



---

ICT Department

# ភាសាសរសេរកម្មវិធី Dart

**Instructor:** Oum Saokosal, *Master of Engineering in Information Systems, South Korea '2010*

**Email:** [oumsaokosal@gmail.com](mailto:oumsaokosal@gmail.com)

**Phone:** 012 252 752 (Telegram)

- ភាសាសរសេរកម្មវិធី Dart ត្រូវបានបង្កើតឡើងដោយ Google នាឆ្នាំ២០១១ ដែលជា ភាសាសរសេរកម្មវិធីទូទៅ ដែលដំបូងឡើងគោលដៅបង្កើតសំរាប់សរសេរ web, server, desktop ដែលក្រោយមកគេបានយកប្រើជាមួយនឹង mobile គឺជាមួយ Flutter SDK។
- ទំរង់កូដភាគច្រើនរបស់ Dart គឺដូចគ្នាជាមួយនឹង Java គ្រាន់តែ Dart បានកាត់ បន្ថយចំនុចមិនចាំបាច់មួយចំនួន ហើយបញ្ចូលនូវចំនុចល្អផ្សេងៗនៃភាសា C, C++, C#, Javascript ជាដើម។
- បែបបទនៃការសរសេរកូដ Dart គឺរួមផ្សំចូលគ្នា៣៖ OOP, Functional Programming និង Reactive Programming។



# Dart

- **UpperCamelCase:** សំរាប់ classes, enums, typedefs, និង type parameters

- `class SliderMenu { ... }`
- `class HttpRequest { ... }`
- `typedef Predicate<T> = bool Function(T value);`



- **lowerCamelCase:** សំរាប់ Class members, top-level definitions, variables, និង parameters.

- `const pi = 3.14;`
- `const defaultTimeout = 1000;`
- `final urlScheme = RegExp('^([a-z]+):');`



- **lowercase\_with\_underscores:** សំរាប់ libraries និង source files
  - `library peg_parser.source_scanner;`
  - `import 'file_system.dart';`
  - `import 'slider_menu.dart';`
- **ការ Import តាមលំដាប់:** ត្រូវ import Dart មុន lib ផ្សេង
  - `import 'dart:async';`
  - `import 'dart:html';`
  - `import 'package:bar/bar.dart';`
  - `import 'package:foo/foo.dart';`

- ការប្រកាសអញ្ញតិក្នុង dart អាចប្រើ datatype ក៏បាន មិនដាក់ក៏បាន
  - បើមិនចង់កំណត់ datatype ទេ គឺអាចប្រើ var ឬ dynamic
  - ឬអាចកំណត់ datatype ដោយ int, double, num, bool, String

```
main() {  
  int a = 10;  
  var b = 7.5;  
  double c = 1.5;  
  bool d = false;  
  String title = "Cambodia";  
  dynamic subtitle = 'Kingdom of Wonder';  
  print("$title, $subtitle");  
}
```

Cambodia, Kingdom of Wonder

- **dynamic** គឺអាចប្រើសំរាប់អោយ object ទាំងអស់ក្នុងភាសា Dart
- **var** គឺជា keyword សំរាប់ប្រកាសអញ្ញតិណាមួយក៏បាន
- var ខុសពី dynamic គឺត្រង់ពេលដែលអោយតម្លៃដំបូង (initialize) បើយើងអោយតម្លៃដំបូងជាប្រភេទ integer វានឹងជាប់ជា integer រហូត។ តែ dynamic គឺអាចប្តូរប្រភេទពេលក្រោយបាន។

```
void example(){  
  var a = 10;  
  a = 10.5; //error  
  
  dynamic b = 10;  
  b = 10.5; //no error  
}
```

```
main() {  
  var name;  
  name = "Hello Flutter";  
  print(name);  
  name = 123;  
  print(name);  
  name = ["computer", 'mouse', "keyboard"];  
  print(name);  
}
```

```
main() {  
  
    //Number  
    num age = 23;           // 23  
    num pi = 3.14;          // 3.14  
    int year = 2000;        // 2000  
    double haf1 = 0.5;      // 0.5  
  
    //Boolean  
    bool isTrue = true;     // true  
    bool isFalse = false;   // false  
}
```



- final និង const គឺជាប្រើសំរាប់កំណត់មិនអោយអញ្ញតិណាមួយផ្លាស់ប្តូរតំលៃបានដូចគ្នា គ្រាន់តែថា const គឺសំរាប់ប្រកាសអញ្ញតិនៅក្រៅ class ហើយ final សំរាប់បានទាំងក្នុង និងក្រៅ class។
- ចំនុចសំខាន់មួយទៀតរបស់ const គឺការកំណត់តម្លៃអោយថេរ ក្នុងពេល compile។

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

const double earthGravity = 9.807;
final int million = 1000000;

class MyApp extends StatelessWidget {

  const double pi = 3.14; //error
  final double PI = 3.14; //ok

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
```

- + (បូក)
- - (ដក)
- \* (គុណ)
- / (ចែក)
- ~/ (ចែកយកតំលៃជា integer)
- % (ចែករកសំណល់ | Modulo)
- ++ (កើន)
- -- (ថយ)

Operator	Description	Example
>	Greater than	(A > B) is False
<	Lesser than	(A < B) is True
>=	Greater than or equal to	(A >= B) is False
<=	Lesser than or equal to	(A <= B) is True
==	Equality	(A==B) is True
!=	Not equal	(A!=B) is True

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



ICT Department

# កំណត់លក្ខខណ្ឌ ក្នុង Dart

- if Statement
- for Loop
- while , do while
- break & continue
- switch case

# if Statement

```
if (a > b) {  
    print(a);  
} else if (a < b) {  
    print(b)  
} else if (a > 0 && b >= 0) {  
    print(b)  
} else if (a == 0 || c != 0) {  
    print(b)  
} else {  
    print("equal");  
}
```

```
int a = 5;  
bool b = (a > 0 ? true : false);  
String c = (b ? "Hello" : "Hi");  
int d = a > 0 ? 1 : 2;
```



```
var message = "Hello Dart";  
for (var i = 0; i < 5; i++) {  
  print("$i $message");  
}
```

```
var numbers = [0, 1, 2];  
for (var x in numbers) {  
  print(x); // 0 1 2  
}
```

# while & do while loop

```
var order = 1;
while(order<10) {
  print(`${order++}`);
}
```

```
do {
  print(order);
} while (order > 1);
```

# break & continue

```
var order = 1;
while(order<10) {
  if(order == 5) break;    // stop the loop
  print("${order++}");
}
```

```
var order = 1;
while(order<10) {
  if(order == 5) continue; // skip the loop
  print("${order++}");
}
```

```
var order = 1;
switch (order){
  case 1:
    print(order);
    break;
  case 2:
    print(order);
    break;
  default:
    print("None");
}
```