



เทคโนโลยีดิจิทัล: ภาษาไพทอน

เรียบเรียงโดย รองศาสตราจารย์ ดร. กรุง สีนอริมย์สรานู

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

วันพฤหัสบดีที่ 28 ธันวาคม 2566



แผนการนำเสนอ & ฝึกปฏิบัติการ

บรรยายคลาสวอร์ด โดยวิทยากร

- ทำแบบทดสอบก่อนการบรรยาย
- ฟังการบรรยาย
- ถาม-ตอบ
- ทำแบบทดสอบท้ายการบรรยาย

ปฏิบัติการกู้เกิลโคแลป

- สาธิตการใช้งาน Google colabory
- As the calculator
- As the scientific calculator
- Python programming
- ส่งงานท้ายการอบรม

พักทานอาหารเที่ยง



ปฏิบัติการคลาวด์กู้เกิล

- สาธิตการใช้งาน Google app
- Google Docs
- Google Sheets
- Google Slides
- Google Forms
- ส่งงานท้ายการอบรม

บรรยายภาษาไพทอนโดยวิทยากร

- ทำแบบทดสอบก่อนการบรรยาย
- ฟังการบรรยาย
- ถาม-ตอบ
- ทำแบบทดสอบท้ายการบรรยาย

สารบัญ

- ประวัติของภาษาไพทอน
- หลักการตั้งชื่อและชนิดของวัตถุ
- นิพจน์และตัวดำเนินการในภาษาไพทอน
- การเขียนคำสั่งในภาษาไพทอน
- โจทย์

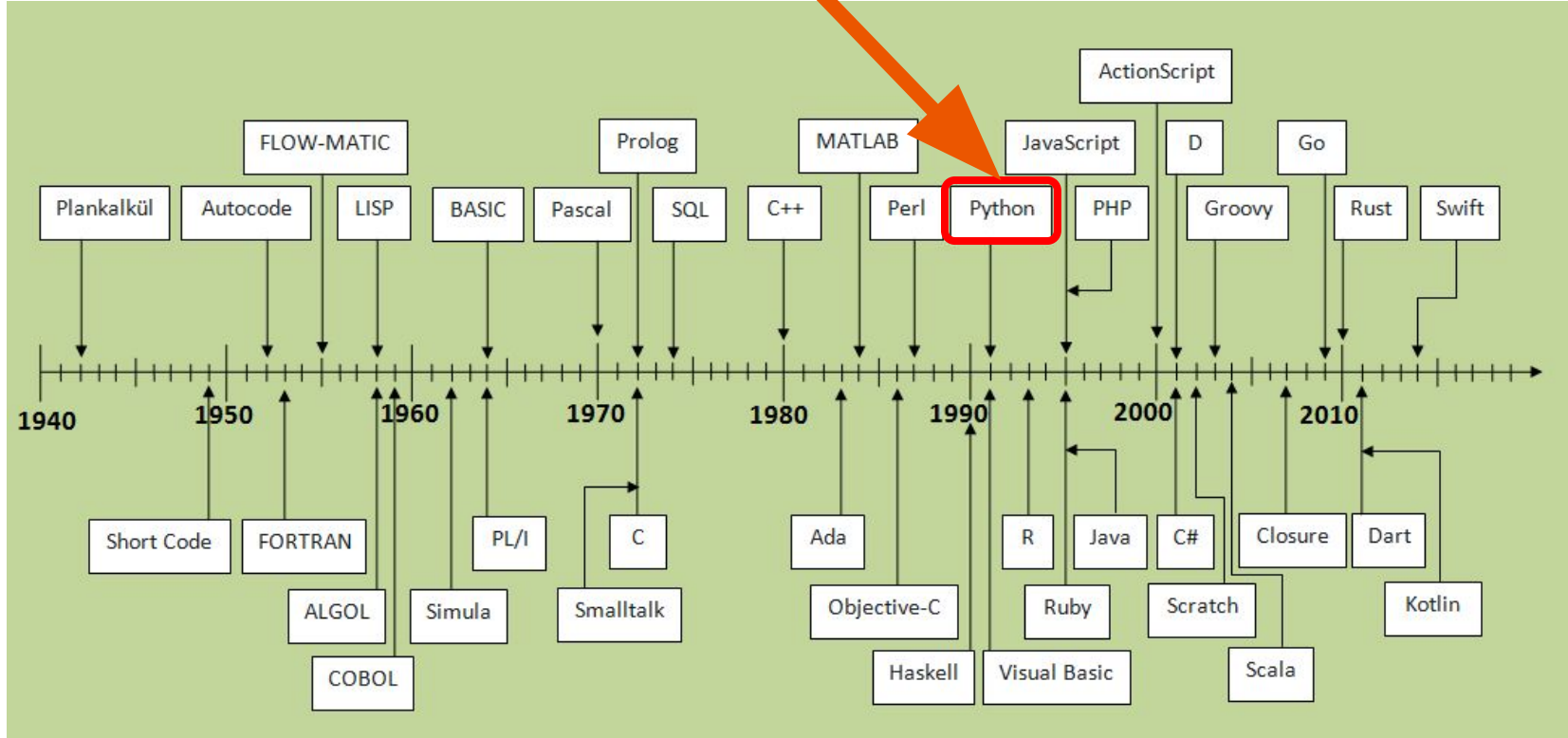


ประวัติของภาษาไพทอน



ภาษาพัฒนาโปรแกรม

<https://javaconceptsoftheday.com/history-of-programming-languages/>



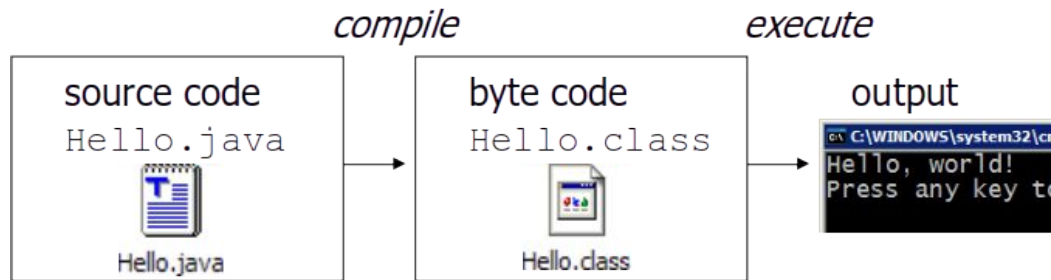
เทอมที่ใช้สำหรับการเขียนโปรแกรม



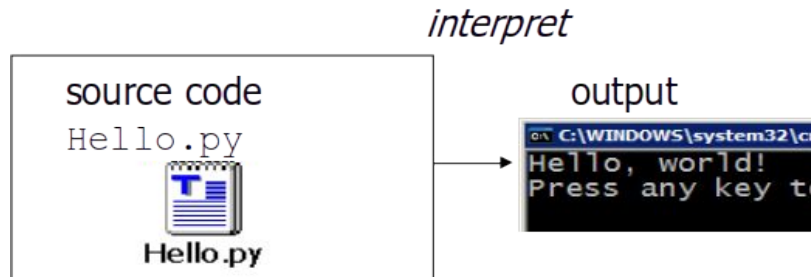
- รหัสโปรแกรม (Source code) เป็นลำดับของคำสั่งที่เขียนเพื่อตอบโจทย์
- ไวยากรณ์ (Syntax) เป็นโครงสร้างของภาษาที่บังคับ แตกต่างกันตามแต่ละภาษา
- ผลลัพธ์ (Output) ปกติจะแสดงเป็นข้อความผ่านฟังก์ชัน print()
- ข้อมูลนำเข้า (Input) มักบันทึกในตัวแปรโปรแกรม และมีการอ้างถึงค่าผ่านชื่อตัวแปรโปรแกรม เพื่อเปลี่ยนให้เป็นผลลัพธ์ที่ต้องการ

ความแตกต่างระหว่าง คอมไพเลอร์ กับ อินเทอร์พรีเตอร์

- ภาษาซี ภาษาจาวา และภาษาอื่น ๆ บังคับให้ผู้ใช้งานแปลงรหัสโปรแกรม เป็นรหัสภาษาเครื่องที่ประมวลผลกับคอมพิวเตอร์ ก่อนสั่งประมวลผล (run) เรียก คอมไพเลอร์ (Compiler)



- ภาษาไพทอนใช้การแปลงรหัสโปรแกรมแต่ละคำสั่งโดยตรง ณ เวลาที่สั่งประมวลผล เรียก อินเทอร์พรีเตอร์ (interpreter)



ประวัติโดยย่อของภาษาไพทอน



- คิดค้นในเนเธอร์แลนด์ต้นยุค 90 โดย Guido van Rossum
- ตั้งชื่อตาม Monty Python
- มีการจัดตั้ง Python Software Foundation เป็น Open source
- เริ่มจากการเป็นภาษาสคริปต์
- จัดการกับข้อมูลขนาดใหญ่ ได้ มีลักษณะการพัฒนาเชิงฟังก์ชันและวัตถุ
- กู้เกิลใช้ภาษาไพทอน ตั้งแต่เริ่มมีการสร้างภาษานี้

ผู้สร้างภาษาไพทอน

“Python is an experiment in how much freedom programmers need. Too much freedom and nobody can read another's code; too little and expressiveness is endangered.”

Guido van Rossum



Download

Download these documents

Docs by version

Python 3.13 (in development)
Python 3.12 (stable)
Python 3.11 (stable)
Python 3.10 (security-fixes)
Python 3.9 (security-fixes)
Python 3.8 (security-fixes)
Python 3.7 (EOL)
Python 3.6 (EOL)
Python 3.5 (EOL)
Python 3.4 (EOL)
Python 3.3 (EOL)
Python 3.2 (EOL)
Python 3.1 (EOL)
Python 3.0 (EOL)
Python 2.7 (EOL)
Python 2.6 (EOL)
All versions

Other resources

PEP Index
Beginner's Guide
Book List
Audio/Visual Talks
Python Developer's Guide

Python 3.12.1 documentation

Welcome! This is the official documentation for Python 3.12.1.

Parts of the documentation:

[What's new in Python 3.12?](#)
or all "What's new" documents since 2.0

[Tutorial](#)
start here

[Library Reference](#)
keep this under your pillow

[Language Reference](#)
describes syntax and language elements

[Python Setup and Usage](#)
how to use Python on different platforms

[Python HOWTOs](#)
in-depth documents on specific topics

[Installing Python Modules](#)
installing from the Python Package Index & other sources

[Distributing Python Modules](#)
publishing modules for installation by others

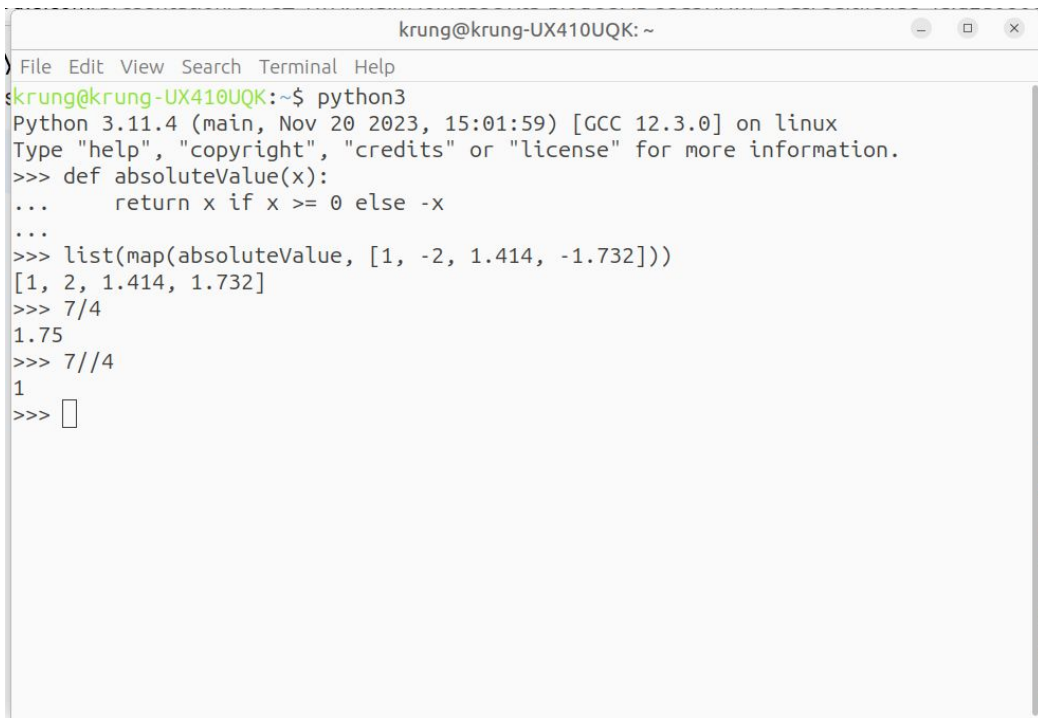
[Extending and Embedding](#)
tutorial for C/C++ programmers

[Python/C API](#)
reference for C/C++ programmers

[FAQs](#)
frequently asked questions (with answers!)

Indices and tables:

ตัวอย่างการทำงานในคอนโซล



```
krung@krung-UX410UQK: ~  
File Edit View Search Terminal Help  
krung@krung-UX410UQK:~$ python3  
Python 3.11.4 (main, Nov 20 2023, 15:01:59) [GCC 12.3.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> def absoluteValue(x):  
...     return x if x >= 0 else -x  
...  
>>> list(map(absoluteValue, [1, -2, 1.414, -1.732]))  
[1, 2, 1.414, 1.732]  
>>> 7/4  
1.75  
>>> 7//4  
1  
>>> □
```

- การพัฒนาโปรแกรมภาษาไพทอนมีทั้งแบบ interpreter และ compiler
- อินเทอร์เฟซทำงานแบบ read-eval-print loop (REPL)
 - Read อ่านคำสั่งพร้อมข้อมูลเข้า
 - Eval ประมวลผลโดยใช้คอมไพเตอร์
 - Print แสดงผลลัพธ์ที่ได้ออกมา
 - Loop วนซ้ำรอคำสั่งใหม่

ภาษาไพทอนบนกูเกิลโคลแลปบอราทอรี

The screenshot displays the Google Colaboratory web interface in a Microsoft Edge browser. The address bar shows the URL: https://colab.research.google.com/drive/1fdhH9_jv_t9L1mkiyyR4TbgylFhKg.... The interface includes a top menu bar with options like File, Edit, View, Insert, Runtime, Tools, and Help. A left sidebar contains a 'Table of contents' and a search bar. The main workspace shows a code cell titled 'เอกสารการเขียนโปรแกรมภาษาไพทอน' (Python Programming Document). The code cell contains two lines of Python code:

```
1 def absoluteValue(x):  
2   return x if x >= 0 else -x
```

 Below the code, the output is displayed:

```
[1, 1, 1.414, 1.732]
```

. The status bar at the bottom indicates '0s completed at 10:08 AM'.

- ใช้ฟรีผ่านบัญชีกูเกิล
- อินเทอร์เฟซแบบสมูทจัด
- เซลล์รหัส (Code cell) ประมวลผลภาษาไพทอน
- เซลล์ข้อความ (Text cell) เขียนเอกสารรองรับภาษา Markdown, HTML, Latex

ภาษาไพทอนบน เครื่องแม่ข่ายเซจเซล

Sage Cell Server and 2 more pages - Profile 1 - Microsoft Edge

3.12.0 Documentati... SimplePython - Cola... Sage Cell Server

https://sagecell.sagemath.org

About SageMathCell

SageMathCell

Type some Sage code below and press Evaluate.

```
1 def absoluteValue(x):  
2     return x if x >= 0 else -x  
3 print(list(map(absoluteValue, [1, -1, 1.414, -1.732])))
```

Evaluate

Language: Sage

Share

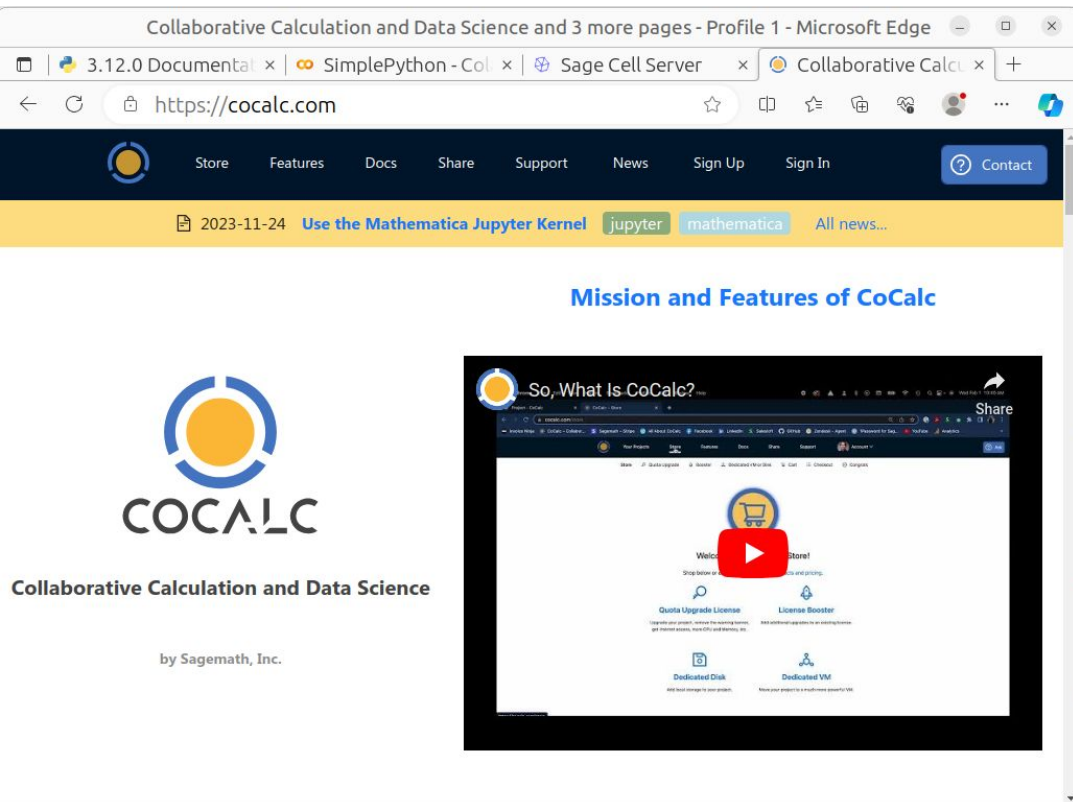
[1, 1, 1.4140000000000000, 1.7320000000000000]

Help | Powered by SageMath

About

- รหัสโปรแกรมภาษาไพทอนทั้งหมด ต้องพิมพ์ในกล่องเพื่อส่งไปประมวลผลที่เครื่องแม่ข่ายเซจเซล
- ผลลัพธ์ที่ได้จากปรากฏในกล่องผลลัพธ์ที่อยู่ใต้กล่อง
- สนับสนุนภาษามากมาย: Sage, Gap, GP, HTML, Maxima, Octave, Python, R, Singular

ภาษาไพทอนบน โคลแคลชี



- ผู้ใช้ต้องมีบัญชีในอินเทอร์เน็ต
- สนับสนุนภาษามากมาย: Sage, Python, R, Julia, Octave, Latex



หลักการตั้งชื่อและชนิดของวัตถุ



การตั้งชื่อและการกำหนดค่า

- ภาษาไพทอนอ้างถึงสิ่งต่าง ๆ ด้วยชื่อ (Identifier) ไม่ว่าจะเป็นตัวแปรโปรแกรม ฟังก์ชัน วัตถุ
- หลักการตั้งชื่อ ต้องเริ่มด้วย ตัวอักษร [a-zA-Z] หรือ ตัวขีดเส้นใต้ _ แล้วตามด้วยตัวอักษร ตัวขีดเส้นใต้ หรือ ตัวเลขที่มาจาก ASCII code ตัวอักษรตัวใหญ่และตัวเล็กมีความแตกต่างกัน
 - เช่น m, total, cost_total (ควรใช้ชื่อที่มีความหมายตอนเขียน)
- การใช้ชื่อในการกำหนดค่าให้ใช้สัญลักษณ์ = ในการกำหนดชื่อ เช่น

```
>> t1 = 12
```

เป็นการกำหนดชื่อตัวแปรโปรแกรม t1 เก็บค่าจำนวนเต็ม 12

ตัวแปรโปรแกรม

- ชนิดของตัวแปรโปรแกรมของภาษาไพทอน ขึ้นกับค่าข้อมูลที่เก็บ
- ผู้เขียนโปรแกรมสามารถกำหนดค่าพร้อมกันมากกว่าหนึ่งตัวแปร โดยการใช้ `=` ร่วมกับ `,` ที่มีจำนวนสมาชิกด้านซ้ายเท่ากับด้านขวา เช่น

`>> it, sum, default = 1, 0.0, -1` เป็นการกำหนดค่าตัวแปรโปรแกรมสามตัวคือ `it` ให้มีค่าเป็นจำนวนเต็ม 1 และ `sum` ให้มีค่าเป็นจำนวนจริง 0.0 และ `default` ให้มีค่าเป็นจำนวนเต็ม -1

`>> a, b = b, a` เป็นการสลับค่าระหว่างตัวแปรโปรแกรม `a` และ `b`

การใช้ย่อหน้ากับ comment

- ผู้เขียนใช้ย่อหน้าแสดงกลุ่มของคำสิ่งที่อยู่ใน Block เดียวกัน
- การใช้ = ครั้งแรกกับตัวแปรโปรแกรมจะเป็นการสร้างที่เก็บข้อมูล พร้อมชนิดแบบพลวัต (ผู้เขียนโปรแกรมไม่ต้องบอชนิดข้อมูลก่อนการใช้)
- เครื่องหมาย = ใช้กับการกำหนดค่า แต่ == ใช้กับการเปรียบเทียบค่า
- การใช้ # คือการกำหนดให้ข้อความที่ตามต่อท้ายไม่ถูกนำไปประมวลผล


```
>> sum = 0
```

```
>> for i in [1, 2, 3]:
```

```
>>     sum += i    # Increment sum by i
```

```
>> print(sum)      # Show the result of the sum
```

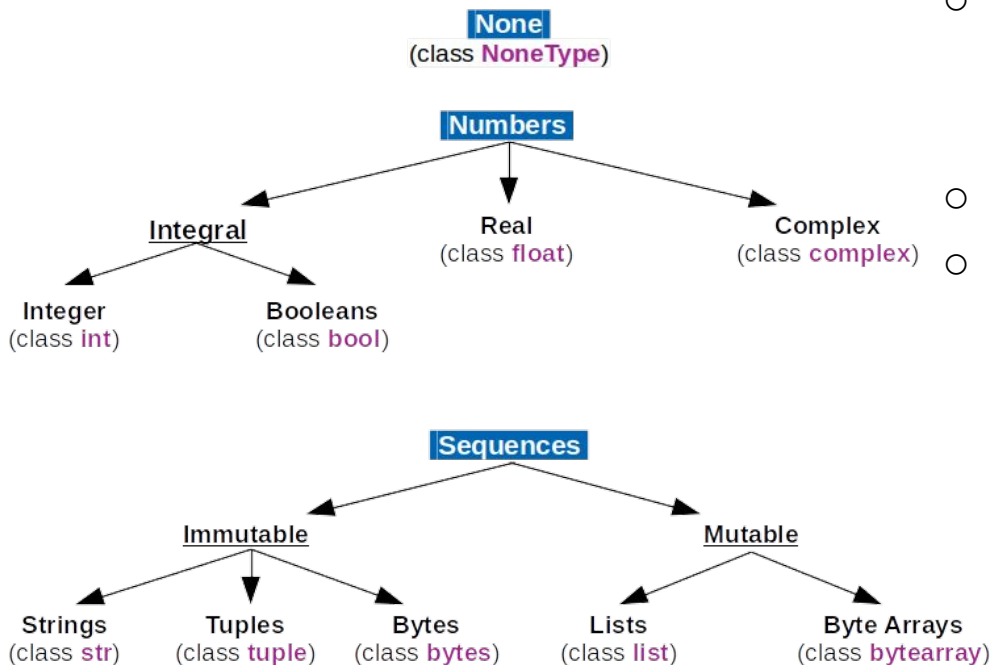
ข้อมูลดั้งเดิม 7 รูปแบบ



- รูปแบบ None ใช้กับข้อมูลที่ขาดหายไป (class NoneType)
- รูปแบบจำนวน แยกออกเป็น จำนวนเต็ม (class int) ตรรกะ (class bool) จำนวนจริง (class float) จำนวนเชิงซ้อน (class complex)
- รูปแบบลำดับ แยกออกเป็น สายอักขระ (class string) หลายสิ่งลำดับ (class tuple) สายบิต (class bytes) รายการ (class list) รายการไบนารี (class bytearray)
- รูปแบบเซต แยกออกเป็น เซต (class set) กับ เซตที่ไม่เปลี่ยนแปลง (class frozenset)
- รูปแบบการส่ง ใช้กับข้อมูลพจนานุกรม (class dict)
- รูปแบบที่เรียกใช้ฟังก์ชันอื่น
- รูปแบบที่เรียกใช้โมดูลอื่น

ภาษาไพทอน ข้อมูลจำนวน

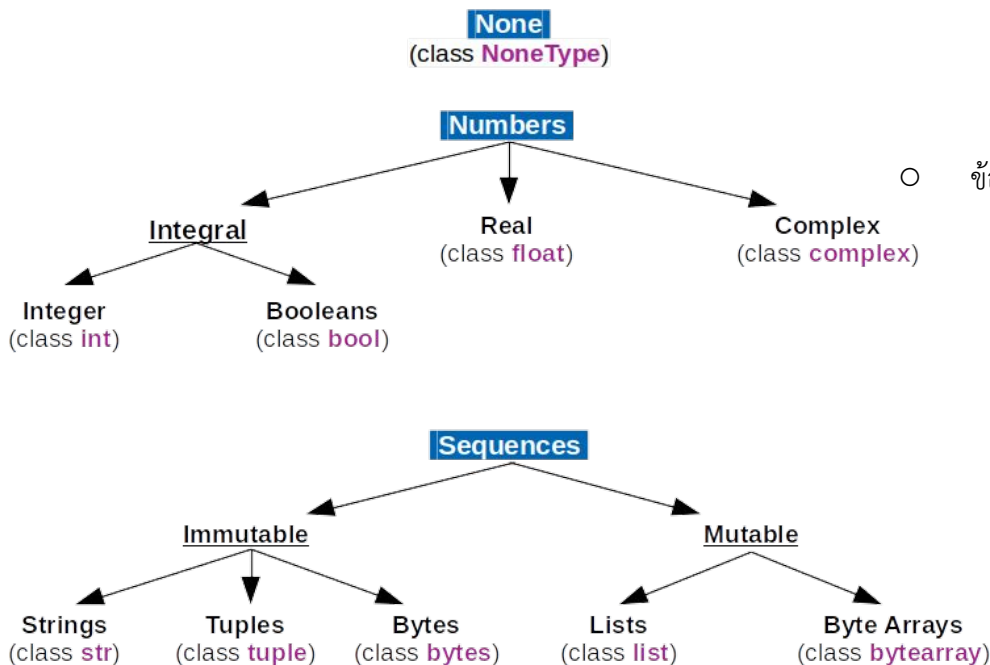
Python 3 The standard type hierarchy



- ข้อมูลจำนวน แยกออกได้เป็นสามลักษณะคือ
 - ข้อมูลจำนวนเต็มส่วน (integral) ที่แยกออกได้เป็น
 - ข้อมูลจำนวนเต็ม (integer)
 - ข้อมูลตรรกะ (booleans) ค่า T/F
 - ข้อมูลจำนวนจริง (Real) ที่แสดงโดยมีการใช้จุดทศนิยม
 - ข้อมูลจำนวนเชิงซ้อน (Complex) ที่แสดงค่าสองส่วนคือ ส่วนของจำนวนจริง (real part) กับส่วนของจำนวนจินตภาพ (imaginary part) ที่นำหน้าตัวอักษร j

ภาษาไพทอน ข้อมูลลำดับ

Python 3 The standard type hierarchy



- ข้อมูลลำดับที่ใช้ในภาษาไพทอน แทนกลุ่มของข้อมูลที่มีได้มากกว่า 1 สมาชิก ตามลำดับ
 - ข้อมูลลำดับที่ไม่เปลี่ยนรูป (immutable)
 - ข้อมูลสายอักขระ (string) ที่สมาชิกทุกตัวเป็นตัวอักษร
 - ข้อมูลทูเปิล (tuple) แยกสมาชิกระหว่างเครื่องหมายวงเล็บเปิดกับปิด
 - ข้อมูลไบต์ (bytes) สมาชิก = ASCII
 - ข้อมูลลำดับที่เปลี่ยนได้ (mutable)
 - ข้อมูลรายการ (list) ที่มีการเพิ่มหรือลดจำนวนสมาชิกได้หลังการสร้าง
 - ข้อมูลแถวลำดับไบต์ (byte arrays) ที่สมาชิกแทนด้วยรหัส ASCII และสามารถเพิ่มหรือลดสมาชิกได้

ภาษาไพทอน ข้อมูลเซตและอื่น ๆ

Set types

Sets

(class `set`)

Frozen sets

(class `frozenset`)

Mappings

Dictionaries
(class `dict`)

Callable

< Functions, Methods, Classes >

Modules

- ข้อมูลเซตในภาษาไพทอน แทนกลุ่มของข้อมูลที่มีได้มากกว่า 1 สมาชิก โดยไม่มีลำดับ และทุกสมาชิกต้องต่างกันทั้งหมด แยกออกได้เป็น
 - ข้อมูลเซต (set) ที่สมาชิกอยู่ระหว่างเครื่องหมายปีกกา คั่นแต่ละสมาชิกด้วยคอมมา และสามารถเพิ่มหรือลดจำนวนสมาชิกได้ ลำดับของสมาชิกไม่มีความสำคัญขึ้นกับ hashing
 - ข้อมูลเซตฟรอสเซิน (frozen set) ที่สร้างด้วยฟังก์ชัน `frozenset({s1, s2, ..., sn})` หลังจากสร้างแล้วไม่สามารถเพิ่มหรือลดจำนวนสมาชิกได้
- ข้อมูลการส่ง (mapping) ในภาษาไพทอน แทนกลุ่มของข้อมูลที่มีได้มากกว่า 1 สมาชิก โดยมีการใช้คีย์ในการเข้าถึง คีย์ต้อง hashable
- ข้อมูลการส่ง สร้างโดยใช้เครื่องหมายวงเล็บปีกกาเปิด และปีกกาปิด และสมาชิกประกอบด้วยคีย์กับค่าที่ใช้ colon คั่นแต่ละสมาชิกคั่นด้วยคอมมา



นิพจน์และตัวดำเนินการ



นิพจน์ในภาษาไพทอน

● นิยาม นิพจน์ในภาษาไพทอนคือ

- ชื่อวัตถุ (อ้างถึงค่า) หรือค่า (จำนวนเต็ม จำนวนจริง สายอักขระ)
- ผลลัพธ์ที่ได้จากเทอมภายในวงเล็บ (สื่อลำดับการประมวลผล จากวงเล็บในสุดก่อน)
- ผลลัพธ์ที่ได้จากการประมวลผลของตัวดำเนินการกับเทอม
- ผลลัพธ์จากการเรียกฟังก์ชัน

● ตัวอย่างของนิพจน์

>> 5 # คือนิพจน์ที่ส่งค่าเป็นจำนวนเต็ม 5

>> 5 * 3.0 # คือนิพจน์ที่ได้จากการคูณกันของจำนวนเต็มกับจำนวนจริง

>> 5*(4+2) # ลำดับการประมวลผลจะเกิดจากการคำนวณในวงเล็บก่อน

ตัวดำเนินการจำนวนในภาษาไพทอน

- จำนวนมีตัวดำเนินการที่ใช้คือ + (บวก) - (ลบ) * (คูณ) / (หาร) ** (ยกกำลัง)

```
>> 5 - 3 * 2 # ได้ผลลัพธ์เป็น -1
```

```
>> 5/2**3 + 3/8 # ได้ผลลัพธ์เป็น 1.0
```

- สำหรับจำนวนเต็มสองจำนวนใด ๆ
 - // ใช้สำหรับการหารเอาส่วน
 - % ใช้สำหรับการหารเอาเศษ
 - ^ ใช้สำหรับการหา Exclusive-or เมื่อจำนวนเต็มทั้งหมดถูกเปลี่ยนเป็นจำนวนฐานสอง

```
>> 5 // 3 # ได้ผลลัพธ์เป็น 1
```

```
>> 5 % 3 # ได้ผลลัพธ์เป็น 2
```

```
>> 5 ^ 3 # ได้ผลลัพธ์เป็น 6 = 1012 112 = 1102 = 6
```

ตัวดำเนินการสายอักขระในภาษาไพทอน



- สายอักขระในภาษาไพทอนเป็นวัตถุที่ใช้ร่วมกับฟังก์ชัน (method) ต่าง ๆ เช่น

```
>> "Krung".lower() # ได้ผลลัพธ์เป็น 'krung'
```

```
>> "My name is Krung".split() # ได้ผลลัพธ์เป็น ['My', 'name', 'is', 'Krung']
```

- ตัวดำเนินการในสายอักขระคือ + แทนการต่อกันของสายอักขระสองสาย

```
>> 'Krung' + ' ' + 'Sinapiromsaran' # ได้ผลลัพธ์เป็น 'Krung Sinapiromsaran'
```

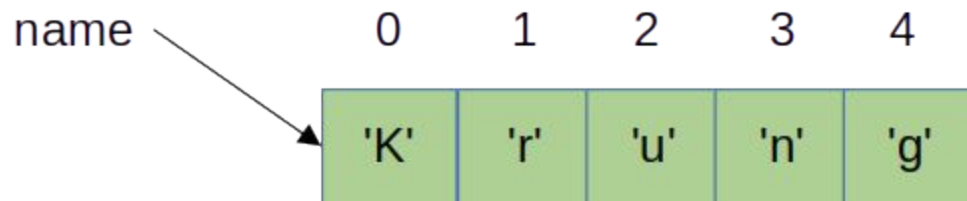
- สำหรับตัวดำเนินการ * ใช้ร่วมกับจำนวนเต็มเพื่อแสดงการสร้างสายอักขระซ้ำ

```
>> '='*10 # ได้ผลลัพธ์เป็น '=========='
```

ตัวดำเนินการสายอักขระในภาษาไพทอน

- สำหรับข้อมูลสายอักขระในภาษาไพทอน สร้างโดยใช้สัญลักษณ์ ฟนทอง หรือฟันทูสองตัว
- ข้อมูลสายอักขระ มีการอ้างอิงสมาชิกโดยใช้ดัชนี เริ่มจาก 0 ถึง จำนวนสมาชิกลบหนึ่ง เช่น

```
>> name = "Krung"
```



```
>> name[1] # ได้ค่าคือ 'r'
```

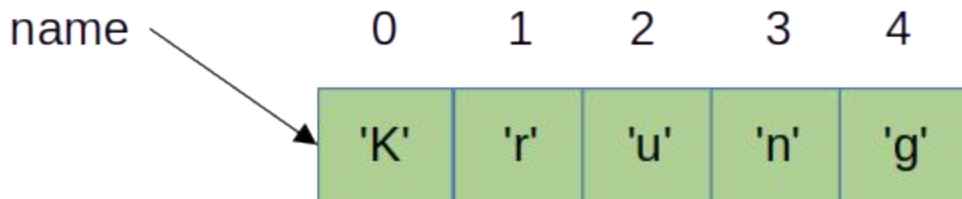
```
>> name[3] # ได้ค่าคือ 'n'
```

```
>> name[4] # ได้ค่าคือ 'g'
```

ตัวดำเนินการสายอักขระในภาษาไพทอน

- สายอักขระเป็นข้อมูลลำดับที่แต่ละสมาชิกคือตัวอักขระ มีหลักในการอ้างถึงสมาชิก โดยการใช้ดัชนี
- ในสายอักขระ ตัวอักขระแรกสุดมีค่าดัชนีคือ 0 ไปจนถึงจำนวนทั้งหมดลบหนึ่ง

```
>> name = "Krung"
```



- ฟังก์ชันที่มักใช้ร่วมกับสายอักขระคือ `len(S)` ส่งค่าจำนวนสมาชิกในสายอักขระ

```
>> len(name) # ได้ผลลัพธ์เป็น 5
```

ตัวดำเนินการสายอักขระในภาษาไพทอน

- การใช้ a:b ใช้สำหรับอ้างถึงดัชนีแรก a ไปจบที่ดัชนี b-1
- การใช้ a:b:inc ใช้สำหรับการอ้างถึงดัชนีแรก a เพิ่มครั้งละ inc จนค่าไม่เกิน b-1

```
>> name = "Krung"
```

```
>> name[2:4] # ได้ค่าเป็น 'un'
```

- ผู้เขียนสามารถใช้ : กับสายอักขระได้เช่นกัน

```
>> name[:3] # ได้ค่าเป็น 'Kru'
```

```
>> name[4:2:-1] # ได้ค่าเป็น 'gn'
```

```
>> name[::-1] # ได้สายอักขระย้อนลำดับ 'gnurK'
```

ตัวดำเนินการตรรกะในภาษาไพทอน



- ค่าตรรกะจะแสดงผลออกมาสองแบบคือ True หรือ False

```
>> True | False # ได้ค่าเป็น True
```

```
>> False & True # ได้ค่าเป็น False
```

```
>> not True # ได้ค่าเป็น False
```

```
>> True ^ True # ได้ค่าเป็น False
```

- ค่าของตัวแปรตรรกะ เมื่อใช้ร่วมกับจำนวนจะถูกแปลงเป็นจำนวนเต็มโดยที่ True = 1 และ False = 0

```
>> True + 2 # ได้ผลลัพธ์เป็น 3
```

```
>> False * 4.0 # ได้ผลลัพธ์เป็น 0.0
```

ตัวดำเนินการลำดับในภาษาไพทอน

- สำหรับข้อมูลลำดับในภาษาไพทอน ผู้เขียนอ้างอิงถึงสมาชิกโดยการใช้ดัชนีจำนวนเต็มที่ไม่เป็นลบ ระหว่างเครื่องหมายวงเล็บปีกกึ่งเปิดและปิด โดยที่สมาชิกตัวแรกมีดัชนีเป็นศูนย์ เช่น

```
>> L = (3, 4, 5, 6, 7, 8)
```

```
>> L[1] # ได้ค่าเป็น 4
```

```
>> L[0] + L[2] # ได้ค่าเป็น 8
```

- ผู้เขียนสามารถใช้ : แทนการสร้างดัชนีในภาษาไพทอน สามารถใช้ร่วมกับค่าลบ เช่น

```
>> L[2:4] # ได้ค่าเป็น [5, 6]
```

```
>> L[4:2:-1] # ได้ค่าเป็น [6, 5]
```

```
>> L[::-1] # ได้ลำดับสมาชิกลย้อนกลับ (8, 7, 6, 5, 4, 3)
```



การเขียนคำสั่งในภาษาไพทอน



ตัวดำเนินการลำดับในภาษาไพทอน




- ภาษาไพทอนเป็นภาษาการพัฒนาระดับสูง ทำงานแบบแปลที่ละบรรทัดโดยไม่มีการเปลี่ยนเป็นรหัสภาษาเครื่อง
- ภาษาไพทอนทำงานในสภาพแวดล้อมของภาษาไพทอนที่สามารถอ้างอิงถึง
 - ข้อมูลที่เป็นจำนวน สายอักขระ ค่าตรรกะ
 - ข้อมูลที่บันทึกในหน่วยความจำที่อ้างอิงโดยชื่อ
 - คำสั่งและการควบคุมการประมวลผล
 - ฟังก์ชันและโมดูลต่าง ๆ ที่อ้างอิงโดยชื่อ

คำสั่งและการควบคุมการประมวลผล

- คำสั่งกำหนดค่า (Assignment) ที่ใช้เครื่องหมาย =
- คำสั่ง if เพื่อการทดสอบเงื่อนไขก่อนการประมวลผล รวมถึง else, elif
- คำสั่ง for เพื่อใช้ในการวนซ้ำของวัตถุที่วนซ้ำได้
- คำสั่ง while สำหรับรวมกลุ่มของคำสั่งเมื่อเงื่อนไขเป็นจริง
- คำสั่ง try ใช้ในการส่งสัญญาณเมื่อพบว่ามีความผิดปกติในการประมวลผลคู่กับ except
- คำสั่ง raise สำหรับการส่งสัญญาณความผิดปกติออกไป
- คำสั่ง class ในการสร้างวัตถุใหม่ในสภาพแวดล้อมภาษาไพทอน
- คำสั่ง def ใช้ในการนิยามฟังก์ชัน หรือการเรียกฟังก์ชันจากวัตถุ

การควบคุมโปรแกรมภาษาไพทอน



- การควบคุมการทำงานของโปรแกรมภาษาไพทอนประกอบด้วยสามลักษณะ
- การประมวลผลคำสั่งที่ละบรรทัดตามลำดับ (Sequential control construct) คำสั่งจะถูกประมวลผลจากบรรทัดแรกไล่ลงมา
- การประมวลผลแบบมีเงื่อนไข (Conditional control construct) กลุ่มของคำสั่งที่มีเงื่อนไขที่เท่ากัน จะถูกประมวลผลขึ้นกับเงื่อนไข
- การประมวลผลแบบวนซ้ำ (Iterative control construct) กลุ่มของคำสั่งที่มีเงื่อนไขที่เท่ากัน จะถูกประมวลผลแบบวนซ้ำจนกระทั่งเงื่อนไขไม่จริง

คำสั่งประมวลผลตามเงื่อนไข



- คำสั่ง if-elif-else เป็นคำสั่งที่จะให้ภาษาไพทอนประมวลผลคำสั่งตามเงื่อนไข

if condition 1:

Statement 1

elif condition 2:

Statement 2

else:

statement 3

- ในกรณีที่ condition 1 มีค่าเป็น True ให้ประมวลผล statement 1 มิฉะนั้นถ้า condition 2 เป็นจริงให้ประมวลผล statement 2 มิฉะนั้นให้ประมวลผล statement 3

คำสั่งประมวลผลตามเงื่อนไข



- ตัวอย่าง

```
>> x = -2
>> if x > 0:
>>     print(x, ' is positive.')
>> elif x < 0:
>>     print(x, ' is negative.')
>> else:
>>     print(x, ' is zero.')
```

- เงื่อนไขใน if ต้องส่งค่าความจริงเป็น จริงหรือเท็จ ผู้เขียนสามารถเขียนนิพจน์ตรรกะได้ ด้วยตัวดำเนินการ and or not

คำสั่ง if ในหนึ่งบรรทัด

- ภาษาไพทอนยังสนับสนุนการประมวลผลคำสั่ง if ในบรรทัดเดียว
- สมมติค่าของตัวแปรโปรแกรม x ถ้ามีค่าเป็นบวกให้ส่งค่าออกมาเป็น +1 มิฉะนั้นให้ส่งค่าออกมาเป็นลบ เก็บไว้ในตัวแปรโปรแกรม signx

```
>> x = -2
```

```
>>> signx = +1 if x > 0 else -1; signx # ได้ค่า -1
```

```
>>> x = 3
```

```
>>> signx = +1 if x > 0 else -1; signx # ได้ค่า 1
```

- วิธีการอ่านนิพจน์นี้คือ ไฮยีนเอ็กซ์ เท่ากับ บวกหนึ่ง ถ้า เอ็กซ์มากกว่าศูนย์ มิฉะนั้นเป็น ลบหนึ่ง

คำสั่ง for

- คำสั่ง for ใช้กับวัตถุที่วนซ้ำได้ เช่น รายการ ทูเปิ้ล เซต

for x in list:

statement 1

- คำสั่งดังกล่าวยังใช้ใน list-comprehension กับ dictionary-comprehension ที่ใช้อ้างถึงข้อมูลที่ละตัว

```
>> sum([i for i in range(11)])
```

- แสดงค่าผลบวกรวมของจำนวนเต็มจาก 0, 1, 2, ..., 10

```
>> dow = {0:'sun', 1:'mon', 2:'tue', 3:'wed', 4:'thu', 5:'fri', 6:'sat'}
```

- สร้างตัวแปรโปรแกรม dow ที่แสดงค่าตัวอักษรสามตัวของวันในสัปดาห์

คำสั่งการวนซ้ำ for

- ตัวอย่าง

```
>> fac, n = 1, 5
```

```
>> for i in range(2,n+1):
```

```
>>     fac *= i
```


- คำนวณค่าแฟคตอเรียลของ n โดยคูณจาก 1 ด้วย 2 วนซ้ำจนครบ n

```
>> n = 5
```

```
>> oddsum = sum(2*i+1 for i in range(n))
```

- คำนวณค่าผลบวกจำนวนคี่ 5 จำนวนแรก

คำสั่ง while



- คำสั่ง while ใช้สำหรับการวนซ้ำในขณะที่เงื่อนไขเป็นจริง

while condition:

statement 1

- เงื่อนไข condition จะถูกประมวลผลก่อนการวนซ้ำ ในกรณีที่เงื่อนไขเป็นจริงจะเกิดการวนซ้ำของการประมวลผล statement 1 และคำสั่งใน statement 1 ต้องมีการเปลี่ยนแปลงเพื่อให้เงื่อนไขมีโอกาสเป็นเท็จได้ มิฉะนั้นการวนซ้ำจะไม่มีที่สิ้นสุด

คำสั่งการวนซ้ำ while

- ตัวอย่าง

```
>> fac, n = 1, 5
```

```
>> i = 1
```

```
>> while i < n+1:
```

```
>>     fac *= i
```

```
>>     i += 1
```

คำสั่ง while

- คำสั่ง while ใช้สำหรับการวนซ้ำในขณะที่เงื่อนไขเป็นจริง

while condition:

statement 1

- เงื่อนไข condition จะถูกประมวลผลก่อนการวนซ้ำ ในกรณีที่เงื่อนไขเป็นจริงจะเกิดการวนซ้ำของการประมวลผล statement 1 และคำสั่งใน statement 1 ต้องมีการเปลี่ยนแปลงเพื่อให้เงื่อนไขมีโอกาสเป็นเท็จได้ มิฉะนั้นการวนซ้ำจะไม่มีที่สิ้นสุด

โจทย์ ๑



- กำหนดคำสั่งในภาษาไพทอนต่อไปนี้

Width, Height, Delimiter = 24, 15.0, ''

- จงคำนวณนิพจน์ที่กำหนดให้ต่อไปนี้
 - Width/3
 - Width/3.0
 - Height / 4
 - Delimiter * 6

โจทย์ ๒

- ปริมาตรของทรงกลมมีสูตรเป็น $\frac{4}{3}\pi r^3$ เมื่อ r แทนรัศมีของทรงกลม จงคำนวณหาปริมาตรของทรงกลมรัศมี 4

โจทย์ ๓

- กำหนดให้ราคาที่อยู่ปกหน้าของหนังสือราคา 50.00 บาท ในขณะที่ร้านขายหนังสือได้ส่วนลด 30% จากราคาหน้าปก สมมติว่าค่าขนส่งของหนังสือเล่มแรกคิด 5 บาท และหนังสือเล่มอื่น ๆ คิด 1 บาทต่อเล่ม จงหาราคาสั่งหนังสือมาขายทั้งหมด 100 เล่ม

โจทย์ ๔



- จงเขียนโปรแกรมภาษาไพทอนในการคำนวณค่าทางสถิติของข้อมูลจำนวน
- จงเขียนโปรแกรมภาษาไพทอนหาค่าสูงสุดและหาค่าต่ำสุดของข้อมูลจำนวน
- จงเขียนโปรแกรมภาษาไพทอนเพื่อเรียงข้อมูลจำนวนจากมากไปหาน้อย
 - Selection sort
 - Insertion sort
 - Quick sort
 - Heap sort
 - Merge sort



คำถาม

