|  |
| --- |
| **Worksheet 4 (19 มิ.ย. 2562)** |

ในการใช้งานฟังก์ชัน loop ในบางครั้ง เราอาจจะต้องการหยุดการทำงานของฟังก์ชัน loop หรือ ต้องการให้ ฟังก์ชัน loop ทำงานตามเงื่อนไขที่กำหนด เช่น



จากตัวอย่างด้านบนเราต้องการวน loop ทั้งหมดสิบครั้ง และมีการตั้งเงื่อนไขว่าถ้าค่า i น้อยกว่า 5 ให้ continue และพิมพ์ค่า i แล้วถ้า i<8 ให้พิมพ์ค่า i ถ้าไม่เข้าทั้งสองเงื่อนไขก็ให้ break และ พิมพ์ค่า i สิ่งที่เกิดขึ้นคือ code นี้จะพิมพ์ค่า 5 6 7 ออกมา นักเรียนอาจสงสัยว่า ในเมื่อเรามีเงื่อนไข i<5 แล้ว และภายใต้เงื่อนไขนั้นมีคำสั่ง print(i) ทำไมถึงไม่การพิมพ์ออกมา

เนื่องจากคำสั่ง continue จะเป็นคำสั่งให้จบ loop ปัจจุบัน และเริ่ม loop ถัดไปทันทีโดยไม่สนคำสั่งถัดไป ทำให้ในเมื่อค่า i ยังน้อยกว่า 5 loop ปัจจุบันก็จะจบและเริ่ม loop ใหม่ทันที ส่วนคำสั่ง break เป็นการจบการทำงานของ loop ทันที โดยไม่สนว่า loop จะวนไปแล้วกี่ครั้ง

**1) ให้นักเรียนลองสลับ code ระหว่างบรรทัด 3 และ 4 และสลับ code ระหว่างบรรทัด 8 และ 9 และลองดู output**

**2) ให้นักเรียนเขียน Python code เพื่อพิมพ์เลขที่หารด้วย 5 ลงตัว และเป็นเลขที่อยู่ในช่วง 21 – 39 และ 61 – 79 (ให้ประยุกต์ใช้ for loop, continue และ break)**

**จาก Python code ด้านล่าง**

****

Loop i

Loop j

**Loop i เป็น loop สำหรับควบคุมการพิมพ์ในแต่ละแถว (row) และ loop j เป็น loop สำหรับพิมพ์ค่าในแต่ละ column โดยที่ loop i หนึ่ง loop (นับเป็น outer loop) โปรแกรมจะทำ loop j (inner loop) จนเสร็จ เช่น เริ่มต้นที่ outer loop โดยค่า i เริมที่ 1 (จาก range(1, 11)) โปรแกรมจะมาทำคำสั่ง loop j โดยที่ j มีค่าเริ่มตั้งแต่ 1 ถึง 10 (จบที่ 10 เพราะ 11 - 1) เมื่อจบ loop j เราก็ print() เพื่อขึ้นบรรทัดใหม่ และเริ่ม loop i โดยที่ i มีค่าเท่ากับ 2 แล้วก็มาวน loop j จนครบ ทำอย่างนี้ไปเรื่อย ๆ จนจบ loop i**

**ในตัวอย่างถัดไปเป็นการเพิ่มเงื่อนไข เพื่อให้พิมพ์เลขบางตัวในแต่ละบรรทัด ในกรณีนี้เราระบุเงื่อนไขว่า**

**if j < i + 1 ให้พิมพ์ค่า i**

****

**3) ให้นักเรียนกำหนดเงื่อนไขเพื่อให้ได้ผลลัพธ์ดังต่อไปนี้**

**แบบที่ 1 แบบที่ 2**

** **

**จาก Python code ด้านล่าง เราสามารถพิมพ์ค่าให้มีลักษณะคล้ายตัวอย่างด้านบน โดยไม่ต้องใช้เงื่อนไขเข้ามาช่วย (ใช้ได้ในบางกรณีเท่านั้น)**



**4) ให้ปรับแก้เพื่อให้ได้ผลดังต่อไปนี้**



**5) ให้ปรับแก้ Python code ในข้อก่อนหน้านี้เพื่อให้พิมพ์เฉพาะเลขคี่ และพิมพ์เครื่องหมาย \* แทนเลขคู่ ตามตัวอย่างด้านล่าง**



จากตัวอย่าง Python code ในการใช้ for-loop ด้านล่าง



หากต้องการใช้ while-loop สามารถเขียน code ได้ดังต่อไปนี้



ข้อแตกต่างที่ชัดเจนสำหรับการใช้ while-loop เพื่อให้ได้ผลลัพธ์ที่เหมือนกับ for-loop ในตัวอย่างข้างบนคือ เราต้องกำหนดค่าเริ่มต้นให้ตัวแปร i = 1 และทำการเพิ่มค่า i ในแต่ละ loop ให้กับการใช้ for-loop

**6) จาก for-loop ด้านล่าง จงเขียน Python code โดยใช้ while-loop เพื่อให้ได้ผลลัพธ์เดียวกัน**



 **7) จาก Python code ในข้อ 5 ให้เขียน Python code ใหม่ด้วย while-loop เพื่อให้ได้ผลลัพธ์ต่อไปนี้**



จาก Python code ตัวอย่างด้านล่าง ซึ่งรับค่าอุณหภูมิและหน่วย (เป็นเซลเซียส (C) หรือ ฟาเรนไฮต์ (F)) และทำการแปลงอุณหภูมิจากเซลเซียสเป็นฟาเรนไฮต์ หรือ ฟาเรนไฮต์เป็นเซลเซียส

อันดับแรกรับค่าเข้ามาและเก็บไว้ที่ตัวแปร temp หลังจากนั้นให้ตัวแปร degree มีค่าเท่ากับตัวอักษรตั้งแต่ตัวแรกจนถึงอักษรก่อนตัวสุดท้ายและแปลงเป็นเลขจำนวนเต็มด้วยฟังก์ชัน int() เช่น temp = “456c”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ตำแหน่ง | 0 | 1 | 2 | 3 |
| ค่า | 4 | 5 | 6 | c |
| ตำแหน่ง | -4 | -3 | -2 | -1 |

degree = int(temp[:-1])) ค่า degree จะมีค่าเท่ากับ temp[ : -1] คือค่าเริ่มจากตำแหน่ง 0 คือค่า 4 ไปจนถึงตำแหน่ง -1 (ไม่รวมค่าที่ตำแหน่ง -1) คือค่า 6 เท่ากับว่าค่า degree จะมีค่าคือ 456 ซึ่งยังเป็นค่าแบบ string และก็เปลี่ยนให้เป็นเลขจำนวนเต็มด้วยฟังก์ชัน int( )

สำหรับ i\_convention คือตัวแปรที่มีค่าเท่ากับ temp ที่ตำแหน่ง -1 ในกรณีนี้คือ c



เช่นถ้าใส่ค่า 35c จะแปลงเป็น 95 ฟาเรนไฮต์ สำหรับฟังก์ชัน quit( ) ใช้สำหรับหยุดการทำงานของโปรแกรมโดยทันที (ข้อแตกต่างระหว่าง quit() กับ break คือ break จะใช้เพื่อหยุดการทำงานของ loop แต่ถ้าเป็น quit() โปรแกรมจะหยุดทันที)

**8) ให้เขียน Python code เพื่อใช้สำหรับแปลงระยะทางจากกิโลเมตรเป็นไมล์ หรือ ไมล์เป็นกิโลเมตร โดยที่ค่า input ต้องมีหน่วยเป็น km (สำหรับกิโลเมตร) หรือ mi (สำหรับไมล์) ยกตัวอย่างเช่น ถ้าใส่ค่า 10km ระยะทางเป็นไมล์คือ 6.2137 ไมล์ (ให้แสดงผลมีจุดทศนิยมสี่ตำแหน่ง)**