Computer Programming and
Developing Applications for Education

Chapter 6: Immutable and Mutable Data Types: Tuple, List, Dictionary

Pasawut Cheerapakorn
Suan Sunandha Rajabhat University

DTI3302 Computer Programming and Developing Applications for Education
Department of Digital Technology for Education
Faculty of Education, Suan Sunandha Rajabhat University

Content Credit By: Asst.Prof.Nutthapat Kaewrattanapat, PhD.

Course Description:

หลักการ ทฤษฎี ที่เกี่ยวข้องการเขียนโปรแกรมและพัฒนาแอปพลิเคชัน หลัก การพัฒนาโปรแกรมคอมพิวเตอร์ คุณสมบัติของโปรแกรมภาษาชนิดต่าง ๆ หลักการเบื้องต้นเกี่ยวกับองค์ประกอบ ลักษณะคำสั่ง การเขียนโปรแกรม ขั้นตอนวิธี การวิเคราะห์ การออกแบบ แอปพลิเคชันเพื่อการศึกษา การประเมิน ซอฟต์แวร์ สามารถพัฒนาแอปพลิเคชันเพื่อการศึกษา

Principles, theories associated with computer programming and development applications, computer programming principles, computer language, elements of computer language, syntax, computer programming, algorithms, analysis and design application for education, software evaluation, candidate teachers able to developing applications for education.

Algorithm Design

Python Programming

AI for Programming

Course Outline:

- Chapter 1 – Computer Programming
- Chapter 2 – Introduction to Python Programming
- Chapter 3 – Conditional or Decision Statement
- Chapter 4 – Iteration Statement
- Chapter 5 – Strings
- **Chapter 6 – Lists, Tuples, Sets, Dictionaries**
- Chapter 7 – Functions
- Chapter 8 – Object-Oriented Programming: OOP

Measurement and Evaluation:

การวัดและประเมินผล

1. ระหว่างการจัดการเรียนรู้
   - สอบ Pre-test 0%
   - การมอบหมายงาน 20%
   - สอบ Post-test 15%
   - การมีส่วนร่วมในชั้นเรียน 5%

2. การสอบกลางภาค (Midterm Examination)
   - ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน) 20%

3. โครงงานประจำภาคเรียน (Term Project)
   - โครงงานและการนำเสนอ 20%

4. การสอบปลายภาค (Final Examination)
   - ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน) 20%

ร้อยละ	ระดับผลการเรียน	ความหมาย
86 – 100	A	ดียอดเยี่ยม
82 – 85	A-	ดีเยี่ยม
78 -81	B+	ดีมาก
74 -77	B	ดี
70 – 73	B-	ค่อนข้างดี
66 – 69	C+	ปานกลางค่อนข้างดี
62 – 65	C	ปานกลาง
58 – 61	C-	ปานกลางค่อนข้างอ่อน
54 – 57	D+	ค่อนข้างอ่อน
50 – 53	D	อ่อน
46 – 49	D-	อ่อนมาก
0 - 45	F	ตก

Measurement and Evaluation:

ครั้งที่ / สัปดาห์	บทเรียน / หัวข้อ
1	แนะนำรายวิชา การวัดและการประเมินผล หัวข้อเรียนรู้ (Introduction to Course)
2	บทที่ 1 การโปรแกรมคอมพิวเตอร์ (Computer Programming)
3	บทที่ 2 พื้นฐานการโปรแกรมภาษาไพทอน (Introduction to Python Programming)
4	บทที่ 3 การโปรแกรมแบบเงื่อนไขหรือตัดสินใจ (Conditional or Decision Statement)
5	บทที่ 4 การโปรแกรมแบบทำซ้ำ (Iteration Statement)
6	บทที่ 5 การโปรแกรมสายอักขระ (String)
7	**บทที่ 6 ลิสต์ ทูเพิล และดิกชันนารี (List, Tuple, and Dictionaries)**
8	บทที่ 7 ฟังก์ชั่น (Function)

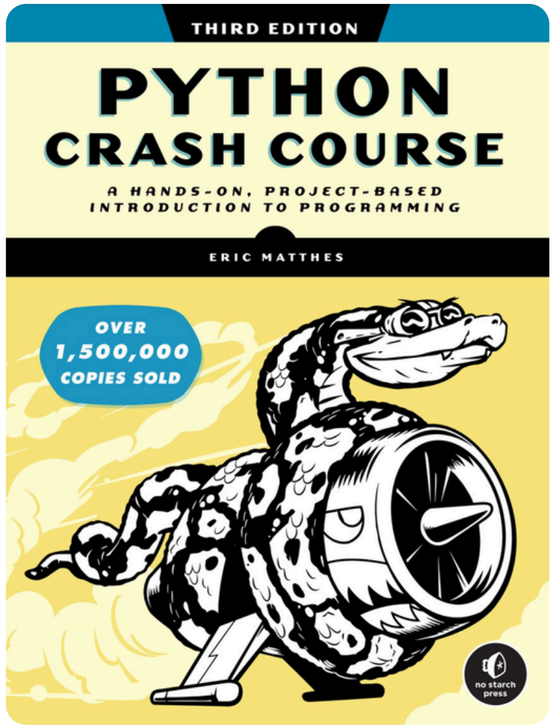
Measurement and Evaluation:

ครั้งที่ / สัปดาห์	บทเรียน / หัวข้อ
9	สอบกลางภาค (Midterm Examination)
10	บทที่ 7 ฟังก์ชั่น (Function) (ต่อ)
11	บทที่ 8 การโปรแกรมเชิงวัตถุ (Object-Oriented Programming: OOP)
12	บทที่ 8 การโปรแกรมเชิงวัตถุ (Object-Oriented Programming: OOP) (ต่อ)
13	การเขียนโปรแกรมด้วยปัญญาประดิษฐ์ (AI for Programming)
14	สอบปลายภาค (Final Examination)
15	นำเสนอและส่งโครงงาน (Project Pitching and Presentation)
16	

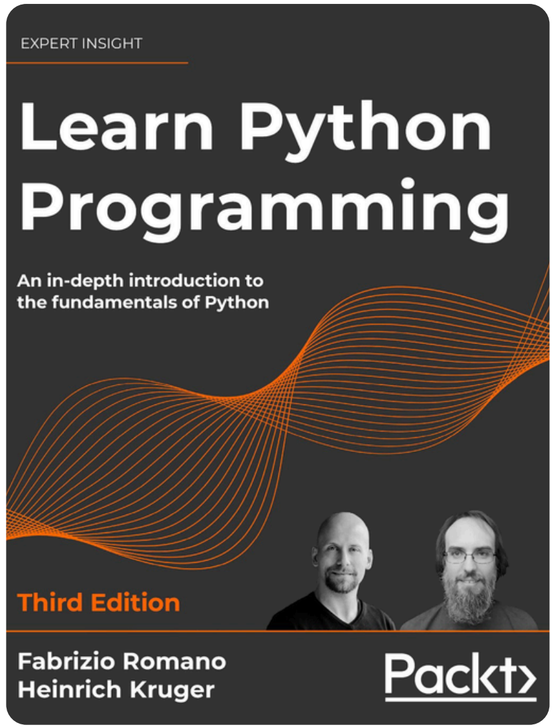
Chapter 6: Immutable and Mutable Data Types: Tuple, List, Dictionary

1. Immutable Data Type
2. Mutable Data Type
3. Immutable and Mutable Data Types Function
4. Tuple
5. List
6. Dictionary

Learning Materials Suggestion:



Python Crash Course: A Hands-On, Project-Based Introduction to Programming [3 ed.]



Learn Python Programming
An in-depth introduction to the fundamentals of Python [3 ed.]

Website

- https://www.w3schools.com/python/
- https://realpython.com/
- https://docs.python.org/3/tutorial/index.html
- https://pythontutor.com/

Pre Test

Question:

1. ใน Python “ลำดับ” (Sequences) คืออะไร

A โครงสร้างข้อมูลที่เก็บข้อมูลเป็นลำดับเรียงต่อกัน

B โครงสร้างของโค้ดที่เขียนต่อเป็นลำดับเรียงต่อกัน

C โครงสร้างโอเปอร์เรเตอร์ที่เก็บข้อมูลเป็นลำดับเรียงต่อกัน

D ไม่มีข้อใดถูก

Question:

2. ข้อใด <u>ไม่ใช่</u> ประโยชน์ของประเภทข้อมูลแบบลำดับ

A ช่วยในการจัดเก็บข้อมูลที่มีลำดับ

B ช่วยในการเข้าถึงข้อมูลโดยอ้างอิงลำดับหรือดัชนี

C ช่วยในการเก็บข้อมูลที่มีรูปแบบแบบซับซ้อน

D ช่วยในการคำนวณตัวเลขที่เกี่ยวข้องได้ง่าย

Question:

3. ข้อใด ไม่ใช่ ลำดับที่ไม่สามารถเปลี่ยนแปลงได้ (Immutable Sequences)

A ประเภทจำนวนเต็ม (Integers)

B ประเภทบูลีน (Boolean)

C ประเภทสายอักขระ (String)

D ประเภทรายการ (List)

Question:

4. ข้อใด ไม่ใช่ ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

A ประเภทรายการ (List)

B ประเภทจำนวนทศนิยม (Floating-point numbers)

C ประเภทเซ็ต (Set)

D ประเภทพจนานุกรม (Dictionary)

Question:

5. ข้อใด <u>ไม่ใช่</u> ลักษณะสำคัญของประเภทข้อมูลแบบทูเพิล (Tuple)

A ไม่สามารถแก้ไขหรือเปลี่ยนแปลงค่าของสมาชิกภายใน tuple หลังจากที่สร้างขึ้น

B สามารถแก้ไขหรือเปลี่ยนแปลงค่าของสมาชิกภายใน tuple หลังจากที่สร้างขึ้นได้

C Tuples สามารถมีสมาชิกหลายประเภทข้อมูลและสามารถมีขนาดต่าง ๆ ได้

D Tuple สร้างโดยใช้วงเล็บ (), และสามารถมีหรือไม่มีวงเล็บรอบสมาชิก

Question:

6. จากคำสั่ง แสดงผลข้อใด

```
countries = ("ไทย", "สหรัฐอเมริกา", "อังกฤษ", "ญี่ปุ่น", "จีน", "อินเดีย", "เบลเยียม", "เกาหลีใต้")
if "ญี่ปุ่น" in countries:
  print("เป็นสมาชิก")
else:
  print("ไม่เป็นสมาชิก")
```

A เป็นสมาชิก

B ไม่เป็นสมาชิก

C เป็นสมาชิก และไม่เป็นสมาชิก

D ไม่เป็นสมาชิก และเป็นสมาชิก

Question:

7. จากคำสั่ง แสดงผลข้อใด

```
dog_breeds1 = ("บีเกิ้ล", "โกลเด้นรีทรีฟเวอร์", "ชิวาว่า")
dog_breeds2 = ("พุดเดิ้ล", "บูลด็อก", "โรตวายเลีย")
all_dog_breeds = dog_breeds1 + dog_breeds2
print(all_dog_breeds.count("ชิวาว่า"))
```

A 0

C 2

B 1

D 3

Question:

8. จากคำสั่ง แสดงผลข้อใด

```
student_height = [163.5, 150.0, 167.0, 161.25, 170.0]
maximum_height = max(student_height)
print("ส่วนสูงสูงสุดของนักเรียนคือ:", maximum_height)
```

A ส่วนสูงสูงสุดของนักเรียนคือ: 163.5

B ส่วนสูงสูงสุดของนักเรียนคือ: 167.0

C ส่วนสูงสูงสุดของนักเรียนคือ: 170.0

D ส่วนสูงสูงสุดของนักเรียนคือ: 150.0

Question:

9. จากคำสั่ง แสดงผลข้อใด

```
students = ["Alice", "Bob", "Charlie", "David", "Eve"]
math_scores = [85, 92, 78, 88, 95]
average = sum(math_scores)/len(math_scores)
print(f"ค่าเฉลี่ย( = {average}")
```

A ค่าเฉลี่ย: 59.3

C ค่าเฉลี่ย: 87.6

B ค่าเฉลี่ย: 43.5

D ค่าเฉลี่ย: 100.4

Question:

10. จากคำสั่ง แสดงผลข้อใด

```
student = {
   "name": "Nutthapat",
   "age": 20,
   "grade": "A"
}
print(len(student))
```

A 1

B 2

C 3

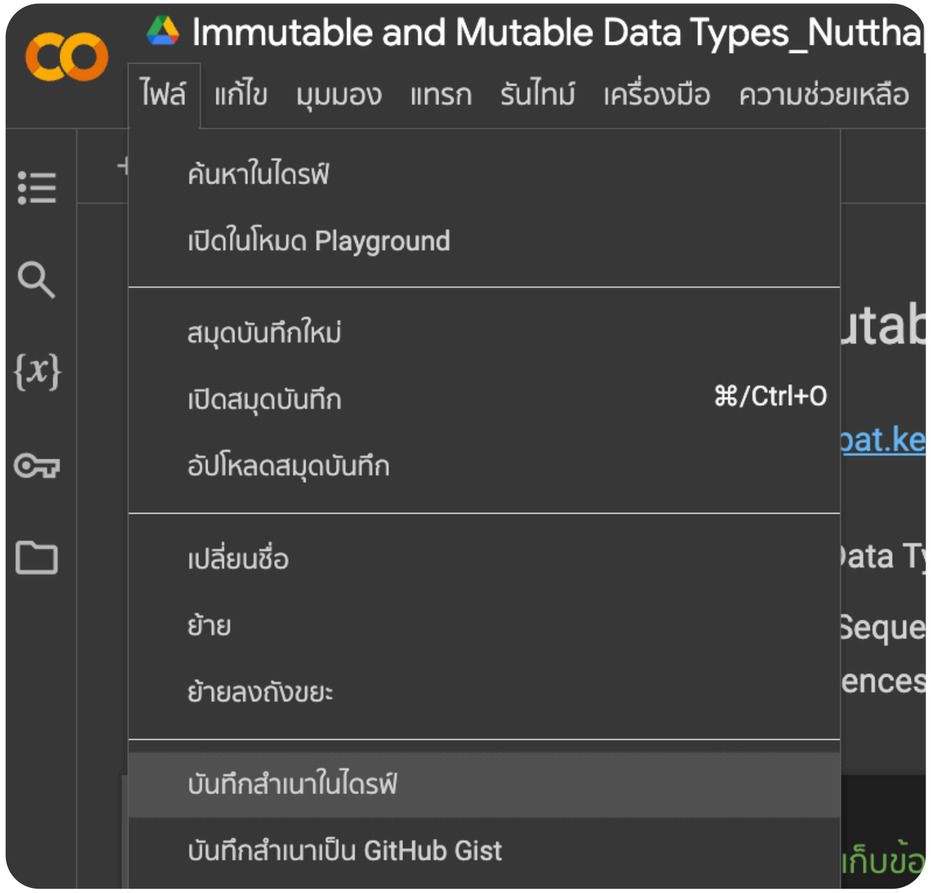
D 4

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

ไฟล์ตัวอย่างคำสั่ง (Code) ในบทเรียนที่ 6

https://cutt.ly/ajnuth_python6

หากต้องการไฟล์นี้เป็นของนักศึกษาเอง ให้คลิกตามภาพ เพื่อสำเนาไฟล์คำสั่งนี้ให้เป็นของนักศึกษาเอง จากนั้นให้ทำการเปลยี่ นชื่อไฟล์เป็นชื่อของนักศึกษาเพื่อใช้ส่วนตัวได้



ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

ใน Python “ลำดับ” (Sequences) คือ โครงสร้างข้อมูลที่เก็บข้อมูลเป็นลำดับเรียงต่อกัน เช่น text = “Hello, World” มีการเรียงลำดับของข้อมูลที่เป็นตัวอักษร

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

ประโยชน์ของประเภทข้อมูลแบบลำดับ

1. การจัดเก็บข้อมูลที่มีลำดับ ตัวแปรแบบลำดับช่วยในการจัดเก็บข้อมูลที่มีลำดับ เช่น รายการสินค้า, ลำดับของวันที่, รายการเพลงในเพลย์ลิสต์

2. การเข้าถึงข้อมูล ตัวแปรแบบลำดับช่วยในการเข้าถึงข้อมูลโดยอ้างอิงลำดับหรือดัชนี นี่เป็นวิธีการใช้งานข้อมูลในลำดับโดยใช้การเข้าถึงดัชนี เช่น my_list[0] เพื่อเข้าถึงสมาชิกแรกของรายการ

3. การปรับปรุงข้อมูล ตัวแปรแบบลำดับที่เป็น mutable (เช่น lists) ช่วยในการปรับปรุงหรือแก้ไขข้อมูลในลำดับ ซึ่งเปิดโอกาสในการปรับปรุงข้อมูลหรือการเรียงลำดับใหม่

4. การวนรอบ (Iteration) ตัวแปรแบบลำดับช่วยในการวนรอบข้อมูล โดยสามารถใช้ลูป (loop) เพื่อเข้าถึงและประมวลผลข้อมูลในลำดับได้ เช่น for item in my_list:

5. การทำงานกับข้อมูลในลำดับ ตัวแปรแบบลำดับช่วยในการดำเนินการต่าง ๆ กับข้อมูลที่อยู่ในลำดับ เช่น การค้นหาข้อมูล, การเรียงลำดับข้อมูล, หรือการรวมข้อมูลจากลำดับต่าง ๆ ไปยังลำดับใหม่

6. การจัดการข้อมูลที่มีความสัมพันธ์ (Associative Data) ในกรณีข้อมูลที่มีความสัมพันธ์ระหว่างคีย์และค่า เช่น dictionaries ที่ใช้คีย์เพื่อเข้าถึงค่า ตัวแปรแบบลำดับช่วยในการจัดการข้อมูลที่มีความสัมพันธ์นี้ได้อย่างมีประสิทธิภาพ

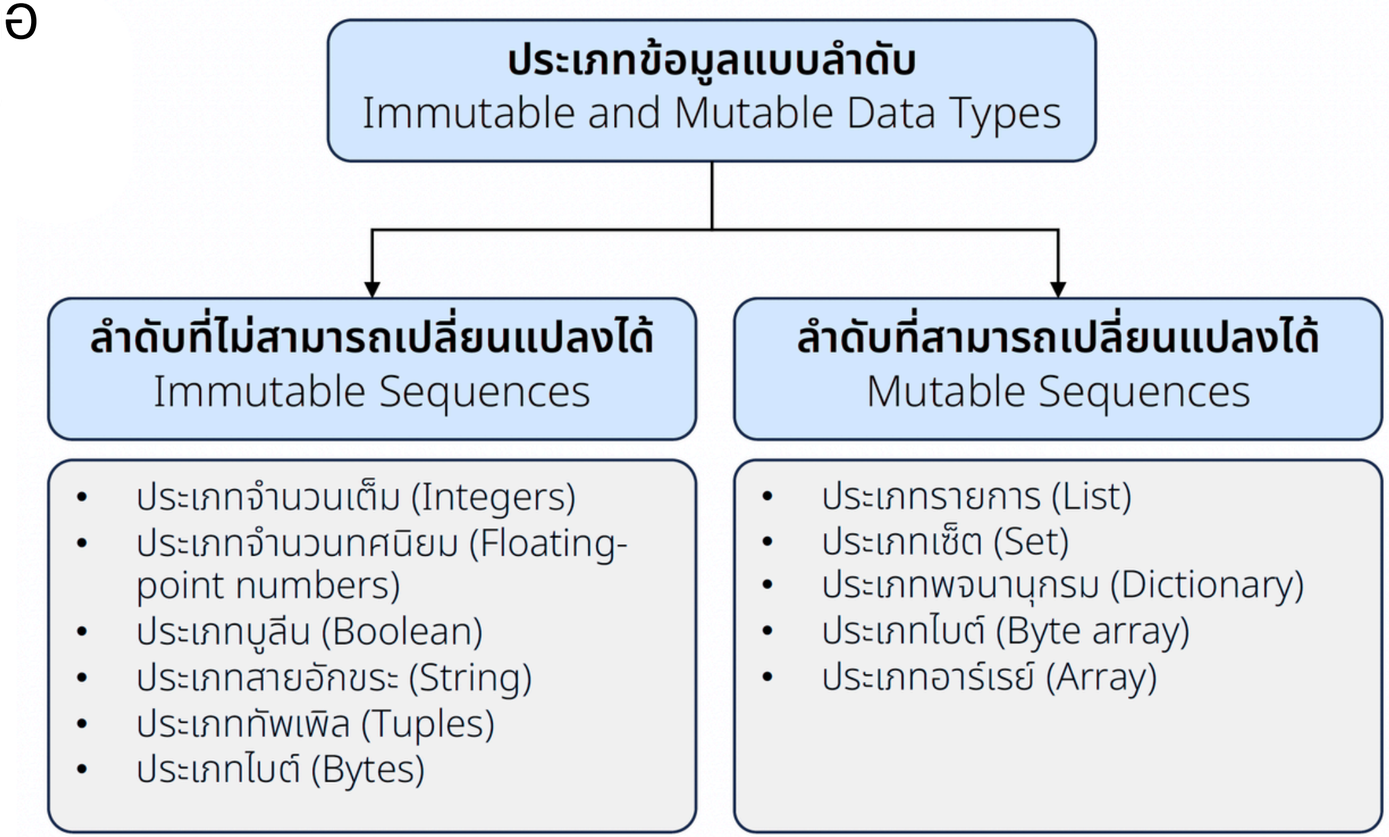
7. การเก็บข้อมูลที่มีรูปแบบแบบซับซ้อน การเก็บข้อมูลที่มีรูปแบบแบบซับซ้อน เช่น ข้อมูลที่มีโครงสร้างแถวและคอลัมน์คล้ายกับ Matrics

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

ตัวแปรแบบลำดับ มี 2 ประเภทหลัก คือ

1. ลำดับที่ไม่สามารถเปลี่ยนแปลงได้
(Immutable Sequences)

2. ลำดับที่สามารถเปลี่ยนแปลงได้
(Mutable Sequences)

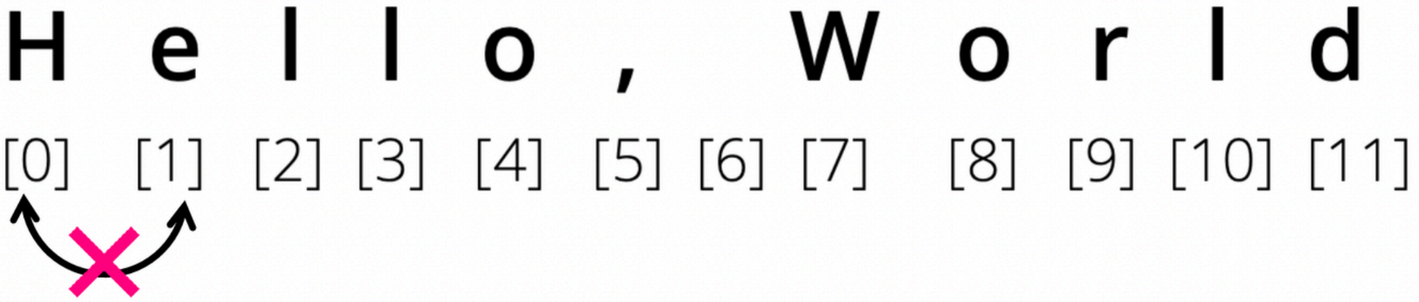


ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

1. ลำดับที่ไม่สามารถเปลี่ยนแปลงได้ (Immutable Sequences)

- Immutable sequences คือ ลำดับที่ไม่สามารถเปลี่ยนแปลงค่าได้หลังจากสร้างประเภทตัวแปรนั้นๆ
- ตัวอย่างของ Immutable sequences ได้แก่ tuples (ทูเพิล) และ strings (สายอักขระ)
- ลำดับที่ไม่สามารถเปลี่ยนแปลงลำดับได้มีความปลอดภัยและเหมาะสำหรับการใช้งานที่ไม่ต้องการการเปลี่ยนแปลงค่าหรือลำดับ

text = “Hello, World”



ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

1. ลำดับที่ไม่สามารถเปลี่ยนแปลงได้ (Immutable Sequences)

ประเภทข้อมูลแบบทูเพิล (Tuple)

- Tuples เป็นลำดับของข้อมูลที่ไม่สามารถเปลี่ยนแปลงค่าได้หลังจากสร้าง
- Tuples สามารถมีสมาชิกหลายประเภทข้อมูลและสามารถมีขนาดต่าง ๆ ได้
- เราสามารถใช้วงเล็บ ( ) เพื่อสร้าง tuples

```
1. my_tuple = (1, 2, 3)
2. print(my_tuple[0])
3. # ผลลัพธ์: 1
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

1. ลำดับที่ไม่สามารถเปลี่ยนแปลงได้ (Immutable Sequences)

ประเภทข้อมูลแบบทูเพิล (Tuple)

```
1. # การสร้าง tuple เพื่อเก็บข้อมูลขนาดคงที่ของวันเดือนปี
2. date_of_birth = (8, "มีนาคม", 1983)
3. # เข้าถึงข้อมูลใน tuple
4. day = date_of_birth[0]
5. month = date_of_birth[1]
6. year = date_of_birth[2]
7. print(f"วันเกิด: {day} {month} {year}")
```

วันเกิด: 8 มีนาคม 1983

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

1. ลำดับที่ไม่สามารถเปลี่ยนแปลงได้ (Immutable Sequences)

ประเภทข้อมูลแบบทูเพิล (Tuple)

```
1. #ตัวอย่างที่ 2 หาจำนวนหรือความยาวของข้อมูลใน Tuple
2. lenght = len(date_of_birth) #นับจำนวนหรือความยาวของข้อมูลใน Tuple
3. print(lenght)
```

```
3
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

1. ลำดับที่ไม่สามารถเปลี่ยนแปลงได้ (Immutable Sequences)

ประเภทข้อมูลแบบทูเพิล (Tuple)

```
1. #ตัวอย่างที่ 3 สร้าง tuple เพื่อเก็บชื่อเดือนและเข้าถึงชื่อเดือนตามระยะดัชนีที่ต้องการ (range)
2. #สร้าง tuple เพื่อเก็บชื่อเดือน
3. months = ("มกราคม", "กุมภาพันธ์", "มีนาคม", "เมษายน", "พฤษภาคม", "มิถุนายน",
   "กรกฎาคม", "สิงหาคม", "กันยายน", "ตุลาคม", "พฤศจิกายน", "ธันวาคม")
4. # แสดงผลชื่อเดือนตามระยะดัชนีที่ต้องการ (range)
5. print(months[6:11+1])
```

('กรกฎาคม', 'สิงหาคม', 'กันยายน', 'ตุลาคม', 'พฤศจิกายน', 'ธันวาคม')

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

1. ลำดับที่ไม่สามารถเปลี่ยนแปลงได้ (Immutable Sequences)

ประเภทข้อมูลแบบทูเพิล (Tuple)

```
1. #ตัวอย่างที่ 4 สร้าง tuple เพื่อเก็บชื่อเดือนและเข้าถึงชื่อเดือนตามระยะดัชนีที่ต้องการ (range)
2. #สร้าง tuple เพื่อเก็บชื่อเดือน
3. months = ("มกราคม", "กุมภาพันธ์", "มีนาคม", "เมษายน", "พฤษภาคม", "มิถุนายน",
   "กรกฎาคม", "สิงหาคม", "กันยายน", "ตุลาคม", "พฤศจิกายน", "ธันวาคม")
4. # แสดงผลชื่อเดือนตามระยะดัชนีที่ต้องการ (range)
5. print(months[:6])
```

('มกราคม', 'กุมภาพันธ์' , 'มีนาคม', 'เมษายน', 'พฤษภาคม', 'มิถุนายน')

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

1. ลำดับที่ไม่สามารถเปลี่ยนแปลงได้ (Immutable Sequences)

ประเภทข้อมูลแบบทูเพิล (Tuple)

```
1. #ตัวอย่างที่ 5 การแสดงสมาชิกของข้อมูลแบบไม่ใช้การวนลูป
2. #สร้าง tuple เพื่อเก็บชื่อเดือน
3. months = ("มกราคม", "กุมภาพันธ์", "มีนาคม", "เมษายน", "พฤษภาคม", "มิถุนายน",
   "กรกฎาคม", "สิงหาคม", "กันยายน", "ตุลาคม", "พฤศจิกายน", "ธันวาคม")
4. # แสดงผลชื่อเดือนตามระยะดัชนีที่ต้องการ (range)
5. print(*months[:6])
```

มกราคม กุมภาพันธ์ มีนาคม เมษายน พฤษภาคม มิถุนายน

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

1. ลำดับที่ไม่สามารถเปลี่ยนแปลงได้ (Immutable Sequences)

ประเภทข้อมูลแบบทูเพิล (Tuple)

```
1. #ตัวอย่างที่ 6 การตรวจสอบการมีของสมาชิกข้อมูล
2. # สร้าง tuple เพื่อเก็บชื่อประเทศ
3. countries = ("ไทย", "สหรัฐอเมริกา", "อังกฤษ", "ญี่ปุ่น", "จีน", "อินเดีย", "เบลเยียม", "เกาหลีใต้")
4. #ตรวจสอบการมีของสมาชิกข้อมูล
5. if "ญี่ปุ่น" in countries:
6.     print("เป็นสมาชิก")
7. else:
8.     print("ไม่เป็นสมาชิก")
```

เป็นสมาชิก

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

1. ลำดับที่ไม่สามารถเปลี่ยนแปลงได้ (Immutable Sequences)

ประเภทข้อมูลแบบทูเพิล (Tuple)

```
1. #ตัวอย่างที่ 7 การวนลูปเพื่อเข้าถึงสมาชิกข้อมูลแต่ละตัว
2. # สร้าง tuple เพื่อเก็บชื่อประเทศ
3. countries = ("ไทย", "สหรัฐอเมริกา", "อังกฤษ", "ญี่ปุ่น", "จีน", "อินเดีย", "เบลเยียม", "เกาหลีใต้")
4. for c in countries:
5.   print(c)
```

```
ไทย
สหรัฐอเมริกา
อังกฤษ
ญี่ปุ่น
จีน
อินเดีย
เบลเยียม
เกาหลีใต้
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

1. ลำดับที่ไม่สามารถเปลี่ยนแปลงได้ (Immutable Sequences)

ประเภทข้อมูลแบบทูเพิล (Tuple)

```
1. #ตัวอย่างที่ 8 การวนลูปแสดง index ของสมาชิกข้อมูล
2. # สร้าง tuple เพื่อเก็บชื่อประเทศ
3. countries = ("ไทย", "สหรัฐอเมริกา", "อังกฤษ", "ญี่ปุ่น", "จีน", "อินเดีย", "เบลเยียม", "เกาหลีใต้")
4. for i in range(len(countries)):
5.   print(i)
```

```
0
1
2
3
4
5
6
7
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

1. ลำดับที่ไม่สามารถเปลี่ยนแปลงได้ (Immutable Sequences)

ประเภทข้อมูลแบบทูเพิล (Tuple)

```
1. #ตัวอย่างที่ 9 การเชื่อมสมาชิกข้อมูลแบบ Tuple
2. dog_breeds1 = ("บีเกิ้ล", "โกลเด้นรีทรีฟเวอร์" , "ชิวาว่า")
3. dog_breeds2 = ("พุดเดิ้ล", "บูลด็อก", "โรตวายเลีย ")
4. # รวม (join) Tuple ด้วยการเพิ่ม
5. all_dog_breeds = dog_breeds1 + dog_breeds2
6. # แสดงผล Tuple ที่รวมกัน
7. print(all_dog_breeds)
```

('บีเกิ้ล', 'โกลเด้นรีทรีฟเวอร์' , 'ชิวาว่า', 'พุดเดิ้ล', 'บูลด็อก', 'โรตวายเลีย ')

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

1. ลำดับที่ไม่สามารถเปลี่ยนแปลงได้ (Immutable Sequences)

ประเภทข้อมูลแบบทูเพิล (Tuple)

```
1. #ตัวอย่างที่ 10 การนับจำนวนสมาชิก หรือ count
2. dog_breeds1 = ("บีเกิ้ล", "โกลเด้นรีทรีฟเวอร์" , "ชิวาว่า")
3. dog_breeds2 = ("พุดเดิ้ล", "บูลด็อก", "โรตวายเลีย")
4. all_dog_breeds = dog_breeds1 + dog_breeds2
5. print(all_dog_breeds.count("ชิวาว่า"))
```

1

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

- ลำดับที่สามารถเปลี่ยนแปลงได้ หมายถึง สามารถแก้ไขค่าของข้อมูลในลำดับได้หลังจากการสร้างตัวแปรและข้อมูลนั้น ๆ ขึ้นมา
- ตัวอย่างของลำดับที่สามารถเปลี่ยนแปลงได้ได้แก่ lists (ลิสต์) dictionary (ดิกชันนารี) และ byte arrays (อาเรย์ข้อมูลไบต์)
- สามารถเพิ่ม, ลบ, หรือแก้ไขข้อมูลในลำดับที่สามารถเปลี่ยนแปลงได้ตามต้องการโดยไม่จำเป็นต้องสร้างลำดับใหม่

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

Variable
ตัวแปรทั่วไป เก็บได้ 1 ค่า

```
fruit = “banana”
```

List variable
ตัวแปรลิสต์ เก็บได้มากกว่า 1 ค่า

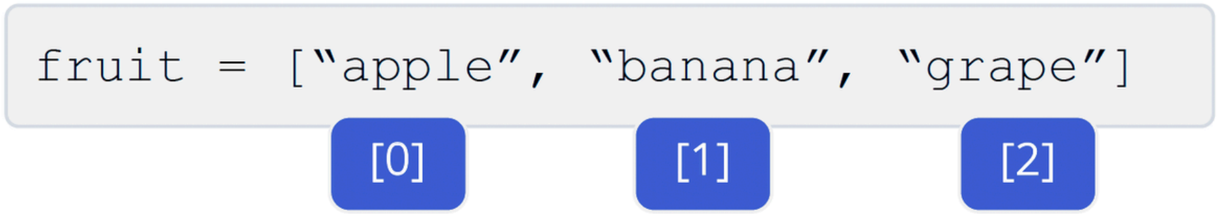
```
fruit = [“apple”, “banana”, “grape”]
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. # List
2. # ตัวอย่างที่ 1
3. #การสร้างตัวแปรลิสต์เพื่อเก็บข้อมูล 3 ค่า
4. fruit = ["apple", "banana", "grape"]
5. # การเข้าถึงข้อมูลบางค่าในตัวแปรลิสต์
6. print(fruit[1])
```



banana

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 2 สร้างตัวแปรเก็บส่วนสูงของนักเรียน 5 คน
2. student_height = [163.5, 150.0, 167.0, 161.25, 170.0]
3. print(student_height[3])
```

161.25

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 3 สร้างตัวแปรเก็บส่วนสูงของนักเรียน 5 คน และหาส่วนสูงมากที่สุด
2. student_height = [163.5, 150.0, 167.0, 161.25, 170.0]
3. # หาค่าสูงสุดของส่วนสูง
4. maximum_height = max(student_height)
5. print("ส่วนสูงสูงสุดของนักเรียนคือ:", maximum_height)
```

ส่วนสูงสูงสุดของนักเรียนคือ: 170.0

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 4 การเปลี่ยนแปลงค่าในลำดับที่ต้องการ
2. fruit = ["apple", "banana", "cherry"]
3. print(fruit)
4. #เปลี่ยนค่า ลำดับที่ 1 จาก banana เป็น blackcurrant
5. fruit[1] = "blackcurrant"
6. print(fruit)
```

['apple', **'banana'**, 'cherry']

['apple', **'blackcurrant'**, 'cherry']

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 5 การเพิ่มค่าเข้าไปต่อท้าย เรียกว่า append (แอพเพ็น)
2. fruit = ["apple", "banana", "cherry"]
3. print(fruit)
4. #เพิ่มค่า orange เข้าไปต่อท้าย
5. fruit.append("orange")
6. print(fruit)
```

['apple', 'banana', 'cherry']
['apple', 'banana', 'cherry', '**orange**']

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 6 การแทรกค่าเข้าไปในดัชนีลำดับที่ต้องการ
2. fruit = ["apple", "banana", "cherry"]
3. print(fruit)
4. #เพิ่มค่า orange เข้าไปที่ดัชนีลำดับที่ 2
5. fruit.insert(2,"orange")
6. print(fruit)
```

['apple', 'banana', 'cherry’]

['apple', 'banana', '**orange**', 'cherry']

[0] [1] [2] [3]

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 7 การรวมลิสต์ เรียกว่า เอ็กเทน (Extend) แปลว่าขยาย
2. coffee = ["Espresso", "Americano", "Turkish Coffee", "Cold Brew", "Black Coffee"]
3. milk_coffee = ["Latte", "Cappuccino", "Macchiato", "Mocha", "Flat White"]
4. #การขยายลิสต์ coffee ด้วย milk_coffee
5. coffee.extend(milk_coffee)
6. print(coffee)
```

การใช้ extend() เป็นวิธีที่สะดวกในการรวมข้อมูลจากลิสต์หนึ่งไปยังอีกลิสต์หนึ่งใน Python โดยที่ไม่ต้องสร้างลิสต์ใหม่

['Espresso', 'Americano', 'Turkish Coffee', 'Cold Brew', 'Black Coffee’, 'Latte', 'Cappuccino', 'Macchiato', 'Mocha', 'Flat White']

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 8 การลบบางค่าออกจากลิสต์ (Remove)
2. coffee = ["Espresso", "Americano", "Turkish Coffee", "Cold Brew", "Black Coffee"]
3. print(coffee)
4. #การลบค่า Americano ออกจากลสิ ต์รายการในตัวแปร coffee
5. coffee.remove("Americano")
6. print(coffee)
```

```
                remove("Americano")
['Espresso', 'Americano', 'Turkish Coffee', 'Cold Brew', 'Black Coffee’]
['Espresso', 'Turkish Coffee', 'Cold Brew', 'Black Coffee']
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 9 การลบบางค่าออกจากลิสต์ (Remove) ด้วยการระบุดัชนีลำดับ (Pop)
2. coffee = ["Espresso", "Americano", "Turkish Coffee", "Cold Brew", "Black Coffee"]
3. print(coffee)
4. #การลบค่า ในดัชนีลำดับที่ 0 ออกจากลิสต์รายการในตัวแปร coffee
5. coffee.pop(0)
6. print(coffee)
```

```
   pop(0)
['Espresso', 'Americano', 'Turkish Coffee', 'Cold Brew', 'Black Coffee’]
['Americano', 'Turkish Coffee', 'Cold Brew', 'Black Coffee']
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 10 การลบบางค่าออกจากลสิต์ (Remove) ด้วยการระบุดัชนีลำดับ (Pop)
2. coffee = ["Espresso", "Americano", "Turkish Coffee", "Cold Brew", "Black Coffee"]
3. print(coffee)
4. #การลบค่า ในดัชนีลำดับที่ -1 ออกจากลิสต์รายการในตัวแปร coffee
5. coffee.pop(-1)
6. print(coffee)
```

pop(-1)

```
['Espresso', 'Americano', 'Turkish Coffee', 'Cold Brew', 'Black Coffee']
['Espresso', 'Americano', 'Turkish Coffee', 'Cold Brew']
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 11 การลบค่าที่อยู่ปลายสุด ด้วยการระบุดัชนีลำดับ (Pop)
2. coffee = ["Espresso", "Americano", "Turkish Coffee", "Cold Brew", "Black Coffee"]
3. print(coffee)
4. # การลบค่า ในดัชนีลำดับที่ที่อยู่ปลายสุด ออกจากลิสต์รายการในตัวแปร coffee
5. coffee.pop()
6. print(coffee)
```

pop()

```
['Espresso', 'Americano', 'Turkish Coffee', 'Cold Brew', 'Black Coffee']
['Espresso', 'Americano', 'Turkish Coffee', 'Cold Brew']
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 12 การหาจำนวนของค่าที่อยู่ในลิสต์ เรียกว่า ความยาว (lenght: len)
2. coffee = ["Espresso", "Americano", "Turkish Coffee", "Cold Brew", "Black Coffee"]
3. print(coffee)
4. #การหาขนาดใช้ฟังก์ชัน len()
5. coffee_size = len(coffee)
6. print(coffee_size)
```

```
['Espresso', 'Americano', 'Turkish Coffee', 'Cold Brew', 'Black Coffee’]
5
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 13 การเรียงข้อมูลในลิสต์
2. student_height = [163.5, 150.0, 167.0, 161.25, 170.0]
3. print(student_height)
4. # การเรียงจากน้อยไปหามากใช้ฟังก์ชัน sort()
5. student_height.sort()
6. print(student_height)

[163.5, 150.0, 167.0, 161.25, 170.0]
[150.0, 161.25, 163.5, 167.0, 170.0]
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 14 การเรียงข้อมูลในลิสต์ จากมากไปหาน้อย
2. student_height = [163.5, 150.0, 167.0, 161.25, 170.0]
3. print(student_height)
4. # การเรียงจากมากไปหาน้อยใช้ฟังก์ชัน sort(reverse = True)
5. student_height.sort(reverse = True)
6. print(student_height)

[163.5, 150.0, 167.0, 161.25, 170.0]
[170.0, 167.0, 163.5, 161.25, 150.0]
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 15 การกลับตำแหน่งหลังมาหน้า และ หน้าไปหลัง
2. flowers = ["Rose", "Lily", "Lotus", "Orange Blossom", "Sunflower"]
3. print(flowers)
4. # ใช้เมธอด reverse() เพื่อสลับลำดับของลิสต์
5. flowers.reverse()
6. print(flowers)
```

```
['Rose', 'Lily', 'Lotus', 'Orange Blossom', 'Sunflower’]
['Sunflower', 'Orange Blossom', 'Lotus', 'Lily', 'Rose']
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 16 การตรวจสอบข้อมูลใน list
2. flowers = ["Rose", "Lily", "Lotus", "Orange Blossom", "Sunflower"]
3. if "Rose" in flowers:
4.   print("Rose in List")
5. else:
6.   print("Out of Rose")
```

Rose in List

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่างที่ 17 การวนเข้าถึงค่าข้อมูลใน list โดยการใช้ for loops
2. flowers = ["Rose", "Lily", "Lotus", "Orange Blossom", "Sunflower"]
3. for flower in flowers:
4.   print(flower)
```

```
Rose
Lily
Lotus
Orange Blossom
Sunflower
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่าง 18 ใช้ลูป for ร่วมกับฟังก์ชัน enumerate() เพื่อระบุดัชนีของข้อมูล
2. flowerslist = ["Rose", "Lily", "Lotus", "Orange Blossom", "Sunflower"]
3. for index, flower in enumerate(flowerslist):
4.     print(f"Index {index}: {flower}")
```

```
Index 0: Rose
Index 1: Lily
Index 2: Lotus
Index 3: Orange Blossom
Index 4: Sunflower
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบลิสต์ (List)

```
1. #ตัวอย่าง 19 ใช้ลูป for ร่วมกับฟังก์ชัน range() เพื่อเข้าถึงข้อมูลด้วยดัชนีลำดับ
2. flowers = ["Rose", "Lily", "Lotus", "Orange Blossom", "Sunflower"]
3. for i in range(len(flowers)):
4.     print(i , flowers[i])
```

```
0 Rose
1 Lily
2 Lotus
3 Orange Blossom
4 Sunflower
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

โจทย์: จงสร้างรายชื่อนักศึกษาและคะแนนตามตัวอย่างต่อไปนี้ และให้แสดงข้อมูล
ชื่อ: XX คะแนน

```
1. students = ["Alice", "Bob", "Charlie", "David", "Eve"]
2. math_scores = [85, 92, 78, 88, 95]
```

```
Alice: 85 คะแนน
Bob: 92 คะแนน
Charlie: 78 คะแนน
David: 88 คะแนน
Eve: 95 คะแนน
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

เฉลยโจทย์: จงสร้างรายชื่อนักศึกษาและคะแนนตามตัวอย่างต่อไปนี้ และให้แสดงข้อมูล
ชื่อ: XX คะแนน

```
1. students = ["Alice", "Bob", "Charlie", "David", "Eve"]
2. math_scores = [85, 92, 78, 88, 95]
3. for i in range(len(students)):
4.     print(f"{students[i]}: {math_scores[i]} คะแนน")
```

```
Alice: 85 คะแนน
Bob: 92 คะแนน
Charlie: 78 คะแนน
David: 88 คะแนน
Eve: 95 คะแนน
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

โจทย์: ให้แสดงเฉพาะคนที่มีคะแนนมากกว่า 90 คะแนนขึ้นไปเท่านั้น

```
1. students = ["Alice", "Bob", "Charlie", "David", "Eve"]
2. math_scores = [85, 92, 78, 88, 95]
```

```
Bob: 92 คะแนน
Eve: 95 คะแนน
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

เฉลยโจทย์: ให้แสดงเฉพาะคนที่มีคะแนนมากกว่า 90 คะแนนขึ้นไปเท่านั้น

```
1. students = ["Alice", "Bob", "Charlie", "David", "Eve"]
2. math_scores = [85, 92, 78, 88, 95]
3. for i in range(len(students)):
4.    if(math_scores[i]>90):
5.       print(f"{students[i]}: {math_scores[i]} คะแนน")
```

```
Bob: 92 คะแนน
Eve: 95 คะแนน
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

โจทย์: ต้องการหาค่าเฉลี่ยของคะแนนทั้งหมด

```
1. students = ["Alice", "Bob", "Charlie", "David", "Eve"]
2. math_scores = [85, 92, 78, 88, 95]
```

```
ค่าเฉลี่ย = 87.6
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

เฉลยโจทย์: ต้องการหาค่าเฉลี่ยของคะแนนทั้งหมด

```
1. students = ["Alice", "Bob", "Charlie", "David", "Eve"]
2. math_scores = [85, 92, 78, 88, 95]
3. average = sum(math_scores)/len(math_scores)
4. print(f"ค่าเฉลีย( = {average}")
```

```
ค่าเฉลี่ย = 87.6
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

โจทย์: ต้องการรายชื่อและคะแนนของนักเรียนที่มีคะแนนสูงหรือเท่ากับค่าเฉลี่ย

```
1. students = ["Alice", "Bob", "Charlie", "David", "Eve"]
2. math_scores = [85, 92, 78, 88, 95]
3. average = sum(math_scores)/len(math_scores)
4. print(f"ค่าเฉลีย( = {average}คะแนน")
5. print("-"*20)
6. for i in range(len(math_scores)):
7.    if math_scores[i] >= average:
8.       print(f"{students[i]}: {math_scores[i]} คะแนน")
```

```
ค่าเฉลี่ย = 87.6 คะแนน
--------------------
Bob: 92 คะแนน
David: 88 คะแนน
Eve: 95 คะแนน
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบ ดิกชันนารี (Dictionary)

- Dictionary ใน Python เป็นโครงสร้างข้อมูลที่ใช้เก็บข้อมูลในรูปแบบคู่ key-value (คีย์-ค่า) ซึ่งมีลักษณะการจัดเก็บข้อมูลในรูปแบบไม่ต่อเนื่อง และสามารถเข้า ถึงข้อมูลด้วย key ที่กำหนดได้
- Dictionary เป็นสิ่งที่มีความสำคัญอย่างมากในการเขียนโปรแกรม Python เนื่องจากมีความสามารถในการจัดเก็บข้อมูลที่ช่วยในการแก้ปัญหาและการจัดการข้อมูลในแบบต่าง ๆ ได้ดี

```
student = {
     "Name": "Alice",
     "Age": 20,
     "Grade": "A"
}
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบ ดิกชันนารี (Dictionary)

```
1.#การสร้าง Dictionary
2.student = {
3.     "name": "Nutthapat",
4.     "age": 20,
5.     "grade": "A"
6.}
7.print(student)
```

```
{'name': 'Nutthapat', 'age': 20, 'grade': 'A'}
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบ ดิกชันนารี (Dictionary)

```
#การเข้าถึงด้วย key ใน Dictionary
student = {
    "name": "Nutthapat",
    "age": 20,
    "grade": "A"
}
print(student["name"])
```

```
Nutthapat
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบ ดิกชันนารี (Dictionary)

```
1. #การเข้าถึงด้วย key ใน Dictionary
2. student = {
3.     "name": "Nutthapat",
4.     "age": 20,
5.     "grade": "A"
6. }
7. print(student["name"])
8. print(student["age"])
9. print(student["grade"])
```

```
Nutthapat
20
A
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบ ดิกชันนารี (Dictionary)

```
1. #การลบข้อมูลด้วย key ใน Dictionary
2. student = {
3.     "name": "Nutthapat",
4.     "age": 20,
5.     "grade": "A"
6. }
7. print(student)
8. del student["grade"]
9. print(student)
```

```
{'name': 'Nutthapat', 'age': 20, 'grade': 'A'}
{'name': 'Nutthapat', 'age': 20}
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบ ดิกชันนารี (Dictionary)

```
1. #การวนลูกเข้าถึง key และ value ใน Dictionary
2. student = {
3.     "name": "Nutthapat",
4.     "age": 20,
5.     "grade": "A"
6. }

7. for key, value in student.items():
8.   print(f"{key}: {value}")
```

```
name: Nutthapat
age: 20
grade: A
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบ ดิกชันนารี (Dictionary)

```
#การตรวจสอบการมีอยู่ของ key ใน Dictionary
student = {
    "name": "Nutthapat",
    "age": 20,
    "grade": "A"
}

if "grade" in student:
  print("มีข้อมูลเกี่ยวกับ grade ใน Dictionary")
else:
  print("ไม่มีข้อมูลเกี่ยวกับ grade ใน Dictionary")
```

```
มีข้อมูลเกี่ยวกับ grade ใน Dictionary
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบ ดิกชันนารี (Dictionary)

```
1. #การตรวจสอบการมีอยู่ของ key ใน Dictionary
2. student = {
3.     "name": "Nutthapat",
4.     "age": 20,
5.     "grade": "A"
6. }

7. print(len(student))
```

```
3
```

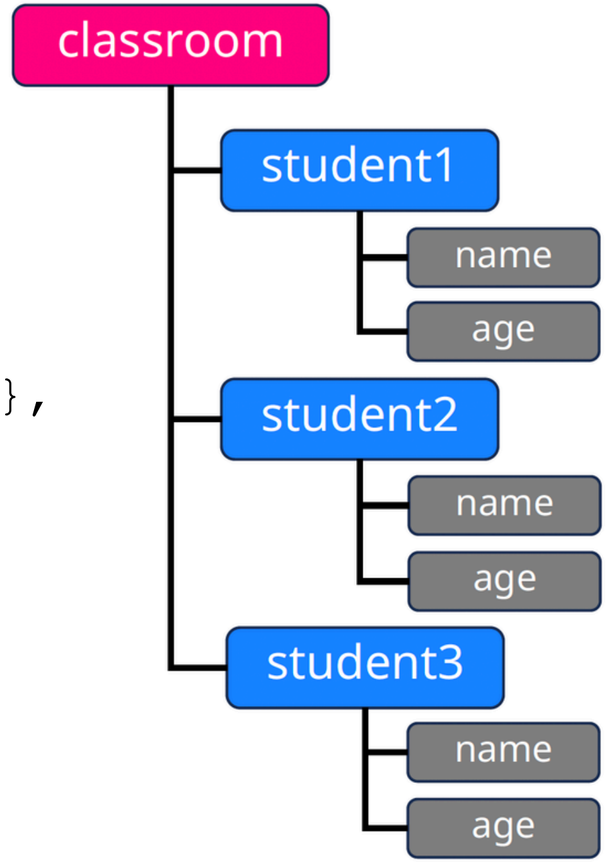
ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบ ดิกชันนารี (Dictionary)

```
1. #การสร้างดิกชันารีแบบซ้อน (Nested Dictionary)
2. classroom = {
3.         "student1": {"name": "Nutthapat", "age": 20},
4.         "student2": {"name": "Mongkolchai", "age": 22},
5.         "student3": {"name": "Wichuda", "age": 21}
6. }
7. stu_name = classroom["student3"]["name"]
8. stu_age = classroom["student3"]["age"]
9. print(f"{stu_name}: {stu_age}")
```

```
Wichuda: 21
```

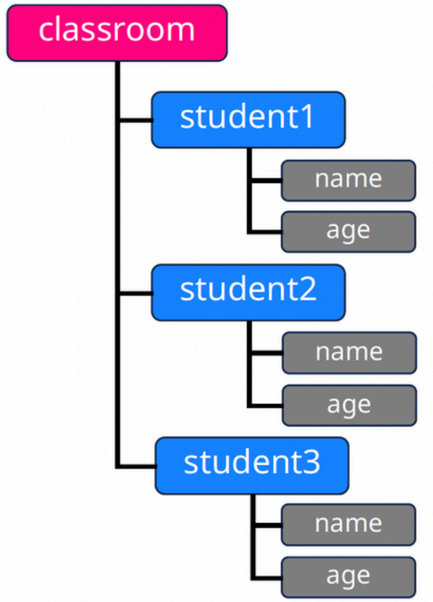


ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบ ดิกชันนารี (Dictionary)

```
#การสร้างดิกชันนารีแบบซ้อน (Nested Dictionary) และวนลูปด้วย for เพื่อเข้าถึงข้อมูล
classroom = {
    "student1": {"name": "Nutthapat", "age": 20},
    "student2": {"name": "Mongkolchai", "age": 22},
    "student3": {"name": "Wichuda", "age": 21}
}

for student_id, student_info in classroom.items():
    print(f"Student ID: {student_id}")
    print(f"Name: {student_info['name']}")
    print(f"Age: {student_info['age']}")
    print()
```

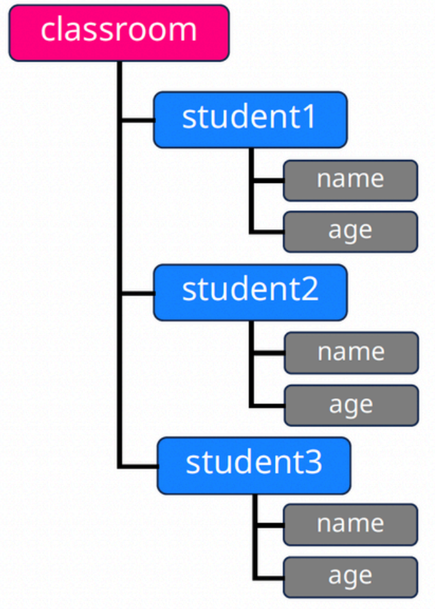



```
Student ID: student1
Name: Nutthapat
Age: 20

Student ID: student2
Name: Mongkolchai
Age: 22

Student ID: student3
Name: Wichuda
Age: 21
```

ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types)

2. ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

ประเภทข้อมูลแบบ ดิกชันนารี (Dictionary)




```
1. # สร้างตัวแปรเพื่อเก็บผลรวมอายุของนักเรียน
2. total_age = 0
3. # วนลูปผ่านดิกชันนารี classroom
4. for student_key, student_info in classroom.items():
5.     # แสดงผลข้อมูลของนักเรียน
6.     print(f"นักเรียน {student_key}: ชื่อ {student_info['name']}, อายุ {student_info['age']} ปี")
7.     # เพิ่มอายุของนักเรียนในผลรวม
8.     total_age += student_info['age']
9. # หาค่าอายุเฉลี่ย
10. average_age = total_age / len(classroom)
11. print(f"ค่าอายุเฉลี่ยของนักเรียนทั้งหมดคือ {average_age} ปี")
```

```
นักเรียน student1: ชื่อ Nutthapat, อายุ 20 ปี
นักเรียน student2: ชื่อ Mongkolchai, อายุ 22 ปี
นักเรียน student3: ชื่อ Wichuda, อายุ 21 ปี
ค่าอายุเฉลี่ยของนักเรียนทั้งหมดคือ 21.0 ปี
```

Assignment 5 - มอบหมายงานครั้งที่ 5

1. จากข้อมูลพนักงานเหล่านี้ที่อยู่ใน dictionary data types ให้เขียนโปรแกรมเพื่อแสดงเฉพาะ ชื่อ และแผนกออกมา เฉพาะ 5 คนแรกเท่านั้น

```
employees = {
1: {"Name": "John", "TelephoneNumber": "123-456-7890", "Department": "HR", "Salary": 50000},
2: {"Name": "Alice", "TelephoneNumber": "234-567-8901", "Department": "IT", "Salary": 60000},
3: {"Name": "Bob", "TelephoneNumber": "345-678-9012", "Department": "Sales", "Salary": 55000},
4: {"Name": "Sarah", "TelephoneNumber": "456-789-0123", "Department": "Marketing", "Salary": 52000},
5: {"Name": "Mike", "TelephoneNumber": "567-890-1234", "Department": "Finance", "Salary": 58000},
6: {"Name": "Emily", "TelephoneNumber": "678-901-2345", "Department": "IT", "Salary": 62000},
7: {"Name": "David", "TelephoneNumber": "789-012-3456", "Department": "Sales", "Salary": 53000},
8: {"Name": "Linda", "TelephoneNumber": "890-123-4567", "Department": "Marketing", "Salary": 51000},
9: {"Name": "Chris", "TelephoneNumber": "901-234-5678", "Department": "Finance", "Salary": 59000},
10: {"Name": "Julia", "TelephoneNumber": "012-345-6789", "Department": "HR", "Salary": 51000}
}
```

Assignment 5 - มอบหมายงานครั้งที่ 5

2. จากข้อมูลพนักงานเหล่านี้ที่อยู่ใน dictionary data types ให้เขียนโปรแกรมเพื่อแสดงรายชื่อพนักงาน ที่ทำงานแผนก “IT” ออกมาเท่านั้น

```
employees = {
1: {"Name": "John", "TelephoneNumber": "123-456-7890", "Department": "HR", "Salary": 50000},
2: {"Name": "Alice", "TelephoneNumber": "234-567-8901", "Department": "IT", "Salary": 60000},
3: {"Name": "Bob", "TelephoneNumber": "345-678-9012", "Department": "Sales", "Salary": 55000},
4: {"Name": "Sarah", "TelephoneNumber": "456-789-0123", "Department": "Marketing", "Salary": 52000},
5: {"Name": "Mike", "TelephoneNumber": "567-890-1234", "Department": "Finance", "Salary": 58000},
6: {"Name": "Emily", "TelephoneNumber": "678-901-2345", "Department": "IT", "Salary": 62000},
7: {"Name": "David", "TelephoneNumber": "789-012-3456", "Department": "Sales", "Salary": 53000},
8: {"Name": "Linda", "TelephoneNumber": "890-123-4567", "Department": "Marketing", "Salary": 51000},
9: {"Name": "Chris", "TelephoneNumber": "901-234-5678", "Department": "Finance", "Salary": 59000},
10: {"Name": "Julia", "TelephoneNumber": "012-345-6789", "Department": "HR", "Salary": 51000}
}
```

Assignment 5 - มอบหมายงานครั้งที่ 5

3. จากข้อมูลพนักงานเหล่านี้ที่อยู่ใน dictionary data types ให้เขียนโปรแกรมเพื่อแสดงเงินเดือนสูงที่สุด เงินเดือนต่ำที่สุด และเงินเดือนเฉลี่ย ของพนักงานทั้งหมด

```
employees = {
1: {"Name": "John", "TelephoneNumber": "123-456-7890", "Department": "HR", "Salary": 50000},
2: {"Name": "Alice", "TelephoneNumber": "234-567-8901", "Department": "IT", "Salary": 60000},
3: {"Name": "Bob", "TelephoneNumber": "345-678-9012", "Department": "Sales", "Salary": 55000},
4: {"Name": "Sarah", "TelephoneNumber": "456-789-0123", "Department": "Marketing", "Salary": 52000},
5: {"Name": "Mike", "TelephoneNumber": "567-890-1234", "Department": "Finance", "Salary": 58000},
6: {"Name": "Emily", "TelephoneNumber": "678-901-2345", "Department": "IT", "Salary": 62000},
7: {"Name": "David", "TelephoneNumber": "789-012-3456", "Department": "Sales", "Salary": 53000},
8: {"Name": "Linda", "TelephoneNumber": "890-123-4567", "Department": "Marketing", "Salary": 51000},
9: {"Name": "Chris", "TelephoneNumber": "901-234-5678", "Department": "Finance", "Salary": 59000},
10: {"Name": "Julia", "TelephoneNumber": "012-345-6789", "Department": "HR", "Salary": 51000}
}
```

Assignment 5 - มอบหมายงานครั้งที่ 5

4. ให้นักศึกษาออกแบบโจทย์เกี่ยวกับ list จำนวน 1 ข้อ จากนั้นเขียนอธิบายเกี่ยวกับวิธีแก้ปัญหาของโจทย์นี้ ให้กับนักเรียนระดับชั้น ม.1

Post Test

Question:

1. ใน Python “ลำดับ” (Sequences) คืออะไร

A โครงสร้างข้อมูลที่เก็บข้อมูลเป็นลำดับเรียงต่อกัน

B โครงสร้างของโค้ดที่เขียนต่อเป็นลำดับเรียงต่อกัน

C โครงสร้างโอเปอร์เรเตอร์ที่เก็บข้อมูลเป็นลำดับเรียงต่อกัน

D ไม่มีข้อใดถูก

Question:

2. ข้อใด <u>ไม่ใช่</u> ประโยชน์ของประเภทข้อมูลแบบลำดับ

A ช่วยในการจัดเก็บข้อมูลที่มีลำดับ

B ช่วยในการเข้าถึงข้อมูลโดยอ้างอิงลำดับหรือดัชนี

C ช่วยในการเก็บข้อมูลที่มีรูปแบบแบบซับซ้อน

D ช่วยในการคำนวณตัวเลขที่เกี่ยวข้องได้ง่าย

Question:

3. ข้อใด ไม่ใช่ ลำดับที่ไม่สามารถเปลี่ยนแปลงได้ (Immutable Sequences)

A ประเภทจำนวนเต็ม (Integers)

B ประเภทบูลีน (Boolean)

C ประเภทสายอักขระ (String)

D ประเภทรายการ (List)

Question:

4. ข้อใด ไม่ใช่ ลำดับที่สามารถเปลี่ยนแปลงได้ (Mutable Sequences)

A ประเภทรายการ (List)

B ประเภทจำนวนทศนิยม (Floating-point numbers)

C ประเภทเซ็ต (Set)

D ประเภทพจนานุกรม (Dictionary)

Question:

5. ข้อใด <u>ไม่ใช่</u> ลักษณะสำคัญของประเภทข้อมูลแบบทูเพิล (Tuple)

A ไม่สามารถแก้ไขหรือเปลี่ยนแปลงค่าของสมาชิกภายใน tuple หลังจากที่สร้างขึ้น

B สามารถแก้ไขหรือเปลี่ยนแปลงค่าของสมาชิกภายใน tuple หลังจากที่สร้างขึ้นได้

C Tuples สามารถมีสมาชิกหลายประเภทข้อมูลและสามารถมีขนาดต่าง ๆ ได้

D Tuple สร้างโดยใช้วงเล็บ (), และสามารถมีหรือไม่มีวงเล็บรอบสมาชิก

Question:

6. จากคำสั่ง แสดงผลข้อใด

```
countries = ("ไทย", "สหรัฐอเมริกา", "อังกฤษ", "ญี่ปุ่น", "จีน", "อินเดีย", "เบลเยียม", "เกาหลีใต้")
if "ญี่ปุ่น" in countries:
  print("เป็นสมาชิก")
else:
  print("ไม่เป็นสมาชิก")
```

A เป็นสมาชิก

B ไม่เป็นสมาชิก

C เป็นสมาชิก และไม่เป็นสมาชิก

D ไม่เป็นสมาชิก และเป็นสมาชิก

Question:

7. จากคำสั่ง แสดงผลข้อใด

```
dog_breeds1 = ("บีเกิ้ล", "โกลเด้นรีทรีฟเวอร์", "ชิวาว่า")
dog_breeds2 = ("พุดเดิ้ล", "บูลด็อก", "โรตวายเลีย")
all_dog_breeds = dog_breeds1 + dog_breeds2
print(all_dog_breeds.count("ชิวาว่า"))
```

A 0

C 2

B 1

D 3

Question:

8. จากคำสั่ง แสดงผลข้อใด

```
student_height = [163.5, 150.0, 167.0, 161.25, 170.0]
maximum_height = max(student_height)
print("ส่วนสูงสูงสุดของนักเรียนคือ:", maximum_height)
```

A ส่วนสูงสูงสุดของนักเรียนคือ: 163.5

B ส่วนสูงสูงสุดของนักเรียนคือ: 167.0

C ส่วนสูงสูงสุดของนักเรียนคือ: 170.0

D ส่วนสูงสูงสุดของนักเรียนคือ: 150.0

Question:

9. จากคำสั่ง แสดงผลข้อใด

```
students = ["Alice", "Bob", "Charlie", "David", "Eve"]
math_scores = [85, 92, 78, 88, 95]
average = sum(math_scores)/len(math_scores)
print(f"ค่าเฉลีย( = {average}")
```

A ค่าเฉลี่ย: 59.3

C ค่าเฉลี่ย: 87.6

B ค่าเฉลี่ย: 43.5

D ค่าเฉลี่ย: 100.4

Question:

10. จากคำสั่ง แสดงผลข้อใด

```
student = {
   "name": "Nutthapat",
   "age": 20,
   "grade": "A"
}
print(len(student))
```

A 1

B 2

C 3

D 4

Computer Programming and
Developing Applications for Education

Thank You

Pasawut Cheerapakorn
Suan Sunandha Rajabhat University

DTI3302 Computer Programming and Developing Applications for Education
Department of Digital Technology for Education
Faculty of Education, Suan Sunandha Rajabhat University