

ฝึกปฏิบัติการครั้งที่ 1

พื้นฐานภาษา Dart

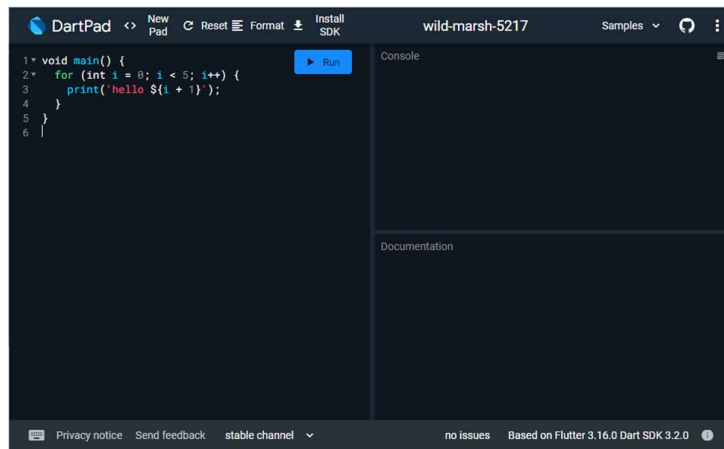
วัตถุประสงค์

1. เพื่อให้มีความเข้าใจเรื่องตัวแปร ชนิดของตัวแปร การประกาศตัวแปร
2. เพื่อศึกษาการทำงานของตัวดำเนินการแบบต่างๆ
3. เพื่อศึกษาการทำงานแบบควบคุม

ตอนที่ 1 เครื่องมือการพัฒนาโปรแกรมด้วยภาษา Dart เบื้องต้น

การเขียนโปรแกรมด้วยภาษา Dart

การฝึกปฏิบัติการเขียนโปรแกรมด้วยภาษา Dart เบื้องต้นใช้เครื่องมือออนไลน์คือ <https://dartpad.dev/>



ฝึกโปรแกรมแรกด้วยการแสดงข้อความว่า “Hello, World!”

```
void main() {  
  print("Hello, World!");  
}
```

ตอนที่ 2 ตัวแปร ชนิดของตัวแปร และการประกาศตัวแปร

2.1 พิจารณาชื่อตัวแปรต่อไปนี้ว่าตั้งชื่อตัวแปรตามหลักในภาษาซีหรือไม่ โดยทำเครื่องหมาย ✓ หรือ ✗ หน้าหมายเลขข้อ พร้อมทั้งให้เหตุผลประกอบ

- | | |
|---------------------|-------------|
| 1. sci_room | เหตุผล..... |
| 2. n-sync | เหตุผล..... |
| 3. 148py | เหตุผล..... |
| 4. _computer | เหตุผล..... |
| 5. year# | เหตุผล..... |
| 6. const | เหตุผล..... |
| 7. \$book | เหตุผล..... |
| 8. NetPrice | เหตุผล..... |
| 9. box.001 | เหตุผล..... |
| 10. Program 1 | เหตุผล..... |

2.2 สังเกตผลลัพธ์จากโค้ดต่อไปนี้

```
void main() {
    dynamic x = 123;
    x = 4.567;
    x = 'hello';
    x = true;

    var y = 123;
    y = 4.567;
    y = 'hello';
    y = true;
}
```

อธิบายผลลัพธ์ที่เกิดขึ้น เหตุใดตัวแปร x สามารถเปลี่ยนค่าข้อมูลที่มีชนิดต่างกันได้ และตัวแปร y ไม่สามารถเปลี่ยนค่าข้อมูลที่มีชนิดต่างกันได้

2.2 สังเกตผลลัพธ์จากโค้ดต่อไปนี้

```
void main() {
    dynamic x;
    x = 99;
    x = 3.141;
    x = 'Dart';
    x = false;

    var y;
    y = 99;
    y = 3.141;
    y = 'Dart';
    y = false;
}
```

อธิบายผลลัพธ์ที่เกิดขึ้น เหตุใดทั้งตัวแปร x และตัวแปร y สามารถเปลี่ยนค่าข้อมูลที่มีชนิดต่างกันได้

2.3 สังเกตผลลัพธ์จากโค้ดต่อไปนี้

```
void main() {
    final int MIN = 20;
    const int MAX = 100;
    MIN = 30;
    MAX = 50;

    final String sec;
    sec = '800';
    sec = '870';

    const String id;
    id = "01418344";
}
```

อธิบายผลลัพธ์ที่เกิดขึ้น ข้อผิดพลาด (Error) ที่เกิดขึ้นคืออะไรบ้าง และเหตุใดจึงเป็นเช่นนั้น

2.4 สังเกตผลลัพธ์จากโค้ดต่อไปนี้

```
void main() {
  String s1 = '01418344';
  String s2 = "Mobile Dev.";
  String s3 = '''Computer Science''';
  String s4 = """KU Sriracha""";

  String q1 = 'Flutter on
    mobile';
  String q2 = "Dart programming
    language";
  String q3 = '''Mobile app development
    for Android and iOS''';
  String q4 = """Building Cross-Platform Apps
    with Flutter and Dart""";
}
```

อธิบายผลลัพธ์ความเหมือนและความแตกต่างของ ' ', " ", ' ', และ " " เมื่อใช้กับข้อความหรือตัวแปรชนิด String

2.5 สังเกตผลลัพธ์จากโค้ดสำหรับการเชื่อมต่อข้อความต่อไปนี้

```
void main() {
  String s1 = 'Hello' + ' ' + 'World!';
  String s2 = 'Flutter' ' ' 'Dart';
  String s3 = 'Programming';
  print(s1);
  print(s2);
  print("I'm learning " + s3);
  print("I'm learning $s3");

  int num = 123;
  String s4 = 'Number is ' + num.toString();
  print(s4);
}
```

ผลลัพธ์ที่ได้คือ

ลำดับอักขระในข้อความ

ข้อความคือรายการของตัวอักษร (List of Characters) และสามารถใช้อันดับ (Index) ในการอ้างถึงตัวอักษรแต่ละตัวที่ประกอบกันเป็นข้อความนั้นๆ ได้โดยระบุเลขลำดับในเครื่องหมาย [] ซึ่งตัวอักษรตัวแรกเริ่มเลขลำดับที่ 0 (ศูนย์) เช่น String str = 'Programming'; สามารถอ้างถึงตัวอักษรได้คือ str[0] = 'P' หรือ str[5] = 'a' เป็นต้น

ตอนที่ 3 การเขียนโปรแกรมด้วยภาษา Dart

3.1 โปรแกรม inverse of integer หาค่าส่วนกลับของตัวเลข num (inverse, $\frac{1}{num}$) กำหนดให้ inverse ของ num เป็น 0 เมื่อ num = 0 ให้นักศึกษาพิมพ์โค้ดต่อไปนี้และรันโปรแกรม

```
1. void main() {  
2.     int num = 19;  
3.     double inv_n;  
4.     inv_n = num != 0 ? 1.0/num : 0;  
5.     print("inverse num = $inv_n");  
6. }
```

สังเกตผลการทำงานของบรรทัดที่ 4 จากนั้นแก้ไขโปรแกรมให้ทำงานเหมือนเดิม แต่ให้ใช้คำสั่ง if-else แทนการใช้ Ternary operator (เงื่อนไข ? กรณีเงื่อนไขที่เป็นจริง : กรณีเงื่อนไขที่เป็นเท็จ)

Hint if(____) inv_n = _____ else inv_n = _____

3.2 เขียนโปรแกรมเพื่อตรวจสอบตัวเลขจำนวนเต็ม (num) ว่าเป็นเลขคู่ (even) หรือ เลขคี่ (odd) และเป็นจำนวนบวก (positive) หรือจำนวนลบ (negative) หรือเป็นค่าศูนย์ โดยสามารถแสดงผลซึ่งแบ่งออกเป็น 5 กรณี คือ positive even number, positive odd number, zero, negative even number และ negative odd number

Hint ใช้โครงสร้าง if-else และตัวดำเนินการ OR หรือ AND ในการกำหนดเงื่อนไขร่วมด้วย เช่น ถ้าจำนวนเต็มมากกว่าศูนย์ และเป็นเลขคู่ แสดงว่าเป็น positive even number เป็นต้น (ตัวแปร num ให้นักศึกษาหาค่าเป็นตัวเลขใดก็ได้)

ตัวอย่างผลลัพธ์

Number is 30 30 is positive even number.	Number is 0 0 is zero.
Number is -28 -28 is negative even number.	Number is -1805 -1805 is negative odd number.

3.3 เขียนโปรแกรมเพื่อให้ได้ผลลัพธ์ตามด้านล่าง โดยใช้คำสั่งการวนลูปใดก็ได้ (for, while, หรือ do-while)

1	7
2	14
3	21
4	28
5	35

3.4 พิมพ์โปรแกรมต่อไปนี้และรันเพื่อสังเกตผลลัพธ์ที่เกิดขึ้น

```
void main() {  
    // Nullable variables  
    String? nullableString;
```

```

int? nullableInt;
double? nullableDouble;
bool? nullableBool;

// Nullable variables can be assigned null
nullableString = null;
nullableInt = null;
nullableDouble = null;
nullableBool = null;

// Nullable variables can also hold non-null values
nullableString = "Hello, Dart!";
nullableInt = 42;
nullableDouble = 3.14;
nullableBool = true;

// Use null-aware operators with nullable types
String nonNullableString = nullableString ?? "Default Value";
int nonNullableInt = nullableInt ?? 0;

// Check for null before using the value
if (nullableBool != null) {
    print("Boolean value: $nullableBool");
} else {
    print("Boolean value is null");
}

// Functions can have nullable return types
String? getResult(bool success) {
    return success ? "Operation succeeded!" : null;
}

// Using a function with a nullable return type
String? result = getResult(true);
print(result ?? "No result");

// Lists and collections can contain nullable types
List<int?> nullableNumbers = [1, null, 3, null, 5];

// Iterate through the list and use null-aware access
for (var number in nullableNumbers) {
    print("Number: $number");
}
}

```

อธิบายผลลัพธ์ที่เกิดขึ้นจากการใช้เครื่องหมาย ? และ ??

.....

.....

.....

.....

.....

ฝึกปฏิบัติทำการทดลองนอกห้องเรียน

เขียนโปรแกรมสำหรับการตรวจสอบตัวเลขว่าเป็นตัวเลข Armstrong หรือไม่ ตัวเลข Armstrong เป็นตัวเลขที่หาผลรวมของเลขแต่ละหลักที่ยกกำลังด้วยจำนวนหลัก แล้วผลลัพธ์ก็เท่ากับตัวมันเอง ตัวอย่างเช่น

- 5 เป็นตัวเลข Armstrong เพราะ $5^1 = 5$ ($5 = 5$ จึงเป็นตัวเลข Armstrong)
- 13 ไม่เป็นตัวเลข Armstrong เพราะ $(1^2) + (3^2) = 1 + 9 = 10$ ($10 \neq 13$)
- 152 ไม่เป็นตัวเลข Armstrong เพราะ $(1^3) + (5^3) + (2^3) = 1 + 125 + 8 = 134$ ($134 \neq 152$)

หมายเหตุ ให้เขียนโปรแกรมโดยใช้คำสั่งทำซ้ำเพื่อหาว่าตัวเลข 153 เป็นตัวเลข Armstrong หรือไม่