

บทที่ 7

ฟังก์ชัน

Chapter 7 Functions



Asst.Prof.Dr.Nutthapat Kaewrattanapat
Suan Sunandha Rajabhat University

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

อาจารย์บรรยาย



ผู้ช่วยศาสตราจารย์ ดร.ณัฐภัทร แก้วรัตนภัทร

การศึกษา: มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

- 2565 ปริญญาเอก ปรัชญาดุสิตบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศและการสื่อสารเพื่อการศึกษา (GPA. 4.00)
- 2551 ปริญญาโท วิทยาศาสตรมหาบัณฑิต สาขาวิชาระบบสารสนเทศเพื่อการจัดการ (GPA. 3.58)
- 2549 ปริญญาตรี วิทยาศาสตรบัณฑิต เกียรตินิยมอันดับ 1 สาขาวิชาวิทยาการคอมพิวเตอร์ (GPA. 3.66)

คุณวุฒิวิชาชีพและประกาศนียบัตร

- คุณวุฒิวิชาชีพ จาก สถาบันคุณวุฒิวิชาชีพ (องค์การมหาชน) สาขาวิชาชีพอุตสาหกรรมดิจิทัล สาขาธุรกิจดิจิทัลและพาณิชย์อิเล็กทรอนิกส์ อาชีพนักจัดการเทคโนโลยีสารสนเทศสำหรับธุรกิจ ระดับ 6 เลขที่หนังสือรับรอง PQCN-ICT-ECM-0-251100-B-64/000029
- วิทยาศาสตรข้อมูลด้วยภาษาไพทอน, มหาวิทยาลัยธรรมศาสตร์
- การโปรแกรมสำหรับนักภาษาศาสตร์, มหาวิทยาลัยมหิดล
- การโปรแกรมภาษาไพทอน, มหาวิทยาลัยเพนซิลวาเนีย, สหรัฐอเมริกา
- การโปรแกรมสำหรับทุกคน, มหาวิทยาลัยมิชิแกน, สหรัฐอเมริกา

ติดต่อ: nutthapat.ke@ssru.ac.th

Course Description

หลักการ ทฤษฎี ที่เกี่ยวข้องกับการเขียนโปรแกรมและพัฒนาแอปพลิเคชัน หลักการ พัฒนาโปรแกรมคอมพิวเตอร์ คุณสมบัติของโปรแกรมภาษาชนิดต่าง ๆ หลักการ เบื้องต้นเกี่ยวกับองค์ประกอบ ลักษณะคำสั่ง การเขียนโปรแกรม ขั้นตอนวิธี การ วิเคราะห์ การออกแบบ แอปพลิเคชันเพื่อการศึกษา การประเมินซอฟต์แวร์ 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 Language

System Development Life Cycle: SDLC

Reference: <https://edu.ssru.ac.th/useruploads/files/20230724/1772131ed638786bc8d19918b37249af72c36be4.pdf>

Course Outline

- Chapter 1 – Computer Programming
- Chapter 2 – Introduction to Python Programming
- Chapter 3 – Conditional Statement
- Chapter 4 – Iteration Statement
- Chapter 5 – Strings
- Chapter 6 – Immutable and Mutable Data Types
- Chapter 7 – Functions
- Chapter 8 – Object-Oriented Programming: OOP
- Chapter 9 – Python and Database Management System
- Chapter 10 – System Development Life Cycle: SDLC

Measurement and Evaluation

การวัดและประเมินผล

1. ระหว่างการจัดการเรียนรู้

- สอบ Pre-test
- การมอบหมายงาน
- สอบ Post-test
- การมีส่วนร่วมในชั้นเรียน

0%
24%
12%
4%

2. การสอบกลางภาค (Midterm Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน)

20%

3. โครงการประจำภาคเรียน (Term Project)

- โครงการและการนำเสนอ

20%

4. การสอบปลายภาค (Final Examination)

- ปรนัย 35 ข้อ (35 คะแนน) อัตนัย 1 ข้อ (5 คะแนน)

20%

ร้อยละ	ระดับผลการเรียน	ความหมาย
86 – 100	A	ดีเยี่ยม
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/วันที่	บทเรียน/หัวข้อ	มอบหมายงาน (24%)	สอบ Post-test (12%)	การมีส่วนร่วมในชั้นเรียน (4%)
1 พท 7 ธค 66	แนะนำรายวิชา การวัดและการประเมินผล หัวข้อเรียนรู้ บทที่ 1 การโปรแกรมคอมพิวเตอร์	-	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
2 พท 14 ธค 66	บทที่ 2 พื้นฐานการโปรแกรมภาษาไพทอน	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
3 พท 21 ธค 66	บทที่ 3 การโปรแกรมแบบเงื่อนไข (Conditional Statement)	-	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
4 พท 28 ธค 66 Online	บทที่ 3 การโปรแกรมแบบเงื่อนไข (Conditional Statement) (ต่อ)	3 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
5 พท 4 มค 67	บทที่ 4 การโปรแกรมแบบทำซ้ำ (Iteration Statement)	3 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
6 พท 11 มค 67	บทที่ 5 การโปรแกรมสายอักขระ (String)	-	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
7 จ 22 มค 67	สอบกลางภาค ปรนัย 35 ข้อ (35 คะแนน) และอัตนัย 1 ข้อ (5 คะแนน) 20%			
8 พท 1 กพ 67	บทที่ 6 ประเภทข้อมูลแบบลำดับ (Immutable and Mutable Data Types) : Tuple, List และ Dictionary	6 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%

Measurement and Evaluation

ครั้งที่/วันที่	บทเรียน/หัวข้อ	มอบหมายงาน (24%)	สอบ Post-test (12%)	การมีส่วนร่วมในชั้นเรียน (4%)
9 พฤษ 8 กพ 67	บทที่ 7 ฟังก์ชัน	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
10 พฤษ 15 กพ 67	บทที่ 8 การโปรแกรมเชิงวัตถุ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
12 พฤษ 22 กพ 67	บทที่ 9 ภาษาไพทอนกับระบบการจัดการฐานข้อมูล	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
13 พฤษ 29 กพ 67	บทที่ 9 ภาษาไพทอนกับระบบการจัดการฐานข้อมูล ต่อ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
14 พฤษ 7 มีค 67	บทที่ 10 วงจรการพัฒนาระบบ	2 คะแนน	1 คะแนน	ขาด/ไม่ทันเช็คชื่อ -1%
15 พฤษ 14 มีค 67	สอบปลายภาค ปรนัย 35 ข้อ (35 คะแนน) และอัตนัย 1 ข้อ (5 คะแนน) 20%			
16 พฤษ 21 มีค 67	ส่งโครงงาน และนำเสนอ 20%			

ฟังก์ชัน (Functions)

ฟังก์ชัน (Functions) ในภาษา Python คือ ชุดของคำสั่งที่ถูกจัดกลุ่มเพื่อทำงานบางอย่าง เช่น ฟังก์ชันหาค่าเฉลี่ย ฟังก์ชันหาพื้นที่สี่เหลี่ยม

เมื่อเรียกใช้งานฟังก์ชัน โปรแกรมจะทำงานตามคำสั่งภายในฟังก์ชันนั้น ๆ และสามารถรับค่าอินพุต (input) และส่งค่าเอาต์พุต (output) กลับไปยังโปรแกรมหลักหรือส่วนอื่น ๆ ของโค้ดได้ ซึ่งจะเรียนว่า ค่าพารามิเตอร์ (Parameters) และ อาร์กิวเมนต์ (Arguments)

ประโยชน์ของฟังก์ชัน

1. การใช้ซ้ำของโค้ด (Code Reusability)
2. การจัดการโค้ดและอ่านโค้ดได้ดีขึ้น
3. การทดสอบและการดูแลรักษา (Testing and Maintenance)
4. การจัดการทรัพยากร (Resource Management)
5. การออกแบบ (Design)
6. การปรับแต่งและการเปลี่ยนแปลง (Customization and Modification)

การเปรียบเทียบการเขียนโปรแกรมแบบ Procedural และ Function

การเขียนโปรแกรมเชิงกระบวนการ (Procedural Programming)

```
1. length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
2. width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
3. area = length * width
4. print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")

5. length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
6. width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
7. area = length * width
8. print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")

9. length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
10. width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
11. area = length * width
12. print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")
```

การเขียนโปรแกรมเชิงฟังก์ชัน (Functional Programming)

```
1. def rectangle()
2.     length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
3.     width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
4.     area = length * width
5.     print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")

6. rectangle()

7. rectangle()

8. rectangle()
```

การเปรียบเทียบการเขียนโปรแกรมแบบ Procedural และ Function

การเขียนโปรแกรมเชิงกระบวนการ (Procedural Programming)

```
1. length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
2. width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
3. area = length * width
4. print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")

5. length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
6. width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
7. area = length * width
8. print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")

9. length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
10. width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
11. area = length * width
12. print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")
```

การเขียนโปรแกรมเชิงฟังก์ชัน (Functional Programming)

Function

```
1. def rectangle()
2.     length = float(input("กรุณาใส่ความยาวของสี่เหลี่ยม: "))
3.     width = float(input("กรุณาใส่ความกว้างของสี่เหลี่ยม: "))
4.     area = length * width
5.     print(f"พื้นที่ของสี่เหลี่ยมคือ: {area}")
```

Calling a function

```
6. rectangle()

7. rectangle()

8. rectangle()
```

ประเภทของ Function

1. ฟังก์ชันพื้นฐานทั่วไป (Built-in Function)
เช่น `print( )` `len( )` `input( )`
2. ฟังก์ชันที่นิยามใหม่ (User-defined Functions)
 1. ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)
 2. ฟังก์ชันแบบมีค่าพารามิเตอร์ (Function)
 3. ฟังก์ชันแบบคืนค่า (Return Function)

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)

สถานการณ์: ต้องการฟังก์ชันสำหรับแสดงเมนูการทำงาน และรับข้อมูลเมนู ดังนี้

เมนูฟังก์ชันสำหรับการหาพื้นที่

กด 1 หาพื้นที่สี่เหลี่ยม (Rectangle Area)

กด 2 หาพื้นที่สามเหลี่ยม (Triangle Area)

กด 3 หาพื้นที่วงกลม (Circle Area)

```
def show_menu():  
    print("เมนูการทำงาน:")  
    print("กด 1: หาพื้นที่สี่เหลี่ยม (Rectangle Area)")  
    print("กด 2: หาพื้นที่สามเหลี่ยม (Triangle Area)")  
    print("กด 3: หาพื้นที่วงกลม (Circle Area)")  
    choice = input("โปรดเลือกหมายเลขเมนู: ")  
    if choice == '1':  
        print("คุณเลือกหาพื้นที่สี่เหลี่ยม")  
    elif choice == '2':  
        print("คุณเลือกหาพื้นที่สามเหลี่ยม")  
    elif choice == '3':  
        print("คุณเลือกหาพื้นที่วงกลม")  
    else:  
        print("โปรดเลือกหมายเลขเมนูที่ถูกต้อง")
```

```
show_menu()
```

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)

สถานการณ์: ต้องการฟังก์ชันหาพื้นที่สี่เหลี่ยม (Rectangle Area)

```
def rectangle_area():  
    length = float(input("ป้อนความยาวของสี่เหลี่ยม: "))  
    width = float(input("ป้อนความกว้างของสี่เหลี่ยม: "))  
    area = length * width  
    print("พื้นที่ของสี่เหลี่ยมคือ:", area)
```

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)

สถานการณ์: ต้องการฟังก์ชันหาพื้นที่สามเหลี่ยม (Triangle Area)

```
def triangle_area():  
    base = float(input("ป้อนความยาวฐานของสามเหลี่ยม: "))  
    height = float(input("ป้อนความสูงของสามเหลี่ยม: "))  
    area = 0.5 * base * height  
    print("พื้นที่ของสามเหลี่ยมคือ:", area)
```

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)

สถานการณ์: ต้องการฟังก์ชันหาพื้นที่วงกลม (Circle Area)

```
import math

def circle_area():
    radius = float(input("ป้อนรัศมีของวงกลม: "))
    area = math.pi * radius**2
    print("พื้นที่ของวงกลมคือ:", area)
```

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)

```
import math
def show_menu():
    print("เมนูการทำงาน:")
    print("กด 1: หาพื้นที่สี่เหลี่ยม (Rectangle Area)")
    print("กด 2: หาพื้นที่สามเหลี่ยม (Triangle Area)")
    print("กด 3: หาพื้นที่วงกลม (Circle Area)")
    choice = input("โปรดเลือกหมายเลขเมนู: ")
    if choice == '1':
        print("คุณเลือกหาพื้นที่สี่เหลี่ยม")
    elif choice == '2':
        print("คุณเลือกหาพื้นที่สามเหลี่ยม")
    elif choice == '3':
        print("คุณเลือกหาพื้นที่วงกลม")
    else:
        print("โปรดเลือกหมายเลขเมนูที่ถูกต้อง")
```

```
def rectangle_area():
    length = float(input("ป้อนความยาวของสี่เหลี่ยม: "))
    width = float(input("ป้อนความกว้างของสี่เหลี่ยม: "))
    area = length * width
    print("พื้นที่ของสี่เหลี่ยมคือ:", area)

def triangle_area():
    base = float(input("ป้อนความยาวฐานของสามเหลี่ยม: "))
    height = float(input("ป้อนความสูงของสามเหลี่ยม: "))
    area = 0.5 * base * height
    print("พื้นที่ของสามเหลี่ยมคือ:", area)

def circle_area():
    radius = float(input("ป้อนรัศมีของวงกลม: "))
    area = math.pi * radius**2
    print("พื้นที่ของวงกลมคือ:", area)
```

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)

```
import math
def show_menu():
    print("เมนูการทำงาน:")
    print("กด 1: หาพื้นที่สี่เหลี่ยม (Rectangle Area)")
    print("กด 2: หาพื้นที่สามเหลี่ยม (Triangle Area)")
    print("กด 3: หาพื้นที่วงกลม (Circle Area)")
    choice = input("โปรดเลือกหมายเลขเมนู: ")
    if choice == '1':
        rectangle_area()
    elif choice == '2':
        triangle_area()
    elif choice == '3':
        circle_area()
    else:
        print("โปรดเลือกหมายเลขเมนูที่ถูกต้อง")
```

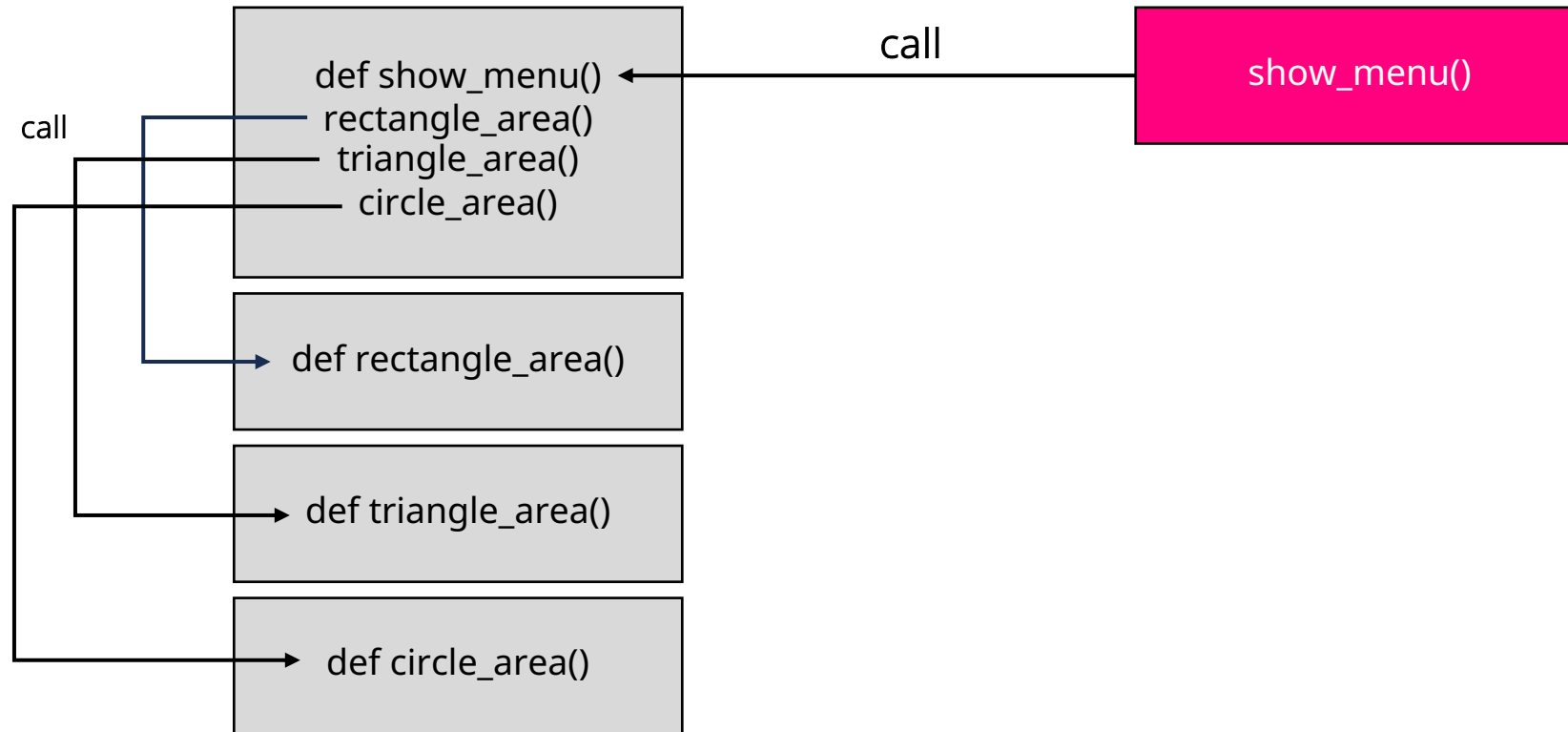
```
def rectangle_area():
    length = float(input("ป้อนความยาวของสี่เหลี่ยม: "))
    width = float(input("ป้อนความกว้างของสี่เหลี่ยม: "))
    area = length * width
    print("พื้นที่ของสี่เหลี่ยมคือ:", area)

def triangle_area():
    base = float(input("ป้อนความยาวฐานของสามเหลี่ยม: "))
    height = float(input("ป้อนความสูงของสามเหลี่ยม: "))
    area = 0.5 * base * height
    print("พื้นที่ของสามเหลี่ยมคือ:", area)

def circle_area():
    radius = float(input("ป้อนรัศมีของวงกลม: "))
    area = math.pi * radius**2
    print("พื้นที่ของวงกลมคือ:", area)
```

show_menu()

ฟังก์ชันแบบไม่มีค่าพารามิเตอร์ (Void Function)



ฟังก์ชันแบบมีค่าพารามิเตอร์ (Function)

Parameter

```
1. def greeting(name) :  
2.     print(f"Hello {name}")  
3.  
4.  
5. greeting ("Nutthapat")
```

Argument

```
1 def greeting(name):  
2     print(f"Hello {name}")  
3  
4 greeting("Nutthapat")
```

Hello Nutthapat

```
1 def greeting(name):  
2     print(f"Hello {name}")  
3  
4 greeting("Nutthapat")  
5 greeting("Mongkolchai")
```

Hello Nutthapat
Hello Mongkolchai

ฟังก์ชันแบบมีค่าพารามิเตอร์ (Function)

สถานการณ์: ต้องการสร้าง function สำหรับทักทาย โดยมีการรับค่า parameter เพศ (gender) และชื่อ (name)

ถ้า gender เป็น male แสดง "Hello, Mr.Name"

ถ้า gender เป็น female แสดง "Hello, Ms.Name"

ฟังก์ชันแบบมีค่าพารามิเตอร์ (Function)

สถานการณ์: ต้องการสร้าง function สำหรับแปลงอุณหภูมิโดยมีการส่งค่า Arguments และแสดงผลการแปลงอุณหภูมิ

- show_menu
โปรดเลือกการแปลง:
 1. Fahrenheit เป็น Celsius
 2. Celsius เป็น Fahrenheit
- fahrenheit_to_celsius สูตร $celsius = (fahrenheit - 32) * 5/9$
- celsius_to_Fahrenheit สูตร $fahrenheit = (celsius * 9/5) + 32$

ฟังก์ชันแบบมีค่าพารามิเตอร์ (Function)

สถานการณ์: ต้องการสร้าง function สำหรับแปลงอุณหภูมิโดยมีการส่งค่า Arguments และแสดงผลการแปลงอุณหภูมิ

- show_menu

โปรดเลือกการแปลง:

1. Fahrenheit เป็น Celsius
2. Celsius เป็น Fahrenheit

โปรดเลือกการแปลง:

1. Fahrenheit เป็น Celsius
2. Celsius เป็น Fahrenheit

1

ป้อนอุณหภูมิในฟาเรนไฮต์: 44

44.0 องศาฟาเรนไฮต์ เท่ากับ 6.67 องศาเซลเซียส

- fahrenheit_to_celsius สูตร $celsius = (fahrenheit - 32) * 5/9$
- celsius_to_Fahrenheit สูตร $fahrenheit = (celsius * 9/5) + 32$

ฟังก์ชันแบบคืนค่า (Return Function)

```
1. def add(num1, num2) :  
2.     result = num1 + num2  
3.     return result  
4.  
5. print(add(5, 6))
```

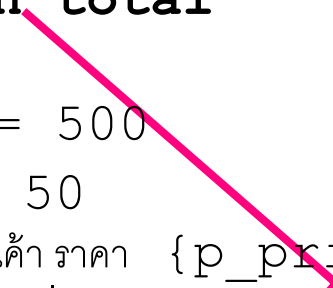
Parameter

Return

Argument

ฟังก์ชันแบบคืนค่า (Return Function)

```
1. def product(price, sale):  
2.     total = price - (price*sale/100)  
3.     return total  
4.  
5. p_price = 500  
6. p_sale = 50  
7. print("สินค้าราคา {p_price} ส่วนลด {p_sale}")  
8. print("ราคาที่ต้องชำระ คือ {product(p_price, p_sale)} บาท")
```



ฟังก์ชันแบบคืนค่า (Return Function)

```
1. def fahrenheit_to_celsius(fahrenheit):
2.     celsius = (fahrenheit - 32) * 5/9
3.     return celsius
4.
5. def celsius_to_fahrenheit(celsius):
6.     fahrenheit = (celsius * 9/5) + 32
7.     return fahrenheit
8.
9. def show_menu():
10.    choice = input("โปรดเลือกการแปลง:\n1. Fahrenheit เป็น Celsius\n2. Celsius เป็น Fahrenheit\n")
11.    if choice == '1':
12.        fahrenheit = float(input("ป้อนอุณหภูมิในฟาเรนไฮต์: "))
13.        celsius = fahrenheit_to_celsius(fahrenheit)
14.        print(f"{fahrenheit} องศาฟาเรนไฮต์ เท่ากับ {celsius:.2f} องศาเซลเซียส")
15.    elif choice == '2':
16.        celsius = float(input("ป้อนอุณหภูมิในเซลเซียส: "))
17.        fahrenheit = celsius_to_fahrenheit(celsius)
18.        print(f"{celsius} องศาเซลเซียส เท่ากับ {fahrenheit:.2f} องศาฟาเรนไฮต์")
19.    else:
20.        print("โปรดเลือกตัวเลือกที่ถูกต้อง")
21.
22. show_menu()
```

ฟังก์ชันแบบมีตัวแปรอาร์กิวเมนต์ที่ไม่แน่นอน Arbitrary Arguments

```
1. def my_function(*args):
2.     for text in args:
3.         print("Country:", text)
4.
5. my_function("Thailand", "Japan", "China")
6. print("*"*50)
7. my_function("Thailand", "Japan", "China", "South Korea", "USA")
```

```
1 def my_function(*args):
2     for arg in args:
3         print("Country:",arg)
4
5 my_function("Thailand", "Japan", "China")
6 print("*"*50)
7 my_function("Thailand", "Japan", "China", "South Korea", "USA")
```

```
Country: Thailand
Country: Japan
Country: China
*****
Country: Thailand
Country: Japan
Country: China
Country: South Korea
Country: USA
```

ฟังก์ชันแบบระบุค่าเฉพาะของอาร์กิวเมนต์ Keyword Arguments

สถานการณ์: ต้องการสร้าง function สำหรับทักทาย โดยมีการรับค่า parameter เพศ (gender) และชื่อ (name)

ถ้า gender เป็น male แสดง "Hello, Mr.Name"

ถ้า gender เป็น female แสดง "Hello, Ms.Name"

ฟังก์ชันแบบระบุค่าเฉพาะของอาร์กิวเมนต์ Keyword Arguments

```
1. def greet(name, gender):  
2.     if gender == "male":  
3.         print(f"Hello, Mr.{name}")  
4.     elif gender == "female":  
5.         print(f"Hello, Ms.{name}")  
6.  
7. greet(gender="male", name="Nutthapat")  
8. greet(name="Wichuda", gender="female")
```

ฟังก์ชันแบบระบุค่าเฉพาะของอาร์กิวเมนต์และมีค่าอาร์กิวเมนต์ไม่แน่นอน Arbitrary Keyword Arguments, **kwargs

```
1. def my_function(**kwargs):
2.     for key, value in kwargs.items():
3.         print(f"{key}: {value}")
4.     print("-"*50)

5. my_function(city="Thailand", region="Asia")
6. my_function(city="Japan", region="Asia", latitude="36°00' N", longitude="138°00' E")
```

```
1 def my_function(**kwargs):
2     for key, value in kwargs.items():
3         print(f"{key}: {value}")
4     print("-"*50)
5
6 my_function(city="Thailand", region="Asia")
7 my_function(city="Japan", region="Asia", latitude="36° 00' N", longitude="138° 00' E")
8
```

```
city: Thailand
region: Asia
-----
city: Japan
region: Asia
latitude: 36° 00' N
longitude: 138° 00' E
-----
```

ฟังก์ชันแบบระบุค่าโดยปริยายให้กับพารามิเตอร์ Default Parameter Value

```
1. def my_function(country = "Thailand") :  
2.     print("I am from " + country)  
3.  
4.  
5. my_function("Sweden")  
6. my_function("India")  
7. my_function()  
8. my_function("Brazil")  
9. my_function()
```

```
1 def my_function(country = "Thailand") :  
2     print("I am from " + country)  
3  
4 my_function("Sweden")  
5 my_function("India")  
6 my_function()  
7 my_function("Brazil")  
8 my_function()  
  
I am from Sweden  
I am from India  
I am from Thailand  
I am from Brazil  
I am from Thailand
```

ฟังก์ชันแบบส่งผ่านค่าประเภทลิสต์อาร์กิวเมนต์ Passing a List as an Argument

```
1. def my_function(food):  
2.     for x in food:  
3.         print(x)  
4.  
5. fruits = ["apple", "banana", "cherry"]  
6. my_function(fruits)
```

```
1 def my_function(food):  
2     for x in food:  
3         print(x)  
4  
5 fruits = ["apple", "banana", "cherry"]  
6 my_function(fruits)
```

```
apple  
banana  
cherry
```

ฟังก์ชันแบบส่งผ่านค่าประเภทลิสต์อาร์กิวเมนต์ Passing a List as an Argument

สถานการณ์: ต้องการสร้าง function สำหรับรับค่าคะแนนนักเรียนแบบ list และคำนวณหาค่าเฉลี่ย

```
1. def _____(_____):
2.     total = _____(scores)
3.     average = total / _____(scores)
4.     return average
5.
6. student_scores = [85, 90, 92, 88, 95]
7. _____ = calculate_average(_____)
8. print("คะแนนเฉลี่ยของนักเรียน:", average_score)
```