



Ruby Range Intervals

in Ruby Programming

650710837 Pattanapong Duangmang

Ruby Range Intervals คือ?

Ruby Range Intervals คือ การกำหนดช่วงของค่า เช่น เลข 1 ถึง 20 หรือ ตัวอักษร a ถึง z เป็นต้น เพื่อให้เราสามารถตรวจสอบได้ว่าสิ่งที่เราสนใจอยู่ในช่วงนั้นหรือไม่ โดยจะใช้ "===" เพื่อตรวจสอบ หากค่าอยู่ในช่วงจะคืนค่าเป็น true แต่หากค่านั้น ไม่ได้อยู่ในช่วงก็จะคืนค่าเป็น false ซึ่งการกำหนดช่วงของค่าจะมีด้วยกัน 2 แบบ คือ

1. การใช้ ..
2. การใช้ ...

โดยทั้ง 2 วิธีต่างกันโดยที่วิธีที่ 1 นั้นจะนับรวมตัวข้างหลัง แต่วิธีที่ 2 จะไม่นับรวมตัวข้างหลัง ดังตัวอย่างในรูป

```
(1..5).to_a    # => [1, 2, 3, 4, 5]
(1...5).to_a   # => [1, 2, 3, 4]
```



คณะวิทยาศาสตร์
มหาวิทยาลัยศิลปากร

```
# ตัวอย่างเลขของค่าที่กำหนด อยู่ในช่วงของค่า
if ((1..10) === 5) # หากอยู่ในช่วงจะทำตามคำสั่งใน if
  puts "5 lies in (1..10)" # แสดงผลลัพธ์
end
# ตัวอย่างเลขของค่าที่กำหนด ไม่ได้อยู่ในช่วงของค่า
if ((1..10) === 12) # หากอยู่ในช่วงจะทำตามคำสั่งใน if
  puts "12 is not lies in (1..10)" # แสดงผลลัพธ์
end
# ตัวอย่างตัวอักษรของค่าที่กำหนด อยู่ในช่วงของค่า
if (('a'..'j') === 'c') # หากอยู่ในช่วงจะทำตามคำสั่งใน if
  puts "c lies in ('a'..'j')" # แสดงผลลัพธ์
end
# ตัวอย่างตัวอักษรของค่าที่กำหนด ไม่ได้อยู่ในช่วงของค่า
if (('a'..'j') === 'z') # หากอยู่ในช่วงจะทำตามคำสั่งใน if
  puts "z is not lies in ('a'..'j')" # แสดงผลลัพธ์
end
```

ตัวอย่างการใช้งาน Ruby Range Intervals

ความแตกต่างกับภาษา โปรแกรมมิ่ง อื่นๆ

- Ruby มี Range เป็น built-in syntax ที่ใช้ง่าย
- ส่วนใน Python ก็มี range แต่ไม่รวมค่าปลายสุด
- ส่วน Java ต้องใช้ IntStream
- ขณะที่ภาษา C ไม่มี range โดยตรง ทำให้ต้องใช้การเขียนเช็คค่าเอง



```
import java.util.stream.IntStream;

public class Demo {
    public static void main(String[] args) {
        boolean isInRane = IntStream.rangeClosed(1, 10).anyMatch(n -> n == 5);
        if(isInRange){
            System.out.println("5 5 lies in (1..10)");
        }
    }
}
```

Range ใน Java

ใน Java เราสามารถใช้ IntStream เพื่อสร้างช่วงตัวเลขได้ โดยการใช้ method range() หรือ rangeClosed()

```
#include <stdio.h>

int main() {
    int x = 5;
    for (int i = 1; i <= 10; i++) {
        if(i == x){
            printf("%d lies in (1..10)\n", x);
        }
    }
    return 0;
}
```

Range ใน C

ภาษา C ไม่มี Range โดยตรง จึงต้องใช้ if เพื่อเช็คค่า หรือใช้ loop ถ้าต้องการประมวลผลค่าทั้งช่วง

```
x = 5
if x in range(1, 11):
    print(f"{x} lies in (1..10)")
```

Range ใน Python

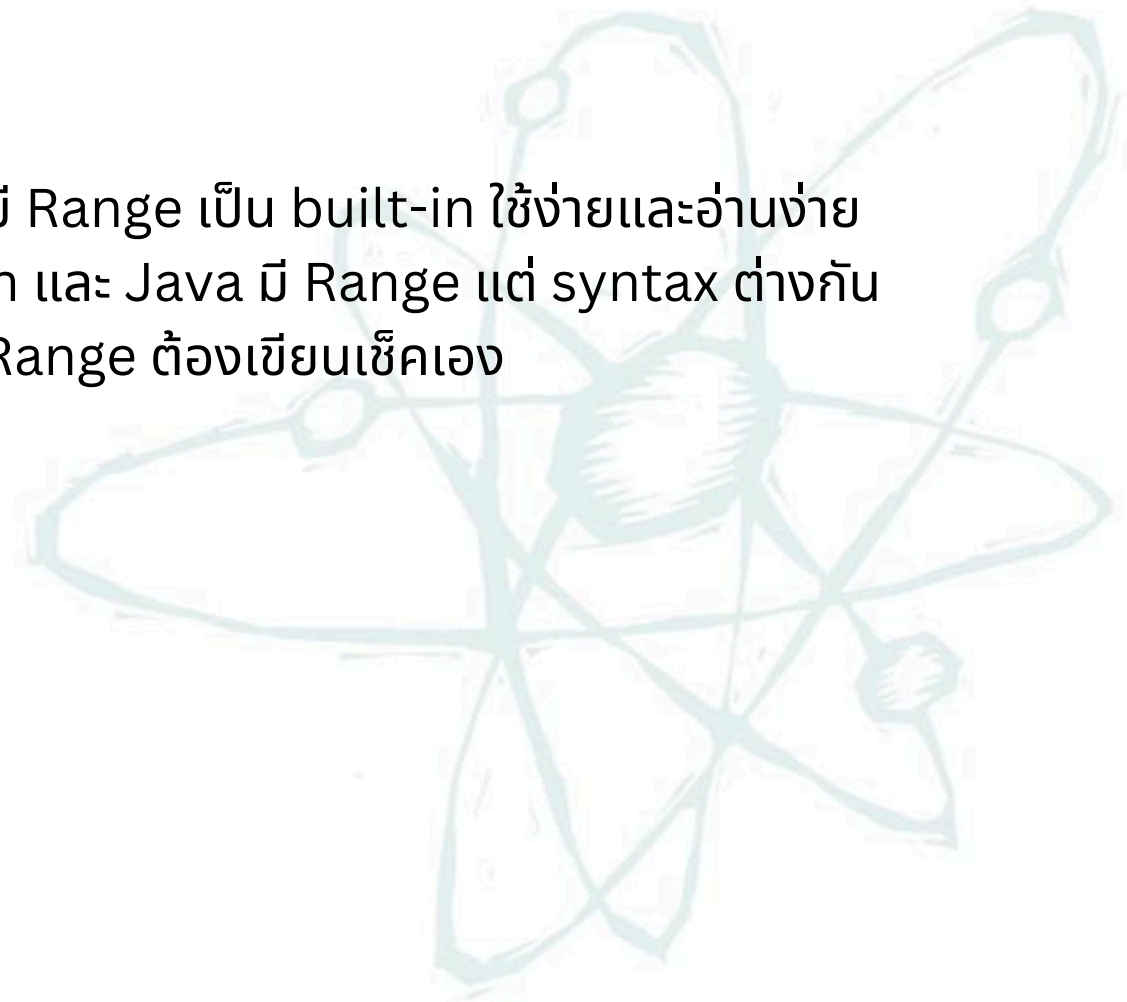
Python ใช้ range() เพื่อสร้างลำดับตัวเลข และสามารถใช้ in ตรวจสอบว่าค่าอยู่ในช่วงหรือไม่ รวมถึงใช้กับ loop ได้โดยตรง

ประโยชน์ของ Range

1. ตรวจสอบค่า – เช่น ตรวจสอบ input ว่าอยู่ในช่วงที่ยอมรับได้
2. สร้าง sequence – สร้างชุดข้อมูลที่ต่อเนื่อง เช่น 1-100
3. Data Filtering – กรองข้อมูลตามช่วงค่า เช่น ช่วงอายุ, ช่วงคะแนน

สรุปการเปรียบเทียบ

Ruby มี Range เป็น built-in ใช้ง่ายและอ่านง่าย
Python และ Java มี Range แต่ syntax ต่างกัน
C ไม่มี Range ต้องเขียนเช็คเอง



References

Techotopia. Ruby Ranges. Techotopia, n.d.,
https://www.techotopia.com/index.php/Ruby_Ranges

Tutorialspoint. Ruby – Ranges. Tutorialspoint, n.d.,
[ps://www.tutorialspoint.com/ruby/ruby_ranges.htm](https://www.tutorialspoint.com/ruby/ruby_ranges.htm)

Tutorialspoint. IntStream rangeClosed() method in Java. Tutorialspoint, 30 July 2019,
<https://www.tutorialspoint.com/intstream-range-closed-method-in-java>

Oracle. IntStream. Java Platform, Standard Edition 8 API Specification, Oracle, 2014,
<https://docs.oracle.com/javase/8/docs/api/java/util/stream/IntStream.html#range-int-int->

GeeksforGeeks. Python range() function. GeeksforGeeks, 11 July 2025,
<https://www.geeksforgeeks.org/python/python-range-function/>



คณะวิทยาศาสตร์
มหาวิทยาลัยศิลปากร