

Week 5 : Dart Programing

ผศ. ดร. เกื้อแก้ว ธีเนศวร

kejkaew.tha@mail.kmutt.ac.th

ครั้งที่	วันที่	หัวข้อ
6	21/2/2566	Flutter – Basics of Dart Programming
7	28/2/2566	Article: Methodology + submit background
8	7/3/2566	Flutter – OOP, asynchronous programming, and Introduction to Widgets
9	14/3/2566	Flutter - Introduction to Layouts
10	21/3/2566	Article: Experiment and results + submit methodology
12	28/3/2566	Flutter - Animations and Graphics
11	4/4/2566	Article: Conclusions and future work + submit experiment and results
13	11/4/2566	ไม่มีเรียน (GDM443/DMT443)
14	18/4/2566	Flutter - Data Storage and Management
15	25/4/2566	Article: Abstract
16	2/5/2566	Flutter - Data Storage and Management (2)
17	9/5/2566	ส่ง research paper
18	16/5/2566	Flutter - Deploying Flutter Applications
19	23/5/2566	Flutter - ส่งโปรเจค

Today outline

Variables and Operation

Null Safety

Conditions and Loops

Functions

File: `dart_application_week5.dart`

วิธีการ Create new Dart project

Open the Command Palette (Ctrl+Shift+P on Windows หรือ Cmd+Shift+P on macOS).

Select the "Dart: New Project" command and press Enter.

Select Console application

Select folder and type a name of project

หรือจะใช้ DartPad ในการเขียนก็ได้ <https://dartpad.dartlang.org/>

Hello World!

ตัวโปรแกรมจะเริ่มทำงานที่ฟังก์ชัน main เป็นหลัก เราไม่สามารถเขียน statement นอกฟังก์ชันได้

การแสดงผลมาตรฐานจะใช้คำสั่ง print (คำสั่งนี้ auto-newline เสมอ)

ควรจำไว้ว่า ภาษา Dart นั้นการเขียน ; (semi-colon) ไม่ใช่ optional คือจำเป็นต้องใส่ ; ทุกครั้งหลังจบ statement ไม่สามารถละ ; ได้แบบ ภาษาตระกูล C ยุคใหม่ๆ เช่น JavaScript หรือ Kotlin

Example

```
3 void main(List<String> arguments) {  
4   print("Hello World!");  
}
```

Output:

```
Hello World!  
Exited
```

Comment

การใส่คอมเมนต์ทำได้เหมือนภาษา C ทุกอย่างคือ

// สำหรับ inline comment

เปิดด้วย /* และปิดด้วย */ สำหรับ multi-line comment (ไม่สามารถ nested ได้นะ)

Variable และ Data Type

type	คำอธิบาย	ตัวอย่าง
int	เลขจำนวนเต็ม	0, 1, -5, 86400
double	เลขทศนิยม	0.0, 0.1, 0.14, -12.34
num	เลขทศนิยม หรือ เลขจำนวนเต็ม	123, 0.123
bool	ค่าทางตรรกศาสตร์	true, false
String	สายอักขระ (ประโยค)	'hello world!', "This is a book" <-- ในภาษา Dart สามารถใช้ได้ทั้ง " (double quote) และ ' (single quote)

Example

```
6 // Variable and data type
7 int y = 5;
8 double d = 1.0;
9 bool isDone;
10 String name;
11 print(y);
12 print(isDone);
```

ตัวแปรของ Dart ทั้งหมดเป็นแบบ reference type ทั้งหมด ทำให้สามารถมีค่าเป็น null ได้ทั้งหมด ทำให้อาจจะเกิด error ได้ง่าย และทำให้ user ไม่ happy ดังนั้น เราต้องมีวิธีจัดการ!!

Variable

แต่ใน Dart ยังมีชนิดของตัวแปรแบบพิเศษ ซึ่งไม่จำเป็นต้องประกาศ type เลย แต่ตัวภาษาจะ auto assign ชนิดของตัวแปรให้เอง

type	คำอธิบาย
var	เป็นการละ type เอาไว้ให้โปรแกรมกำหนดให้ (ตาม value) หลังจากนั้นตัวแปรจะถูกกำหนดเป็น type นั้นไปตลอด ไม่สามารถเปลี่ยนแปลงได้แล้ว
final	เหมือน var แต่ไม่สามารถเปลี่ยนแปลงค่าได้
const	ค่าคงที่

ข้อแตกต่างระหว่าง final vs const

final เป็นการกำหนดว่าตัวแปรนี้ ไม่สามารถเปลี่ยนแปลงค่าได้ กำหนดค่าแล้ว กำหนดเลย (immutable) ซึ่งเป็นตัวแปรประเภท runtime ดังนั้นเราสามารถ กำหนดค่า final จากตัวแปรหรือฟังก์ชันอื่นได้

const เป็นการประกาศค่าคงที่ โดยค่าที่กำหนดให้จะต้องเป็นค่า เท่านั้น (เช่น 10, 'value') เพราะเป็นตัวแปรที่กำหนดค่าตั้งแต่ตอน compile-time

Example

```
14 // variable var const final
15 var k = 20; //var
16 print(k);
17
18 int x = 10;
19 final int f1 = 1; //กำหนดตัวแปร int ให้เป็นค่าคงที่
20 final f2 = 'final-val'; //ใช้เหมือน var คือไม่กำหนด type ก็ได้
21 final f3 = x + 20; //กำหนดค่าจากกโดยคำนวณมาจากตัวแปรอื่นอีกที
22
23 const int c1 = 1; //กำหนดตัวแปร int ให้เป็นค่าคงที่
24 const c2 = 'const-val'; //ใช้เหมือน var คือไม่กำหนด type ก็ได้
25 const c3 = x + 20; //Error: Not a constant expression. เพราะ x เป็นตัวแปรที่ value
```

String

เราจะดู operation, properties และ methods
บางอย่างที่ใช้สำหรับจัดการ string

เช่น

- String Concatenation
- isEmpty/ isEmpty
- length
- toUppercase() / toLowercase()

- replaceAll()
- split()
- toString()

Example

```
27 // String
28 String firstName = 'John';
29 String lastName = 'Doe';
30 print("Using +, Full Name is " + firstName + " " + lastName + ".");
31 print("Using interpolation, full name is $firstName $lastName.");
32
33 // // empty and length
34 String str = "Hi";
35 print(str.isEmpty); //Example of isEmpty
36 print(str.isNotEmpty); //Example of isNotEmpty
37 print("The length of the string is: ${str.length}"); //Example of Length
```

Output:

```
Using +, Full Name is John Doe.
Using interpolation, full name is John Doe.
false
true
The length of the string is: 2
```

Example

```
39 // capital letter
40 String address1 = "Florida"; // Here F is capital
41 String address2 = "Texas"; // Here T and A are capital
42 print("Address 1 in uppercase: ${address1.toUpperCase()}");
43 print("Address 1 in lowercase: ${address1.toLowerCase()}");
44 print("Address 2 in uppercase: ${address2.toUpperCase()}");
45 print("Address 2 in lowercase: ${address2.toLowerCase()}");
46
47 // replaceAll method
48 String text =
49 |   "I like milk and milk is good for health.";
50 String newText = text.replaceAll("milk", "water");
51 print("Original Text: $text");
52 print("Replaced Text: $newText");
```

Output:

```
Address 1 in uppercase: FLORIDA
Address 1 in lowercase: florida
Address 2 in uppercase: TEXAS
Address 2 in lowercase: texas
Original Text: I like milk and milk is good for health.
Replaced Text: I like water and water is good for health.
```

Example

```
53 // Split method
54 String allNames = "James, Jane, John, Max";
55 List<String> listNames = allNames.split(",");
56 print("Value of listName is $listNames");
57
58 // toString method
59 int number = 20;
60 String result = number.toString();
61 print("Type of number is ${number.runtimeType}");
62 print("Type of result is ${result.runtimeType}");
```

Output:

```
Value of listName is [James, Jane, John, Max]
Type of number is int
Type of result is String
```


List

List หรือเรียกอีกอย่างว่า Array สามารถสร้างได้ดังนี้

Example

```
64 // List and Map
65 List<String> names = ["Jane", "John", "Max"];
66 print("Value of names is $names");
67 print("Value of names[0] is ${names[0]}"); // index 0
68 print("Value of names[1] is ${names[1]}"); // index 1
69 print("Value of names[2] is ${names[2]}"); // index 2
70
71 // Finding Length of List
72 int length = names.length;
73 print("The Length of names is $length");
```

```
Value of names is [Jane, John, Max]
Value of names[0] is Jane
Value of names[1] is John
Value of names[2] is Max
The Length of names is 3
```

Map

Map เป็น object สำหรับเก็บ key-value pairs เหมือน JSON

Example

```
75 // Map
76 Map<String, String> myDetails = {
77     'name': 'John Doe',
78     'address': 'USA',
79     'fathername': 'Soe Doe'
80 };
81
82 print(myDetails['name']);
```

Output:

John Doe

Math Operation

การใช้ +, -, *, / และ % เหมือนกับภาษาอื่นๆ แต่มีข้อควรระวังที่ตัว /
สำหรับภาษาอื่นถ้าเรานำ int / int ผลที่ออกมาจะได้เป็น int แน่ๆ แต่
สำหรับ Dart นั้นการหารจะได้ค่าออกมาเป็น double เสมอ
ตัวอย่าง

```
84 // Math operation
85 int p = 4 + 3;
86 int x = 4 / 2; // Error: A value of type 'int' is not a 'double'.
87 int x = 4 ~/ 2;
88 print('x value = $x');
```

Logical and Relational Operators

Operator	Description
>	Greater than
<	Lesser than
>=	Greater than or equal to
<=	Lesser than or equal to
==	Equality
!=	Not equal

Operator	Meaning
is	True if the object has the specified type
is!	False if the object has the specified type

Logical Operators

Operator	Description	Example
&&	And – The operator returns true only if all the expressions specified return true	(A > 10 && B > 10) is False.
	OR – The operator returns true if at least one of the expressions specified return true	(A > 10 B > 10) is True.
!	NOT – The operator returns the inverse of the expression's result. For E.g.: !(7>5) returns false	!(A > 10) is True.

Example

```
90 // Logic and relational operators
91 int a = 10;
92 int b = 20;
93 bool k = a > b;
94 print("a > b is $k");
95 print(a is int);
96 k = a > 10 && b == 10;
97 print(k);
```

Output:

```
a > b is false
true
false
```

Question 1

ข้อใดต่อไปนี้ เมื่อกด run แล้ว ไม่เกิด Error

```
int c = 50;  
const c1 = c + 20;
```

```
var c = 50;  
final c1 = c + 20;
```

```
int x;  
x = 6 / 3;
```

Activity 1

เขียนโปรแกรมหาผลหารและเศษที่เหลือจากการหาของจำนวนเต็มสอง
จำนวน โดยใช้ Dart

Null Handling

ใน Dart 2 ตัวแปรปกติจะเป็น non-nullable ซึ่งหมายความว่า จะเป็น null ไม่ได้

ดังนั้น ภาษา Dart มีจึงมี operation สำหรับจัดการค่า null เพื่อให้ผู้พัฒนาโปรแกรมหลีกเลี่ยงปัญหาที่เกิดจาก null

Null Handling

ตัวอย่างการจัดการ null

- เครื่องหมาย ?? เป็นการเช็กว่าตัวแปรตัวนี้ ถ้ามีค่าเป็น null ให้ใช้ค่า default ที่กำหนดให้แทน
- เครื่องหมาย .? เป็นการเช็คว่า หากตัวแปรของเราเป็น object ซึ่งสามารถเรียกใช้งาน method ต่างๆ ได้ ถ้าเป็น null จะใช้ action นั้นไม่ได้
- เครื่องหมาย ? ใช้สำหรับประกาศตัวแปรที่เป็น null value

Example

Output:

```
Name is null  
Stranger
```

```
99 // Null safty  
100 int productid = 20; // non-nullable  
101 // productid = null; // give error  
102 // Declaring a nullable variable by using ?  
103 int? productid_2 = 30;  
104 productid_2 = null;  
105 // Declaring a nullable variable by using ?  
106 String? name;  
107 // Assigning John to name  
108 name = "John";  
109 // Assigning null to name  
110 name = null;  
111 // Checking if name is null using if statement  
112 if (name == null) {  
113     print("Name is null");  
114 }  
115 // Using ?? operator to assign a default value  
116 String name1 = name ?? "Stranger";  
117 print(name1);
```

Example

Output:

```
0  
[1, 2, null, 4]
```

```
119 // ?. Null Conditional  
120 List<int>? a;  
121 int b = a?.length ?? 0;  
122 print(b);  
123  
124 // list of nullable ints  
125 List<int?> items = [1, 2, null, 4];  
126 print(items);
```

Question 2

สองข้อนี้รันแล้วจะเกิด error เราจะต้องแก้ที่ตรงไหน

```
void main() {  
    int age;  
    age = null;  
    print("Age is $age");  
}
```

```
void main() {  
    List<int> items = [1, 2, null, 4];  
    print(items);  
}
```

Conditions

if-else if- else

Syntax

```
if(condition1){  
    statements1;  
}else if(condition2){  
    statements2;  
}else if(condition3){  
    statements3;  
}  
.  
.  
.  
else(conditionN){  
    statementsN;  
}
```

Example

```
128 // // condition
129 int num1 = 1200;
130 int num2 = 1000;
131 int num3 = 150;
132
133 if (num1 > num2 && num1 > num3) {
134     print("Num 1 is greater: i.e $num1");
135 } else if (num2 > num1 && num2 > num3) {
136     print("Num2 is greater: i.e $num2");
137 } else if (num3 > num1 && num3 > num2) {
138     print("Num3 is greater: i.e $num3");
139 }
```

Output:

```
Num 1 is greater: i.e 1200
```

For Loop

Syntax

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

Example

```
141 // for loop  
142 for (int i = 1; i <= 10; i++) {  
143     print(i);  
144 }
```

Output:

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


For In Loop

Loop ใน list element หรือ variables

Example:

```
146 // For In Loop
147 List<String> footballplayers = ['Ronaldo', 'Messi', 'Neymar', 'Hazard'];
148 for (String player in footballplayers) {
149     print(player);
150 }
```

Output:

```
Ronaldo
Messi
Neymar
Hazard
```

While Loop

Syntax

```
while(condition){  
    //statement(s);  
    // Increment (++) or Decrement (--) Operation;  
}
```

Example

```
152 // While loop  
153 int i = 1;  
154 while (i <= 10) {  
155     print(i);  
156     i++;  
157 }
```

Output:

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

Activity 2

เขียนโค้ดเพื่อเช็คค่าแต่ละ character ของ string เป็นสระ [a, e, i, o, u] หรือ
เปล่า

ถ้าต้องการเช็คค่า มี x อยู่ใน list หรือไม่ให้ใช้ `list.contains(x)`

Function

Functions are the block of code that performs a specific task.
The function helps reusability of the code in the program.

Syntax

```
returntype functionName(parameter1,parameter2, ...){  
    // function body  
}
```

Example

```
182 // Function
183 void add(int num1, int num2) {
184     int sum = num1 + num2;
185     print("The sum is $sum");
186 }
187
188 int add_1(int a, int b) {
189     int sum = a + b;
190     return sum;
191 }
```

```
159 // Call function
160 add(1, 2);
161 int k = add_1(5, 6);
162 print(k);
```

Output:

```
The sum is 3
11
```

Arrow Function

Syntax

The syntax for the dart arrow function.

```
returnType functionName(parameters...) => expression;
```

Example

```
164 // Calling Arrow function
165 print('The sum is ${add_2(9, 5)}');
```

Output:

```
The sum is 14
```

```
193 int add_2(int a, int b) => a + b;
```

List: generate method

ใช้สำหรับ generate a list of values

Syntax การใช้ generate

```
List<E>.generate(  
    int length,  
    E generator(  
        int index  
    ),  
    {bool growable = true}  
)
```

```
167 // List generate  
168 var listA = List<int>.generate(10, (i) => i + 1);  
169 print(listA);
```

Output:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

List: map method

ใช้สำหรับจัดการ list ได้

Example

```
171 // Map method
172 List<int> list = [10, 20, 30, 40, 50];
173 var douledList = list.map((n) => n * 2);
174 print(douledList);
```

Output:

```
(20, 40, 60, 80, 100)
```


List: where method

ใช้สำหรับหาค่าตามที่กำหนด

```
176 // Where method
177 List<int> numbers = [2, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14];
178 List<int> even = numbers.where((number) => number.isEven).toList();
179 print(even);
```

Output:

```
[2, 4, 6, 8, 10, 12, 14]
```

Activity 3

เขียนฟังก์ชันสำหรับแสดงจำนวนคู่ ที่อยู่ในช่วงที่กำหนด

เขียนฟังก์ชันสำหรับหาอักษร a จากข้อมูลใน List ให้ตอบว่ามีหรือไม่มี เช่น
['a', 'b'] คำตอบคือ [true, false]

Week 5: Classroom game

ตอบคำถามตาม link นี้เลย...

<https://forms.gle/ZgrVS16SyVOPYPar7>

Reference

<https://dart-tutorial.com/introduction-and-basics/>