



Full-Stack Service Programming

Lecture 1

Dart 언어의 이해 (기초)

2023. 09. 01

Sungwon Lee
Department of Software Convergence

Welcome

Enjoy Dart & Flutter Life



교재 목차

- Volume.A 시작에 앞서
- Volume.B Dart 언어 기초 이해



TEXT BOOK & SOURCE CODE

소스 코드

- <https://github.com/drsungwon/DART-FLUTTER-BOOK>

The screenshot shows a GitHub repository page for the user drsungwon named DART-FLUTTER-BOOK. The repository is public and has 1 branch and 0 tags. The 'Code' tab is selected. A commit from drsungwon dated May 9 at 62b9a46 is shown, updating the README.md file. The repository has 12 commits in total. The 'About' section includes a description in Korean: '플스택 개발이 쉬워지는 다트 & 플러터 (영진단 캠)'. It also lists the Readme, MIT license, 3 stars, 1 watching, and 1 fork. There is a link to report the repository. The 'Releases' section indicates no releases have been published.

GitHub - drsungwon/DART-FLUTTER-BOOK

Product Solutions Open Source Pricing

Search Sign in Sign up

drsunwon / DART-FLUTTER-BOOK Public

Notifications Fork 1 Star 3

Code Issues Pull requests Actions Projects Security Insights

main 1 branch 0 tags Go to file Code

drsunwon Update README.md 62b9a46 on May 9 12 commits

BOOKTITLE Add files via upload 2 months ago

volume-B-chapter-02 Add files via upload 2 months ago

volume-B-chapter-03 Add files via upload 2 months ago

volume-B-chapter-04 Add files via upload 2 months ago

volume-B-chapter-05 Add files via upload 2 months ago

volume-B-chapter-06 Add files via upload 2 months ago

volume-B-chapter-07 Add files via upload 2 months ago

volume-B-chapter-08 Add files via upload 2 months ago

About

플스택 개발이 쉬워지는 다트 & 플러터 (영진단 캠)

Readme

MIT license

3 stars

1 watching

1 fork

Report repository

Releases

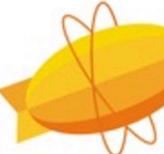
No releases published

Full Stack 개발자에게 Dart 언어를 권하는 이유

- Full-Stack 프로그래머가 될 수 있는 언어
- 구글이 만든 언어
- Adobe가 큰 관심을 갖는 언어
- 비교적 최근에 만들어진 언어
- 정량적인 장점 (수행 시간, 요구 메모리 량, 실행 파일 크기 등)

VOLUME.A CHAPTER.1

Full Stack 개발자에게 Dart 언어를 권하는 이유

UI/UX	Mobile Application (Front-End)	Server Application (HTTP based Back-End)	Desktop (with GUI)	Web
	2014 Swift with Cocoa (iOS)	Swift (MacOS, Linux)	Swift with Cocoa (MacOS)	
Adobe XD   	2011 Dart with Flutter (iOS, Android)	Dart (MacOS, Windows, Linux)	Dart with Flutter (MacOS)	Dart with Flutter
	2009 Go with gomobile (iOS, Android)	Go (MacOS, Windows, Linux)	Go with {plugins} (Case by Case)	
	2000 C# with Unity/Xamarin (iOS, Android)	C# with .Net (Windows)	C# with Unity/XamarinNet (MacOS, Windows)	
	1995 Javascript with jQuery/React/Cordova/Ionic (iOS, Android)	Javascript with Node.js (MacOS, Windows, Linux)	Javascript with Electron (MacOS, Windows, Linux)	Javascript with jQuery/React/Vue.js
	1995 Java with Android SDK (Android)	Java (MacOS, Windows, Linux)	Java with Swing (MacOS, Windows, Linux)	
	1991 Python with kivy (iOS, Android)	Python (MacOS, Windows, Linux)	Python with Tk/PyQt (MacOS, Windows, Linux)	
	1984 Objective-C with Cocoa (iOS)	Objective-C (MacOS, Linux)	Objective-C with Cocoa (MacOS)	
	1983 C++ with MS CrossPlatform (iOS, Android)	C++ with {3rd-party libraries} (MacOS, Windows, Linux)	C++ with Cocoa/MFC/QT/GTK (MacOS, Windows, Linux)	

앞으로 자주 방문하게 될 웹 사이트 소개

- Dart (<https://dart.dev/>)
- Flutter (<https://flutter.dev/>)
- DartPad (<https://dartpad.dev/?>)
- Visual Studio Code (<https://code.visualstudio.com/>)
- Dart Code (<https://dartcode.org/>)
- pub.dev (<https://pub.dev/>)
- Flutter YouTube (<https://www.youtube.com/flutterdev>)
- Google YouTube (<https://www.youtube.com/googlecode>)
- Dart GitHub (<https://github.com/dart-lang>)
- Flutter GitHub (<https://github.com/flutter>)
- Adobe XD (<https://helpx.adobe.com/kr/support/xd.html>)
- XD to Flutter (<https://github.com/AdobeXD/xd-to-flutter-plugin>)
- Medium (<https://medium.com/>)

DartPad를 이용하여 개발 환경 구축하기

- DartPad 사이트 접속 (<https://dartpad.dev/?>)
- 기본 제공 샘플 선택 및 실행

The screenshot shows a browser window with the DartPad website open at <https://dartpad.dev/?id=c0f7c578204d61e08ec0fbc4d63456cd>. The code editor contains a Dart script that prints 'hello' four times. The console output shows the results of the execution.

```
1 // Copyright 2015 the Dart project authors. All rights reserved.
2 // Use of this source code is governed by a BSD-style license
3 // that can be found in the LICENSE file.
4
5 void main() {
6   for (var i = 0; i < 4; i++) {
7     print('hello $i');
8   }
9 }
10
```

Console output:

```
hello 0
hello 1
hello 2
hello 3
```

The 'Samples' button in the top right corner is highlighted with a red box.

Hello World 프로그램 개발하기

● Hello, World 타이핑 후 실행

The screenshot shows a web-based code editor, DartPad, running on a Mac OS X browser window. The URL is <https://dartpad.dev/?id=c0f7c578204d61e08ec0fbc4d63456cd>. The code in the editor is:

```
1 void main() {  
2     print('Hello, World');  
3 }  
4
```

A red box highlights the code area. Below the code is a 'Run' button. To the right is a 'Console' window with a red box around it, displaying the output:

Hello, World

At the bottom of the page, there are links for 'Privacy notice', 'Send feedback', 'stable channel', 'no issues', 'Based on Flutter 3.10.5 Dart SDK 3.0.5', and a page number '9'.

기초적인 숫자와 문자 다루기

문법	의미	주요 특징
int	정수	-9,007,199,254,740,992 ~ 9,007,199,254,740,992
double	실수	최대 1.7976931348623157e+308
num	숫자	정수 혹은 실수
String	문자열	글자, 단어 혹은 문장
var	변수	정수, 실수, 문자열 등을 저장하며, 값의 변경 가능 (한번 값을 저장하고 나면, 같은 타입의 값을 저장해야 함)
dynamic	변수	정수, 실수, 문자열 등을 저장하며, 값의 변경 가능 (저장하는 값의 타입에 제한 없음)
constant	상수	처음 만드는 시점에 값을 설정하며, 값의 변경 불가능

[표 1] Dart 언어의 대표적인 변수와 상수 타입

기초적인 숫자와 문자 다루기 (리뷰 & 실습)

● volume-B-chapter-03.dart

ONE MORE : final

- final 은 상수 생성시 초기 값을 지정하지 않고, 추후 지정함
- const는 상수 생성시 초기 값을 지정함

숫자 자세히 다루기

연산자	의미	기능
+	더하기	두 값을 더함
-	빼기	왼쪽 값에서 오른쪽 값을 뺌
*	곱하기	두 값을 곱함
/	나누기(division)	왼쪽 값을 오른쪽 값으로 나눔
~/	나누기(division)	왼쪽 값을 오른쪽 값으로 나눈 몫을 정수로 계산
%	나누기(modulus)	왼쪽 값을 오른쪽 값으로 나눈 나머지를 계산
++	더하기	연산자가 적용된 값을 1만큼 증가
--	빼기	연산자가 적용된 값을 1만큼 감소
+=	더하기	연산자 왼쪽의 값에 연산자 오른쪽의 값을 더함
-=	빼기	연산자 왼쪽의 값에서 연산자 오른쪽의 값을 뺌
*=	곱하기	연산자 왼쪽의 값에 연산자 오른쪽의 값을 곱함
/=	나누기	연산자 왼쪽의 값을 연산자 오른쪽의 값으로 나눔

[표 1] Dart 언어의 대표적인 연산자

숫자 자세히 다루기 (리뷰 & 실습)

- volume-B-chapter-04.dart

문자 자세히 다루기 (리뷰 & 실습)

- volume-B-chapter-05.dart

비교 연산자 이해하기

연산자	수학기호	의미
<code>==</code>	$=$	두 값이 같으면 true, 다르면 false를 계산함
<code>!=</code>	\neq	두 값이 다르면 true, 같으면 false를 계산함
<code>>=</code>	\geq	왼쪽 값이 오른쪽 값보다 크거나 같으면 true, 반대면 false를 계산함
<code><=</code>	\leq	왼쪽 값이 오른쪽 값보다 작거나 같으면 true, 반대면 false를 계산함
<code>></code>	$>$	왼쪽 값이 오른쪽 값보다 크면 true, 반대면 false를 계산함
<code><</code>	$<$	왼쪽 값이 오른쪽 값보다 작으면 true, 반대면 false를 계산함
<code>&&</code>	AND	두 값이 모두 true이면 true, 아니면 false를 계산함
<code> </code>	OR	두 값 중 하나 이상이 true이면 true, 아니면 false를 계산함

[표 1] Dart 언어의 대표적인 비교 연산자

조건문 이해하기 (if-else, switch)

```
if (조건문A) {  
    작업1  
} else if (조건문B) {  
    작업2  
} else if (조건문C) {  
    작업3  
} else {  
    작업4  
}
```

```
switch(조건문) {  
    case 조건A:  
        작업1  
        break;  
    case 조건B:  
    case 조건C:  
        작업2  
        break;  
    default:  
        작업3  
        break;  
}
```

조건문 이해하기 (리뷰 & 실습)

● volume-B-chapter-06.dart

ONE MORE : bool Class

- true 혹은 false를 저장하는 Class

ONE MORE : assert

- assert(true) : 프로그램의 수행에 영향을 주지 않음
- assert(false) : 프로그램의 수행을 중지함 (동작 오류 확인시 활용함)

반복문으로 작업하기

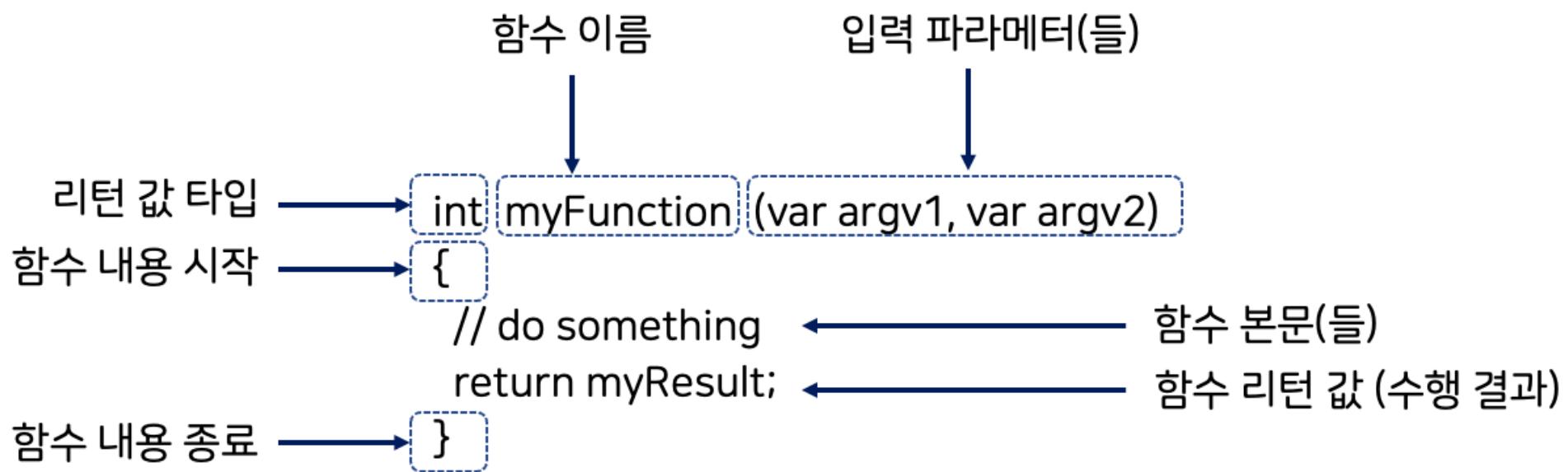
- for, while, do-while
- break, continue

반복문으로 작업하기 (리뷰 & 실습)

- volume-B-chapter-07.dart

함수를 이용하여 반복 작업하기 (리뷰 & 실습)

● C/C++ 언어의 형태와 유사함



함수를 이용하여 반복 작업하기 (리뷰 & 실습)

● volume-B-chapter-08.dart

ONE MORE : default & named Parameters

- default : 함수 호출시 입력 파라메타를 주지 않으면, 초기 값으로 대체함
- named : 함수 호출시 입력 파라메타의 이름과 이의 값을 적는 형태로 실행함

Class를 이용한 OOP 이해하기

클래스	메서드	의미
int	abs	절대 값을 계산함
int	gcd	최대 공약수를 계산함
int	toString	문자열로 변환함
double	abs	절대 값을 계산함
double	toString	문자열로 변환함
double	floor	소수점 이하를 버리고 가까운 정수를 계산함
double	round	가까운 정수를 계산함
String	toLowerCase	소문자로 변환함
String	toUpperCase	대문자로 변환함
String	length	길이를 계산함
String	substring	문자열의 일부를 추출해서 새로운 문자열을 생성함
String	[]	문자열의 특정 글자를 지칭함
bool	toString	문자열로 변환함
bool	is	왼쪽의 변수/상수가 오른쪽의 타입인지 검사함
bool	is!	왼쪽의 변수/상수가 오른쪽의 타입이 아닌지 검사함

[표 1] 소스 코드에서 다루는 클래스들과 클래스의 함수(메서드)들

Class를 이용한 OOP 이해하기 (공식 사이트)

The screenshot shows a browser window displaying the Dart API documentation for the `int` class. The URL in the address bar is `https://api.dart.dev/stable/3.0.5/dart-core/int-class.html`. The page title is "int class - dart:core library - Dart". The left sidebar lists various Dart classes, with "int class" highlighted. The main content area shows the `int` class definition, which is abstract and final. It describes an integer number and its 64-bit two's complement implementation. A note explains that when compiling to JavaScript, integers are restricted to values representable as double-precision floating point numbers. The right sidebar lists properties like `bitLength`, `hashCode`, and `isEven`.

int class abstract final

An integer number.

The default implementation of `int` is 64-bit two's complement integers with operations that wrap to that range on overflow.

Note: When compiling to JavaScript, integers are restricted to values that can be represented exactly by double-precision floating point values. The available integer values include all integers between -2^{53} and 2^{53} , and some integers with larger magnitude. That includes some integers larger than 2^{63} . The behavior of the operators and methods in the `int` class therefore sometimes differs between the Dart VM and Dart code compiled to JavaScript. For example, the bitwise operators truncate their operands to 32-bit integers when compiled to JavaScript.

Classes cannot extend, implement, or mix in `int`.

See also:

- `num` the super class for `int`.

CONSTRUCTORS

- `fromEnvironment`

PROPERTIES

- `bitLength`
- `hashCode`
- `isEven`
- `isFinite`
- `isInfinite`
- `isNaN`
- `isNegative`
- `isOdd`
- `runtimeType`
- `sign`

METHODS

- volume-B-chapter-09.dart

복수 데이터 처리하기 (List 클래스)

기능	의미
length	리스트에 속한 항목의 개수를 계산
first	리스트에 속한 첫번째 항목을 확인
last	리스트에 속한 마지막 항목을 확인
indexOf	리스트에 속한 특정 위치의 항목을 확인
insert	리스트의 특정 위치에 새로운 항목을 추가
add	리스트의 끝에 새로운 항목을 추가
addAll	리스트의 끝에 (전달받은 리스트의) 모든 항목을 추가
sort	리스트의 항목을 순서대로 정렬
clear	리스트의 모든 항목을 삭제

[표 1] 소스 코드에서 다루는 List의 기능들

복수 데이터 처리하기 (리뷰 & 실습)

- volume-B-chapter-10.dart

집합 데이터 처리하기 (Set 클래스)

기능	의미
add	집합에 새로운 항목을 추가
addAll	전달받은 집합의 모든 항목을 새로운 항목으로 추가
contains	집합에 특정 항목이 있는지 확인
intersection	두 집합의 교집합을 계산
difference	두 집합의 차집합을 계산
union	두 집합의 합집합을 계산
remove	집합에서 특정 항목을 삭제
runtimeType	클래스 객체의 타입을 알려줌

[표 1] 소스 코드에서 다루는 Set의 기능들

- volume-B-chapter-11.dart

사전 데이터 처리하기 (Map 클래스)

기능	의미
length	Map에 포함된 항목의 개수를 계산
runtimeType	Map 클래스의 타입을 확인
isNotEmpty	Map에 포함된 항목이 비어 있지 않음을 확인
isEmpty	Map에 포함된 항목이 비어 있음을 확인
addAll	Map에 전달받은 항목(들)을 추가
containsKey	Map에 포함된 항목에 특정 key가 포함되어 있는지 확인
containsValue	Map에 포함된 항목에 특정 value가 포함되어 있는지 확인
remove	Map에서 특정 항목을 삭제
clear	Map에 포함된 모든 항목을 삭제

[표 1] 소스 코드에서 다루는 Map의 기능들

- volume-B-chapter-12.dart

요긴한 Dart 언어 기능

- Unicode (유니코드)
- Cascade Operators (캐스캐이드 연산자)
- forEach 메서드
- Nested Function (중첩 함수)
- Conditional Expression (조건적 표현)
- Bitwise Operators (비트 처리 연산자)
- Hexa-Decimal Presentation (16진법 표현)
- Exponential Presentation (지수 표현)
- String-to-Number Conversion (문자열을 숫자로 변환)
- Enumerator (나열형 데이터)

Null-Safety (Null에 대한 프로그램 안정성 보장)

- Dart 언어는 Null-Safety를 기본으로 함
 - Dart 언어의 변수들은 기본적으로 null 값을 가질 수 없음
 - 변수들은 사용하기 전에 반드시 어떤 값을 가지고 있어야 함
- 아래에서, 102/103번 줄의 주석 처리 제거 후 실행시 에러 발생함

```
97  print("\n[11] Null Safety");  
98  
99  int iTemp = 3;  
100 print(iTemp);  
101 int iNonNullableInt;  
102 //print(iNonNullableInt);  
103 //iTemp = iNonNullableInt;  
104
```

```
Error: Non-nullable variable 'iNonNullableInt' must be assigned before it can  
be used.
```

```
    print(iNonNullableInt);
```

```
Error: Non-nullable variable 'iNonNullableInt' must be assigned before it can  
be used.
```

```
    iTemp = iNonNullableInt;
```

Null-Safety (Null에 대한 프로그램 안정성 보장)

- Dart 언어에서 강제로 null 값을 갖는 변수를 정의할 수 있음
 - ◆ int? i; 경우, i는 **int 객체 혹은 null**을 저장할 수 있음
 - null은 “의미없는 값을 저장하는 **객체**”로 정의함
 - ◆ Dart 언어의 표준 함수/클래스는 대부분 Null-Safety 함
- 아래에서, 108번 줄의 주석 처리 제거 후 실행시 에러 발생함
 - ◆ 소스 코드에서 iNullableInt는 null을 저장하고 있음
 - ◆ null은 abs() 메소드를 지원하지 않음

```

105  int? iNullableInt;
106  iNullableInt = null;
107  print(iNullableInt);
108  //print(iNullableInt.abs());
109

```

Error: Method 'abs' cannot be called on 'int?' because it is potentially null.
 print(iNullableInt.abs());

Null-Safety (Null에 대한 프로그램 안정성 보장)

- runtimeType 메소드 (null 객체가 지원하는 유일한 메소드)
 - 주어진 변수/객체가 null인 경우 null을 리턴함
 - 주어진 변수/객체가 null이 아니면, 해당하는 타입을 리턴함
- A ?? B
 - 만약 A가 null이 아니면, A가 가지고 있는 값으로 계산함
 - 만약 A가 null이면, B가 가지고 있는 값으로 계산함
- A = (B ??= C)
 - 만약 B가 null이 아니면, A를 B로 설정함
 - 만약 B가 null 이면, A를 C로 설정함
- '?.' 메소드 호출 연산자
 - 객체가 null이 아니면, 문제없이 메서드/프로퍼티를 접근함
 - 객체가 null이라면 '?' 문법으로 호출한 메서드/프로퍼티는 Dart 언어가 null로 처리함

- volume-B-chapter-13.dart



Thank you