ОНЛАЙН-ОБРАЗОВАНИЕ

Flutter Mobile Developer

Тема 1. Dart. Основы.

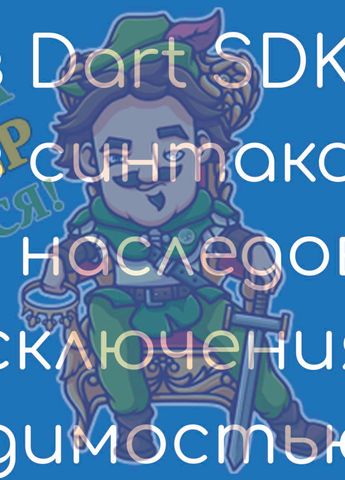
Меня хорошо видно && слышно?

Ставьте + , если все хорошо
Напишите в чат, если есть проблемы

Тема 1. Dart. Основы.

Цель урока

После занятия вы сможете:



- Разбираться в Dart SDK & Tools
- Разбираться в синтаксисе Dart Language
- Использовать наследование в Dart.
- Работать с исключениями.
- Управлять видимостью классов и методов внутри проекта.
- Писать простой синхронный код с помощью Dart.
- Использовать сторонние библиотеки в проекте.

О чем будем говорить

- История возникновения языка и текущее состояние
- Устанавливаем Dart SDK || Flutter
- Знакомимся с Dart Language
- Знакомимся с Dart SDK
- Tooling
- Практика

Dart

10–12 октября 2011 года - Dash. Проект был основан Ларсом Баком и Каспером Лундом.

- Создать структурированный и в то же время гибкий язык для веб-программирования
- Сделать язык похожим на существующие для упрощения обучения
- Высокая производительность получаемых программ как в браузерах, так и в иных окружениях, начиная от смартфонов и заканчивая серверами.
- Изначально было предложено два способа исполнения Dart-программ: с использованием виртуальной машины (в поддерживающих язык браузерах) или с промежуточной трансляцией в javascript


Dart

Dart

15 ноября 2013 года - Dart SDK 1.0

4 июля 2014 года ECMA-408

Dart

В августе 2018 года был выпущен Dart 2.0 с изменениями языка, включая Sound Type System.

https://dart.dev/dart-2

https://medium.com/dartlang/announcing-dart-2-80ba01f43b6

Changelog

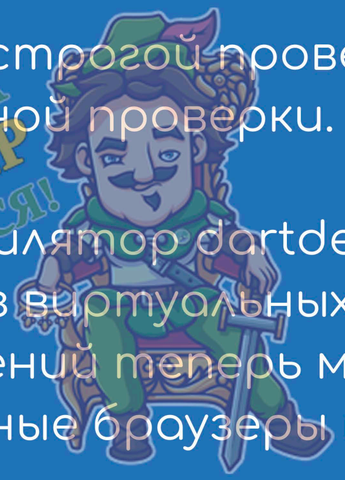
https://github.com/dart-lang/sdk/blob/master/CHANGELOG.md
Первая запись
https://github.com/dart-lang/sdk/blob/master/CHANGELOG.md#170---2014-10-15
- -особо язык не развивался несколько лет

Dart

В августе 2018 года был выпущен Dart 2.0 с изменениями языка, включая Sound Type System.

В Dart 2 режим «**strong**» для строгой проверки типов заменил режим «**checked**» для их ограниченной проверки.

Строгая типизация и компилятор dartdevc обеспечивают быструю компиляцию в JavaScript без виртуальных машин.
Для тестирования приложений теперь можно использовать компиляцию в JavaScript и любые штатные браузеры вместо Dartium.



Reboot of the language to embrace our vision of Dart: **as a language uniquely optimized for client-side development for web and mobile.**

Dart

В августе 2018 года был выпущен Dart 2.0 с изменениями языка, включая Sound Type System.

Flutter

Состояние на 27 feb 2018 - первая beta, dart 2 (Mobile World Congress in Barcelona in February.)
https://medium.com/flutter/announcing-flutter-beta-1-build-beautiful-native-apps-dc142aea74c0
Flutter's beta also works with a pre-release of Dart 2, with improved support for declaring UI in code with minimal language ceremony. For example, Dart 2 infers new and const to remove boilerplate when buildi UI.

Dart

В августе 2018 года был выпущен Dart 2.0 с изменениями языка, включая Sound Type System.

Flutter

Apr 9 2018
https://medium.com/flutter/https-medium-com-flutter-io-announcing-flutters-beta-2-c85ba1557d5e
Dart 2 enabled by default
Our first beta offered a preview of the Dart 2 programming language. Our testing has shown Dart 2 to be near complete, and very stable. Flutter's second beta has Dart 2 enabled by default.

Dart

В августе 2018 года был выпущен Dart 2.0 с изменениями языка, включая Sound Type System.

Flutter

4 Dec 2018 - flutter 1.0 release

Dart 2.6 представил новое расширение, dart2native. Эта функция расширяет встроенную компиляцию для настольных платформ Linux, macOS и Windows.

Dart

В августе 2018 года был выпущен Dart 2.0 с изменениями языка, включая Sound Type System.

Есть четыре способа запустить код Dart:

Dart

В августе 2018 года был выпущен Dart 2.0 с изменениями языка, включая Sound Type System.

1

JavaScript for Dart For Web applications

При запуске кода Dart в веб-браузере код предварительно компилируется в JavaScript с помощью компилятора dart2js.

Скомпилированный как JavaScript, код Dart совместим со всеми основными браузерами.

Благодаря оптимизации скомпилированного вывода JavaScript во избежание дорогостоящих проверок и операций, код, написанный на Dart, в некоторых случаях может работать даже быстрее, чем эквивалентный код, написанный вручную с использованием идиом JavaScript.

Dart

В августе 2018 года был выпущен Dart 2.0 с изменениями языка, включая Sound Type System.

2

DartVM

Комплект разработки программного обеспечения (SDK) Dart поставляется с автономной виртуальной машиной Dart, позволяющей запускать код Dart в среде интерфейса командной строки .

Поскольку языковые инструменты, включенные в Dart SDK, в основном написаны на Dart, виртуальная машина Dart является важной частью SDK.

Эти инструменты включают компилятор dart2js и менеджер пакетов pub.
Dart поставляется с полной стандартной библиотекой, позволяющей пользователям писать полностью рабочие системные приложения, такие как пользовательские веб-серверы.

Dart

В августе 2018 года был выпущен Dart 2.0 с изменениями языка, включая Sound Type System.

3 Ahead-Of-Time compilation

Код Dart может быть AOT-скомпилирован в машинный код (собственные наборы инструкций).

Приложения, созданные с помощью Flutter, SDK мобильного приложения, созданного с помощью Dart, развертываются в аппсторах в виде кода Dart, скомпилированного AOT.

Dart

В августе 2018 года был выпущен Dart 2.0 с изменениями языка, включая Sound Type System.

4

Native

Dart 2.6 с компилятором dart2native для компиляции в автономный собственный код исполняемых файлов.

До Dart 2.6 эта функция предоставляла эту возможность только на мобильных устройствах iOS и Android через Flutter.

Сейчас поддерживается компиляция в Linux, Mac OSX & Windows

Dart

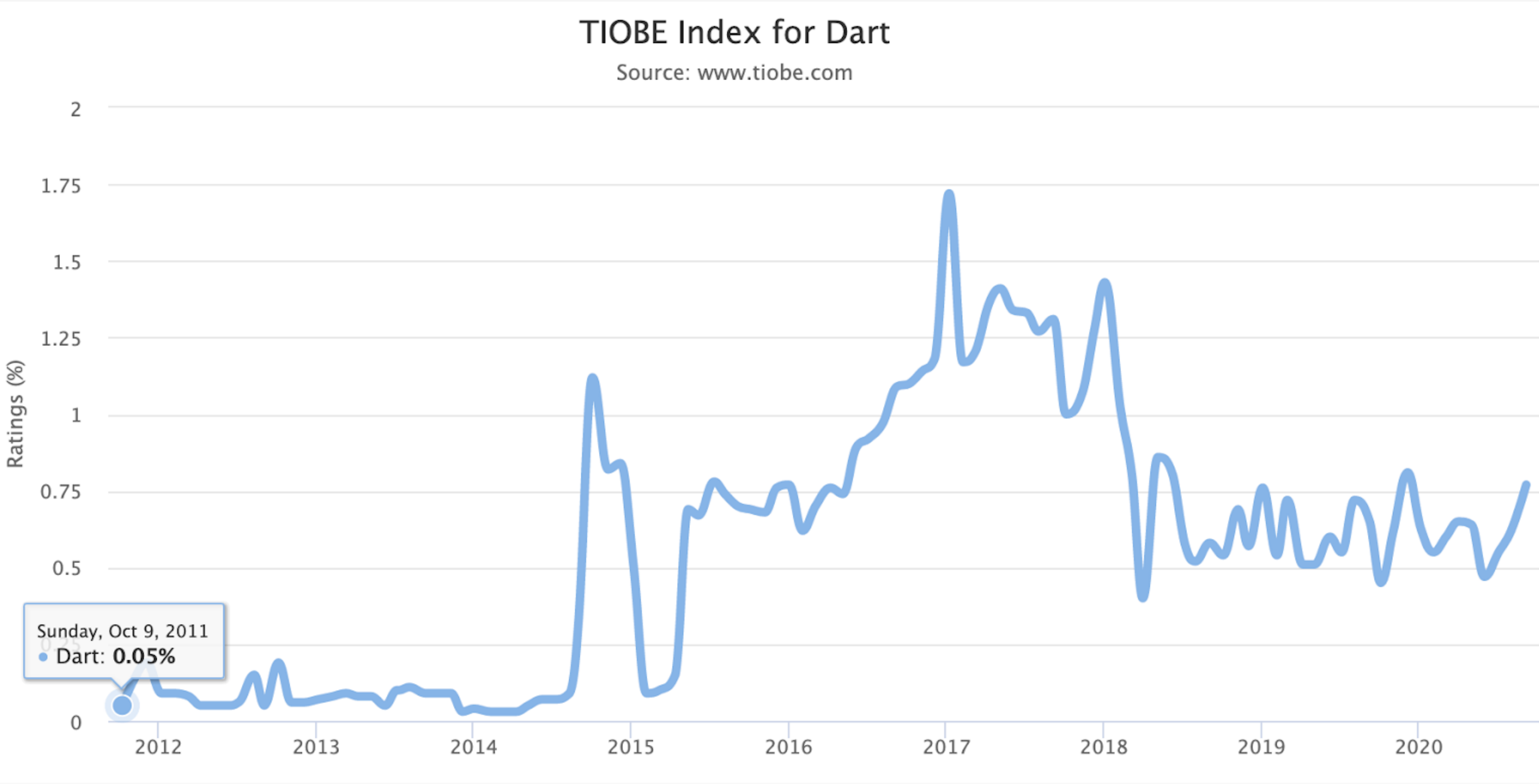
Текущее состояние

https://www.tiobe.com/tiobe-index/

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the best programming language or the language in which most lines of code have been written.
The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system. The definition of the TIOBE index can be found here.

Sep 2020	Sep 2019	Change	Programming Language	Ratings	Change
1	2	↑	C	15.95%	+0.74%
2	1	↓	Java	13.48%	-3.18%
3	3		Python	10.47%	+0.59%
4	4		C++	7.11%	+1.48%
5	5		C#	4.58%	+1.18%
6	6		Visual Basic	4.12%	+0.83%
7	7		JavaScript	2.54%	+0.41%
8	9	↑	PHP	2.49%	+0.62%
9	19	⇈	R	2.37%	+1.33%
10	8	↓	SQL	1.76%	-0.19%
11	14	↑	Go	1.46%	+0.24%
12	16	⇈	Swift	1.38%	+0.28%
13	20	↑	Perl	1.30%	+0.26%

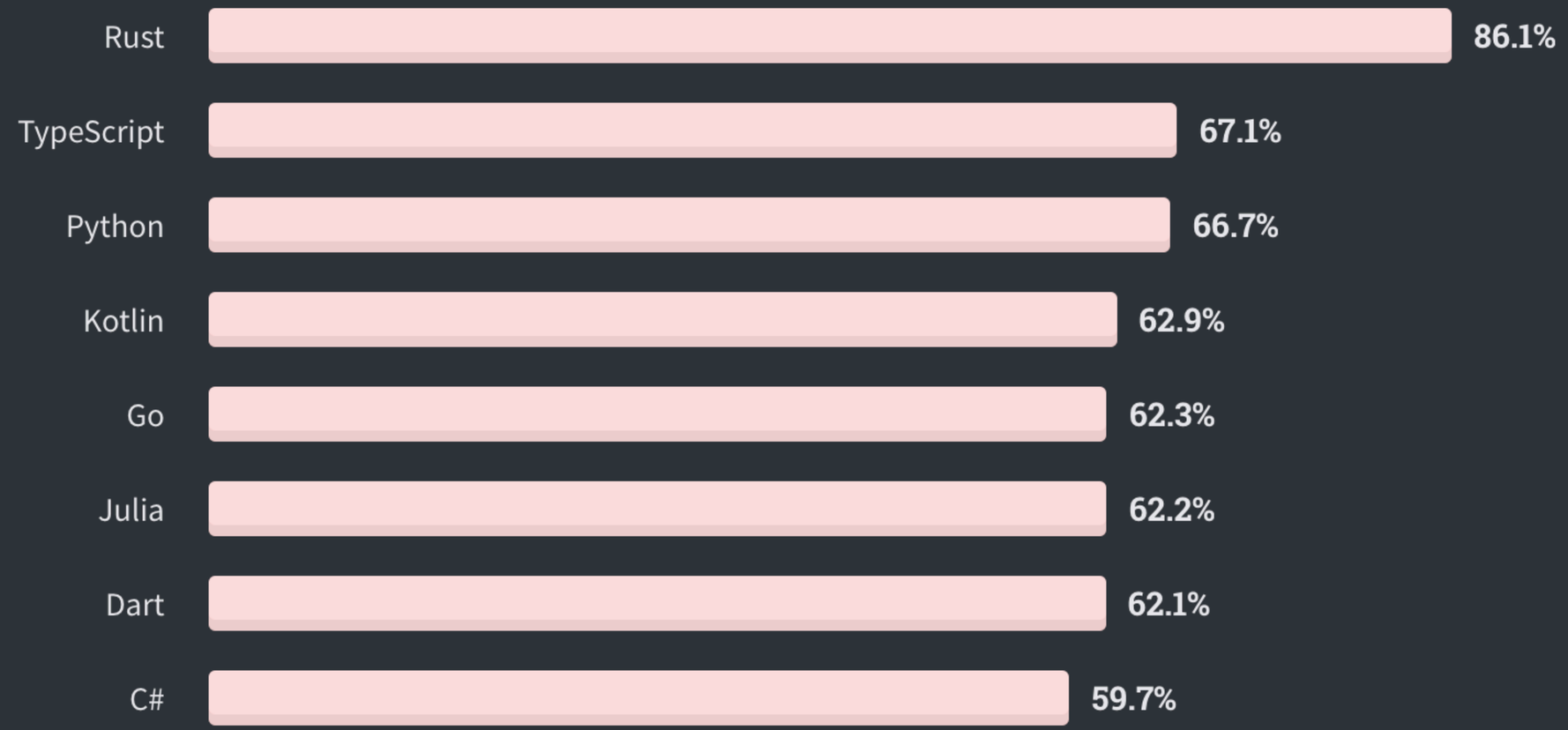
13	20	⇈	Perl	1.30%	+0.26%
14	12	⌄	Assembly language	1.30%	-0.08%
15	15		Ruby	1.24%	+0.03%
16	18	⌃	MATLAB	1.10%	+0.04%
17	11	⇊	Groovy	0.99%	-0.52%
18	33	⇈	Rust	0.92%	+0.55%
19	10	⇊	Objective-C	0.85%	-0.99%
20	**24**	⇈	**Dart**	**0.77%**	**+0.13%**


TIOBE Index for Dart
Source: www.tiobe.com
Ratings (%)
2
1.75
1.5
1.25
1
0.75
0.5
0
2012
2013
2014
2015
2016
2017
2018
2019
2020
Sunday, Oct 9, 2011
Dart: 0.05%

https://insights.stackoverflow.com/survey/2020#technology-most-loved-dreaded-and-wanted-languages-loved

Loved Dreaded Wanted

% of developers who are developing with the language or technology and have expressed interest in continuing to develop with it



Dart

Текущее состояние

Из последних значимых обновлений (July 2020):
Null Safety

https://dart.dev/null-safety
https://dart.dev/null-safety/understanding-null-safety

https://nullsafety.dartpad.dev/ - песочница

Dart

Текущее состояние

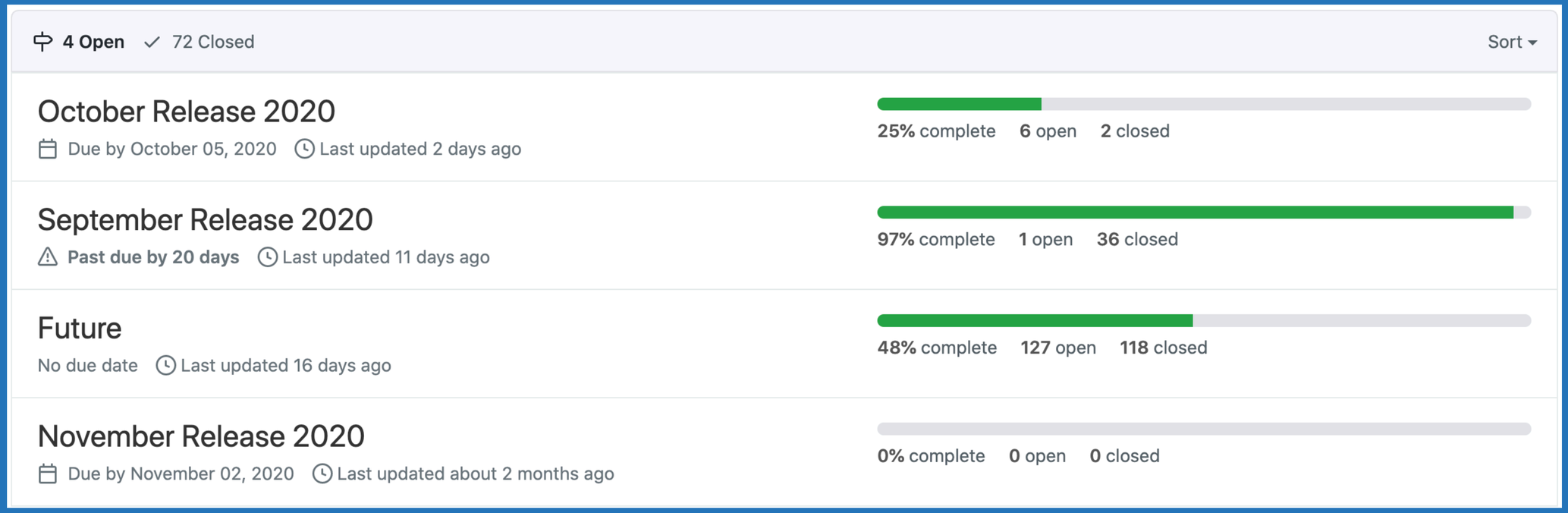
A Dart Language Guide for C# and Java Developers

https://www.toptal.com/dart/dartlang-guide-for-csharp-java-devs

Dart

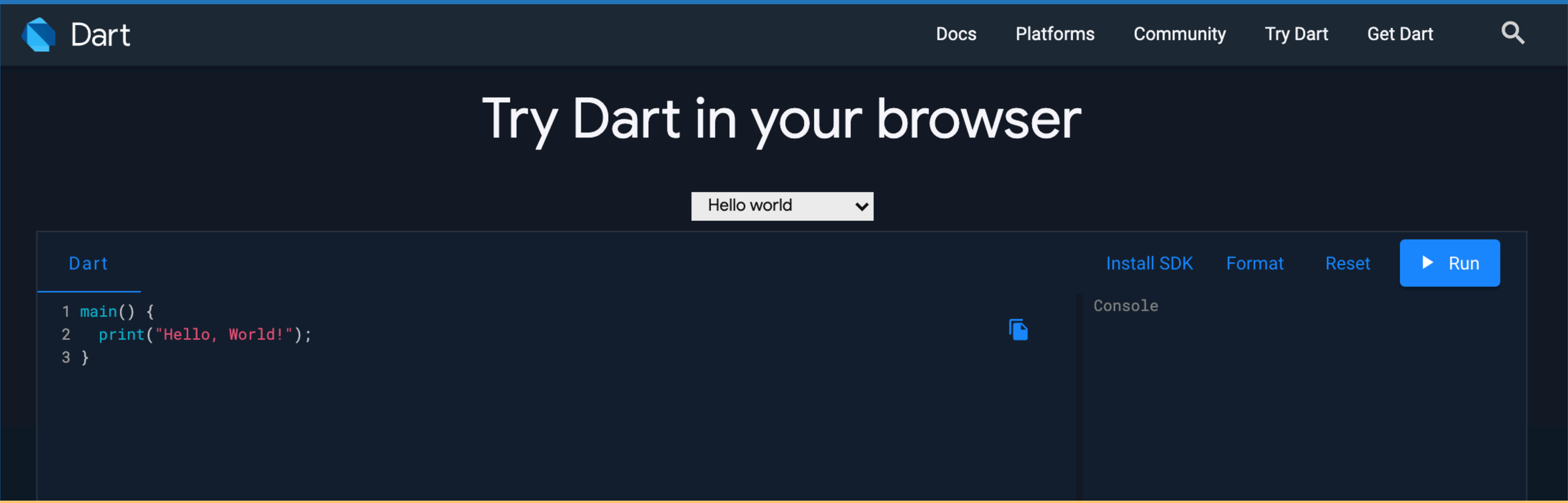
Развитие языка на ближайшее будущее

https://github.com/dart-lang/sdk/milestones



Dart

https://dart.dev/#try-dart - Try Dart in your browser



О чем будем говорить

История возникновения языка и текущее состояние

Устанавливаем Dart SDK || Flutter

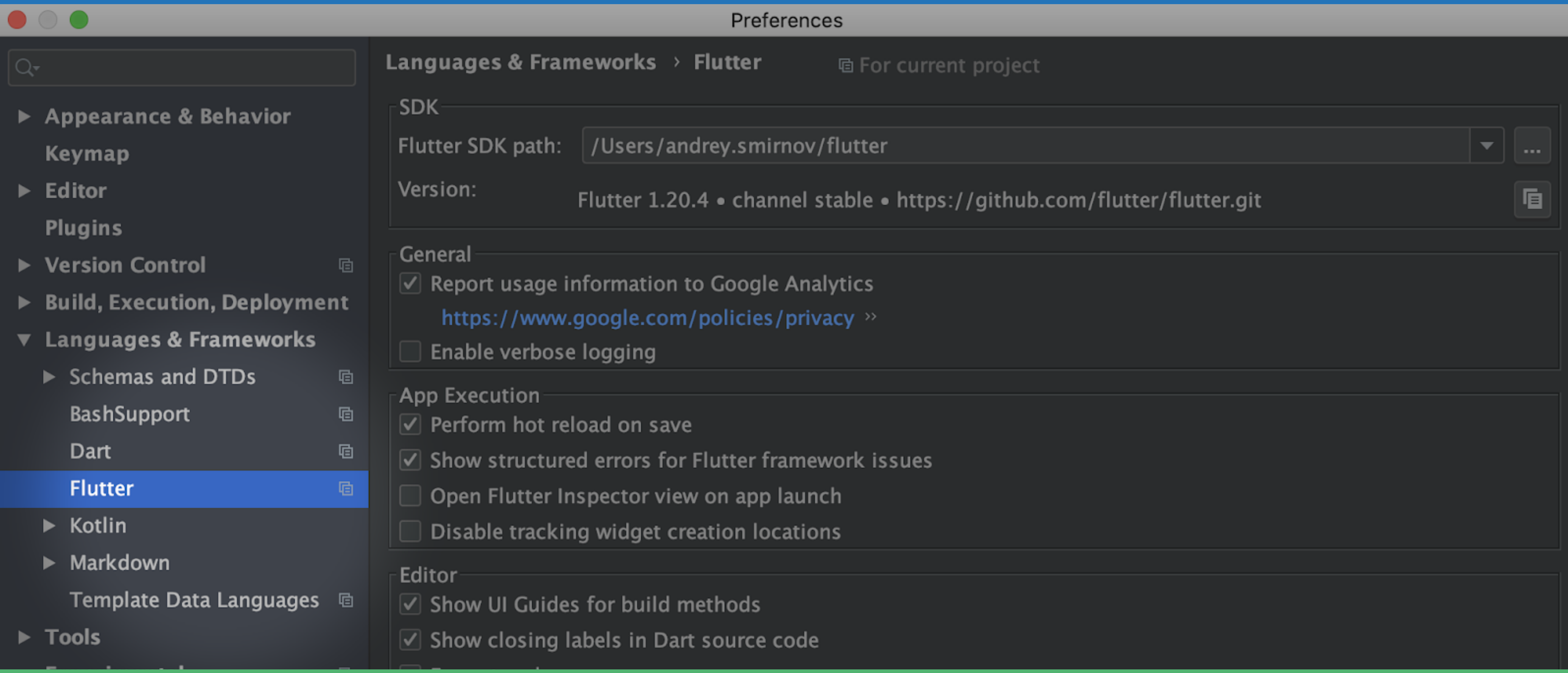
Знакомимся с Dart Language

Знакомимся с Dart SDK

Tooling

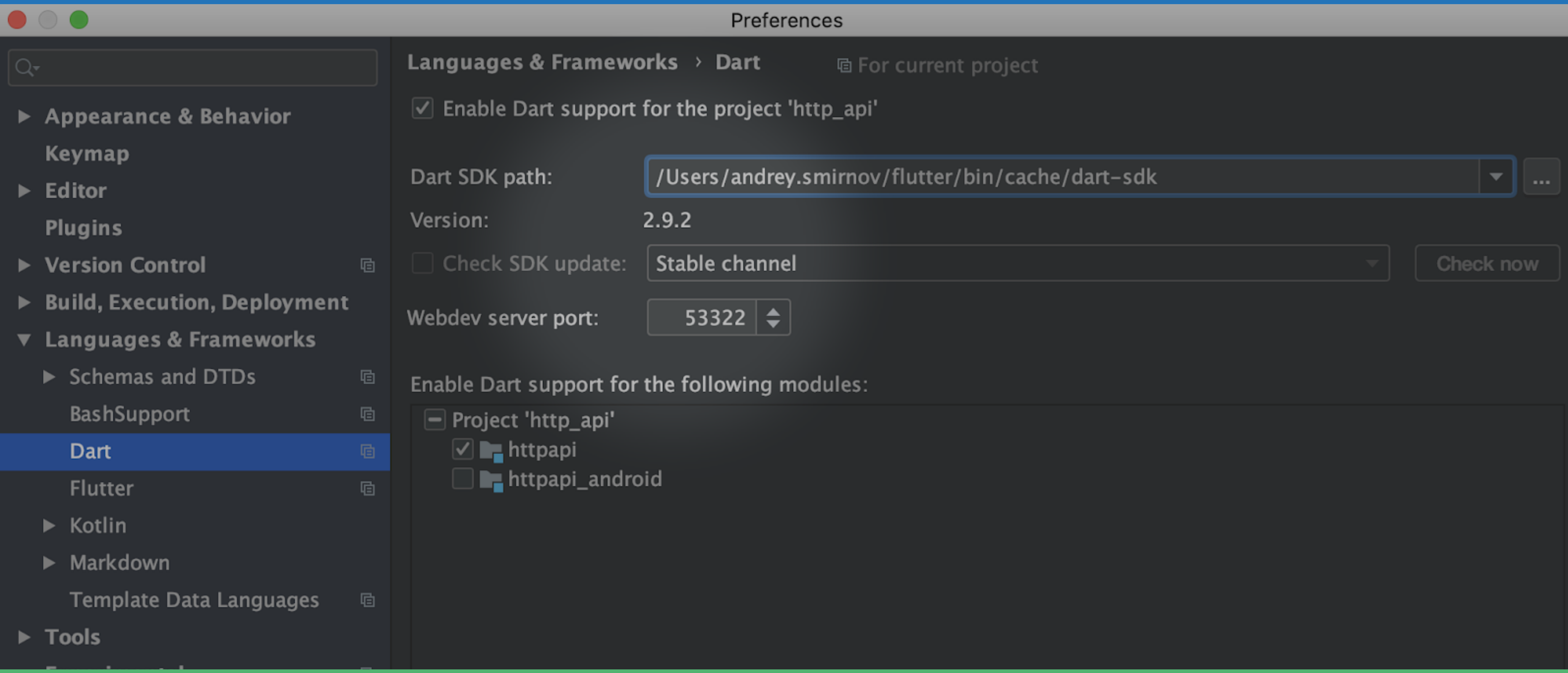
Практика

Flutter



Устанавливаем DartSDK

Flutter



Устанавливаем DartSDK

Dart

Проверим настроенное окружение прямо сейчас в командной строке:

```
❯ dart --version
Dart SDK version: 2.9.2 (stable) (Wed Aug 26 12:44:28 2020 +0200) on "macos_x64"
❯
❯
❯
❯
❯ where dart
/usr/local/bin/dart
/Users/andrey.smirnov/flutter/bin/dart
❯
```

Dart

https://dart.dev/get-dart

Get the Dart SDK

This page describes how to download the Dart SDK. The Dart SDK has the libraries and command-line tools that you need to develop Dart command-line, server, and non-Flutter web apps. For details, see the Dart SDK overview.

As of Flutter 1.21, the Flutter SDK includes the full Dart SDK. So if you have Flutter installed, you might not need to explicitly download the Dart SDK. Consider downloading the Dart SDK if any of the following are true:

- You don't use Flutter.
- You use a pre-1.21 version of Flutter.
- You want to reduce disk space requirements or download time, and your use case doesn't require Flutter. For example, you might have a

Устанавливаем DartSDK

О чем будем говорить

История возникновения языка и текущее состояние

Устанавливаем Dart SDK || Flutter

Знакомимся с Dart Language

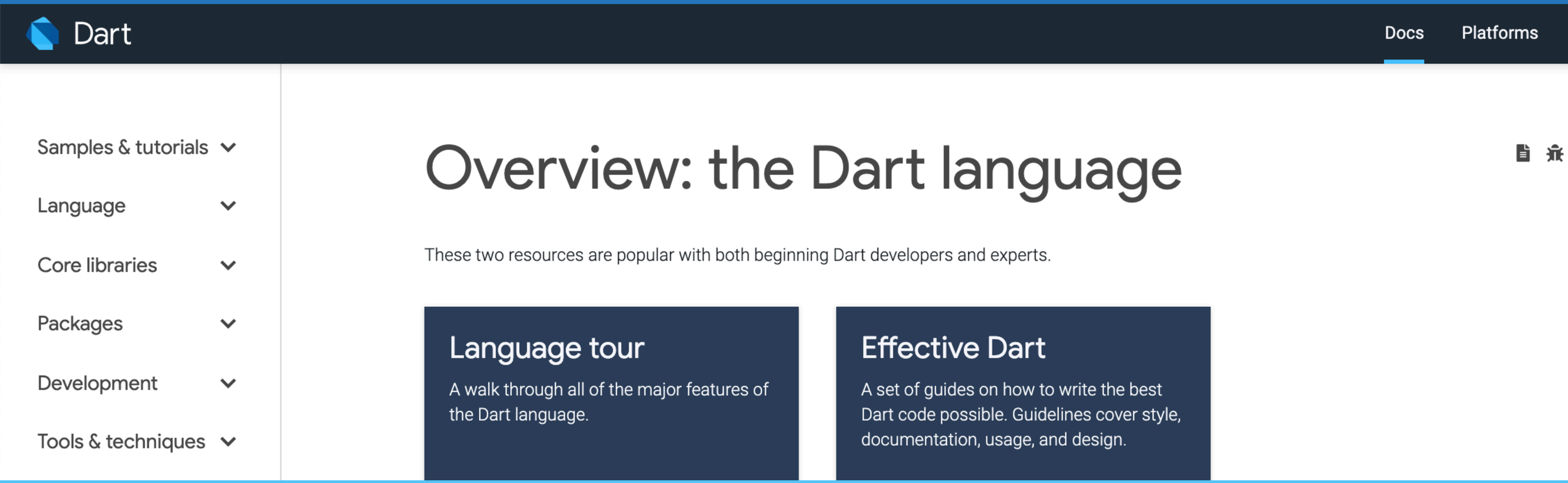
Знакомимся с Dart SDK

Tooling

Практика

Dart

https://dart.dev/guides/language



Dart

https://dart.dev/samples

Language samples

This collection is not exhaustive—it's just a brief introduction to the language for people who like to learn by example. You might also want to check out the language and library tours, or the Dart cheatsheet codelab.

Hello World

Every app has a `main()` function. To display text on the console, you can use the top-level `print()` function:

```
void main() {
```

Dart

https://dart.dev/codelabs/dart-cheatsheet

Dart cheatsheet codelab

The Dart language is designed to be easy to learn for coders coming from other languages, but it has a few unique features. This codelab — which is based on a Dart language cheatsheet written by and for Google engineers — walks you through the most important of these language features.

The embedded editors in this codelab have partially completed code snippets. You can use these editors to test your knowledge by completing the code and clicking the **Run** button. If you need help, click the **Hint** button. To run the code formatter (dartfmt), click **Format**. The **Reset** button erases your work and restores the editor to its original state.

Note: This page uses embedded DartPads to display runnable examples. If you see empty boxes instead of DartPads, go to the DartPad

Dart language - OOP

- demo

Dart language - Null Safety

Samples

Null Safety

https://dart.dev/null-safety
https://dart.dev/null-safety/understanding-null-safety

https://nullsafety.dartpad.dev/ - песочница

О чем будем говорить

История возникновения языка и текущее состояние

Устанавливаем Dart SDK || Flutter

Знакомимся с Dart Language

Знакомимся с Dart SDK

Tooling

Практика

Packages

The Dart ecosystem uses packages to manage shared software such as libraries and tools. To get Dart packages, you use the pub package manager.

You can find publicly available packages on the https://pub.dev site,
or you can load packages from the local file system or elsewhere, such as Git repositories.

Wherever your packages come from, pub manages version dependencies, helping you get package versions that work with each other and with your SDK version.

Packages

https://dart.dev/tools/pub/package-layout

Your project folder/

pubspec.yaml

bin/
doc/
example/
lib/
  export_files.dart
  src/your_private_sources.dart
test/
tool/
  generators.dart
web/

Your project folder/

.dart_tool
.packages
pubspec.lock

...

LICENSE
README.md
CHANGELOG.md

Packages - import/export

Your own package

pubspec.yaml
name: **my_models**

lib/
src/your_private_sources.dart

file_to_declare_public_code.dart

export "src/
your_private_sources.dart";

Using the packages/

pubspec.yaml
dependencies:
my_models: any

lib/
src/class_with_using_the models.dart

import "package: **my_models/**
file_to_declare_public_code.dart";

Packages - import/export

Your own package

pubspec.yaml
name: **my_models**

lib/
src/your_private_sources.dart

file_to_declare_public_code.dart

export "src/
your_private_sources.dart";

Using the packages/

pubspec.yaml
dependencies:
my_models: any

lib/
src/class_with_using_the models.dart

import "package: **my_models/src/**
your_private_sources.dart";

Packages - pubspec.yaml

Your own package

pubspec.yaml

name: <name of your package to use in pub.dev>

version: <semver>

dependencies:

- dependency to code to be used in your project

dev_dependencies:

- dependency to be used in development phase (generators, builders, tests)

dependency_overrides

- override dependencies

https://dart.dev/tools/pub/pubspec

Packages - pubspec.yaml

```yaml
name: cocktail_app_models
version: 1.0.0
description: >-
  Models to used in Cocktail Application.
homepage: https://cocktail.app
documentation: https://cocktail.app/docs
environment:
  sdk: '>=2.9.0 <3.0.0'
dependencies:
  json_annotation: ^3.0.1
dev_dependencies:
  test: '>=0.6.0 <0.12.0'
  effective_dart: ^1.2.0
```

https://dart.dev/tools/pub/pubspec

Packages - core libraries

- import ‘package:package_name_declared_in_pubspec_yaml/file_in_lib_folder.dart’;
- import 'package:json_annotation/json_annotation.dart';

- import '**dart:collection**' show Queue;

```
import 'dart:' show Queue;
```

```
dart:
dart:collection
dart:isolate
dart:wasm
dart:nativewrappers
dart:convert
dart:math
```

Dart

https://dart.dev/guides/libraries/library-tour

A tour of the core libraries

This page shows you how to use the major features in Dart's core libraries. It's just an overview, and by no means comprehensive. Whenever you need more details about a class, consult the Dart API reference.

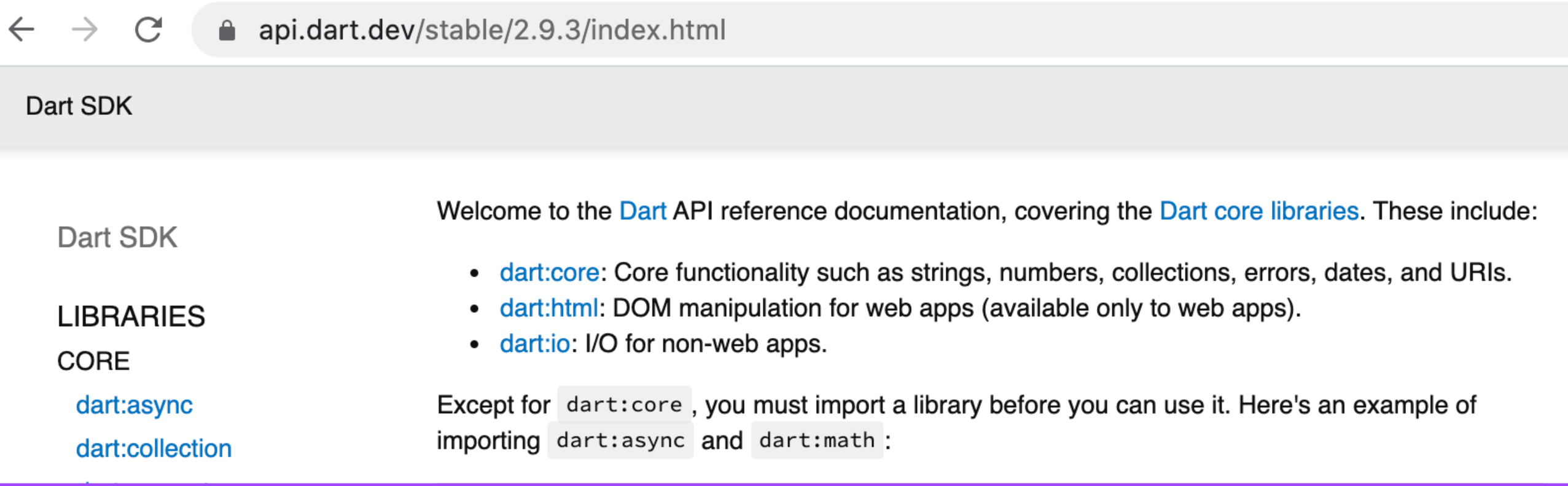
dart:core
Built-in types, collections, and other core functionality. This library is automatically imported into every Dart program.

dart:async
Support for asynchronous programming, with classes such as Future and Stream.

dart:math
Mathematical constants and functions, plus a random number generator.

Dart

https://api.dart.dev/



Dart

```
❯ ls
_http           collection      ffi             isolate         mirrors         web_gl
_internal       convert         html            js              svg             web_sql
api_readme.md   core            indexed_db      js_util         typed_data
async           dev_compiler    internal        libraries.json  wasm
cli             developer       io              math            web_audio
❯ cd core
❯ ls
annotations.dart    duration.dart       iterable.dart       print.dart          string_sink.dart
bigint.dart         errors.dart         iterator.dart       regexp.dart         symbol.dart
bool.dart           exceptions.dart     list.dart           set.dart            type.dart
comparable.dart     expando.dart        map.dart            sink.dart           uri.dart
core.dart           function.dart       null.dart           stacktrace.dart
core_sources.gni    identical.dart      num.dart            stopwatch.dart
date_time.dart      int.dart            object.dart         string.dart
double.dart         invocation.dart     pattern.dart        string_buffer.dart
❯ ~/flutter/b/c/dart-sdk/lib/core 〉 stable                                    ✔
```

Знакомимся с Dart SDK

О чем будем говорить

История возникновения языка и текущее состояние

Устанавливаем Dart SDK || Flutter

Знакомимся с Dart Language

Знакомимся с Dart SDK

Tooling

Практика

Dart

```
❯ ls -la
total 75800
drwxr-xr-x  15 andrey.smirnov  admin       480 Sep  1 02:07 .
drwxr-xr-x  10 andrey.smirnov  admin       320 Sep 23 23:21 ..
-rwxr-xr-x   1 andrey.smirnov  admin  32578256 Sep  1 02:07 dart
-rwxr-xr-x   1 andrey.smirnov  admin      1535 Sep  1 00:41 dart2js
-rwxr-xr-x   1 andrey.smirnov  admin       781 Sep  1 00:41 dart2native
-rwxr-xr-x   1 andrey.smirnov  admin       986 Sep  1 00:41 dartanalyzer
-rwxr-xr-x   1 andrey.smirnov  admin   6198992 Sep  1 02:07 dartaotruntime
-rwxr-xr-x   1 andrey.smirnov  admin      1164 Sep  1 00:41 dartdevc
-rwxr-xr-x   1 andrey.smirnov  admin       857 Sep  1 00:41 dartdoc
-rwxr-xr-x   1 andrey.smirnov  admin       915 Sep  1 00:41 dartfmt
drwxr-xr-x   3 andrey.smirnov  admin        96 Sep  1 00:43 model
-rwxr-xr-x   1 andrey.smirnov  admin      1345 Sep  1 00:41 pub
drwxr-xr-x   3 andrey.smirnov  admin        96 Sep  1 00:43 resources
drwxr-xr-x  16 andrey.smirnov  admin       512 Sep  1 02:07 snapshots
drwxr-xr-x   3 andrey.smirnov  admin        96 Sep  1 02:07 utils
❯ ~/flutter/bin/cache/dart-sdk/bin 〉 stable                              ✔
```

Dart

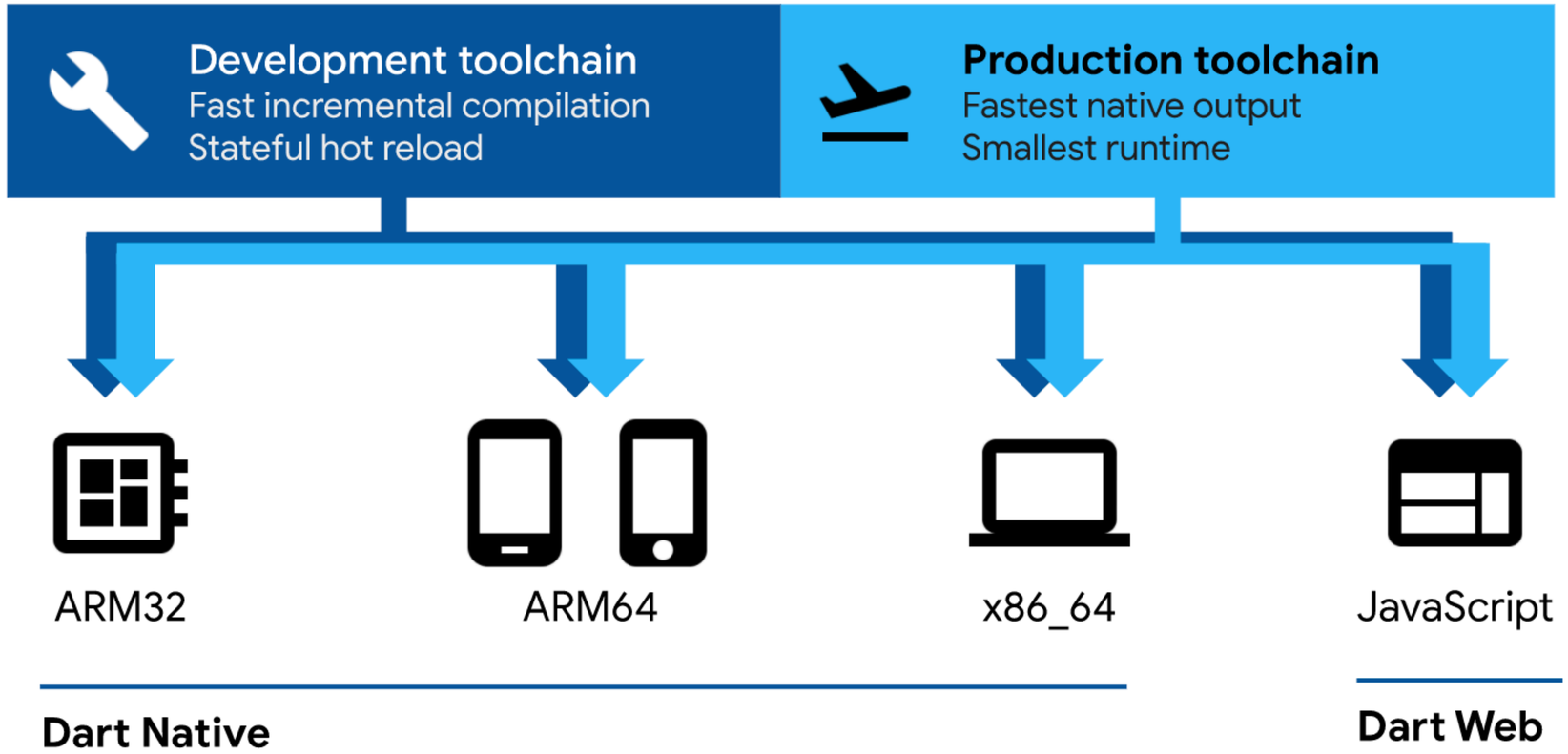
https://dart.dev/platforms

Platforms

You can use Dart to write simple scripts or full-featured apps. Whether you're creating a mobile app, web app, command-line script, or server-side app, there's a Dart solution for that.

Flexible compiler technology lets you run Dart code in different ways, depending on your target platform and goals:

- **Dart Native**: For programs targeting devices (mobile, desktop, server, and more), Dart Native includes both a Dart VM with JIT (just-in-time) compilation and an AOT (ahead-of-time) compiler for producing machine code.
- **Dart Web**: For programs targeting the web, Dart Web includes both a development time compiler (`dartdevc`) and a production time compiler (`dart2js`).



Dart

```
❯ cd cache/dart-sdk/bin
❯ ls -la
total 75800
drwxr-xr-x  15 andrey.smirnov  admin       480 Sep  1 02:07 .
drwxr-xr-x  10 andrey.smirnov  admin       320 Sep 23 23:21 ..
-rwxr-xr-x   1 andrey.smirnov  admin  32578256 Sep  1 02:07 dart
-rwxr-xr-x   1 andrey.smirnov  admin      1535 Sep  1 00:41 dart2js
-rwxr-xr-x   1 andrey.smirnov  admin       781 Sep  1 00:41 dart2native
-rwxr-xr-x   1 andrey.smirnov  admin       986 Sep  1 00:41 dartanalyzer
-rwxr-xr-x   1 andrey.smirnov  admin   6198992 Sep  1 02:07 dartaotruntime
-rwxr-xr-x   1 andrey.smirnov  admin      1164 Sep  1 00:41 dartdevc
-rwxr-xr-x   1 andrey.smirnov  admin       857 Sep  1 00:41 dartdoc
-rwxr-xr-x   1 andrey.smirnov  admin       915 Sep  1 00:41 dartfmt
drwxr-xr-x   3 andrey.smirnov  admin        96 Sep  1 00:43 model
-rwxr-xr-x   1 andrey.smirnov  admin      1345 Sep  1 00:41 pub
```

bin/pub - Настройка pub

https://dart.dev/tools/pub/environment-variables

- demo

- Настройка pub
  - PUB_URL
  - PUB_CACHE

bin/pub stagehand

- demo
- pub global activate stagehand

bin/pub - pub get & pub upgrade

- demo

- Показать .pub-cache

~/.pub-cache/hosted/pub.dev/effective_dart-1.2.4/example

stagehand —console-simple

- demo

bin/dart - running the code in vm

- demo

bin/dart2native

- demo

```
❯ cd ./../console-simple
❯ dart2native bin/main.dart
Generated: /Users/andrey.smirnov/otus/dart-examples/cli/console-simple/bin/main.exe
❯ bin/main.exe
Hello world!
❯ dart bin/main.dart
Hello world!
```

bin/dart - snapshots

- https://github.com/dart-lang/sdk/wiki/snapshots

- Kernel Snapshots
- JIT Application Snapshots (app-jit)
- AOT Application Snapshots (app-aot) => 2.4

```
❯ dart --snapshot-kind=kernel --snapshot=main.dart.snapshot bin/main.dart
❯ dart main.dart.snapshot
Hello world!
 ~/otus/dart-examples/cli/console-simple
```

```
❯ time dart bin/main.dart
Hello world!
dart bin/main.dart  0.72s user 0.11s system 124% cpu 0.667 total
❯ time dart main.dart.snapshot
Hello world!
dart main.dart.snapshot  0.09s user 0.03s system 156% cpu 0.075 total
```

bin/dartanalyzer

- demo

```
Analyzing bin...
  lint • The function main should have a return type but doesn't. • bin/flow.dart:4:1 • always_declare_return_types
  lint • Only use double quotes for strings containing single quotes. • bin/metrics.dart:4:29 • prefer_single_quotes
  lint • Only use double quotes for strings containing single quotes. • bin/metrics.dart:4:39 • prefer_single_quotes
  lint • The function main should have a return type but doesn't. • bin/metrics.dart:6:1 • always_declare_return_types
  hint • The value of the local variable 'isNotUsedVar' isn't used. • bin/main.dart:2:8 • unused_local_variable
4 lints and 1 hint found.
~/otus/dart-examples/cli/console-simple
```

stagehand —web-simple

- demo
- pub global activate webdev

  webdev serve

```
[INFO] Reading cached asset graph completed, took 165ms
[INFO] Checking for updates since last build completed, took 646ms
[INFO] Serving `web` on http://127.0.0.1:8080
[INFO] Running build completed, took 193ms
[INFO] Caching finalized dependency graph completed, took 121ms
[INFO] Succeeded after 322ms with 0 outputs (0 actions)
[INFO] ----------------------------------------------------------------
```

bin/dart2js

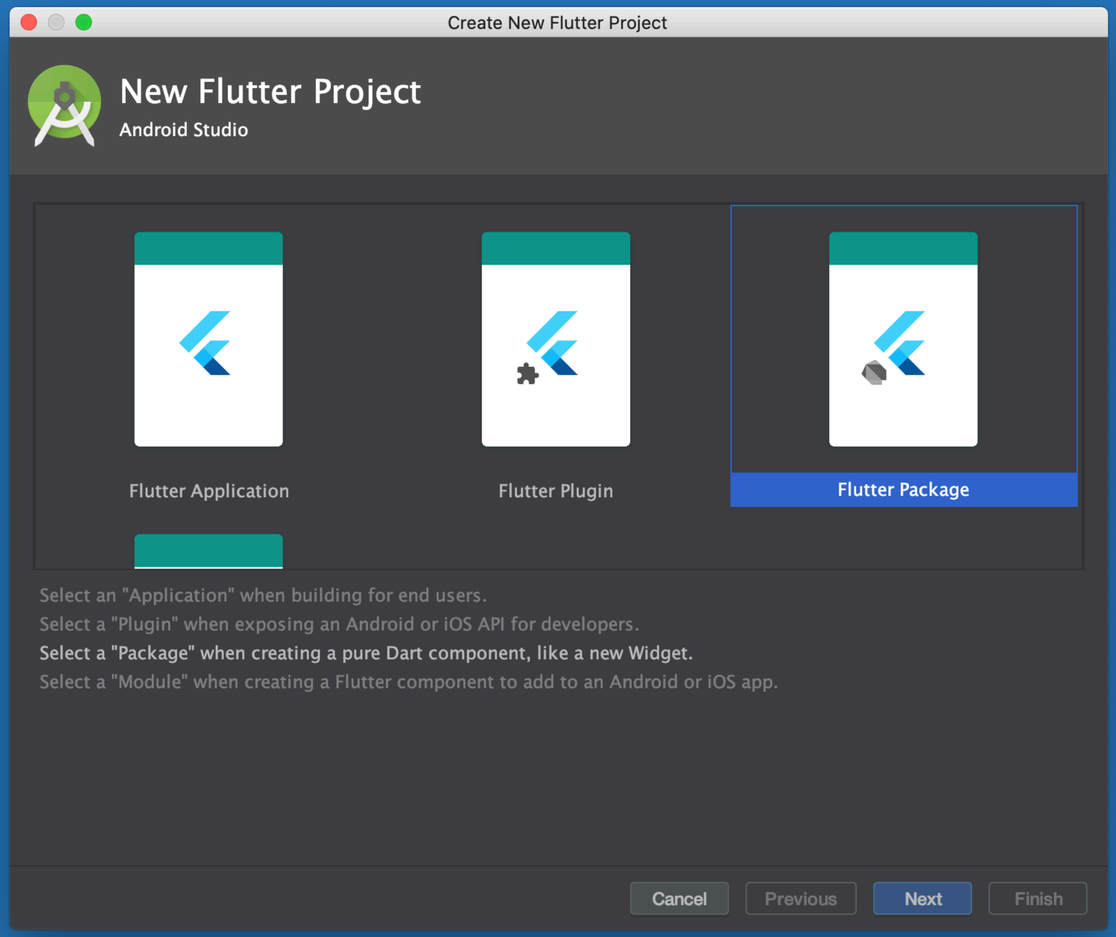
- Dart2js and tree shaking

```
❯ dart2js web/main.dart
Compiled 8,015,282 characters Dart to 23,056 characters JavaScript in 0.75 seconds
 ~/otus/dart-examples/cli/web-simple
```

stagehand --package-simple & flutter create

- demo

flutter create
--template=package flutter_package



bin/pub - pubspec.yaml & pubspec.lock

- После pub get будет pubspec.lock

- https://dart.dev/guides/packages#upgrading-a-dependency

bin/pub - commands

- The pub tool provides the following commands:
  - pub cache
  - pub deps
  - pub downgrade
  - pub get
  - pub global
  - pub outdated
  - pub publish
  - pub run
  - pub upgrade
  - pub uploader

https://dart.dev/guides/packages#pub-commands

```
❯ cd ./../console-simple
❯ pub deps
Dart SDK 2.9.2
console_simple 0.0.0
'-- pedantic 1.9.2
    '-- meta 1.2.3
```

bin/dartanalyzer

https://pub.dev/packages/analyzer

https://dart.dev/guides/language/analysis-options

Demo

~/otus/dart-examples/cli/console-simple

dartanalyzer .
dartanalyzer lib test web

```
❯ dartanalyzer .
Analyzing console-simple...
  lint • Only use double quotes for strings containing single quotes. • bin/metrics.dart:4:29 • prefer_single_quotes
  lint • Only use double quotes for strings containing single quotes. • bin/metrics.dart:4:39 • prefer_single_quotes
  lint • The function main should have a return type but doesn't. • bin/metrics.dart:6:1 • always_declare_return_types
  hint • The value of the local variable 'isNotUsedVar' isn't used. • bin/main.dart:2:8 • unused_local_variable
3 lints and 1 hint found.
❯ ~/otus/dart-examples/cli/console-simple
```

dart analysis server & IDEs

https://dart.dev/tools#ides-and-editors

IDEs and editors

Dart plugins exist for these commonly used IDEs.


Android Studio


Visual Studio Code


IntelliJ IDEA

(and other JetBrains IDEs)

The following Dart plugins are also available, thanks to the Dart community.


Emacs


Atom


Vim


Eclipse

A Language Server Protocol implementation is also available for LSP-capable editors that don't have specific Dart extensions.

dart analysis server & IDEs

https://github.com/dart-lang/sdk/blob/master/pkg/analysis_server/tool/lsp_spec/README.md

Language Server Protocol

Language Server Protocol (LSP) support is available in the Dart analysis server from version 2.2.0 of the SDK (which was included in version 1.2.1 of Flutter). The supported messages are detailed below (for the version of the SDK that matches this README).

Using the Dart LSP server in editors

- Using LSP with Dart-Vim
- Using LSP with Emacs

Running the Server

The analysis server snapshot is included in the `bin/snapshots` folder of the Dart SDK. Pass the `--lsp` flag to start the server in LSP mode and the `--client-id` and `--client-version` flags to identify your editor/plugin and version:

```
dart bin/snapshots/analysis_server.dart.snapshot --lsp --client-id my-editor.my-plugin --client-version 1.2
```

Customizing static analysis

https://dart.dev/guides/language/analysis-options

https://dart-lang.github.io/linter/lints/

pedantic & effective_dart

Вспомним, что делать нельзя:
Packages - import/export

Your own package

pubspec.yaml
name: **my_models**

lib/
src/your_private_sources.dart

file_to_declare_public_code.dart

export “src/
your_private_sources.dart”;

Using the packages/

pubspec.yaml
dependencies:
my_models: any

lib/
src/class_with_using_the models.dart

import “package: **my_models/src/**
your_private_sources.dart”;

Customizing static analysis

analysis_options.yaml

```yaml
# see more details on
# https://pub.dev/packages/pedantic
# https://dart.dev/guides/language/analysis-options
# https://dart-lang.github.io/linter/lints/
#

include: package:effective_dart/analysis_options.yaml

analyzer:
  strong-mode:
    implicit-casts: false
    implicit-dynamic: false

  errors:
    avoid_returning_null_for_future: error
    avoid_void_async: error
    implementation_imports: error # todo: demonstrate warning level

linter:
  rules:
    - avoid_returning_null_for_future
    - avoid_void_async

```

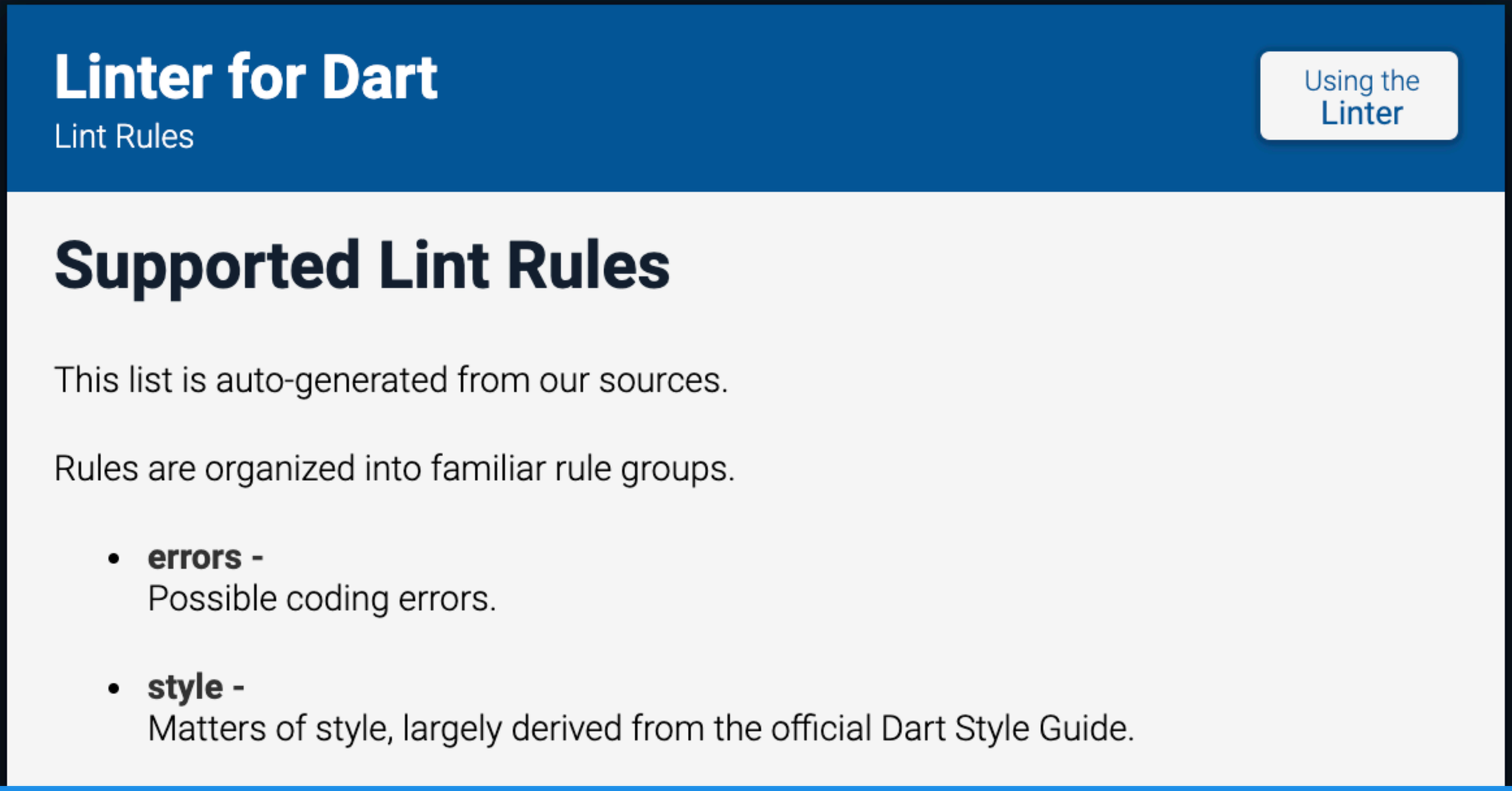
Document 1/1 › analyzer: › errors:

Dart Analysis 1 error

Description	
Don't import implementation files from another package.	[cocktailappmodelsconsumer] lib/main.dart:2

Dart Linter

https://dart-lang.github.io/linter/lints/



Observatory: A Profiler for Dart Apps

https://dart-lang.github.io/observatory/

https://github.com/dart-lang/observatory/blob/gh-pages/metrics.md

```
❯ pwd
/Users/andrey.smirnov/otus/dart-examples/cli/console-simple
❯ dart --observe bin/metrics.dart
Observatory listening on http://127.0.0.1:8181/RwU8CCSdM7M=/
Instance of 'Counter'
Instance of 'Counter'
Instance of 'Counter'
Instance of 'Counter'
Instance of 'Counter'
Instance of 'Counter'
Instance of 'Counter'
Instance of 'Counter'
```

Observatory: A Profiler for Dart Apps

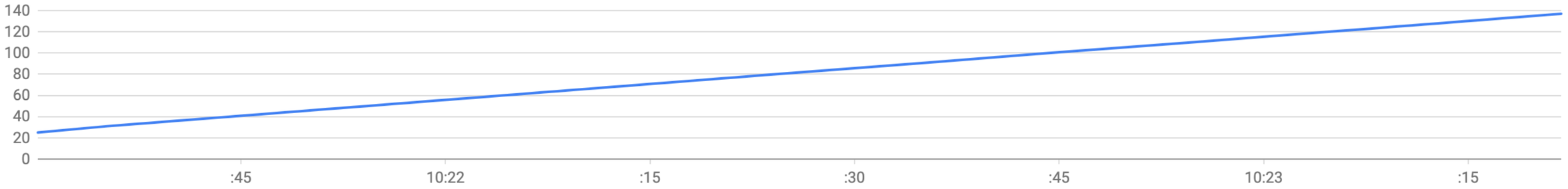
Observatory > vm@ws://127.0.0.1:8181/p-1tPV0HHKA=/ws > main > metrics

Metrics

Metric Bodies

name Bodies
description number of Bodies created
refresh rate Once a second
buffer size 100

current 137



Observatory: A Profiler for Dart Apps

Observatory > vm@ws://127.0.0.1:8181/p-1tPV0HHKA=/ws > main

Refresh | Reload Source

evaluate an expression | Evaluate | Multi-line

```
1  import 'dart:async';
2  import 'dart:developer';
3
4  Counter numBodies = Counter("Bodies", "number of Bodies created");
5
6  main() {
7    Metrics.register(numBodies);
8    numBodies.value = 10;
9    print(numBodies);
10   Timer.periodic(Duration(seconds: 1), (t) {
11     numBodies.value++;
12     print(numBodies);
13   });
14
15   if (numBodies.value < 10) {
16         notExecutedMethod();
17   }
18 }
19
20 void notExecutedMethod() {
21         print('Hello! Run me ((((');
22 }
```

О чем будем говорить

История возникновения языка и текущее состояние

Устанавливаем Dart SDK || Flutter

Знакомимся с Dart Language

Знакомимся с Dart SDK

Tooling

Практика

Начинаем наше мобильное приложение

Описываем предметную область

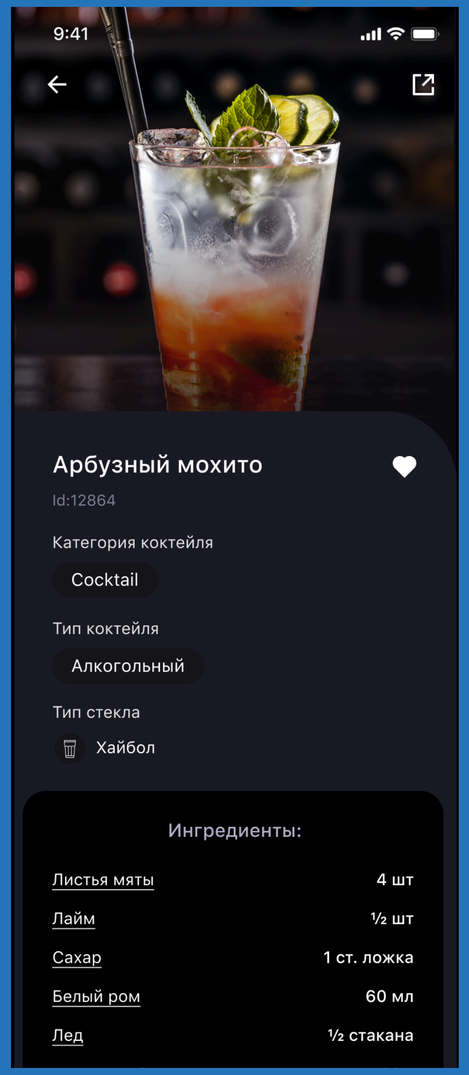
https://rapidapi.com/thecocktaildb/api/the-cocktail-db

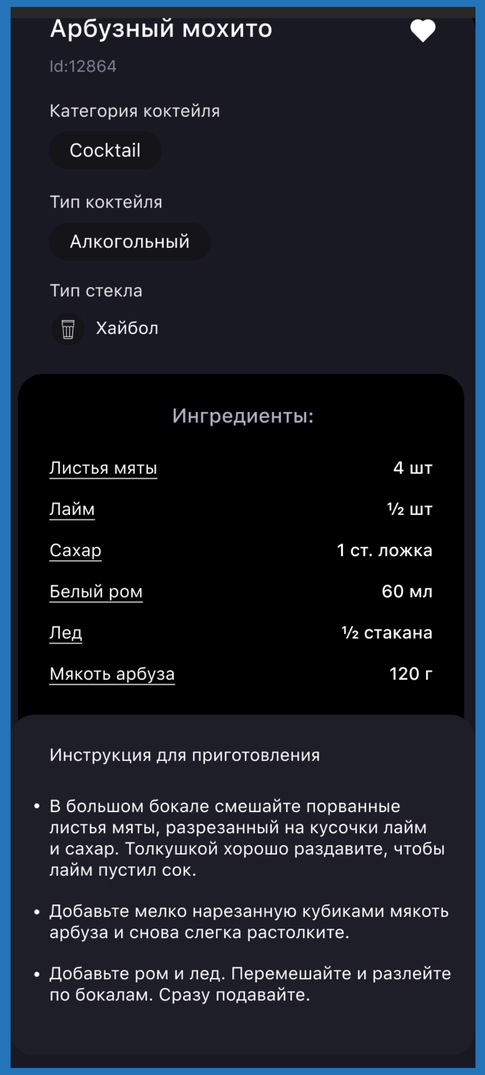
```
❯ curl https://the-cocktail-db.p.rapidapi.com/random.php -H "x-rapidapi-key : e5b7f97a78msh3b1ba27c40d8ccdp105034jsn34e2da32d50b"| jora -p
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1412  100  1412    0     0   3785      0 --:--:-- --:--:-- --:--:--  3785
{
    "drinks": [
        {
            "idDrink": "178329",
            "strDrink": "Captain Kidd's Punch",
            "strDrinkAlternate": null,
            "strDrinkES": null,
            "strDrinkDE": null,
            "strDrinkFR": null,
            "strDrinkZH-HANS": null,
            "strDrinkZH-HANT": null,
            "strTags": null,
            "strVideo": null,
            "strCategory": "Cocktail",
            "strIBA": null,
            "strAlcoholic": "Alcoholic",
            "strGlass": "Collins glass",
            "strInstructions": "Mix all ingredients together in a shaker and strain into a collins glass. Use Caribbean dark Rum for a sweeter
taste.",
            "strInstructionsES": null,
            "strInstructionsDE": null,
            "strInstructionsFR": null,
            "strInstructionsZH-HANS": null,
            "strInstructionsZH-HANT": null,
            "strDrinkThumb": "https://www.thecocktaildb.com/images/media/drink/d83spj1596017390.jpg",
            "strIngredient1": "Rum",
```

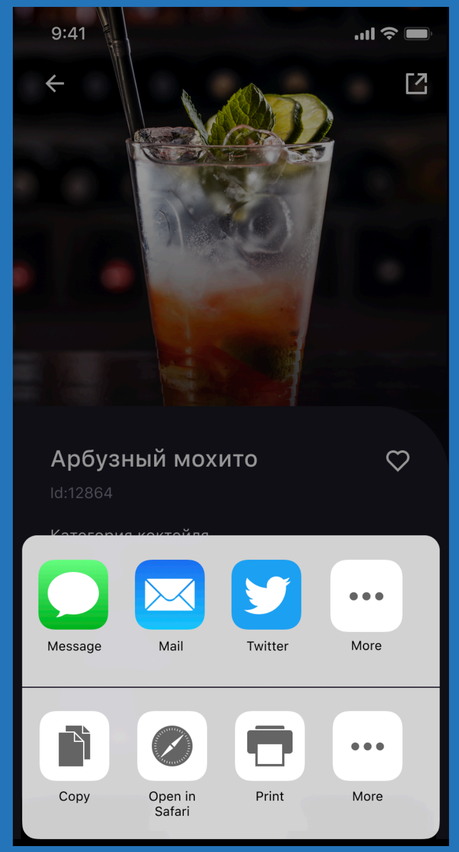
https://rapidapi.com/thecocktaildb/api/the-cocktail-db


```
            "strInstructions": "Mix all ingredients together in a shaker and strain into a collins glass. Use Caribbean dark Rum for a sweeter
taste.",
            "strInstructionsES": null,
            "strInstructionsDE": null,
            "strInstructionsFR": null,
            "strInstructionsZH-HANS": null,
            "strInstructionsZH-HANT": null,
            "strDrinkThumb": "https://www.thecocktaildb.com/images/media/drink/d83spj1596017390.jpg",
            "strIngredient1": "Rum",
            "strIngredient2": "Lime Juice",
            "strIngredient3": "Egg White",
            "strIngredient4": "Bitters",
            "strIngredient5": "Sugar",
            "strIngredient6": "Nutmeg",
            "strIngredient7": "",
            "strIngredient8": null,
            "strIngredient9": null,
            "strIngredient10": null,
            "strIngredient11": null,
            "strIngredient12": null,
            "strIngredient13": null,
            "strIngredient14": null,
            "strIngredient15": null,
            "strMeasure1": "2 shots",
            "strMeasure2": "1 shot",
            "strMeasure3": "1 shot",
            "strMeasure4": "1 dash",
            "strMeasure5": "Ground",
            "strMeasure6": "Top",
            "strMeasure7": "",
```

Модель коктейля



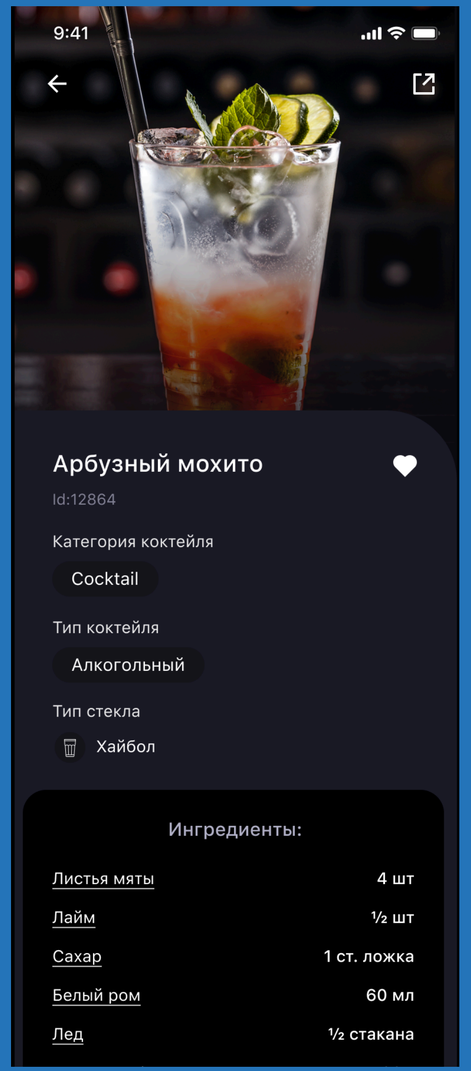


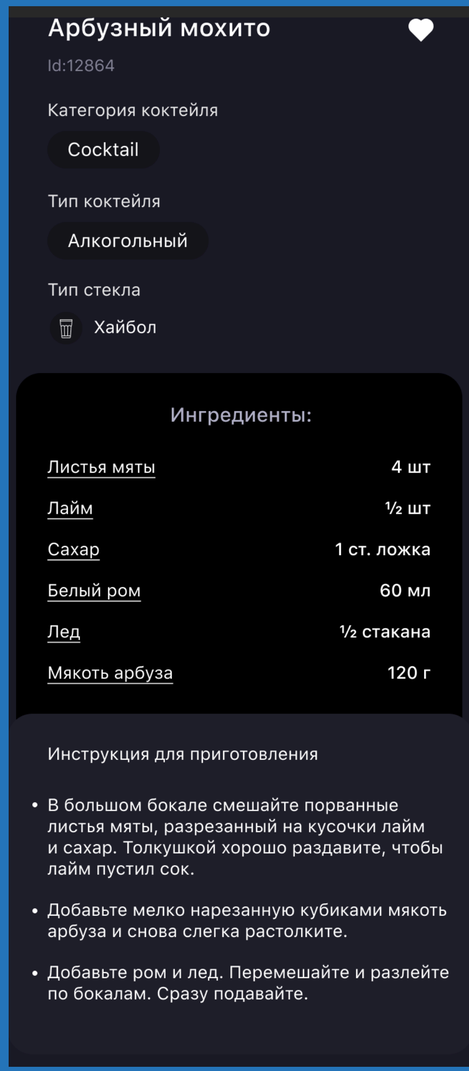


https://rapidapi.com/thecocktaildb/api/the-cocktail-db

<to discuss> min to code and share

1.Создаем flutter (dart) package cocktail_app_models
2.Создаем lib/src
3.Создаем класс коктейля в src
4.Формируем файл экспорта в lib
5.Создаем flutter application cocktail_app
6.Импортируем package с моделью в pubspec.yaml
7.Создаем в main() инстанс модели коктейля





https://rapidapi.com/thecocktaildb/api/the-cocktail-db

Итоги

История возникновения языка и текущее состояние

Устанавливаем Dart SDK || Flutter

Знакомимся с Dart Language

Знакомимся с Dart SDK

Tooling

Практика

Заполните, пожалуйста,
опрос о занятии по ссылке в чате

Спасибо за внимание!
Приходите на следующие вебинары

Смирнов Андрей wrike

Курс **Мобильная разработка на Flutter**

Comments

https://dart.dev/guides/libraries/create-library-packages x

Note: When the `library` directive isn't specified, a unique tag is generated for each library based on its path and filename. Therefore, we suggest that you omit the `library` directive from your code unless you plan to generate library-level documentation.

Note: You may have heard of the `part` directive, which allows you to split a library into multiple Dart files. We recommend that you avoid using `part` and create mini libraries instead.