

(ฉบับเริ่มต้น)

Python

คู่มือเขียนโปรแกรมภาษาไพทอน
สำหรับผู้เริ่มต้น

แองเจล่า เจริญ
อนุชิต ชลโร

สำนักพิมพ์ก๊อปวาง

Python ฉบับเริ่มต้น

คู่มือเขียนโปรแกรมภาษาไพทอน สำหรับ
ผู้เริ่มต้น

แองเจล่า เจริญ

อนุชิต ชโลธร

คำนำสำนักพิมพ์

ในฐานะสำนักพิมพ์ เรามีความยินดีเป็นอย่างยิ่งที่ได้นำเสนอหนังสือ “Python ฉบับเริ่มต้น: คู่มือเขียนโปรแกรมภาษาไพทอนสำหรับผู้เริ่มต้น” เล่มนี้แก่ผู้อ่านทุกท่าน ในยุคที่เทคโนโลยีก้าวหน้าอย่างรวดเร็ว ความสามารถในการเขียนโปรแกรมได้กลายเป็นทักษะพื้นฐานที่จำเป็นในหลากหลายสาขาอาชีพ และภาษา Python ก็ได้พิสูจน์แล้วว่าเป็นหนึ่งในภาษาที่เข้าถึงง่ายที่สุดและทรงพลังที่สุดสำหรับการเริ่มต้นเส้นทางนี้

หนังสือเล่มนี้ถูกออกแบบมาอย่างพิถีพิถันเพื่อให้ผู้ที่ไม่มีพื้นฐานการเขียนโปรแกรมมาก่อนสามารถทำความเข้าใจและเริ่มต้นเขียนโค้ด Python ได้อย่างมั่นใจ ด้วยเนื้อหาที่ครอบคลุมตั้งแต่แนวคิดพื้นฐานไปจนถึงหัวข้อที่ซับซ้อนขึ้น พร้อมด้วยตัวอย่างโค้ดที่ชัดเจนและแบบฝึกหัดที่ช่วยเสริมสร้างความเข้าใจ เราเชื่อมั่นว่าหนังสือเล่มนี้จะเป็นเพื่อนร่วมทางที่ดีที่สุดสำหรับผู้เรียนทุกคน

เราหวังเป็นอย่างยิ่งว่า “Python ฉบับเริ่มต้น: คู่มือเขียนโปรแกรมภาษาไพทอนสำหรับผู้เริ่มต้น” จะเป็นแรงบันดาลใจและเป็นก้าวแรกที่ยืนยันสำหรับคุณในการสำรวจโลกอันกว้างใหญ่ของการเขียนโปรแกรม และนำความรู้ที่ได้ไปประยุกต์ใช้สร้างสรรค์สิ่งใหม่ๆ ในอนาคต

ขอให้ทุกท่านสนุกกับการเรียนรู้และประสบความสำเร็จในเส้นทางสาย
โปรแกรมเมอร์

คำนำ

ยินดีต้อนรับสู่ “Python ฉบับเริ่มต้น: คู่มือเขียนโปรแกรมภาษาไพทอนสำหรับผู้เริ่มต้น” หนังสือเล่มนี้ถูกออกแบบมาสำหรับผู้เริ่มต้นที่ไม่มีประสบการณ์ในการเขียนโปรแกรมมาก่อน หรือผู้ที่ต้องการเรียนรู้ภาษา Python เป็นภาษาแรก

ในโลกปัจจุบันที่ขับเคลื่อนด้วยข้อมูลและเทคโนโลยี ความสามารถในการเขียนโปรแกรมได้กลายเป็นทักษะที่สำคัญอย่างยิ่ง และ Python คือหนึ่งในภาษาโปรแกรมที่ได้รับความนิยมมากที่สุด ด้วยไวยากรณ์ที่อ่านง่ายและใช้งานได้หลากหลาย ทำให้ Python เป็นตัวเลือกที่ยอดเยี่ยมสำหรับการเริ่มต้นเส้นทางในโลกของการเขียนโค้ด ไม่ว่าจะเป็นคุณจะสนใจในการพัฒนาเว็บ วิทยาศาสตร์ข้อมูล ปัญญาประดิษฐ์ การทำงานอัตโนมัติ หรือแม้แต่การสร้างเกม Python ก็มีเครื่องมือและไลบรารีที่พร้อมรองรับ

ตลอดทั้งเล่มนี้ เราจะพาคุณดำดิ่งสู่พื้นฐานของ Python ตั้งแต่แนวคิดพื้นฐานที่สุด เช่น ตัวแปรและประเภทข้อมูล ไปจนถึงหัวข้อที่ซับซ้อนขึ้น เช่น การควบคุมการไหลของโปรแกรม, โครงสร้างข้อมูล, ฟังก์ชัน, โมดูล และการเขียนโปรแกรมเชิงวัตถุ (OOP) แต่ละบทจะมาพร้อมกับคำอธิบายที่ชัดเจน ตัวอย่างโค้ดที่เข้าใจง่าย และแบบฝึกหัดที่จะช่วยให้คุณได้ฝึกฝนและเสริมสร้างความเข้าใจ

เป้าหมายของเราคือทำให้การเรียนรู้ Python เป็นเรื่องที่น่าสนุก เข้า
ถึงได้ และมีประสิทธิภาพ เมื่อคุณอ่านจบ คุณจะมีความเข้าใจที่
แข็งแกร่งในหลักการเขียนโปรแกรมด้วย Python และพร้อมที่จะ
สำรวจโลกของการพัฒนาซอฟต์แวร์ในระดับที่สูงขึ้น

ขอให้สนุกกับการเรียนรู้!

สารบัญ

บทที่ 1 ทำความรู้จักกับภาษา Python	1
Python คืออะไร	1
คุณสมบัติเด่นของ Python	1
ทำไมต้องเขียน Python	2
การตั้งค่าสภาพแวดล้อม Python ของคุณ	3
โปรแกรมแรก “Hello, World!”	5
การใช้งาน Python Interpreter	7
บทที่ 2 พื้นฐาน Python	10
ตัวแปรและประเภทข้อมูล	10
ตัวเลข (Integers, Floats)	11
Integers	11
Floats	12
สตริง	13
บูลีน	16
การแปลงประเภทข้อมูล	16
ตัวดำเนินการพื้นฐาน	18
ตัวดำเนินการเปรียบเทียบ	18
ตัวดำเนินการตรรกะ	19

บทที่ 3 การควบคุมการไหลของโปรแกรม	21
คำสั่งเงื่อนไข (if, elif, else)	21
คำสั่ง if	21
คำสั่ง else	22
คำสั่ง elif (Else If)	22
ลูป for	23
วนซ้ำผ่านสตริง	23
วนซ้ำผ่านลิสต์	23
วนซ้ำด้วย range()	24
ลูป while	25
คำสั่ง break และ continue	25
คำสั่ง break	26
คำสั่ง continue	26
บทที่ 4 โครงสร้างข้อมูล	28
ลิสต์	28
การเข้าถึงรายการในลิสต์	29
การเปลี่ยนแปลงรายการในลิสต์	29
เมธอดของลิสต์ที่สำคัญ	29
ทูเพิล	31
การเข้าถึงรายการในทูเพิล	31
ข้อดีของทูเพิล	31

ดิกชันนารี	32
การเข้าถึงค่าในดิกชันนารี	32
การเปลี่ยนแปลงและเพิ่มรายการในดิกชันนารี	33
การลบรายการในดิกชันนารี	33
เมธอดของดิกชันนารีที่สำคัญ	33
เซต	34
การเพิ่มและลบรายการในเซต	35
การดำเนินการของเซต	35
บทที่ 5 ฟังก์ชัน	38
การนิยามและการเรียกใช้ฟังก์ชัน	38
การนิยามฟังก์ชัน	38
การเรียกใช้ฟังก์ชัน	38
อาร์กิวเมนต์ของฟังก์ชัน	39
อาร์กิวเมนต์แบบตำแหน่ง	39
อาร์กิวเมนต์แบบคีย์เวิร์ด	40
ค่าเริ่มต้นของพารามิเตอร์	40
ค่าที่ส่งคืน	41
ขอบเขต	42
ขอบเขตท้องถิ่น (Local)	42
ขอบเขตส่วนกลาง (Global)	43
Docstrings	44

บทที่ 6 โมดูลและแพ็คเกจ	47
โมดูลคืออะไร?	47
การนำเข้าโมดูล	48
import module_name	48
from module_name import function_name	49
from module_name import *	49
import module_name as alias	50
ไลบรารีมาตรฐานของ Python	50
math	50
random	51
datetime	51
การติดตั้งแพ็คเกจภายนอกด้วย pip	52
การตรวจสอบ pip	52
การติดตั้งแพ็คเกจ	53
การถอนการติดตั้งแพ็คเกจ	54
การแสดงแพ็คเกจที่ติดตั้ง	54
การติดตั้งแพ็คเกจจาก requirements.txt	54
บทที่ 7 การจัดการไฟล์	56
การอ่านจากไฟล์	56
การเปิดไฟล์	56
การอ่านทีละบรรทัด	57

การวนลูปอ่านไฟล์	58
การเขียนลงในไฟล์	58
โหมด “w” (โหมดเขียน)	58
โหมด “a” (โหมดเพิ่มต่อท้าย)	59
การทำงานกับพาธไฟล์	59
พาธสัมพัทธ์และพาธสมบูรณ์	60
โมดูล os.path	60
โมดูล pathlib	61
คำสั่ง with	62
บทที่ 8 การเขียนโปรแกรมเชิงวัตถุ	64
การเขียนโปรแกรมเชิงวัตถุ	64
คลาสและวัตถุ	65
การนิยามคลาส	65
เมธอด __init__	66
คุณสมบัติอินสแตนซ์และเมธอด	67
คุณสมบัติอินสแตนซ์	67
เมธอด	68
การสืบทอด	69
การห่อหุ้ม	71
บทที่ 9 การจัดการข้อผิดพลาด	75
ทำความเข้าใจข้อยกเว้น	75

บล็อก try...except	76
การจับข้อยกเว้นทั่วไป	77
การจัดการข้อยกเว้นเฉพาะ	78
บล็อก else (สำหรับ try...except)	79
บล็อก finally	80
บทที่ 10 การสร้างโครงสร้างข้อมูลแบบย่อ	83
การสร้างลิสต์แบบย่อ (List Comprehensions)	83
ตัวอย่าง List Comprehensions	84
การสร้างดิกชันนารีแบบย่อ (Dictionary Comprehensions)	86
ตัวอย่าง Dictionary Comprehensions	86
การสร้างเซตแบบย่อ (Set Comprehensions)	87
ตัวอย่าง Set Comprehensions	87
ข้อดีของการใช้ Comprehensions	88
บทที่ 11 ฟังก์ชันแบบย่อด้วย Lambda Functions	90
Lambda Functions คืออะไร	90
ตัวอย่างการเปรียบเทียบกับฟังก์ชันปกติ	91
ทำไมต้องใช้ Lambda Functions	91
การใช้ lambda กับ map()	92
การใช้ lambda กับ filter()	93
การใช้ lambda กับ sorted()	93
ข้อจำกัดและข้อควรพิจารณา	94

บทที่ 12 ไวยากรณ์พิเศษ *args และ **kwargs	96
*args	96
ตัวอย่างการใช้ *args	96
**kwargs	98
ตัวอย่างการใช้ **kwargs	98
การใช้ *args และ **kwargs ร่วมกัน	99
ตัวอย่างการใช้ร่วมกัน	100
การ Unpacking ด้วย * และ **	101
Unpacking Lists/Tuples ด้วย *	101
Unpacking Dictionaries ด้วย **	102
บทที่ 13 Decorators	103
ฟังก์ชันเป็น First-Class Citizens	103
ตัวอย่างแนวคิด	104
Decorator คืออะไร?	105
การสร้าง Decorator แบบพื้นฐาน	105
ไวยากรณ์ @ (Pie Syntax)	106
การจัดการอาร์กิวเมนต์ใน Decorator	107
การเก็บข้อมูลเมตาของฟังก์ชันดั้งเดิม	109
ตัวอย่างการใช้ functools.wraps	109
กรณีการใช้งานจริงของ Decorators	110
ตัวอย่าง: Decorator จับเวลา	110

บทที่ 14 Generators และ yield	113
Iterables และ Iterators คืออะไร?	113
Generators คืออะไร?	114
ตัวอย่าง Generator Function	115
ข้อดีของ Generators	117
กรณีการใช้งานจริง	118
Generator Expressions	119
บทที่ 15 การเขียนเทส	122
ทำไมการเขียนเทสจึงสำคัญ?	122
ประเภทของการเทส	123
โมดูล unittest	123
โครงสร้างพื้นฐานของ unittest	123
ตัวอย่าง: การทดสอบฟังก์ชันคำนวณ	124
Assert Methods ที่ใช้บ่อย	126
การรันเทส	127
แนะนำ pytest	128
การติดตั้ง pytest	128
การเขียนเทสด้วย pytest	128
การรัน pytest	129
บทที่ 16 Type Hinting	131
Type Hinting คืออะไร?	131

ไวยากรณ์พื้นฐานของ Type Hinting	132
การระบุประเภทข้อมูลของตัวแปร	132
การระบุประเภทข้อมูลในฟังก์ชัน	132
การใช้โมดูล typing	133
ประเภทข้อมูลพื้นฐานใน typing	134
ตัวอย่างการใช้โมดูล typing	134
การใช้ Static Type Checker (Mypy)	136
การติดตั้ง Mypy	136
การใช้งาน Mypy	136
บทที่ 17 ตัวอย่างโปรเจกต์	138
แนวคิดโปรเจกต์ แอปพลิเคชัน To-Do List	138
การวางแผนโปรเจกต์	139
การสร้างโปรเจกต์ทีละขั้นตอน	140
การตั้งค่าเริ่มต้นและฟังก์ชันการจัดการไฟล์	140
ฟังก์ชันการจัดการงาน (เพิ่ม, ดู, ทำเครื่องหมาย, ลบ)	141
เมนูและฟังก์ชันหลัก	143
การรันและการทดสอบโปรเจกต์	144
บทส่งท้าย: ก้าวต่อไปในเส้นทางของ Python	147
สำรวจพลังของระบบนิเวศ Python	147
NumPy	147
Pandas	148

Matplotlib / Seaborn	148
Flask / Django	148
Requests	148
Beautiful Soup	148
Scikit-learn	149
แหล่งเรียนรู้ที่คุณควรรู้จัก	149
เริ่มมีส่วนร่วมกับโอเพนซอร์ส	150
การเดินทางเพิ่งเริ่มต้น	150
ดาวนโหลดซอร์สโค้ด	152

บทที่ 1 ทำความรู้จักกับภาษา Python

ยินดีต้อนรับสู่บทแรกของหนังสือ ในบทนี้ เราจะมาทำความรู้จักกับภาษา Python ตั้งแต่พื้นฐานว่า Python คืออะไร ทำไมถึงได้รับความนิยม และจะเริ่มต้นใช้งาน Python ได้อย่างไร

Python คืออะไร

Python คือภาษาโปรแกรมระดับสูงที่พัฒนาโดย *Guido van Rossum* และเปิดตัวครั้งแรกในปี 1991 จุดเด่นสำคัญของ Python คือไวยากรณ์ที่อ่านง่ายคล้ายภาษาอังกฤษ ทำให้เป็นภาษาที่เหมาะสมสำหรับผู้เริ่มต้นและนักพัฒนามืออาชีพ

คุณสมบัติเด่นของ Python

- **อ่านง่ายเป็นธรรมชาติ:** โครงสร้างภาษาเรียบง่าย ไม่ซับซ้อน ลดภาระของผู้เรียน และทำให้โค้ดดูสะอาดตา
- **ใช้งานได้หลากหลาย:** รองรับงานหลายประเภท เช่น พัฒนาเว็บ, วิทยาศาสตร์ข้อมูล (Data Science), ปัญญาประดิษฐ์, Machine Learning, งานอัตโนมัติ (Automation), การวิเคราะห์ข้อมูล และการพัฒนาเกม