

10 แบบฝึกหัด Python พื้นฐานสำหรับผู้เริ่มต้น

ชุดแบบฝึกหัดนี้ถูกออกแบบมาสำหรับผู้เพิ่งเริ่มต้นเขียนโปรแกรม Python โดยเน้นที่โจทย์พื้นฐานที่เข้าใจง่าย และสามารถนำความรู้จากห้องเรียนมาประยุกต์ใช้ได้โดยตรง

HW01: โปรแกรมทักทายและคำนวณอายุ

สถานการณ์: สร้างโปรแกรมง่ายๆ เพื่อทักทายผู้ใช้และคำนวณอายุจากปีเกิด

ภารกิจ:

1. รับ ชื่อเล่น ของผู้ใช้
2. รับ ปีเกิด (พ.ศ.) ของผู้ใช้
3. คำนวณอายุโดยใช้ปีปัจจุบัน (พ.ศ. 2567)
4. แสดงข้อความทักทายพร้อมบอกอายุ

ความรู้ที่ครอบคลุม: input(), print(), ตัวแปร, int(), f-string

ตัวแปรที่แนะนำ:

- nickname: สำหรับเก็บชื่อเล่น
- birth_year_str: สำหรับเก็บปีเกิดที่รับมาเป็นข้อความ
- birth_year: สำหรับเก็บปีเกิดหลังแปลงเป็นตัวเลข
- current_year: สำหรับเก็บปีปัจจุบัน (พ.ศ.)
- age: สำหรับเก็บผลลัพธ์อายุ

HW02: โปรแกรมคำนวณค่าอาหาร (หารเท่ากัน)

สถานการณ์: คุณและเพื่อนไปทานอาหารและต้องการโปรแกรมช่วยหารค่าอาหารพร้อมทิป

ภารกิจ:

1. รับ ราคารวมของอาหาร
2. รับ เปอร์เซนต์ทิป ที่ต้องการให้ (เช่น 10 หรือ 15)
3. รับ จำนวนคน ที่จะหาร
4. คำนวณยอดที่แต่ละคนต้องจ่าย และแสดงผล

ความรู้ที่ครอบคลุม: float(), int(), ตัวดำเนินการคณิตศาสตร์, round()

ตัวแปรที่แนะนำ:

- total_cost: สำหรับเก็บราคารวมของอาหาร
- tip_pct: สำหรับเก็บเปอร์เซนต์ทิป
- num_people: สำหรับเก็บจำนวนคน
- tip_amount: สำหรับเก็บจำนวนเงินทิป
- final_bill: สำหรับเก็บยอดรวมสุทธิ (อาหาร + ทิป)
- cost_per_person: สำหรับเก็บค่าใช้จ่ายต่อคน

HW03: โปรแกรมสร้างชื่อเล่นเท่ๆ

สถานการณ์: สร้างโปรแกรมสนุกๆ สำหรับสร้างชื่อเล่นใหม่จากชื่อจริงและเดือนเกิด

ภารกิจ:

1. รับ ชื่อจริง ของผู้ใช้
2. รับ เดือนเกิด ของผู้ใช้ (เช่น มกราคม)
3. สร้างชื่อเล่นโดยนำ 3 ตัวอักษรแรกของชื่อมาทำเป็นตัวพิมพ์ใหญ่ ต่อด้วยความยาวของเดือนเกิด
4. แสดงผลชื่อเล่นที่สร้างขึ้น

ความรู้ที่ครอบคลุม: String Slicing ([0:3]), len(), .upper(), การต่อข้อความ

ตัวแปรที่แนะนำ:

- first_name: สำหรับเก็บชื่อจริง
- birth_month: สำหรับเก็บเดือนเกิด
- name_part: สำหรับเก็บ 3 ตัวอักษรแรกของชื่อ
- month_part: สำหรับเก็บความยาวของเดือนเกิด
- cool_nickname: สำหรับเก็บชื่อเล่นที่สร้างขึ้น

HW04: โปรแกรมแปลงวินาทีเป็น ชั่วโมง:นาที:วินาที

สถานการณ์: เขียนโปรแกรมเพื่อแปลงหน่วยเวลาจากวินาทีทั้งหมด ให้เป็นรูปแบบที่มนุษย์อ่านเข้าใจง่าย

ภารกิจ:

1. รับ จำนวนวินาทีทั้งหมด
2. คำนวณหาจำนวน ชั่วโมง, นาที, และวินาที
3. แสดงผลในรูปแบบ X ชั่วโมง Y นาที Z วินาที

ความรู้ที่ครอบคลุม: ตัวดำเนินการ // (หารปัดเศษลง) และ % (หารเอาเศษ)

ตัวแปรที่แนะนำ:

- total_seconds: สำหรับเก็บจำนวนวินาทีทั้งหมด
- hours, minutes, seconds: สำหรับเก็บผลลัพธ์แต่ละหน่วย
- remaining_seconds: ตัวแปรช่วยสำหรับเก็บวินาทีที่เหลือ

HW05: โปรแกรมตรวจสอบคะแนนสอบ

สถานการณ์: เขียนโปรแกรมรับคะแนนสอบ และแจ้งผลสอบเป็นค่าความจริง (True/False) โดยโปรแกรมต้องไม่พังถ้าผู้ใช้กรอกข้อมูลผิด

ภารกิจ:

1. รับ คะแนนสอบ จากผู้ใช้
2. ใช้ try-except เพื่อป้องกันกรณีผู้ใช้กรอกข้อมูลที่ไม่ใช่ตัวเลข
3. ถ้าข้อมูลถูกต้อง: สร้างตัวแปรเก็บผลลัพธ์แบบบูลีน (True/False) จากการตรวจสอบว่าคะแนนมากกว่าหรือเท่ากับ 50 หรือไม่ แล้วแสดงผลลัพธ์นั้น
4. ถ้าข้อมูลไม่ถูกต้อง: แสดงข้อความแจ้งเตือน

ความรู้ที่ครอบคลุม: try-except, float(), ตัวดำเนินการเปรียบเทียบ \geq , ผลลัพธ์แบบบูลีน

ตัวแปรที่แนะนำ:

- score_input: สำหรับเก็บข้อมูลดิบที่รับมา
- score: สำหรับเก็บคะแนนหลังแปลงเป็น float
- is_passed: สำหรับเก็บผลลัพธ์การตรวจสอบ (True หรือ False)

HW06: โปรแกรมคำนวณดัชนีมวลกาย (BMI)

สถานการณ์: เขียนโปรแกรมคำนวณค่า BMI ซึ่งเป็นค่าที่นิยมใช้เพื่อประเมินภาวะน้ำหนักตัว

ภารกิจ:

1. รับ น้ำหนัก (หน่วยกิโลกรัม)
2. รับ ส่วนสูง (หน่วยเซนติเมตร)
3. คำนวณ BMI ตามสูตร: น้ำหนัก (kg) / (ส่วนสูง (m) * ส่วนสูง (m))
4. แสดงผลค่า BMI โดยปัดเศษ 1 ตำแหน่ง

ความรู้ที่ครอบคลุม: float(), ตัวดำเนินการคณิตศาสตร์, round()

ตัวแปรที่แนะนำ:

- weight_kg: สำหรับเก็บน้ำหนัก (กก.)
- height_cm: สำหรับเก็บส่วนสูง (ซม.)
- height_m: สำหรับเก็บส่วนสูงหลังแปลงเป็นเมตร
- bmi: สำหรับเก็บผลลัพธ์ BMI

HW07: โปรแกรมสร้างรหัสผ่านแนะนำ

สถานการณ์: สร้างโปรแกรมช่วยคิดรหัสผ่านง่ายๆ จากข้อมูลที่ใช้คุ้นเคย

ภารกิจ:

1. รับ ชื่อสัตว์เลี้ยง
2. รับ เลข 2 ตัวสุดท้ายของเบอร์โทรศัพท์
3. สร้างรหัสผ่านโดยนำ 4 ตัวอักษรแรกของชื่อสัตว์เลี้ยง + เลขที่รับมา + สัญลักษณ์พิเศษ _X
4. แสดงผลรหัสผ่านที่ได้

ความรู้ที่ครอบคลุม: String Slicing, การต่อข้อความ

ตัวแปรที่แนะนำ:

- pet_name: สำหรับเก็บชื่อสัตว์เลี้ยง
- phone_digits: สำหรับเก็บเลขท้ายเบอร์โทร
- password: สำหรับเก็บรหัสผ่านที่สร้างขึ้น

HW08: โปรแกรมวิเคราะห์ประโยค

สถานการณ์: เขียนโปรแกรมเพื่อวิเคราะห์ประโยคง่ายๆ ที่ผู้ใช้ป้อนเข้ามา

ภารกิจ:

1. รับ 1 ประโยค จากผู้ใช้
2. นับ จำนวนอักขระทั้งหมด (รวมเว้นวรรค)
3. นับ จำนวนคำทั้งหมด ในประโยค
4. แสดงผลการวิเคราะห์

ความรู้ที่ครอบคลุม: `len()`, `.split()`

ตัวแปรที่แนะนำ:

- `sentence`: สำหรับเก็บประโยคที่รับมา
- `char_count`: สำหรับเก็บจำนวนอักขระ
- `words`: สำหรับเก็บ list ของคำที่แยกออกมา
- `word_count`: สำหรับเก็บจำนวนคำ

HW09: โปรแกรมคำนวณส่วนลด

สถานการณ์: ร้านค้าแห่งหนึ่งมีโปรโมชั่น ซื้อครบ 500 บาท จะได้รับส่วนลด 10%

ภารกิจ:

1. รับ ราคารวมของสินค้า
2. ตรวจสอบว่าราคาถึง 500 บาทหรือไม่
3. ถ้าถึง ให้คำนวณราคาสุทธิหลังหักส่วนลด 10%
4. ถ้าไม่ถึง ให้แสดงราคาเดิม
5. แสดงราคาสุทธิที่ต้องจ่าย

ความรู้ที่ครอบคลุม: if-else, ตัวดำเนินการเปรียบเทียบ, การคำนวณเปอร์เซ็นต์

ตัวแปรที่แนะนำ:

- price: สำหรับเก็บราคาสินค้า
- final_price: สำหรับเก็บราคาสุทธิ

HW10: โปรแกรมสลับชื่อ-นามสกุล

สถานการณ์: เขียนโปรแกรมเพื่อรับชื่อเต็ม แล้วสลับลำดับเพื่อแสดงในรูปแบบ "นามสกุล ชื่อ"

ภารกิจ:

1. รับ ชื่อ-นามสกุลเต็ม (คั่นด้วยการเว้นวรรค)
2. แยกชื่อและนามสกุลออกจากกัน
3. นำมาต่อกันใหม่ในรูปแบบ นามสกุล ชื่อ
4. แสดงผลลัพธ์

ความรู้ที่ครอบคลุม: `.split()`, การเข้าถึงข้อมูลใน List (Indexing), f-string

ตัวแปรที่แนะนำ:

- `full_name`: สำหรับเก็บชื่อเต็ม
- `name_parts`: สำหรับเก็บ list ของชื่อที่แยกแล้ว
- `first_name`: สำหรับเก็บชื่อ
- `last_name`: สำหรับเก็บนามสกุล
- `formatted_name`: สำหรับเก็บชื่อที่จัดรูปแบบใหม่