาพที่ 1.3 ในเบื้องต้นได้มีการใช้สื่อบอกลักษณะให้เห็นความแตกต่าง เช่น การรับคำสั่งใช้สัญลักษณ์ >>> สิ้นน้ำตาล คำสั่งที่ใช้สีดำ ผลลัพธ์ใช้สีน้ำเงินและการผิดพลาดของโปรแกรมใช้สีแดง เป็นต้น เพื่อแสดงสถานะของคำสั่งที่ใช้ ทำให้สามารถตรวจสอบการใช้คำสั่งเบื้องต้นได้สะดวกต่อการใช้งานเพิ่มมากขึ้น แต่สีเหล่านี้นักเขียนโปรแกรมสามารถกำหนดขึ้นเองได้โดยเลือกในรายการใช้ใน IDLE

2.2.2 IDLE File คือ การสร้างชุดคำสั่งใน IDLE Shell ที่เป็นลักษณะของการทดสอบการทำงานของคำสั่งและการประมวลผล เพื่อตรวจสอบผลลัพธ์ ซึ่งเป็นการใช้งานพื้นฐานการนำไปใช้งานจริงไม่สามารถนำไปใช้งานได้ จะต้องมีการสร้างการทำงานร่วมกันของชุดคำสั่ง ซึ่งจะเก็บไว้ในรูปแบบของไฟล์ ดังภาพที่ 1.4



ภาพที่ 1.4 การใช้งาน IDLE File

การประมวลผลของ IDLE File เป็นการประมวลที่แตกต่างจาก IDLE Shell เนื่องจากต้องเขียนตามขั้นตอนการทำงานของโปรแกรม ซึ่งผลลัพธ์ของค่าที่แสดงจะต้องมีส่วนที่เกี่ยวข้องกับการนำคำสั่งในการแสดงค่ามาใช้ ดังภาพที่ 1.5

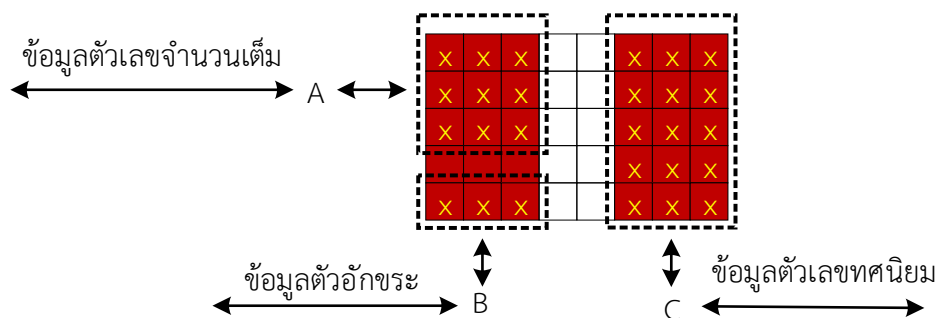
```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:57:36) [MSC v.1900 64
bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/python_file/test.py =====
=====
12.0
>>>
```

ภาพที่ 1.5 การประมวลผลโปรแกรม IDLE File

การแสดงผลลัพธ์ของข้อมูล จะแสดงเมื่อมีการใช้คำสั่ง `print` ซึ่งเป็นคำสั่งในการแสดงข้อมูลเท่านั้น ส่วนค่าอื่นที่ได้ผ่านการประมวลผลทางคณิตศาสตร์ ถูกเก็บไว้ในหน่วยความจำ แต่ไม่ได้ใช้คำสั่งในการแสดงผลลัพธ์ของข้อมูลออกมาจึงไม่แสดง ดังนั้นการสร้างชุดคำสั่งในส่วนของ IDLE File จะเป็นขั้นตอนการทำงานของโปรแกรม การเขียนโปรแกรมควรต้องมีการใช้คำสั่งที่เกี่ยวข้องกับการทำงานของโปรแกรมพื้นฐาน เช่น การรับค่าข้อมูลและแสดงผลค่าข้อมูล คำสั่งนี้มีลักษณะสร้างปฏิสัมพันธ์กับผู้ใช้งานโปรแกรม ดังภาพที่ 1.5 ซึ่งตัวอย่างของโปรแกรมรหัสต้นฉบับ (Source Code) (ราชบัณฑิตยสภา, 2564) ประกอบการอธิบายเนื้อหาแต่ละบท สามารถนำมาทดสอบการใช้งานได้ที่ <http://www.com-tech.ubru.ac.th/~ekarach/pcp.php>

ตัวแปร

ตัวแปร (Variable) หมายถึง การนิยามชื่อที่ทำหน้าที่อ้างอิงการเก็บข้อมูลในตำแหน่งของหน่วยความจำจริง แทนการใช้ตำแหน่งที่อยู่จริงของหน่วยความจำ (Sebesta, 2016) ซึ่งยากต่อการใช้งานและจดจำ ดังนั้นเพื่อสะดวกในการเรียกใช้งานและตรวจสอบ จึงกำหนดชื่อที่สามารถจดจำได้ง่าย โดยการอ้างอิงผ่านชื่อและชนิดของข้อมูลที่ใช้เก็บ รูปแบบอักษรหรือสัญลักษณ์ที่ใช้ในการอ้างอิงถึงข้อมูลนำมาเก็บในตัวแปร จะต้องตรงกับชนิดข้อมูลที่กำหนดให้กับตัวแปรนั้นและการจะนำข้อมูลไปใช้งานก็ต้องพิจารณาชนิดของตัวแปรนั้นเช่นกัน (บัญชา ปะสีละเตสัง, 2558) นอกเหนือจากนี้ ตัวแปรจะทำหน้าที่ในการจัดเก็บข้อมูลไว้ชั่วคราวในหน่วยความจำและการกำหนดชื่อให้กับตัวแปร ทำให้สามารถอ้างอิงชื่อตัวแปรเหล่านั้นเพื่อนำค่าที่เก็บไว้อยู่ใน ตัวแปรดังกล่าวมาใช้งานได้ (โอภาส เอี่ยมสิริวงศ์ และ สมโภชน์ ชื่นเอี่ยม, 2560) ลักษณะการทำงานของตัวแปรดังกล่าวแสดงลักษณะแผนภาพได้ดังต่อไปนี้



ภาพที่ 1.6 การใช้ตัวแปรอ้างอิงกับหน่วยความจำ

จากภาพที่ 1.6 การนำข้อมูลไปเก็บไว้ในหน่วยความจำของคอมพิวเตอร์ และการดึงข้อมูลออกมาใช้ จะต้องมิตัวแปรซึ่งเป็นตัวกลางอ้างอิงกับหน่วยความจำ การกำหนดตัวแปรนั้นจะต้องมีการพิจารณาถึงความเหมาะสมของข้อมูลที่จะนำผ่านตัวแปรเข้าไป เพราะการกำหนดตัวแปรนอกเหนือจากชื่อของตัวแปร จะต้องกำหนดชนิดของตัวแปรที่ใช้ในการเก็บข้อมูล ตัวอย่างจากภาพที่ 1.6 ถ้าต้องการเก็บข้อมูลอายุของนักเรียนควรเก็บไว้ในรูปแบบของตัวเลขจำนวนเต็มดังตัวแปร A เพราะถ้าหากเก็บเป็นข้อมูลตัวเลขทศนิยม จะต้องจองพื้นที่ในหน่วยความจำมากกว่าและไม่มีความเป็นในการเก็บหลังจุดทศนิยม ซึ่งถ้าเป็นโปรแกรมขนาดใหญ่และมีการกำหนดไม่เหมาะสมกับการเก็บข้อมูล จะทำให้สูญเสียทรัพยากรและความเร็วในการประมวลผลจะช้าลง ดังนั้นนำไปเขียนโปรแกรม ต้องพิจารณาความเหมาะสมในด้านการใช้ในหน่วยความจำ เพื่อให้ประสิทธิภาพในการทำงานสูงสุด

การสร้างโปรแกรมสิ่งที่มีความสำคัญนั่นคือ ข้อมูลที่เกิดขึ้น อาจจะเกิดขึ้นจากข้อมูลนำเข้า ข้อมูลที่ได้จากการประมวลผลและข้อมูลนำออก จะต้องมีการพักหรือเก็บไว้ในหน่วยความจำก่อน จากนั้นจึงนำข้อมูลออกไปใช้งานต่อ ซึ่งข้อมูลในหน่วยความจำดังกล่าว ตัวแปรจะเป็นสิ่งสำคัญในการอ้างอิงข้อมูลในหน่วยความจำ ตัวอย่างการทำงานของตัวแปร ดังโปรแกรมตัวอย่างที่ 1.1

ตัวอย่างที่ 1.1 การใช้งานตัวแปร

บรรทัดที่	คำสั่ง
1	a = 5
2	b = 4
3	c = a+b
4	print (c)
ผลลัพธ์	
9	

ตัวอย่างที่ 1.1 กำหนดตัวแปรทั้งหมด 3 ตัวแปร ประกอบด้วย a b และ c โดยบรรทัดแรกให้ a มีค่าเท่ากับ 5 ตัวแปร a จะเก็บค่านี้เอาไว้ไปเก็บในหน่วยความจำ ตัวแปร b จะมีค่าเท่ากับ 4 และเก็บค่า 4 ไว้ในหน่วยความจำ จากนั้นบรรทัดที่ 3 ทำการดึงค่าที่อยู่ในตัวแปร a และตัวแปร b ซึ่งตัวแปรแต่ละตัวแปรจะดึงข้อมูลที่เก็บเอาไว้ในข้างต้นมาทำการบวกกัน ผลลัพธ์ที่ได้เอามาเก็บไว้ในตัวแปร c และตัวแปร c จะเอาผลลัพธ์ที่ได้คือ 9 มาเก็บไว้ในหน่วยความจำและบรรทัดที่ 4 ใช้คำสั่ง print ให้ทำการดึงข้อมูลของตัวแปร c มาแสดงผล ดังนั้นผลลัพธ์ที่ได้จะมีค่าเท่ากับ 9

จากการทำงานดังกล่าว ตัวแปรมีความสำคัญมากในการเขียนโปรแกรมคอมพิวเตอร์ ลักษณะของการใช้งานตัวแปรในภาษาไพธอน มีรายละเอียดการใช้งานตัวแปรดังต่อไปนี้

1. การตั้งชื่อตัวแปร

การตั้งชื่อของตัวแปรนั้น ภาษาไพธอนเข้มงวดในการกำหนดตัวแปร โดยมีข้อกำหนดในการตั้งชื่อตัวแปร (Halterman, 2016) มีรายละเอียดตามข้อกำหนดต่อไปนี้

1.1 ชื่อตัวแปรต้องขึ้นต้นด้วยอักษรตัวพิมพ์เล็กหรือตัวพิมพ์ใหญ่ และตามด้วยอักษรหรือตัวเลข ต่าง ๆ แต่ห้ามเว้นช่องว่าง

1.2 ชื่อตัวแปรสามารถขึ้นต้นด้วยสัญลักษณ์ขีดเส้นใต้ (ราชบัณฑิตยสถาน, 2564) หรือ underscore (_) เพียงสัญลักษณ์เดียวเท่านั้น

1.3 การตั้งชื่อตัวแปร โดยใช้ตัวอักษรพิมพ์เล็กและพิมพ์ใหญ่ จะมีความหมายในชื่อตัวแปรต่างกัน

1.4 การตั้งชื่อของตัวแปรจะต้องไม่ซ้ำกับคำสงวน

1.5 การตั้งชื่อตัวแปร ตั้งชื่อให้สอดคล้องกับข้อมูลที่ใช้ในการเก็บ เพราะเมื่อมีการตรวจสอบและความความเข้าใจย้อนหลัง จะสามารถทำความเข้าใจได้ง่ายและรวดเร็วเพิ่มมากขึ้น

หลักการตั้งชื่อของตัวแปรตามหลักเกณฑ์ที่กำหนดไว้ สามารถนำไปสร้างเป็นตัวแปรใช้งานในภาษาไพธอน มีรายละเอียดดังตัวอย่างตารางที่ 1.1

ตารางที่ 1.1 ตัวอย่างการตั้งชื่อตัวแปร

ชื่อตัวแปร	ผลลัพธ์	สาเหตุ
name_11= "Ekarach"	Ekarach	ใช้ได้
Surname2= "Thammasa"	Thammasa	ใช้ได้
_number=12	12	ใช้ได้
aaa= "Computer Programming"	Computer Programming	ใช้ได้ : ไม่เหมาะสม ชื่อไม่สื่อ ความหมาย
pass = "True"	invalid syntax	ใช้ไม่ได้ : ซ้ำกับคำสงวน

ตารางที่ 1.1 ตัวอย่างการตั้งชื่อตัวแปร (ต่อ)

ชื่อตัวแปร	ผลลัพธ์	สาเหตุ
tel number="0817603302"	invalid syntax	ใช้ไม่ได้ : มีช่องว่าง
0address="Ubon Ratchathani"	invalid syntax	ใช้ไม่ได้ : ขึ้นต้นด้วยตัวเลข
@email="ekarach19@gmail.com"	invalid syntax	ใช้ไม่ได้ : ขึ้นต้นด้วยตัวอักษร

การตั้งชื่อตัวแปรในตารางที่ 1.1 เมื่อนำไปใช้งาน โปรแกรมจะทำการตรวจสอบ หากพบผิดกฎเกณฑ์ที่กำหนด โปรแกรมจะทำการแจ้งเตือน invalid syntax แจ้งการกำหนดรูปแบบที่ไม่ถูกต้อง และไม่สามารถนำไปใช้งานได้

2. ประเภทของตัวแปรพื้นฐาน

ตัวแปรทำหน้าที่ในการเก็บข้อมูล ลักษณะของข้อมูลที่ใช้ในการเก็บของแต่ละตัวแปร มีลักษณะที่แตกต่างกัน โดยสามารถแยกประเภทของตัวแปร มีรายละเอียดดังตารางที่ 1.2

ตารางที่ 1.2 ประเภทของตัวแปรพื้นฐาน

ประเภท	การเก็บข้อมูล
str	ข้อมูลประเภทตัวอักษร อักษรหรือข้อความ
int	ข้อมูลประเภทจำนวนเต็มบวกและเต็มลบ
float	ข้อมูลประเภทจำนวนจริง ซึ่งได้แก่ ทศนิยม ยกกำลัง เป็นต้น
bool	ข้อมูลประเภทบูลีน (จริง,เท็จ)
complex	ข้อมูลประเภทตัวเลขเชิงซ้อน

ที่มา : Kalb (2016)

ประเภทตัวแปรตารางที่ 1.2 สามารถเก็บข้อมูลได้หลายประเภท เมื่อนำไปใช้ในการเขียนโปรแกรม จะต้องกำหนดประเภทของตัวแปรให้เหมาะสม มีรูปแบบและรายละเอียดการเก็บข้อมูลของตัวแปร ดังโปรแกรมตัวอย่างที่ 1.2

ตัวอย่างที่ 1.2 การแสดงชนิดของตัวแปรตามข้อมูลที่เก็บ

บรรทัดที่	ชุดคำสั่ง
1	a="Computer"

2	b=4
3	c=4.0
4	d= "4.0"
5	e=True
6	f=70.7+2.7j
7	print (type(a))
8	print (type(b))
9	print (type(c))
10	print (type(d))
11	print (type(e))
12	print (type(f))
ผลลัพธ์	
<pre><class 'str'> <class 'int'> <class 'float'> <class 'str'> <class 'bool'> <class 'complex'></pre>	

โปรแกรมตัวอย่างที่ 1.2 แสดงชนิดของตัวแปรที่ได้ทำการเก็บตามข้อมูลที่อยู่ในตัวแปรนั้น ซึ่งเมื่อโปรแกรมประมวลผล จะวิเคราะห์ข้อมูลและรูปแบบที่อยู่ในตัวแปรมีรายละเอียดดังต่อไปนี้

ตัวแปร a และ d โปรแกรมวิเคราะห์เป็นข้อมูลประเภท str เพราะข้อมูลอยู่ภายใต้เครื่องหมาย “ ” โปรแกรมจะวิเคราะห์เป็นข้อมูลประเภท str ทั้งหมด

ตัวแปร b โปรแกรมวิเคราะห์เป็นข้อมูลประเภท int เนื่องจากไม่ได้อยู่ภายใต้เครื่องหมาย “ ” และไม่มีจุดทศนิยม

ตัวแปร c โปรแกรมวิเคราะห์เป็นข้อมูลประเภท float เนื่องจากไม่ได้อยู่ภายใต้เครื่องหมาย “ ” และมีจุดทศนิยม

ตัวแปร e โปรแกรมวิเคราะห์เป็นข้อมูลประเภท bool เนื่องจากไม่ได้อยู่ภายใต้เครื่องหมาย “ ” และมีค่าเงื่อนไขเป็นจริง

ตัวแปร f โปรแกรมวิเคราะห์เป็นข้อมูลประเภท complex เนื่องจากไม่ได้อยู่ภายใต้เครื่องหมาย “ ” และมีการกำหนดค่าเป็นตัวเลขเชิงซ้อน

2.1 ตัวแปรประเภท str

การเก็บข้อมูลในลักษณะของอักขระ อักษร หรือข้อความ ตัวแปรที่ใช้ในการเก็บค่าข้อมูลจะเป็นประเภท str ซึ่งในภาษาไพธอนจะใช้การเข้ารหัสเป็นแบบ UTF-8 Unicode เป็นหลักเพื่อรองรับปริมาณของภาษาที่ใช้งานทั่วโลก การกำหนดข้อมูลให้เป็นลักษณะ str จะใช้เครื่องหมายอัฒประกาศคู่ “ ” หรือ อัฒประกาศเดี่ยว ‘ ’ (ราชบัณฑิตยสถาน, 2564) เป็นการกำหนดการเก็บค่าข้อมูลเป็นแบบ str และใช้งานร่วมกับสัญลักษณ์พิเศษที่จะทำให้ไม่เกิดปัญหาในการประมวลผลที่เกี่ยวข้องกับสัญลักษณ์หรือรูปแบบของการเขียนคำสั่ง เรียกว่า “Escape Sequence Character” ซึ่งสัญลักษณ์ที่ใช้มีรายละเอียดดังต่อไปนี้

ตารางที่ 1.3 สัญลักษณ์พิเศษที่ใช้กับกับตัวแปรแบบ str

สัญลักษณ์	หน้าที่	ตัวอย่าง	ผลลัพธ์
\\	แสดงอักขระ Backslash	print “\\”	\
\'	แสดงอักขระ single-quote	print “\'”	'
\"	แสดง double quote	print “\”	“
\a	รหัส ASCII bell ที่จะสร้างเสียงระฆังเตือน	print “\a”	N/A
\b	รหัส ASCII backspace (BS) ลบอักขระที่อยู่ก่อนหน้า 1 ตัว	print "ab" + "\b" + "c"	ac
\f	รหัส ASCII formfeed (FF)	print "hello\fworld"	hello world
\n	รหัส ASCII linefeed (LF)	print "hello\nworld"	hello world
\N{name}	แสดงอักขระจาก Unicode database	print u"\N{DAGGER}"	†
\r	รหัส ASCII carriage return (CR).	print "123456\rXX_XX"	XX_XX6
\t	รหัส ASCII horizontal tab (TAB)	print "\t* hello"	* hello
\t	รหัส ASCII vertical tab (VT).	N/A	N/A
\uxxxx	แสดงตัวอักขระ Unicode 16-bit บนเลขฐาน 16	print u"\u041b"	Л

ตารางที่ 1.3 สัญลักษณ์พิเศษที่ใช้กับกับตัวแปรแบบ str (ต่อ)

สัญลักษณ์	หน้าที่	ตัวอย่าง	ผลลัพธ์
\Uxxxxxxx	แสดงตัวอักษร Unicode 32-bit บนเลขฐาน 16	print u"\U000001a9"	Σ
\ooo	แสดงอักขระ บนเลขฐาน 8	print "\043"	#
\xhh	แสดงอักขระ บนเลขฐาน 16	print "\x23"	#

ที่มา : Przywóski (2015)

2.2 ตัวแปรประเภท int

ตัวแปรประเภท int คือ ตัวแปรที่ใช้ในการเก็บข้อมูลในลักษณะของตัวเลขที่เป็นจำนวนเต็ม สามารถเก็บค่าข้อมูลทั้งในรูปแบบของฐานสอง ฐานแปดและฐานสิบหก ซึ่งมีคำสั่งให้แสดงรูปแบบของการแสดงค่าและแปลงค่าจากเลขฐาน มีรายละเอียดดังต่อไปนี้

ตารางที่ 1.4 คำสั่งแสดงค่าและแปลงค่าจากเลขฐานตัวแปรประเภท int

เลขฐาน	อักขระนำหน้า	คำสั่ง
สอง (Binary)	0b	bin
แปด (Octal)	0o	oct
สิบ (Decimal)	ไม่มี	int
สิบหก (Hexadecimal)	0x	hex

ที่มา : โชติพันธุ์ หล่อเลิศสุนทร และ ฐิตะพันธุ์ หล่อเลิศสุนทร (2559)

คำสั่งในการจัดการรูปแบบของข้อมูลประเภท int ซึ่งอยู่ในรูปของเลขฐาน สามารถนำคำสั่งไปใช้แสดงค่าและแปลงค่าจากเลขฐานตัวแปรประเภท int ดังรายละเอียดโปรแกรมตัวอย่างที่ 1.3

ตัวอย่างที่ 1.3 การเก็บข้อมูลในรูปแบบของ int

บรรทัดที่	ชุดคำสั่ง
1	a=bin(30)
2	b=oct(50)

3	c=hex(100)
4	d=0b1011
5	e=0o41
6	f=0x2c
7	print(a)
8	print(b)
9	print(c)
10	print(d)
11	print(e)
12	print(f)
ผลลัพธ์	
0b11110	
0o62	
0x64	
11	
33	
44	

โปรแกรมตัวอย่างที่ 1.3 บรรทัดที่ 1 ถึง 3 การเก็บข้อมูลในตัวแปรรูปแบบของการแปลงจากเลขฐานสิบเป็นเลขฐานสอง ฐานแปดและฐานสิบหกและบรรทัดที่ 4 ถึง 6 คำสั่งที่ใช้ในการแปลงจากฐานสอง ฐานแปด และฐานสิบหก เป็นเลขฐานสิบ จากรูปแบบการเก็บข้อมูลประเภท int ในภาษาโปรแกรมแสดงให้เห็นข้อดี มีความยืดหยุ่นต่อการเก็บข้อมูลและเหมาะสำหรับการเก็บข้อมูลที่ซับซ้อน ซึ่งเป็นจุดเด่นของภาษาไพธอน

2.3 ตัวแปรประเภท float

ตัวแปรประเภท float คือ ตัวแปรที่ใช้ในการเก็บข้อมูลในลักษณะของตัวเลขทศนิยม จะใช้เทคนิคการคำนวณของอัลกอริทึมที่เป็นที่นิยมกันทั่วไปของ David Goy จะมีความคลาดเคลื่อนไม่มาก เหมาะสมกับการนำไปใช้ในงานทั่วไป (โชติพันธุ์ หล่อเลิศสุนทร และ จิตะพันธุ์ หล่อเลิศสุนทร, 2559) การใช้งานตัวแปรประเภท float มีรูปแบบ ดังตัวอย่างที่ 1.4

ตัวอย่างที่ 1.4 การเก็บข้อมูลในรูปแบบของ float

บรรทัดที่	ชุดคำสั่ง
1	a,b,c,d =10,3,20,9.9

2	e=a/b
3	f=c/d
4	pi=22/7
5	print(a,',',b,',',c,',',d)
6	print(e)
7	print(f)
8	print(pi)
9	print(pi*(e**2))
10	print(type(a),type(b),type(c),type(d))
11	print(type(e),type(f),type(pi))
ผลลัพธ์	
10 , 3 , 20 , 9.9	
2.3333333333333335	
2.0202020202020203	
2.142857142857143	
34.920634920634924	
<class 'int'> <class 'int'> <class 'int'> <class 'float'>	
<class 'float'> <class 'float'> <class 'float'>	

โปรแกรมตัวอย่างที่ 1.4 ผลลัพธ์ที่ได้จากการประมวลผล พบว่าตัวแปร c เก็บข้อมูลประเภท int และตัวแปร d มีการเก็บข้อมูลประเภท float เมื่อใช้คำสั่ง type() ในการแสดงประเภทของตัวแปร ซึ่งทั้งสองตัวแปรมีรูปแบบการเก็บข้อมูลแตกต่างกัน แต่สามารถนำมาคำนวณร่วมกันได้ ซึ่งมีภาษาคอมไพเลอร์บางประเภทไม่อนุญาตให้ประมวลผลเช่นนี้ได้ การทำงานลักษณะเช่นนี้มีความสะดวกในการใช้งานตัวแปรมากและการปัดเศษนั้นจะมีจำนวนหลักของการปัดเศษไม่เหมือนกัน ทั้งนี้ขึ้นอยู่กับการวิเคราะห์ความเหมาะสมของภาษาไพธอน

2.4 ตัวแปรประเภท bool

การเก็บข้อมูลประเภทตัวแปรบูลีน (Boolean) คือ ตัวแปรที่ใช้ในการเก็บค่าข้อมูล มีลักษณะเป็นจริงและเท็จ เพื่อใช้เป็นเงื่อนไขในการเปรียบเทียบ มีลักษณะการใช้งานพื้นฐานของตัวแปร ดังโปรแกรมตัวอย่างที่ 1.5

ตัวอย่างที่ 1.5 การเก็บข้อมูลในรูปแบบของ bool

บรรทัดที่	ชุดคำสั่ง
1	a=True
2	b=True
3	print (a)
4	print (not(b))
5	print ((a)and(b))
6	print ((a)and not(b))
7	print (not(a)and(b))
8	print (not(a)and not(b))
9	print ((a)or(b))
10	print ((a)or not(b))
11	print (not(a)or(b))
12	print (not(a)or not(b))
ผลลัพธ์	
True	
False	
True	
False	
False	
False	
True	
True	
True	
False	

การใช้งานตัวแปรประเภท bool มีลักษณะของการตรวจสอบเงื่อนไขที่มีการเชื่อมต่อหลายเงื่อนไขร่วมกัน ดังนั้นจึงต้องใช้ตัวดำเนินการในการเปรียบเทียบ ยกตัวอย่างเช่น and or not เป็นต้น ดังโปรแกรมตัวอย่างที่ 1.5

2.5 ตัวแปรประเภท complex

ตัวแปรประเภท complex คือ ตัวแปรเก็บข้อมูลประเภทเชิงซ้อน โครงสร้างลักษณะของรูปแบบของตัวเลขเชิงซ้อน ประกอบไปด้วยค่า Real และ Imaginary มีรูปแบบการทำงานดังโปรแกรมตัวอย่างที่ 1.6

ตัวอย่างที่ 1.6 การเก็บข้อมูลในรูปแบบของ complex

บรรทัดที่	ชุดคำสั่ง
1	<code>a=50.5+2.2j</code>
2	<code>b=5.4+1.7j</code>
3	<code>c=10j</code>
4	<code>d=12</code>
5	<code>print(a)</code>
6	<code>print(b)</code>
7	<code>print(a+b)</code>
8	<code>print(a-b)</code>
9	<code>print(a+c)</code>
10	<code>print(a+d)</code>
ผลลัพธ์	
<code>(50.5+2.2j)</code> <code>(5.4+1.7j)</code> <code>(55.9+2.9000000000000004j)</code> <code>(45.1+0.5000000000000002j)</code> <code>(50.5+12.2j)</code> <code>(62.5+2.2j)</code>	

ตัวเลขเชิงซ้อนจะประกอบไปด้วย 2 ส่วนคือ ตัวเลขปกติจะเรียกค่า Real และตัวเลขที่ลงท้ายด้วยอักษร j ซึ่งจะเรียกค่า Imaginary ซึ่งในการคำนวณภาษาโปรแกรมจะทำการตรวจสอบค่าและทำการคำนวณในกลุ่มเดียวกัน ดังโปรแกรมตัวอย่างที่ 1.6

ลักษณะของตัวแปรในภาษาไพธอน จะมีลักษณะแตกต่างจากภาษาโปรแกรมอื่น โดยการสร้างตัวแปรในโปรแกรมที่นิยมนำมาใช้ เช่น C C++ Java และ Visual Basic จะต้องดำเนินการกำหนดประเภทให้กับตัวแปรก่อน ถึงจะสามารถนำตัวแปรนั้นไปใช้งานได้ ถ้านำไปเก็บค่าที่ไม่ตรงกับ

ชนิดข้อมูลที่กำหนด จะเกิดการผิดพลาดของโปรแกรมหรือการเก็บข้อมูลไม่ตรงกับความต้องการ แต่ในภาษาไพธอน ไม่ต้องกำหนดชนิดของตัวแปร เมื่อมีการประมวลชุดคำสั่งของภาษาไพธอนจะทำการวิเคราะห์ข้อมูลและกำหนดตัวแปรตามลักษณะประเภทข้อมูลที่ใช้ในการเก็บ ซึ่งทำให้สะดวกต่อการใช้งานตัวแปร

คำสงวน

คำสงวน (Reserved Word) หมายถึง กลุ่มคำที่ถูกสร้างขึ้นและใช้งานในภาษาคอมพิวเตอร์ ซึ่งคำเหล่านี้ไม่สามารถนำไปตั้งชื่อเป็นตัวแปรได้ (Langtangen, 2014) โดยคำสั่งที่มีในภาษาคอมพิวเตอร์สงวนไว้ โดยไม่สามารถนำคำเหล่านี้ไปกำหนดเป็นชื่อต่าง ๆ ในการเขียนโปรแกรมได้ (บัญชา ปะสีละเตสัง, 2558)

ดังนั้น คำสงวน เป็นชื่อคำสั่งหรือชื่อฟังก์ชันที่อยู่ในภาษาคอมพิวเตอร์ ซึ่งจะไม่อนุญาตให้นำชื่อนี้ไปใช้ได้ เช่น ตัวแปร ค่าคงที่ เป็นต้น คำสงวนมีรายละเอียดดังต่อไปนี้

ตารางที่ 1.5 คำสงวน

and	False	nonlocal
as	finally	not
assert	for	or
break	from	pass
class	global	raise
continue	if	return
def	import	try
del	in	True
elif	is	while
else	lambda	with
except	None	yield

ที่มา : Halterman (2016)

การนำคำสงวนไปใช้งานการเขียนคำสั่ง ภาษาไพธอนจะไม่อนุญาตให้ใช้งาน ซึ่งเมื่อมีการนำไปใช้ จะเกิดการแจ้งเตือนดังโปรแกรมตัวอย่างที่ 1.7

ตัวอย่างที่ 1.7 การสร้างตัวแปรซ้ำกับคำสงวน

```
>>> in =2
SyntaxError: invalid syntax
>>> False=5
SyntaxError: can't assign to keyword
>>> and = 5
SyntaxError: invalid syntax
>>> class = 'computer'
SyntaxError: invalid syntax
>>> while = 20
SyntaxError: invalid syntax
>>> False = 'true'
SyntaxError: can't assign to keyword
>>> raise=x
SyntaxError: invalid syntax
```

การเขียนโปรแกรมในยุคเริ่มแรก การกำหนดชื่อตัวแปร จะเกิดการซ้ำกับคำสงวนค่อนข้างมาก บางครั้งภาษาโปรแกรม ไม่ได้แจ้งเตือนข้อความระบุตั้งชื่อซ้ำกับคำสงวน ดังนั้นจึงมีความลำบากมากที่จะตรวจสอบความผิดพลาดที่เกิดขึ้น ซึ่งต้องใช้ระยะเวลานานในการตรวจสอบ แต่ในภาษาโปรแกรมปัจจุบัน ได้พัฒนาให้มีการใช้สีเป็นการระบุลักษณะของคำสั่งเป็นชนิดใดในเบื้องต้น ซึ่งอำนวยความสะดวกต่อการตรวจสอบข้อผิดพลาดของโปรแกรม

ค่าคงที่

ค่าคงที่ หมายถึง ตัวแปรที่มีค่ากำหนดขึ้นตายตัว ไม่สามารถเปลี่ยนแปลงค่าได้ (โอภาส เอี่ยมสิริวงศ์ และ สมโภชน์ ชื่นเอี่ยม, 2560) โดยทั่วไปขนาดค่าข้อมูลของตัวแปร จะถูกกำหนดค่าโดยย่อให้กับตัวแปรในลักษณะของคงที่ (Sheppard, 2014) ค่าคงที่ของตัวแปรจะไม่เปลี่ยนแปลงตลอดการทำงานของโปรแกรม มีประโยชน์ คือ เมื่อต้องการเก็บค่าที่ไม่ให้มีการเปลี่ยนแปลง เช่น เก็บค่าคงที่ π ที่ใช้ในการคำนวณเกี่ยวกับวงกลม (ธีรวัฒน์ ประกอบผล, 2558) ดังโปรแกรมตัวอย่างที่ 1.8

ตัวอย่างที่ 1.8 การกำหนดค่าคงที่ให้กับตัวแปร

บรรทัดที่	ชุดคำสั่ง
1	pi=22/7
2	r = float(input("รัศมีของวงกลม = "))
3	area=pi*(r**2)
4	print("ขนาดพื้นที่ = ",area)
ผลลัพธ์	
รัศมีของวงกลม = 1.25 ขนาดพื้นที่ = 4.910714285714286	

การกำหนดค่าคงที่ให้กับตัวแปรโปรแกรมตัวอย่างที่ 1.8 โดยการกำหนดค่าตัวแปร pi เป็นค่าคงที่มีค่าเท่ากับ 22/7 ทุกครั้งที่มีการประมวลผลโปรแกรม ค่าในตัวแปรจะมีค่านี้เสมอไม่มีการเปลี่ยนแปลง ซึ่งแตกต่างจากค่า r ที่ค่าข้อมูลจะเปลี่ยนแปลงตามข้อมูลที่ป้อนข้อมูลเข้าสู่โปรแกรม

ตัวดำเนินการ

ตัวดำเนินการ หรือ โอเปอเรเตอร์ (Operator) หมายถึง สัญลักษณ์พิเศษที่ใช้ทำงานร่วมกับตัวแปร (Halterman, 2016) เพื่อดำเนินการประมวลทางด้านคณิตศาสตร์และตรรกะ (บัญชา ปะสีละเตสัง, 2558) หรือการกระทำบางอย่างกับข้อมูลและได้ผลลัพธ์ออกมา (ธีรวัฒน์ ประกอบผล, 2558) ซึ่งโอเปอเรเตอร์ที่นำไปใช้งานร่วมกับตัวแปร ใช้ในการคำนวณตามชนิดของตัวแปรและสถานการณ์ บางครั้งเครื่องหมายโอเปอเรเตอร์เหมือนกัน แต่การทำงานแตกต่างกัน (โชติพันธุ์ หล่อเลิศสุนทร และ ฐิตะพันธุ์ หล่อเลิศสุนทร, 2559) โอเปอเรเตอร์นั้น ได้แบ่งกลุ่มตามประเภทของการนำไปใช้งานประกอบด้วย ตัวดำเนินการทางคณิตศาสตร์ ตัวดำเนินการกำหนดค่า ตัวดำเนินการเปรียบเทียบ ตัวดำเนินการทางตรรกะ ตัวดำเนินการระบุลักษณะ ตัวดำเนินการสมาชิกและตัวดำเนินการแบบปิด ซึ่งแต่ละตัวดำเนินการจะมีหน้าที่และการทำงานที่แตกต่างกัน การนำไปใช้งานให้ถูกต้อง พิจารณารายละเอียดหน้าที่ตัวดำเนินการแต่ละประเภทดังต่อไปนี้

1. ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators) หมายถึง การใช้สัญลักษณ์ทางคณิตศาสตร์ ดำเนินการระหว่างข้อมูลเพื่อให้ได้ผลลัพธ์ที่ต้องการ สัญลักษณ์ตัวดำเนินการทางคณิตศาสตร์มีรายละเอียดของสัญลักษณ์ดังต่อไปนี้

ตารางที่ 1.6 ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการ	ความหมาย	รูปแบบ	ตัวอย่าง	ผลลัพธ์
+	การบวก	$x + y$	<code>print(7+5)</code>	12
-	การลบ	$x - y$	<code>print(7-5)</code>	2
*	การคูณ	$x * y$	<code>print(7*5)</code>	35
/	การหาร	x / y	<code>print(7/5)</code>	1.4
%	หารเอาเศษ	$x \% y$	<code>print(7%5)</code>	2
**	ยกกำลัง	$x ** y$	<code>print(7**5)</code>	16807
//	หารปัดเศษ	$x // y$	<code>print(7//5)</code>	1

ที่มา : Halterman (2016)

การใช้ตัวดำเนินการทางคณิตศาสตร์ ผลลัพธ์จากประมวลผลที่ได้จะมีความแตกต่างกัน ดังนั้นการนำไปใช้จะต้องพิจารณาถึงความถูกต้องของผลลัพธ์ที่ได้ สัญลักษณ์ทางคณิตศาสตร์บางสัญลักษณ์มีความหมายใกล้เคียงกัน ซึ่งจะทำให้การนำไปใช้ผิดได้ เช่น การหารปกติ การหารเอาเศษ หรือการหารปัดเศษ เป็นต้น

2. ตัวดำเนินการกำหนดค่า

ตัวดำเนินการกำหนดค่า (Assignment Operators) หมายถึง การกำหนดรูปแบบของการประมวลผลข้อมูล ผลลัพธ์ที่ได้จากการประมวลผลเหมือนกัน แต่การกำหนดค่านั้นแตกต่างกันและ การใช้สัญลักษณ์มากกว่าหนึ่งสัญลักษณ์ มีรายละเอียดดังต่อไปนี้

ตารางที่ 1.7 ตัวดำเนินการกำหนดค่า

ตัวดำเนินการ	รูปแบบ	รูปแบบที่เหมือน	ผลลัพธ์ (x = 7)
=	$x = 7$	$x = 7$	7
+=	$x += 5$	$x = x + 5$	12
-=	$x -= 5$	$x = x - 5$	2
*=	$x *= 5$	$x = x * 5$	35
/=	$x /= 5$	$x = x / 5$	1.4
%=	$x \% = 5$	$x = x \% 5$	2
//=	$x //= 5$	$x = x // 5$	1

ตารางที่ 1.7 ตัวดำเนินการกำหนดค่า (ต่อ)

ตัวดำเนินการ	รูปแบบ	รูปแบบที่เหมือน	ผลลัพธ์ (x = 7)
**=	x **= 5	x = x ** 5	16807
&=	x &= 5	x = x & 5	5
=	x = 5	x = x 5	7
^=	x ^= 3	x = x ^ 3	4
>>=	x >>= 3	x = x >> 3	0
<<=	x <<= 3	x = x << 3	56

ที่มา : Przywóski (2015)

ตัวดำเนินการกำหนดค่า ได้ใช้สัญลักษณ์ทางคณิตศาสตร์มาใช้ร่วมกัน เพื่อให้คำสั่งมีขนาดกะทัดรัด ซึ่งก่อนนำตัวดำเนินการกำหนดค่าไปเขียนโปรแกรมจะต้องทำความเข้าใจในตัวดำเนินการ แต่ตัวก่อนนำไปใช้ เนื่องจากรูปแบบของการเขียนจะมีความแตกต่างจากความเข้าใจของมนุษย์โดยทั่วไป

3. ตัวดำเนินการเปรียบเทียบ

ตัวดำเนินการเปรียบเทียบ (Comparison Operators) หมายถึง การใช้สัญลักษณ์ทางด้านคณิตศาสตร์ที่ใช้ในการเปรียบเทียบระหว่างข้อมูล ผลลัพธ์ที่ได้จะมีค่าความเป็นจริงและเท็จ รายละเอียดดังต่อไปนี้

ตารางที่ 1.8 ตัวดำเนินการเปรียบเทียบ

ตัวดำเนินการ	ความหมาย	รูปแบบ	ตัวอย่าง (x=7,y=5)	ผลลัพธ์
==	เท่ากับ	x == y	print(x==y)	False
!=	ไม่เท่ากับ	x != y	print(x != y)	True
>	มากกว่า	x > y	print(x > y)	True
<	น้อยกว่า	x < y	print(x < y)	False
>=	มากกว่าหรือเท่ากับ	x >= y	print(x >= y)	True
<=	น้อยกว่าหรือเท่ากับ	x <= y	print(x <= y)	False

ที่มา : Kalb (2016)

ตัวดำเนินการเปรียบเทียบ มีการนำไปใช้ในโครงสร้างของโปรแกรม ในการกำหนดเงื่อนไขการทำงานของโปรแกรมเป็นหลัก เปรียบเทียบข้อมูลเพื่อหาความเป็นจริงและเป็นเท็จ เพื่อเลือกเส้นทางการทำงานของโปรแกรม

4. ตัวดำเนินการทางตรรกะ

ตัวดำเนินการทางตรรกะ (Logical Operators) หมายถึง การใช้สัญลักษณ์เชื่อมระหว่างข้อมูล 2 เงื่อนไขขึ้นไปเพื่อใช้ในการตรวจสอบ ผลลัพธ์ที่ได้จะมีค่าความเป็นจริงหรือเป็นเท็จ มีรายละเอียดตัวดำเนินการทางตรรกะดังต่อไปนี้

ตารางที่ 1.9 ตัวดำเนินการทางตรรกะ

ตัวดำเนินการ	ความหมาย	ตัวอย่าง ($x=7$)	ผลลัพธ์
and	คืนค่าเป็นจริง เมื่อคำสั่งเงื่อนไขทั้งหมดเป็นจริง	$x < 6$ and $x < 9$	False
or	คืนค่าเป็นจริง เมื่อคำสั่งเงื่อนไข 1 เงื่อนไขเป็นจริง	$x < 6$ or $x < 9$	True
not	กลับค่าผลลัพธ์ที่ได้	not($x < 6$ and $x < 9$)	True

ที่มา : Kalb (2016)

ตัวดำเนินการทางตรรกะมีความสำคัญมากในการนำไปเขียนโปรแกรม เนื่องจากการทำงานของโปรแกรมจะมีเส้นทางในการทำงานที่แตกต่างกัน ดังนั้นจะต้องใช้ตัวดำเนินการทางตรรกะ มาทำการตรวจสอบในการเลือกเส้นทางในการทำงานของโปรแกรม ตัวดำเนินการทางด้านตรรกะที่สำคัญในการนำมาตรวจสอบเงื่อนไข ประกอบไปด้วย and ซึ่งจะให้ผลลัพธ์ที่ได้เป็นจริงก็ต่อเมื่อเงื่อนไขทุกเงื่อนไขจะต้องเป็นจริงเท่านั้น ส่วน or จะให้ผลลัพธ์ที่เป็นจริงก็ต่อเมื่อมีเงื่อนไขเพียงเงื่อนไขเดียวเท่านั้นที่เป็นจริงและ not จะเป็นการกลับค่าผลลัพธ์ที่ได้ให้อยู่ในสถานะที่ตรงกันข้าม

5. ตัวดำเนินการระบุลักษณะ

ตัวดำเนินการระบุลักษณะ (Identity Operators) หมายถึง การใช้คำที่กำหนดลักษณะของความสัมพันธ์กันระหว่างตัวแปร 2 ตัว ผลลัพธ์ที่ได้จะมีค่าความเป็นจริงและเท็จ ดังรายละเอียดต่อไปนี้

ตารางที่ 1.10 ตัวดำเนินการระบุลักษณะ

ตัวดำเนินการ	ความหมาย	ตัวอย่าง (x=7, y=5)	ผลลัพธ์
is	คืนค่าเป็นจริง เมื่อทั้งสองตัวแปรเหมือนกัน	x is y	False
is not	คืนค่าเป็นจริง เมื่อทั้งสองตัวแปรไม่เหมือนกัน	x is not y	True

ที่มา : สุชาติ คุ่มมณี (2558)

ตัวดำเนินการระบุลักษณะเป็นอีกรูปแบบที่ตรวจสอบข้อมูลในการเปรียบเทียบ ซึ่งจากตัวอย่าง is จะมีผลเป็นจริงเมื่อตัวแปรทั้งสองมีข้อมูลเดียวกัน ส่วนตัวดำเนินการ is not จะให้ผลเป็นจริง เมื่อทั้งสองตัวแปรไม่ใช่ข้อมูลเดียวกัน

6. ตัวดำเนินการสมาชิก

ตัวดำเนินการสมาชิก (Membership Operators) หมายถึง การใช้คำตรวจสอบข้อมูลจากตัวแปรหนึ่ง มีข้อมูลหรือเป็นสมาชิกในอีกตัวแปรหนึ่งหรือไม่ ผลลัพธ์ที่ได้จะมีค่าความเป็นจริงและเท็จ ดังรายละเอียดต่อไปนี้

ตารางที่ 1.11 ตัวดำเนินการสมาชิก

ตัวดำเนินการ	ความหมาย	ตัวอย่าง (x=[7, 5])	ผลลัพธ์
in	คืนค่าเป็นจริง เมื่อตรวจสอบมีค่าข้อมูลภายในตัวแปร	print(5 in x)	True
not in	คืนค่าเป็นจริง เมื่อตรวจสอบไม่มีค่าข้อมูลภายในตัวแปร	print(5 not in x)	False

ที่มา : Sheppard (2014)

ตัวดำเนินการสมาชิกทำหน้าที่ในการตรวจสอบการเป็นสมาชิกที่อยู่ในชุดข้อมูลหรือไม่ จากตัวอย่างตัวแปรชุด x มีสมาชิกประกอบด้วย 7 และ 5 ซึ่งจากคำสั่ง print(5 in x) ให้ตรวจสอบข้อมูลตัวเลข 5 มีอยู่ในตัวแปร x หรือไม่ ตัวดำเนินการ in จึงทำการตรวจสอบและพบว่าเป็นสมาชิกของตัวแปร x ดังนั้นผลลัพธ์ที่ได้ก็จะเป็นจริงและคำสั่ง print(5 not in x) ให้ตรวจสอบตัวเลข 5 ไม่ได้มีอยู่ในตัวแปร x แต่พบว่าตัวเลขนี้มีอยู่ในตัวแปร x ดังนั้นผลลัพธ์ที่ได้จึงเป็นเท็จดังกล่าว

7. ตัวดำเนินการแบบบิต

ตัวดำเนินการแบบบิต (Bitwise Operators) หมายถึง การใช้สัญลักษณ์ทางคณิตศาสตร์ในการหาความสัมพันธ์ระหว่างค่าข้อมูลในระดับบิต มีรายละเอียดดังต่อไปนี้

ตารางที่ 1.12 ตัวดำเนินการแบบบิต

ตัวดำเนินการ	ชื่อ	ความหมาย	ตัวอย่าง	ผลลัพธ์
&	AND	กำหนดบิตเป็น 1 ถ้าทั้งสองบิตเป็น 1	print(x & y)	00000111 00000101 00000101 = 5
	OR	กำหนดบิตเป็น 1 ถ้ามีอย่างน้อย 1 บิตเป็น 1	print(x y)	00000111 00000101 00000111 = 7
^	XOR	กำหนดบิตเป็น 1 ต้องมีบิตที่เป็น 1 ค่าเดียวเท่านั้น	print(x ^ y)	00000111 00000101 00000010 = 2
~	NOT	กลับค่าบิตตรงกันข้ามทั้งหมด	print(~x)	00000111 - 00001000 = -8
<<	เติมบิต 0 ทางซ้าย	เลื่อนบิต โดยการใส่ 0 ทางขวาและบิตที่อยู่ทางซ้ายสุดตัดออกไป	print(x << 3)	00000111 00111000 = 56
>>	เติมบิต 0 ทางขวา	เลื่อนบิต โดยการใส่ 0 ทางซ้ายและบิตที่อยู่ทางขวาสุดตัดออกไป	print(x >> 2)	00000111 00000001 = 1

ที่มา : สุชาติ คุ่มมณี (2558)

การทำงานโปรแกรมคอมพิวเตอร์ ตัวดำเนินการมีความสำคัญมาก ผลลัพธ์ที่ได้ขึ้นอยู่กับ การนำตัวดำเนินการไปใช้งาน ซึ่งอาจเกิดจากในรูปของตัวเลข ข้อความ และลักษณะความเป็นจริง หรือเท็จ ในการใช้งานตัวดำเนินการนอกเหนือจากที่นำไปใช้งานทั่วไป จะต้องคำนึงถึงลำดับ ความสำคัญของตัวดำเนินการ เนื่องหลักการประมวลผลของคอมพิวเตอร์จะมีหลักการบางส่วนที่ไม่ เหมือนกับการคำนวณจากมนุษย์ เช่น คอมพิวเตอร์จะให้ตัวดำเนินการ * และ / มีลำดับความสำคัญ มากกว่า + และ - ดังนั้นสัญลักษณ์ลำดับเดียวกันจะคำนวณจากซ้ายไปขวา แต่มนุษย์จะใช้หลักการ

คำนวณจากซ้ายมาขวาทั้งหมดโดยไม่ได้เรียงลำดับความสำคัญ จึงทำให้ผลลัพธ์ที่ได้นั้นไม่ตรงกับความต้องการ ดังโปรแกรมตัวอย่างที่ 1.9

ตัวอย่างที่ 1.9 การประมวลผลเรียงลำดับความสำคัญของตัวดำเนินการ

บรรทัดที่	ชุดคำสั่ง
1	$a=4+2-3-2+8-2-2$
2	$b=4+2*3-2+8/2**2$
3	$c=9+5-3+4/2**2+6/3*3$
4	$d=(9+5-3+4)/2**2+(6/3*3)$
5	print(a)
6	print(b)
7	print(c)
8	print(d)
ผลลัพธ์	
5	
10.0	
18.0	
9.75	

จากโปรแกรมตัวอย่างดังที่ 1.9 การประมวลผลเรียงลำดับความสำคัญของตัวดำเนินการโดยค่าตัวแปร a b c และ d ประมวลผลด้วยลำดับความสำคัญของตัวดำเนินการจะได้ผลลัพธ์ 5 10.0 18.0 และ 9.75 แต่ถ้าคำนวณจากซ้ายมาขวาจะได้ผลลัพธ์ 5 144 62.25 และ 62.25 ซึ่งเป็นลักษณะของการคำนวณของมนุษย์ ซึ่งเห็นว่าจะมีความแตกต่างกัน ดังนั้นการนำตัวดำเนินการต่าง ๆ ไปใช้งาน จะต้องคำนึงถึงคุณสมบัติและลำดับความสำคัญก่อนนำไปใช้ เพื่อให้ได้ประสิทธิภาพและผลลัพธ์ถูกต้องตามที่ต้องการ

คำสั่งรับข้อมูลและแสดงข้อมูล

การทำงานของโปรแกรมคอมพิวเตอร์นั้น โปรแกรมจะสามารถประมวลผล จะต้องมีความรู้เพื่อหน้าที่ได้ไปประมวลผลตามขั้นตอนวิธีการ เมื่อเสร็จสิ้นของการประมวลผล ข้อมูลที่ได้จะถูกเก็บไว้ในหน่วยความจำของคอมพิวเตอร์และเมื่อต้องการแสดงผลลัพธ์ของข้อมูล จะต้องนำข้อมูลที่อยู่ใน

หน่วยความจำนั้นออกไปแสดง การนำข้อมูลเข้าและการนำข้อมูลออกนั้น ในภาษาไพธอน มีคำสั่งที่ใช้ทำงานดังกล่าว มีรูปแบบของคำสั่งใช้งานดังต่อไปนี้

1. คำสั่ง input()

input() คือ คำสั่งที่ใช้ในการรับข้อมูล โดยเมื่อมีการประมวลคำสั่ง โปรแกรมจะรอรับข้อมูลจากผู้ใช้โปรแกรมและทำงานต่อเมื่อผู้ใช้โปรแกรมกดปุ่ม Enter บนแผงแป้นอักขระ โปรแกรมจะนำข้อมูลที่ได้อุปประมวลผลต่อไป (Fior, 2016) ซึ่งการรับข้อมูลสามารถรับได้หลายรูปแบบ แต่ละรูปแบบจะมีหลักการและวิธีการแตกต่างกัน มีโครงสร้างของคำสั่ง ดังต่อไปนี้

```
ตัวแปร = input (ข้อความ)
```

คำสั่ง input() จากรูปแบบของคำสั่ง สามารถนำไปเขียนโปรแกรม มีลักษณะการทำงานของคำสั่งดังโปรแกรมตัวอย่างที่ 1.10

ตัวอย่างที่ 1.10 การรับข้อมูลด้วยคำสั่ง input

บรรทัดที่	คำสั่ง
1	name = input("ชื่อ-นามสกุล : ")
2	salary = input("เงินเดือน : ")
3	print ("คุณชื่อ : ", name)
4	print ("เงินเดือนที่ได้ : ", salary)
5	print(type(name))
6	print(type(salary))
ผลลัพธ์	
ชื่อ-นามสกุล : เอกราช ธรรมชา	รับข้อมูลจากแผงแป้นอักขระ
เงินเดือน : 35000	รับข้อมูลจากแผงแป้นอักขระ
คุณชื่อ : เอกราช ธรรมชา	
เงินเดือนที่ได้ : 35000	
<class 'str'>	
<class 'str'>	

การรับข้อมูลจากคำสั่ง input นั้น จะรับข้อมูลจากแผงแป้นอักขระ ข้อมูลที่ได้ในตัวแปรที่รับเป็นรูปแบบของตัวแปรประเภท str เท่านั้น ถึงแม้ว่าจะมีการรับข้อมูลเป็นตัวเลข แต่ตัวเลขที่ได้จะเป็นข้อมูล str ดังนั้นการให้ข้อมูลที่รับเข้ามาเป็นตัวเลข เพื่อนำไปดำเนินการคำนวณทางคณิตศาสตร์ จะต้องมีการแปลงข้อมูลที่รับเข้ามาดังโปรแกรมตัวอย่างที่ 1.11

ตัวอย่างที่ 1.11 การกำหนดประเภทข้อมูลรับด้วยคำสั่ง input

บรรทัดที่	คำสั่ง
1	<code>a = float(input("ตัวเลขที่ 1 : "))</code>
2	<code>b = int(input("ตัวเลขที่ 2 : "))</code>
3	<code>print("a * b = ",a*b)</code>
4	<code>print("a / b = ",a/b)</code>
5	<code>print("a + b = ",a+b)</code>
6	<code>print("a - b = ",a-b)</code>
7	<code>print(type(a))</code>
8	<code>print(type(b))</code>
ผลลัพธ์	
ตัวเลขที่ 1 : 10	รับข้อมูลจากแผงแป้นอักขระ
ตัวเลขที่ 2 : 5	รับข้อมูลจากแผงแป้นอักขระ
<code>a * b = 50.0</code>	
<code>a / b = 2.0</code>	
<code>a + b = 15.0</code>	
<code>a - b = 5.0</code>	
<code><class 'float'></code>	
<code><class 'int'></code>	

โปรแกรมตัวอย่างที่ 1.11 โปรแกรมให้แสดงผลลัพธ์จากการประมวลผลทางคณิตศาสตร์อย่างง่าย คือ การคูณ การหาร การบวก และการลบ ซึ่งจะมีการรับค่าข้อมูลเข้ามาประมวลผลสองค่านั้นคือประโยคคำสั่งในบรรทัดที่ 1 และ 2 จากความหมายของคำสั่ง input ที่กล่าวไว้ในข้างต้น คำสั่งนี้จะทำการรับค่าข้อมูลที่เป็นแบบตัวอักขระเท่านั้น ดังนั้นเมื่อต้องการเอาข้อมูลที่รับเข้ามาไปทำการประมวลผล จะต้องทำการแปลงข้อมูลที่รับเข้ามาเป็นรูปแบบของตัวเลข ซึ่งในคำสั่งให้ทำการแปลงข้อมูลให้เป็นข้อมูลจำนวนจริง จึงใช้คำสั่ง float และคำสั่ง int ทำการแปลงข้อมูล ซึ่งทั้งสองรูปแบบ

สามารถประมวลรวมกันได้ แต่ถ้าไม่มีการแปลงข้อมูลที่รับเข้ามา จะได้ผลลัพธ์ดังรายละเอียดโปรแกรมตัวอย่างที่ 1.12

ตัวอย่างที่ 1.12 การประมวลผิดพลาดจากการรับข้อมูลที่ยังไม่ได้ทำการแปลงค่า

บรรทัดที่	คำสั่ง
1	print ("----- โปรแกรมคำนวณพื้นฐานด้านคณิตศาสตร์ -----")
2	a = input("ค่าตัวเลขตัวที่ 1 = ")
3	b = input("ค่าตัวเลขตัวที่ 2 = ")
4	print("ผลลัพธ์",a," * ",b," = ",a*b)
5	print("ผลลัพธ์",a," / ",b," = ",a/b)
6	print("ผลลัพธ์",a," + ",b," = ",a+b)
7	print("ผลลัพธ์",a," - ",b," = ",a-b)
8	print("ผลลัพธ์",a," % ",b," = ",a%b)
9	print("ผลลัพธ์",a," // ",b," = ",a//b)
10	print("ผลลัพธ์",a," ** ",b," = ",a**b)
ผลลัพธ์	
----- โปรแกรมคำนวณพื้นฐานด้านคณิตศาสตร์ -----	
ค่าตัวเลขตัวที่ 1 = 10 (รับข้อมูลจากแผงแป้นอักขระ)	
ค่าตัวเลขตัวที่ 2 = 5 (รับข้อมูลจากแผงแป้นอักขระ)	
Traceback (most recent call last):	
File "C:\Python36\tt.py", line 4, in <module>	
print("ผลลัพธ์",a," * ",b," = ",a*b)	
TypeError: can't multiply sequence by non-int of type 'str'	

การรับข้อมูลหลังคำสั่ง input จะต้องทำการระบุประเภทของข้อมูลที่จะทำการรับ การระบุคำสั่ง ใช้ชื่อเดียวกันกับชนิดของตัวแปรที่กล่าวไว้ในข้างต้น จึงจะสามารถเก็บข้อมูลในรูปแบบที่ต้องการได้ ข้อมูลเหล่านี้คือข้อมูลตั้งต้นที่จะนำไปใช้ในการประมวล ซึ่งเป็นกระบวนการที่สำคัญต่อการทำงานของโปรแกรม

2. คำสั่ง print()

print() คือ คำสั่งที่ใช้ในการแสดงข้อมูล ซึ่งสามารถแสดงข้อมูลได้หลายรูปแบบ ทั้งที่อยู่ในรูปข้อมูลตัวแปรหรือชุดคำสั่ง (Fior, 2016) การแสดงข้อมูลมีโครงสร้างของคำสั่ง ดังต่อไปนี้

```
print ("อักขระ อักขระหรือข้อความ" หรือ ตัวแปร)
```

ลักษณะของคำสั่ง print() ที่ใช้ในการแสดงข้อมูล ซึ่งข้อมูลนั้นจะประกอบไปด้วย 2 ส่วน คือ ข้อมูลที่เป็นข้อความ ซึ่งจะต้องอยู่ภายในเครื่องหมาย “ ” หรือ ‘ ’ และข้อมูลที่อยู่ในรูปของตัวแปร ไม่ต้องใส่เครื่องหมายดังกล่าว ดังรายละเอียดโปรแกรมตัวอย่างที่ 2.13

ตัวอย่างที่ 1.13 การแสดงผลลัพธ์ของคำสั่ง print

บรรทัดที่	คำสั่ง
1	print ("----- โปรแกรมคำนวณพื้นฐานด้านคณิตศาสตร์ -----")
2	a = float(input("ค่าตัวเลขตัวที่ 1 = "))
3	b = float(input("ค่าตัวเลขตัวที่ 2 = "))
4	print("ผลลัพธ์",a," * ",b," = ",a*b)
5	print("ผลลัพธ์",a," / ",b," = ",a/b)
6	print("ผลลัพธ์",a," + ",b," = ",a+b)
7	print("ผลลัพธ์",a," - ",b," = ",a-b)
8	print("ผลลัพธ์",a," % ",b," = ",a%b)
9	print("ผลลัพธ์",a," // ",b," = ",a//b)
10	print("ผลลัพธ์",a," ** ",b," = ",a**b)
ผลลัพธ์	
----- โปรแกรมคำนวณพื้นฐานด้านคณิตศาสตร์ -----	
ค่าตัวเลขตัวที่ 1 = 10 (รับข้อมูลจากแผงแป้นอักขระ)	
ค่าตัวเลขตัวที่ 2 = 5 (รับข้อมูลจากแผงแป้นอักขระ)	
ผลลัพธ์ 10.0 * 5.0 = 50.0	
ผลลัพธ์ 10.0 / 5.0 = 2.0	
ผลลัพธ์ 10.0 + 5.0 = 15.0	
ผลลัพธ์ 10.0 - 5.0 = 5.0	

บรรทัดที่	คำสั่ง
	ผลลัพธ์ 10.0 % 5.0 = 0.0
	ผลลัพธ์ 10.0 // 5.0 = 2.0
	ผลลัพธ์ 10.0 ** 5.0 = 100000.0

การใช้สัญลักษณ์ทางคณิตศาสตร์กับข้อมูล สามารถประยุกต์ใช้ เช่น ให้แสดงข้อความซ้ำกันหลายครั้ง ภาษาไพธอน อนุญาตให้นำข้อความนั้นมาทำการประมวลผลกับตัวดำเนินการทางคณิตศาสตร์ได้ ดังโปรแกรมตัวอย่างที่ 1.14

ตัวอย่างที่ 1.14 การแสดงข้อมูลจากตัวดำเนินการทางคณิตศาสตร์

บรรทัดที่	คำสั่ง
1	print ("Hellow " * 3)
2	print ("Hellow " / 3)
3	print ("Hellow " + 3)
4	print ("Hellow " - 3)
ผลลัพธ์	
Hellow Hellow Hellow	
Traceback (most recent call last):	
File "C:/Python34/t.py", line 2, in <module>	
print ("Hellow " / 3)	
TypeError: unsupported operand type(s) for /: 'str' and 'int'	

โปรแกรมตัวอย่างที่ 1.14 ภาษาไพธอน อนุญาตให้ใช้ตัวดำเนินการทางคณิตศาสตร์ ข้อมูลประเภท str โดยใช้สัญลักษณ์ “*” ซึ่งมีจุดมุ่งหมายของการนำไปใช้ โดยให้แสดงข้อความตามจำนวนที่ต้องการและเครื่องหมาย “+” เป็นสัญลักษณ์ในการเชื่อมข้อมูลประเภท str ด้วยกัน มีรายละเอียดในการใช้งานดังต่อไปนี้

ตัวอย่างที่ 1.15 การใช้ตัวดำเนินการข้อมูลประเภท str

บรรทัดที่	คำสั่ง
1	print (" Hellow " + "student")
2	print (" Hellow " + "student" * 3)

3	<code>print ((" Hellow " + "student" * 3)*2)</code>
4	<code>print (" Hellow " + "student" * 3*2)</code>
5	<code>print (" Hellow " + "student" * (3*2))</code>
ผลลัพธ์	
Hellow student Hellow studentstudentstudent Hellow studentstudentstudent Hellow studentstudentstudent Hellow studentstudentstudentstudentstudentstudent Hellow studentstudentstudentstudentstudentstudent	

การใช้ตัวดำเนินการ “+” และ “*” ซึ่งเป็นตัวดำเนินการที่สามารถนำมาใช้กับข้อมูลประเภท str ได้ การใช้งานร่วมกันนั้น โปรแกรมจะให้ความสำคัญตัวดำเนินการ “*” ก่อน “+” แต่การประมวลผลที่ได้จะต้องอยู่ในกลุ่มคำสั่งเดียวกัน คือ อยู่ภายใต้เครื่องหมาย () หรือ วงเล็บ (ราชบัณฑิตยสถาน, 2564) ของคำสั่ง print ซึ่งสามารถซ้อนกันได้ไม่จำกัด โดยห้ามแยกออกจากกลุ่ม ซึ่งถ้าแยกออกจากกลุ่ม จะทำให้เกิดการผิดพลาดของโปรแกรมทันที ดังผลลัพธ์ที่ได้ของโปรแกรมตัวอย่างที่ 1.6

ตัวอย่างที่ 1.16 การทำงานผิดพลาดของโปรแกรมในการใช้ตัวดำเนินการทางคณิตศาสตร์

บรรทัดที่	คำสั่ง
1	<code>print (" Hellow " + "student"+" Good"+" Morning"))</code>
2	<code>print (" Hellow " + "student"+" Good"+" Morning"))</code>
3	<code>print (" Hellow " + "student")+(" Good Morning")</code>
ผลลัพธ์	
Hellow student Good Morning Hellow student Good Morning Hellow student Traceback (most recent call last): File "D:\python_file\test.py", line 3, in <module> print (" Hellow " + "student")+(" Good Morning") TypeError: unsupported operand type(s) for +: 'NoneType' and 'str'	

กระบวนการของการทำงานของโปรแกรม ส่วนที่สำคัญซึ่งจะสัมพันธ์กับผู้ใช้โปรแกรมโดยตรง คือ ข้อมูลนำเข้าที่ได้จากผู้ใช้โปรแกรมและการนำข้อมูลออกเพื่อแสดงผลลัพธ์ที่ได้จากการทำงานของโปรแกรมต่อผู้ใช้โปรแกรม ภาษาคอมพิวเตอร์โดยทั่วไปที่มีลักษณะการทำงานรับคำสั่งที่ละบรรทัด จะมีคำสั่งที่ใช้ในการนำข้อมูลเข้าและคำสั่งนำข้อมูลออก ภาษาไพธอนก็เช่นเดียวกัน การทำงานบน IDLE จะมีลักษณะการรับคำสั่งที่ละบรรทัด ดังนั้นจึงได้เตรียมคำสั่ง input ในการนำข้อมูลเข้าและคำสั่ง print ในการนำข้อมูลออก ซึ่งกระบวนการนี้เป็นพื้นฐานของการทำงานของโปรแกรมคอมพิวเตอร์

สรุป

ภาษาไพธอน คือ ภาษาคอมพิวเตอร์ที่นำมาใช้งานได้ฟรี จากเว็บไซต์ของผู้พัฒนาภาษาไพธอน การทำงานของภาษาไพธอนจะใช้งานในลักษณะของโปรแกรมประยุกต์ทั่วไปและการทำงานบนเว็บ ซึ่งสามารถใช้งานได้ทั้งระบบปฏิบัติการต่างชนิดกันที่เป็น 32 บิต และ 64 บิตได้ เมื่อติดตั้งโปรแกรม การใช้งานภาษาโปรแกรม สามารถใช้งานผ่านทาง Python IDE ที่มีลักษณะการทำงานรับคำสั่งที่ละบรรทัดผ่าน IDLE Shell และสามารถสร้างเป็นชุดคำสั่งผ่านทาง IDLE File ที่มีลักษณะเก็บกลุ่มคำสั่งก่อนประมวลผล ซึ่งเป็นเครื่องมือในการพัฒนาของภาษาไพธอน นอกเหนือจากนี้ยังมีการใช้งานขั้นสูงที่นอกเหนือจากการประมวลผลคำสั่งที่อยู่ในภาษาโปรแกรม โดยสามารถใช้งานร่วมกับเครื่องมือที่สนับสนุนกับภาษาไพธอน ผ่านการติดตั้งเครื่องมือเสริมที่จะสนับสนุนการทำงานร่วมกับภาษาไพธอน ให้การนำไปพัฒนาโปรแกรมมีประสิทธิภาพเพิ่มมากขึ้นผ่านทาง Python Command Line ก่อนการเขียนโปรแกรมในภาษาโปรแกรม จะต้องมีความเข้าใจเกี่ยวกับพื้นฐานการใช้งานของภาษาโปรแกรม ซึ่งประกอบด้วย ตัวแปร มีหน้าที่อ้างอิงกับหน่วยความจำในการเก็บข้อมูล ซึ่งก่อนนำไปใช้ในการเขียนโปรแกรม จะต้องมีความเข้าใจเกี่ยวกับประเภทของตัวแปร กฎเกณฑ์การตั้งชื่อตัวแปร คำสงวน ซึ่งเป็นคำสั่งและฟังก์ชันที่มีใช้งานในโปรแกรม การใช้งานตัวแปรบางกรณีอาจไม่ต้องการเปลี่ยนแปลงข้อมูลที่อยู่ตัวแปร สามารถกำหนดค่าให้ค่าตายตัวได้ โดยกำหนดให้เป็นลักษณะของค่าคงที่ ข้อมูลที่ถูกเก็บในตัวแปรจะสามารถประมวลผลระหว่างตัวแปรได้ จะต้องใช้ตัวดำเนินการ ซึ่งประกอบไปด้วยตัวดำเนินการหลายชนิด ขึ้นอยู่กับการนำไปใช้งาน ประกอบด้วย ตัวดำเนินการทางคณิตศาสตร์ ตัวดำเนินการกำหนดค่า ตัวดำเนินการเปรียบเทียบ ตัวดำเนินการทางตรรกะ ตัวดำเนินการระบุลักษณะ ตัวดำเนินการสมาชิกและตัวดำเนินการแบบบิต ซึ่งผลที่ได้จาก ตัวดำเนินการ ประกอบไปด้วยข้อมูลที่เป็นข้อความ ตัวเลขและความเป็นจริงและเป็นเท็จ การทำงานของโปรแกรมจะประกอบไปด้วย การนำข้อมูลเข้า การประมวลผล การนำข้อมูลออก ซึ่งส่วนที่มีความเกี่ยวข้องกับผู้ใช้โปรแกรม คือ การนำข้อมูลเข้าจากผู้ใช้โปรแกรมและการนำข้อมูลออก เพื่อให้ผู้ใช้โปรแกรมได้เห็นผลลัพธ์ ดังนั้นภาษาไพธอนพื้นฐานที่ทำงานบน IDLE จะเป็น

ลักษณะของการนำข้อมูลเข้าที่ละบรรทัดซึ่งคำสั่งที่ใช้ในการรับนำข้อมูลเข้าคือคำสั่ง `input()` และคำสั่งนำข้อมูลออก คือคำสั่ง `print()` ซึ่งแต่ละคำสั่งจะมีกฎเกณฑ์ในการนำไปใช้ในการประมวลผลตามลักษณะการทำงานของโปรแกรม

อ.ดร.เอกราช บรรณานา

แบบฝึกหัด

1. ให้ผู้เรียนอธิบายคุณลักษณะของภาษาไพธอน พร้อมทั้งบอกข้อดี ข้อจำกัด และประโยชน์ของการนำมาใช้ในการเขียนโปรแกรม
2. ให้ผู้เรียนอธิบายลักษณะเครื่องมือที่มีให้ใช้งานแต่ละประเภทในภาษาไพธอน
3. ให้ผู้เรียนอธิบายความหมายและหลักการทำงานของตัวแปร
4. ให้ผู้เรียนอธิบายกฎเกณฑ์และระเบียบในการตั้งชื่อตัวแปร
5. ให้ผู้เรียนบอกประเภทของตัวแปรที่มีทั้งหมดกี่ประเภท แต่ละประเภทมีลักษณะการเก็บข้อมูลและการนำไปใช้งานแตกต่างกันอย่างไร
6. ให้ผู้เรียนอธิบายคุณลักษณะของค่าคงที่ ประโยชน์ จุดประสงค์การนำไปใช้ พร้อมทั้งยกตัวอย่างประกอบการอธิบาย
7. ให้ผู้เรียนอธิบายความหมาย หน้าที่ และการทำงานของตัวดำเนินการ พร้อมทั้งยกตัวอย่างการนำไปใช้แต่ละประเภทของตัวดำเนินการ
8. ให้ผู้เรียนอธิบายคำสั่งที่ใช้ในการนำข้อมูลเข้า พร้อมทั้งกฎระเบียบและหลักเกณฑ์การใช้คำสั่ง
9. ให้ผู้เรียนอธิบายคำสั่งที่ใช้ในการนำข้อมูลออก พร้อมทั้งกฎระเบียบและหลักเกณฑ์การใช้คำสั่ง
10. ให้ผู้เรียนเขียนโปรแกรมให้ห้มีนำเข้าข้อมูลและให้แสดงผลลัพธ์ที่ได้จากประมวลผล
ตัวเลขที่ 1 / ตัวเลขที่ 2 + ตัวเลขที่ 3 - ตัวเลขที่ 4 * ตัวเลขที่ 5

เอกสารอ้างอิง

- โชติพันธุ์ หล่อเลิศสุนทร และ ฐิตะพันธุ์ หล่อเลิศสุนทร. (2559). **คู่มือเรียนเขียนโปรแกรม python (ภาคปฏิบัติ)**. กรุงเทพฯ: คอร์ฟังก์ชั่น.
- ธีรวัฒน์ ประกอบผล. (2558). **การเขียนแอปพลิเคชันด้วย Visual Basic 2010**. กรุงเทพฯ: ชิมพลิฟลาย.
- บัญชา ปะสีละเตสัง. (2558). **สร้าง Windows Application ด้วย Visual Basic 2015**. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.
- ราชบัณฑิตยสภา. ศัพท์บัญญัติสำนักงานราชบัณฑิตยสภา. (ออนไลน์) 10 เมษายน 2564 (อ้างเมื่อ 10 เมษายน 2564). จาก: <https://coined-word.orst.go.th/>
- สุชาติ คุ่มมณี. (2558). **เชี่ยวชาญการเรียนรู้ด้วยภาษาไพธอน**. มหาสารคาม. คณะวิทยาการสารสนเทศ: มหาวิทยาลัยมหาสารคาม.
- โอภาส เอี่ยมสิริวงศ์ และ สมโภชน์ ชื่นเอี่ยม. (2560). **การเขียนโปรแกรมคอมพิวเตอร์**. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.
- Fangohr, Hans. (2015). **Introduction to Python for Computational Science and Engineering**. Southampton: University of Southampton.
- Fior, James M. (2016). **Computer Programming with Python and Multisim**. 3rded. London: Pearson.
- Halterman, Richard L. (2016). **Fundamentals of Python Programming**. Tennessee: Southern Adventist University.
- Kalb, Irv. (2016). **Learn to Program with Python**. California: Apress.
- Langtangen, Hans P. (2014). **A Primer on Scientific Programming with Python**. Oslo: University of Oslo.
- Przywowski, Jakub. (2015). **Escape Characters**. (online) (cited 15 July 2016). Available from: <https://python-reference.readthedocs.io/en/latest/docs/str/escapes.html>
- Python. (2021) (online) (cited 15 June 2021). Available from: <https://www.python.org/>
- Rossum, Guido van. (2013). **Python Tutorial Release 3.3.2**. Amsterdam: Python Software Foundation.
- Sebesta, Robert W. (2016). **Concepts of Programming Languages**. 11thed, London: Pearson.

Sheppard, Kevin. (2014). **Introduction to Python for Econometrics, Statistics and Data Analysis**. Oxford: University of Oxford.

อ.ดร.เอกราช ธรรมนุชา