

DR.AMORNTHAP THONGCHIW

PH.D.(RESEARCH & STATISTIC IN COGNITIVE SCIENCE)

ใช้เตรียมสอบ ให้เข้าใจ PYTHON

ใช้เตรียมสอบสัมภาษณ์ ให้ได้งาน

ใช้ทบทวน ตอนทำงานให้ได้จริง

คู่มือทบทวนความรู้



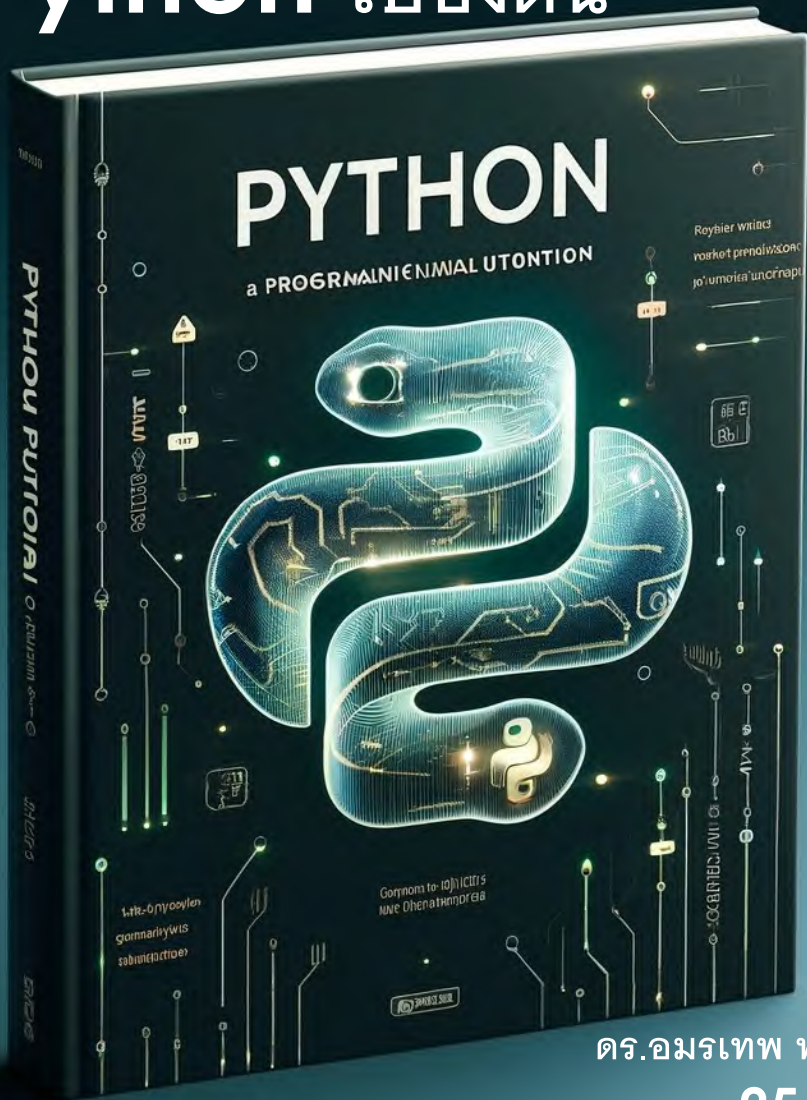
PYTHON

FUNDAMENTAL

(ฉบับเตรียมสอบ แบบเร่งรัด)

มี **QUIZ** ครบทุกหัวข้อ และคำศัพท์ควรรู้

คู่มือทบทวนความรู้ Python เบื้องต้น



ดร.อมรเทพ ทองชีว

เมษายน 2567

ใช้เตรียมตัวก่อนสอบ หรือใช้เตรียมพร้อมสมัครงาน

บทนำผู้เรียบเรียง

ในยุคที่เทคโนโลยีเปลี่ยนแปลงไปอย่างรวดเร็ว ทักษะการเขียนโปรแกรม ก็เป็นสิ่งสำคัญที่จะช่วยให้ท่านเป็นที่ต้องการของตลาดแรงงานและมีค่าตัวแพง เพราะคุณทำงานด้วยการใช้ Programming ได้ สามารถแก้ปัญหาขององค์กรได้

หนังสือเล่มนี้จึงถูกสร้างขึ้นมาเพื่อเป็นเครื่องมือในการเรียนรู้และทบทวนความรู้ด้านการเขียนโปรแกรม Python ที่สำคัญ ไม่ว่าจะเป็นเพื่อเตรียมสอบ สมัครงาน หรือใช้ในการทำงาน โดยมีเป้าหมายจากการทำ Quiz แล้วได้คะแนนเกิน 85% ขึ้นไป

ผมได้พยายามอย่างมากในการรวบรวมและจัดทำเนื้อหาให้มีความเข้าใจง่าย และเป็นประโยชน์สูงสุด โดยมีการนำเทคโนโลยี AI มาช่วยในการเขียนและจัดทำหนังสือเล่มนี้ เพื่อให้ครอบคลุมถึงเทคนิค และประเด็นสำคัญต่างๆ ที่ทันสมัยและตอบโจทย์การใช้งานจริงในปัจจุบัน

อย่างไรก็ตามผู้อ่านควรตรวจสอบ Code Python โดยตนเองอีกสักหน่อย เพื่อให้แน่ใจว่าเราได้ลองเห็นผลลัพธ์จริงจากภาษา Python ผู้เรียบเรียงหวังเป็นอย่างยิ่งว่าหนังสือเล่มนี้จะเป็นประโยชน์ในการพัฒนาศักยภาพของท่านในการเป็นนักพัฒนาระบบ, ทำงานในสาขาที่เกี่ยวข้อง หรือสำหรับความท้าทายในอนาคต

ขอให้ทุกท่านมีความสุขกับการเรียนรู้และการพัฒนาทักษะการเขียนโปรแกรม Python ผ่านหนังสือเล่มนี้ครับ.

Dr.Amornthep Thongchiw

Ph.d.(Research & Statistic in Cognitive Science)

สารบัญ

	p.
Part 1: Introduction to Python	
Ch. 1: Numbers and Expressions	4
Ch. 2: Operators and Precedence	14
Ch. 3: Variables and Assignment	25
Ch. 4: Introduction to Conditional Statements	38
Part 2: Working with Strings and Functions	
Ch. 5: Strings, Concatenation, and String Manipulation	51
Ch. 6: Functions: Building Blocks of Your Code	61
Part 3: Control Flow and User Input	
Ch. 7: Conditional Statements: if, else, and elif	73
Ch. 8: Loops: Repetitive Tasks Made Easy	85

สารบัญ

	p.
Part 4: Data Structures – Lists, Tuples, Sets, Dicts	
Ch. 9: Lists: Ordered Collections of Items	95
Ch. 10: Tuples: Immutable Ordered Sequences	106
Ch. 11: Dictionaries and Sets: Powerful Data Structures	117
Part 5: Programming Concepts	
Ch. 12: Exceptions: Handling Errors Gracefully	128
Ch. 13: Namespaces and Object-Oriented Programming	139
Ch. 14: Functions in Depth: Arguments, Scope, and More	150
Ch. 15: Modules and Packages: Reusability in Python	161

Recap knowledge

Part 1: Introduction to Python Ch. 1: Numbers and Expressions

พื้นฐานของตัวเลขและการดำเนินการทางคณิตศาสตร์ใน Python
ซึ่งจะเป็นประโยชน์ในการทำความเข้าใจวิธีการจัดการข้อมูลและการ
ประมวลผลข้อมูลอย่างมีประสิทธิภาพใน Python.

1. Basic Numeric Types

ใน Python, มีสามประเภทหลักของตัวเลข ได้แก่ integers (จำนวนเต็ม), floating-point numbers (จำนวนทศนิยม), และ complex numbers (จำนวนเชิงซ้อน).

Example Code:

```
python Copy code

# Integer
x = 10
print("Integer:", x)

# Floating-point
y = 20.5
print("Floating-point:", y)

# Complex number
z = 1 + 2j
print("Complex number:", z)
```

Recap knowledge

Part 1: Introduction to Python

Ch. 1: Numbers and Expressions

2. Arithmetic Operators

Python รองรับตัวดำเนินการทางคณิตศาสตร์พื้นฐานเช่น บวก ('+'), ลบ ('-'), คูณ ('*'), และหาร ('/'). นอกจากนี้ยังมีตัวดำเนินการหารเอาเศษ ('//'), หารเอาเศษ ('%'), และยกกำลัง ('**').

Example Code:

```
python Copy code

a = 8
b = 3

# Addition
print("Addition:", a + b)

# Subtraction
print("Subtraction:", a - b)

# Multiplication
print("Multiplication:", a * b)

# Division
print("Division:", a / b)

# Floor Division
print("Floor Division:", a // b)

# Modulus
print("Modulus:", a % b)

# Exponent
print("Exponent:", a ** b)
```

Recap knowledge

Part 1: Introduction to Python

Ch. 1: Numbers and Expressions

3. Numeric Functions

Python มีฟังก์ชันต่างๆ ที่ใช้สำหรับการคำนวณ ได้แก่ `abs()`, `round()`, และ `pow()`, ซึ่งเป็นตัวอย่างของฟังก์ชันที่ใช้บ่อยในการจัดการข้อมูล.

Example Code:

```
python Copy code

number = -5.5
power = 2

# Absolute value
print("Absolute Value:", abs(number))

# Round
print("Round:", round(number))

# Power
print("Power:", pow(number, power))
```

Recap knowledge

Part 1: Introduction to Python

Ch. 1: Numbers and Expressions

4. Working with Math Module

โมดูล `math` ให้ฟังก์ชันเชิงคณิตศาสตร์มากมาย เช่น `sqrt()`, `sin()`, `cos()`, และ `tan()`, ซึ่งมีประโยชน์มากในการประมวลผลข้อมูล.

Example Code:

python

 Copy code

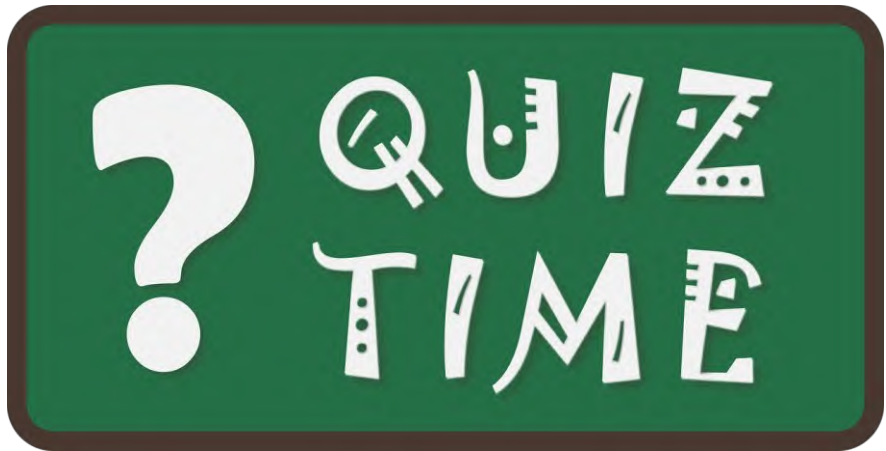
```
import math

# Square root
print("Square root:", math.sqrt(16))

# Sine, Cosine, and Tangent
angle = math.pi / 4 # 45 degrees
print("Sine:", math.sin(angle))
print("Cosine:", math.cos(angle))
print("Tangent:", math.tan(angle))
```

Part 1: Introduction to Python

Ch. 1: Numbers and Expressions



Part 1: Introduction to Python

Ch. 1: Numbers and Expressions

Question 1:

What is the output of the following expression?

```
python Copy code  
  
2 + 3 * 4
```

- A) 14
- B) 20
- C) 5
- D) None of the above

Question 2:

Evaluate the following expression:

```
python Copy code  
  
18 % 4
```

- A) 4
- B) 2
- C) 0
- D) 1

Question 3:

What does the expression `3 ** 2` compute?

- A) 6
- B) 9
- C) 8
- D) 5

Part 1: Introduction to Python

Ch. 1: Numbers and Expressions

Question 4:

Which of the following is the correct result of `'8 / 2'` in Python 3?

- A) 4.0
- B) 4
- C) 8
- D) 2

Question 5:

What type of number is the result of the operation `'5 / 2'` in Python 3?

- A) Integer
- B) Floating-point
- C) String
- D) Boolean

Question 6:

What will be the output of `'7 // 2'`?

- A) 3.5
- B) 3
- C) 2
- D) 4

Question 7:

Evaluate the following expression in Python:

```
python Copy code  
  
(2 + 3) * 4
```

- A) 14
- B) 20
- C) 5
- D) None of the above

Part 1: Introduction to Python

Ch. 1: Numbers and Expressions

Question 8:

What is the result of the expression `int(3.8)`?

- A) 3.0
- B) 4
- C) 3
- D) 4.0

Question 9:

What does `float('3.5')` convert to?

- A) '3.5'
- B) 3.5
- C) 3
- D) 4

Question 10:

What is the output of `round(3.14159, 2)`?

- A) 3
- B) 3.14
- C) 3.1
- D) 3.2

Part 1: Introduction to Python

Ch. 1: Numbers and Expressions

Answers:

Question 1:

Answer: A) 14

Explanation: การคำนวณใช้กฎการคูณก่อนแล้วจึงบวก, ดังนั้น $3 * 4 = 12$ และ $12 + 2 = 14$.

Question 2:

Answer: B) 2

Explanation: 18 หาร 4 มีเศษเป็น 2.

Question 3:

Answer: B) 9

Explanation: `3 ** 2` หมายถึง 3 ยกกำลัง 2, ซึ่งเท่ากับ 9.

Question 4:

Answer: A) 4.0

Explanation: ใน Python 3, การหารจะให้ผลลัพธ์เป็น floating-point.

Question 5:

Answer: B) Floating-point

Explanation: ใน Python 3, การปกติ (`/`) จะให้ผลลัพธ์เป็น floating-point เสมอ.

Part 1: Introduction to Python

Ch. 1: Numbers and Expressions

Answers:

Question 6:

Answer: B) 3

Explanation: การหารทศนิยมลงตัว (`//`) ให้ผลเป็นจำนวนเต็ม, 7 หาร 2 ได้ 3 ไม่เอาเศษ.

Question 7:

Answer: B) 20

Explanation: ใช้วงเล็บกำหนดลำดับการคำนวณก่อน, $(2 + 3)$ เป็น 5 และ $5 * 4$ เป็น 20.

Question 8:

Answer: C) 3

Explanation: การใช้ `int()` จะทำการตัดทศนิยมทิ้ง.

Question 9:

Answer: B) 3.5

Explanation: `float()` แปลงสตริง '3.5' เป็นตัวเลขทศนิยม 3.5.

Question 10:

Answer: B) 3.14

Explanation: ฟังก์ชัน `round()` ใน Python ใช้ในการปัดเศษทศนิยม, บัด 3.14159 เป็นทศนิยมสองตำแหน่งเป็น 3.14.

Recap knowledge

Part 1: Introduction to Python

Ch. 2: Operators and Precedence

เมื่อเขียนนิพจน์ที่มีหลาย operators, การเข้าใจเรื่อง precedence และการใช้วงเล็บเพื่อกำหนดการดำเนินการของแต่ละส่วนจะช่วยให้โค้ดของคุณทำงานได้ถูกต้องตามที่ต้องการ นอกจากนี้ยังช่วยป้องกันข้อผิดพลาดที่อาจเกิดขึ้นจากการอ่านโค้ดผิดได้อีกด้วย.

1. ประเภทของ Operators

ใน Python, มีหลายประเภทของ operators ซึ่งแบ่งออกเป็นหมวดหมู่ต่างๆ ได้แก่:

- **Arithmetic Operators:** ใช้สำหรับการคำนวณพื้นฐาน เช่น บวก ('+'), ลบ ('-'), คูณ ('*'), หาร ('/'), ยกกำลัง ('**'), หารเอาเศษ ('%'), และหารปัดเศษลง ('//').
- **Comparison Operators:** ใช้เพื่อเปรียบเทียบค่าระหว่างสองตัวแปร เช่น เท่ากับ ('=='), ไม่เท่ากับ ('!='), มากกว่า ('>'), น้อยกว่า ('<'), มากกว่าหรือเท่ากับ ('>='), น้อยกว่าหรือเท่ากับ ('<=').
- **Logical Operators:** ใช้เพื่อรวมเงื่อนไขทางตรรกะ เช่น AND ('and'), OR ('or'), NOT ('not').
- **Assignment Operators:** ใช้ในการกำหนดค่า เช่น ('='), ('+='), ('-='), และอื่นๆ.
- **Bitwise Operators:** ใช้สำหรับการดำเนินการตามบิต เช่น AND ('&'), OR ('|'), XOR ('^'), NOT ('~'), Left Shift ('<<'), Right Shift ('>>').

Recap knowledge

Part 1: Introduction to Python

Ch. 2: Operators and Precedence

2. Precedence ของ Operators

Precedence หมายถึงลำดับความสำคัญในการประมวลผลของ operators ในนิพจน์ (expression). การเข้าใจ precedence เป็นสิ่งสำคัญในการเขียนนิพจน์ที่มีหลาย operators เพื่อให้แน่ใจว่าการดำเนินการจะเกิดขึ้นตามที่คาดการณ์ไว้.

ตัวอย่างเช่น, ในนิพจน์ `1 + 3 * 2`, การคูณ (`*`) จะมี precedence สูงกว่าการบวก (`+`) ดังนั้นจะถูกประมวลผลก่อน ผลลัพธ์จึงเป็น `1 + 6 = 7`.

Example Code:

python

 Copy code

```
result = 1 + 3 * 2
print("Result of 1 + 3 * 2:", result)

# Using parentheses to change precedence
result = (1 + 3) * 2
print("Result of (1 + 3) * 2:", result)
```

Recap knowledge

Part 1: Introduction to Python

Ch. 2: Operators and Precedence

3. การใช้ Logical Operators

Logical operators มักใช้ในการรวมเงื่อนไขหลายๆ เงื่อนไข เช่น ในการตรวจสอบข้อมูลตามเงื่อนไขที่ซับซ้อนมากขึ้น.

Example Code:

```
python Copy code

x = 10
y = 20

# Logical AND
if x > 5 and y < 25:
    print("Both conditions are True")

# Logical OR
if x > 5 or y > 25:
    print("At least one of the conditions is True")

# Logical NOT
if not (x > 15):
    print("x is not greater than 15")
```

Part 1: Introduction to Python

Ch. 2: Operators and Precedence

