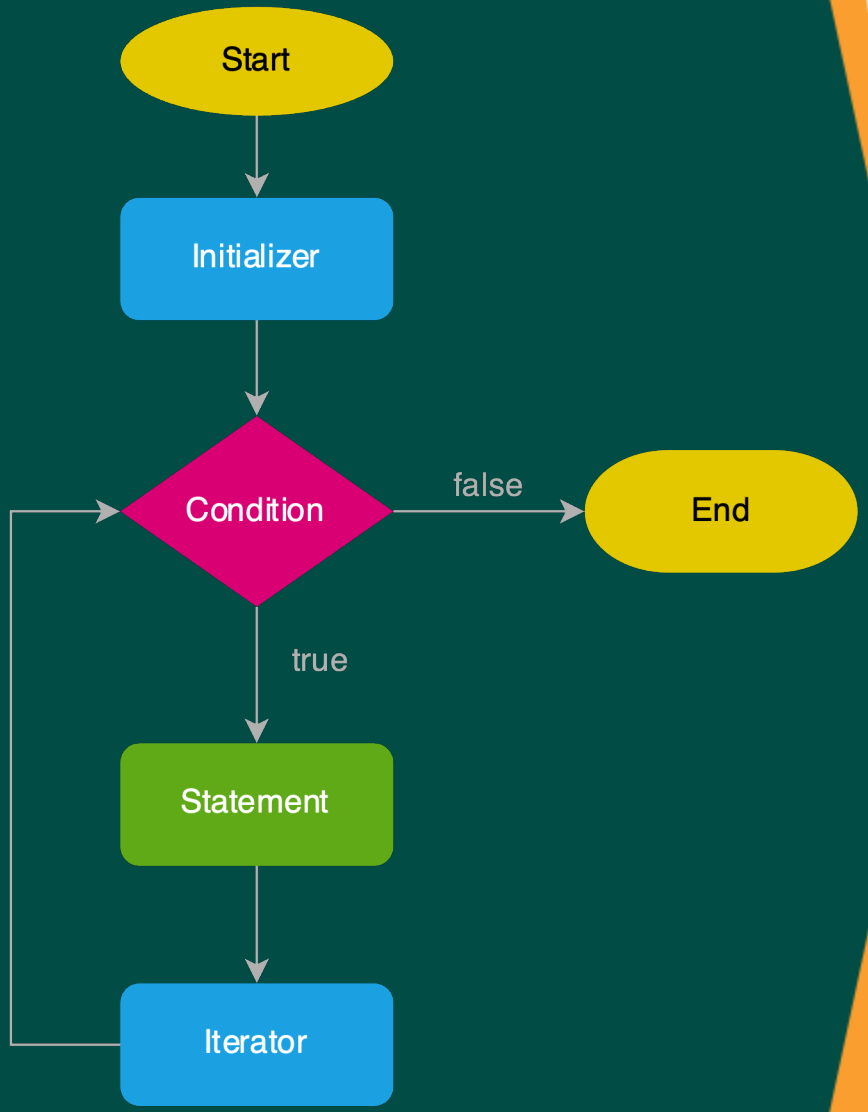
FOR LOOP IN DART

จัดทำโดย
วัชรินทร์ คำโคตรสูนย์ 620710820



For loop คืออะไร ?

เป็นประเภทของลูป (loop) ที่นิยมและใช้บ่อยที่สุด ซึ่งช่วยให้โค้ดนั้นสามารถทำงานตามคำสั่งซ้ำได้หลายครั้ง

โครงสร้างภาษา (Syntax)

เงื่อนไขเริ่มต้น เงื่อนไขสำหรับการทำงานซ้ำ การเปลี่ยนแปลงหลังจากการทำงานซ้ำ

```
for(initialization; condition; increment/decrement){
    statements;   // โค้ดที่จะทำงานในแต่ละรอบของ loop
}
```

เงื่อนไขเริ่มต้น

```
for(initialization; condition; increment/decrement){
        statements;
}
```

- **Initialization (เงื่อนไขเริ่มต้น)** : ส่วนนี้เป็นส่วนที่กำหนดค่าเริ่มต้นของตัวแปรที่จะใช้ในการควบคุมลูป ส่วนนี้จะถูกทำครั้งเดียวก่อนที่ลูปจะเริ่มทำงาน เช่น การกำหนดค่าเริ่มต้นให้กับตัวแปรนับ

เงื่อนไขสำหรับการทำงานซ้ำ

```
for(initialization; condition; increment/decrement){
            statements;
}
```

- **Condition (เงื่อนไขสำหรับการทำงานซ้ำ)** : เป็นเงื่อนไขที่ต้องเป็นจริงเพื่อให้ลูปทำงานซ้ำในแต่ละรอบ หากเงื่อนไขนี้เป็นเท็จ ลูป for จะสิ้นสุดการทำงาน

การเปลี่ยนแปลงหลังจากการทำงานซ้ำ

```
for(initialization; condition; increment/decrement){
            statements;
}
```

- **Increment/Decrement (การเปลี่ยนแปลงหลังจากการทำงานซ้ำ)** : ส่วนนี้ใช้ในการเปลี่ยนค่าของตัวแปรที่ใช้ในการควบคุมลูปหลังจากที่บล็อกของโค้ดในแต่ละรอบทำงานเสร็จสิ้น เช่น การเพิ่มค่าหรือลดค่าของตัวแปรนับ

“ แนวคิดหลักของ for loop คือสามารถทำงานซ้ำกับบล็อกของคำสั่งต่าง ๆ โดยที่กำหนดเงื่อนไขเริ่มต้น เงื่อนไขการทำงานซ้ำ และการเปลี่ยนแปลงหลังจากการทำงานซ้ำได้ ”

ตัวอย่างที่ 1

การพิมพ์ตัวเลข 1 ถึง 10 โดยใช้ลูป for

ใน Dart เราใช้ for ในการเรียกใช้ลูป

- เริ่มต้นที่ int i = 1; ซึ่งเป็นการประกาศและกำหนดค่าตัวแปรเริ่มต้นเป็น i มีค่า 1
- เงื่อนไข i <= 10; จะทำให้ลูปทำงานตราบใดที่ i มีค่าน้อยกว่าหรือเท่ากับ 10
- i++ คือการเพิ่มค่า i ทีละหนึ่งหลังจากแต่ละรอบ

```dart
void main() {
  for (int i = 1; i <= 10; i++)
{    print(i);
  }
}
```

```
1
2
3
4
5
6
7
8
9
10
```

ตัวอย่างภาษา DART :

```dart
void main() {
  for (int i = 1; i <= 10; i++)
{   print(i);
  }
}
```

```java
//ตัวอย่างภาษา JAVA
public class Main {
    public static void main(String[] args)
{       for (int i = 1; i <= 10; i++) {
            System.out.println(i);
        }
    }
}
```

```c
//ตัวอย่างภาษา C
#include <stdio.h>
int main() {
    for (int i = 1; i <= 10; i++)
{       printf("%d\n", i);
    }
    return 0;
}
```

ตัวอย่างที่ 2

การพิมพ์ตัวเลข 10 ถึง 1 โดยใช้ลูป for

- ตรงข้ามจากตัวอย่างที่แล้ว ถ้าเราต้องการพิมพ์ตัวเลข 10 ถึง 1 เราจะใช้การลดค่า (decrement) ของ i
- i-- คือการลดค่า i ทีละหนึ่งหลังจากแต่ละรอบ

```
void main() {
  for (int i = 10; i >= 1; i--)
{    print(i);
  }
}
```

```
10
9
8
7
6
5
4
3
2
1
```

ตัวอย่างที่ 3

พิมพ์ชื่อ 10 ครั้งโดยใช้ลูป for

- ในตัวอย่างนี้จะแสดงการพิมพ์ชื่อ 10 ครั้งโดยใช้ลูป for เพื่อวนรอบการทำงานตามเงื่อนไข

```
void main() {
  for (int i = 0; i < 10; i++)
{    print("John Doe");
  }
}
```

```
John Doe
John Doe
John Doe
John Doe
John Doe
John Doe
John Doe
John Doe
John Doe
John Doe
```

ตัวอย่างที่ 4

แสดงผลรวมของจำนวนเต็มบวก n โดยใช้ลูป for

- ในตัวอย่างนี้จะแสดงการคำนวณผลรวมของจำนวนเต็มบวก n โดยใช้ลูป for เพื่อวนรอบการทำงาน

```
void main(){

  int total = 0;
  int n = 100; // change as per
required
  for(int i=1; i<=n; i++){
    total = total + i;
  }

  print("Total is $total");

}
```

```
Total is 5050
```

ตัวอย่างที่ 5

แสดงจำนวนคู่ระหว่าง 50 ถึง 100 โดยใช้ลูป for

- ในตัวอย่างนี้จะแสดงการพิมพ์จำนวนคู่ที่อยู่ระหว่าง 50 ถึง 100 โดยใช้ลูป for เพื่อวนรอบการทำงาน

```
void main(){
  for(int i=50; i<=100; i++)
{   if(i%2 == 0){
      print(i);
    }
  }
}
```

```
50
52
54
...
98
100
```

Infinite Loop in DART

การทำงานวนซ้ำซึ่งเงื่อนไขไม่เคยเป็นเท็จ นั่นหมายความว่าโปรแกรมจะทำงานในลูปไปเรื่อยๆ โดยใช้ทรัพยากรคอมพิวเตอร์มาก จะถูกดำเนินการซ้ำๆ จนกว่าจะหมดหน่วยความจำที่ใช้ไป

ตัวอย่างที่ 6

พิมพ์ตัวเลข 1 ไปสู่อนันต์

- โปรแกรมนี้จะพิมพ์ตัวเลข 1 ไปสู่อนันต์ เนื่องจากเงื่อนไขที่ใช้ในลูปคือ i >= 1 ซึ่งจะเป็นจริงเสมอกับการที่ i จะเพิ่มขึ้นทีละหนึ่งในทุกครั้งที่ลูปทำงาน

```
void main() {
  for (int i = 1; i >= 1; i++)
{   print(i);
  }
}
```

```
1
1
1
1
1
...
```

DART FOR ... IN LOOP

```
for (var element in expression)
{ // คำสั่งที่จะถูกทำซ้ำเพื่อแต่ละสมาชิก
}
```

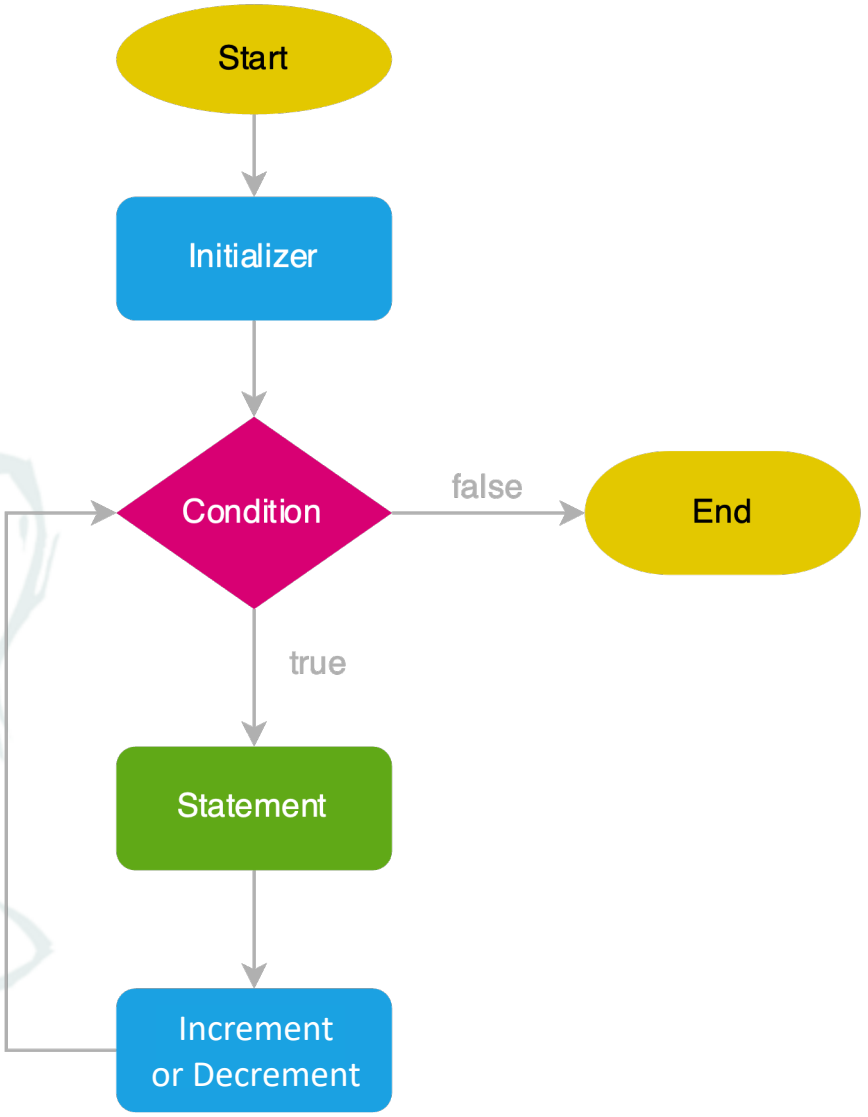
ตัวอย่างที่ 7

- ในตัวอย่างเมื่อลูปถูกเรียกใช้งาน จะมีการวนรอบผ่าน List ของตัวเลขแต่ละตัว และในแต่ละรอบค่าของตัวเลขจะถูกผูกกับตัวแปร number และถูกแสดงผลออกทางหน้าจอ

```
void main() {
  List<int> numbers = [1, 2, 3, 4, 5];
  for (var number in numbers) {
    print(number); // พิมพ์ค่าแต่ละสมาชิกใน List
  }
}
```

```
1
2
3
4
5
```

SUMMARY


Start
Initializer
Condition
false
End
true
Statement
Increment
or Decrement

Reference

- Dart tutorial. "LOOPS IN DART." https://dart-tutorial.com/conditions-and-loops/for-loop-in-dart
- Aditya Taparia , GeeksforGeeks. "Dart Loops." https://www.geeksforgeeks.org/dart-loops/
- Eric Windmill , Flutter by Example. "Loops: for and while." https://flutterbyexample.com/lesson/loops-for-and-while
- Dart Tutorials. "Loops." https://dart.dev/language/loops