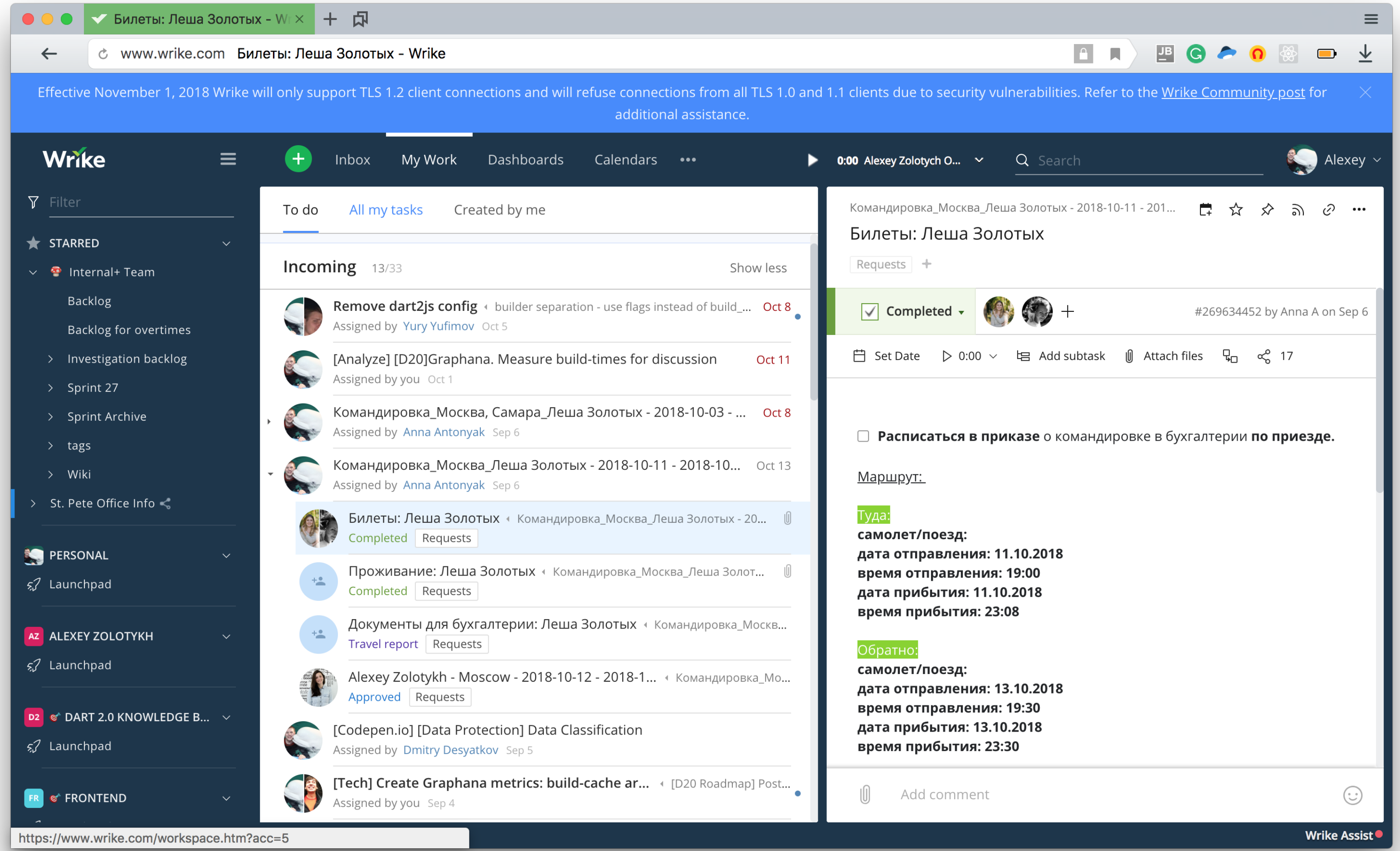
DART

Один язык — все платформы

Алексей Золотых, Wrike



Билеты: Леша Золотых - W

www.wrike.com Билеты: Леша Золотых - Wrike

Effective November 1, 2018 Wrike will only support TLS 1.2 client connections and will refuse connections from all TLS 1.0 and 1.1 clients due to security vulnerabilities. Refer to the Wrike Community post for additional assistance.

Wrike | Inbox | My Work | Dashboards | Calendars | 0:00 Alexey Zolotych O... | Search | Alexey

Filter

STARRED
- Internal+ Team
  - Backlog
  - Backlog for overtimes
  - Investigation backlog
  - Sprint 27
  - Sprint Archive
  - tags
  - Wiki
- St. Pete Office Info

PERSONAL
- Launchpad

ALEXEY ZOLOTYKH
- Launchpad

DART 2.0 KNOWLEDGE B...
- Launchpad

FRONTEND

To do | All my tasks | Created by me

Incoming 13/33 — Show less

- Remove dart2js config · builder separation - use flags instead of build_... — Oct 8 — Assigned by Yury Yufimov Oct 5
- [Analyze] [D20]Graphana. Measure build-times for discussion — Oct 11 — Assigned by you Oct 1
- Командировка_Москва, Самара_Леша Золотых - 2018-10-03 - ... — Oct 8 — Assigned by Anna Antonyak Sep 6
- Командировка_Москва_Леша Золотых - 2018-10-11 - 2018-10... — Oct 13 — Assigned by Anna Antonyak Sep 6
  - Билеты: Леша Золотых · Командировка_Москва_Леша Золотых - 20... — Completed · Requests
  - Проживание: Леша Золотых · Командировка_Москва_Леша Золот... — Completed · Requests
  - Документы для бухгалтерии: Леша Золотых · Командировка_Москв... — Travel report · Requests
  - Alexey Zolotykh - Moscow - 2018-10-12 - 2018-1... · Командировка_Мо... — Approved · Requests
- [Codepen.io] [Data Protection] Data Classification — Assigned by Dmitry Desyatkov Sep 5
- [Tech] Create Graphana metrics: build-cache ar... · [D20 Roadmap] Post... — Assigned by you Sep 4

Командировка_Москва_Леша Золотых - 2018-10-11 - 201...

Билеты: Леша Золотых

Requests +

Completed — #269634452 by Anna A on Sep 6

Set Date | 0:00 | Add subtask | Attach files | 17

☐ **Расписаться в приказе** о командировке в бухгалтерии **по приезде.**

Маршрут:

Туда:
самолет/поезд:
дата отправления: 11.10.2018
время отправления: 19:00
дата прибытия: 11.10.2018
время прибытия: 23:08

Обратно:
самолет/поезд:
дата отправления: 13.10.2018
время отправления: 19:30
дата прибытия: 13.10.2018
время прибытия: 23:30

Add comment

https://www.wrike.com/workspace.htm?acc=5

Wrike Assist

Что есть Dart?

- Dart -> что-то еще
- DartVM

ЧТО ЕСТЬ ВСЕ ПЛАТФОРМЫ?

Браузеры

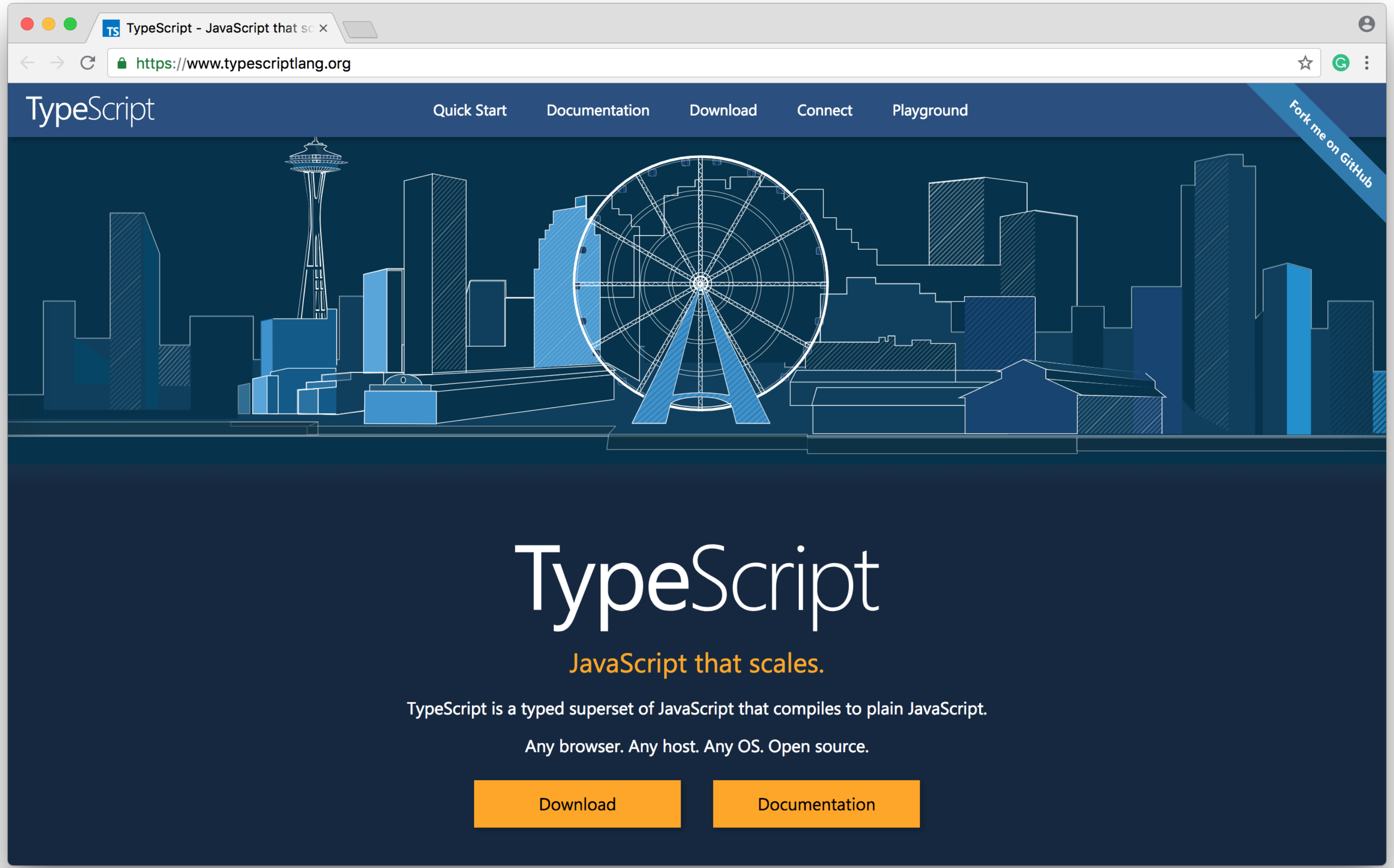
Сервер

Мобильные устройства

Браузеры

Dart как Typescript, только Google его забросил

История



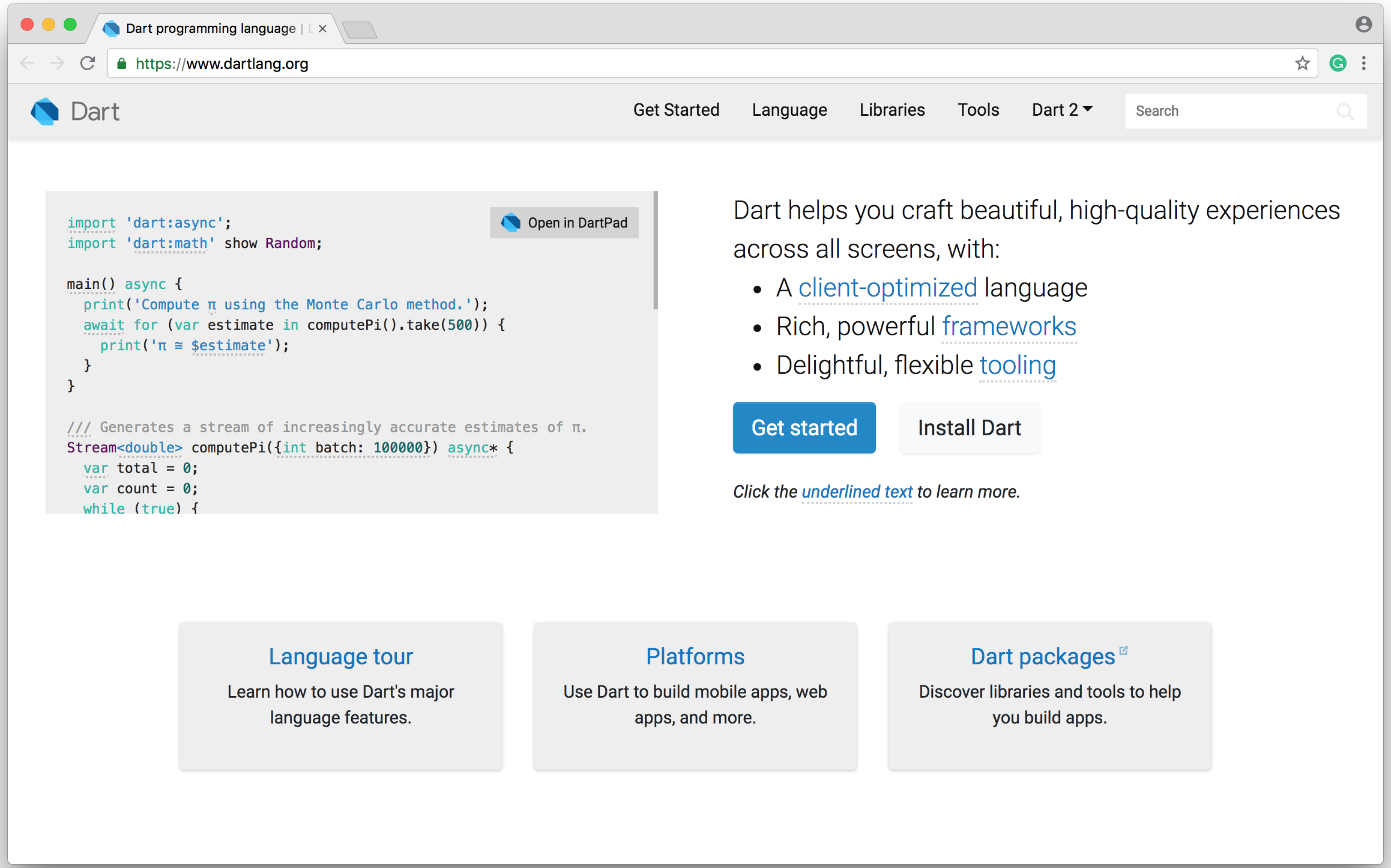
TypeScript

JavaScript that scales.

TypeScript is a typed superset of JavaScript that compiles to plain JavaScript.

Any browser. Any host. Any OS. Open source.

TypeScript is a typed superset of JavaScript that compiles to plain JavaScript.



```
import 'dart:async';
import 'dart:math' show Random;

main() async {
  print('Compute π using the Monte Carlo method.');
  await for (var estimate in computePi().take(500)) {
    print('π ≅ $estimate');
  }
}

/// Generates a stream of increasingly accurate estimates of π.
Stream<double> computePi({int batch: 100000}) async* {
  var total = 0;
  var count = 0;
  while (true) {
```

Dart helps you craft beautiful, high-quality experiences across all screens, with:

- A client-optimized language
- Rich, powerful frameworks
- Delightful, flexible tooling

Get started Install Dart

Click the underlined text to learn more.

Language tour
Learn how to use Dart's major language features.

Platforms
Use Dart to build mobile apps, web apps, and more.

Dart packages
Discover libraries and tools to help you build apps.

Dart helps you craft beautiful, high-quality experiences across all screens, with:

- A client-optimized language
- Rich, powerful frameworks
- Delightful, flexible tooling

Typescript

```
function greeter(person: string) {
    return "Hello, " + person;
}

let user = "Jane User";

document.body.innerHTML = greeter(user);
```

Javascript

```
function greeter(person) {
    return "Hello, " + person;
}

let user = "Jane User";

document.body.innerHTML = greeter(user);
```

Typescript

```
function greeter(person: string) {
    return "Hello, " + person;
}

let user = "Jane User";

document.body.innerHTML = greeter(user);
```

Dart

```
import 'dart:html';

void main() {
  var user = "Jane User";
  document.body.innerHtml = greeter(user);
}
```

SDK

dart:async	dart:svg
dart:collection	dart:typed_data
dart:convert	dart:web_audio
dart:core	dart:web_gl
dart:developer	dart:web_sql
dart:html	dart:cli
dart:indexed_db	dart:io
dart:js	dart:isolate
dart:js_util	dart:mirrors
dart:math	

dart:core

```
import 'dart:core'
```

```
~~import 'dart:core'~~
```

```
void main(){
  final date = DateTime.now();
  final berlinWallFell = new DateTime(1989, 11, 9);
  final Duration diff = date.difference(berlinWallFell);
  print('${diff.inDays} days');
}
```

```
void main(){
  final date = DateTime.now();
  final berlinWallFell = new DateTime(1989, 11, 9);
  final Duration diff = date.difference(berlinWallFell);
  print('${diff.inDays} days');
}
```

```dart
void main(){
  final date = DateTime.now();
  final berlinWallFell = new DateTime(1989, 11, 9);
  final diff = date.difference(berlinWallFell);
  print('${diff.inDays} days');
}
```

```
void main(){
  final date = DateTime.now();
  final berlinWallFell = new DateTime(1989, 11, 9);
  final diff = date.difference(berlinWallFell);
  print('${diff.inDays} days');
}
```

Субъективно удобнее

```
class Hello {
  String foo;
}
```

```
class Hello {
  String _foo;
}
```

```
class Hello {
  String █foo;
}
```

```
class Hello {
  final String _foo;
  Hello(String  input){
    _foo = input;
  }
}
```

```
class Hello {
  final String _foo;
  Hello(this._foo);
}
```

```
new Hello();
```

```
~~new~~ Hello();
```

```
Hello();
```

Future вместо Promise

```
import 'dart:async';

Future<void> main() async {
  print(await asyncOperation());
}

Future<String> asyncOperation() =>
  Future.delayed(
    Duration(seconds: 1),
    () => 'hello',
  );
```

Future вместо Promise

```
import 'dart:async';

Future<void> main() async {
  print(await asyncOperation());
}

Future<String> asyncOperation() =>
  Future.delayed(
    Duration(seconds: 1),
    () => 'hello',
  );
```

FutureOr

```
import 'dart:async';

FutureOr<String> foo() => 'hello';

Future<void> main() async {
  print(await foo());
}
```

Streams

Пока еще не Stream

```
Iterable<int> getList() sync* {
  yield 1;
  yield 2;
}

void main(){
  print(getList().join('\n'));
}
```

Уже Stream

```
Stream<int> getList() async* {
  await 1;
  await 2;
}
...
final stream = getList();
stream.listen((int i) => /* do something */);
...
```

```
//...
final controller = new StreamController<String>();
controller.stream.listen(print);
controller.add('hello');
//...
```

— Ничего не напоминает?

— RxJS

```
@Component(
  selector: 'hero-search',
)
class HeroSearchComponent {
  ...
  Stream<List<Hero>> heroes;
  StreamController<String> _searchTerms = StreamController<String>.broadcast();
  ...
}
```

Весь мир обновляется сразу

Коллекции

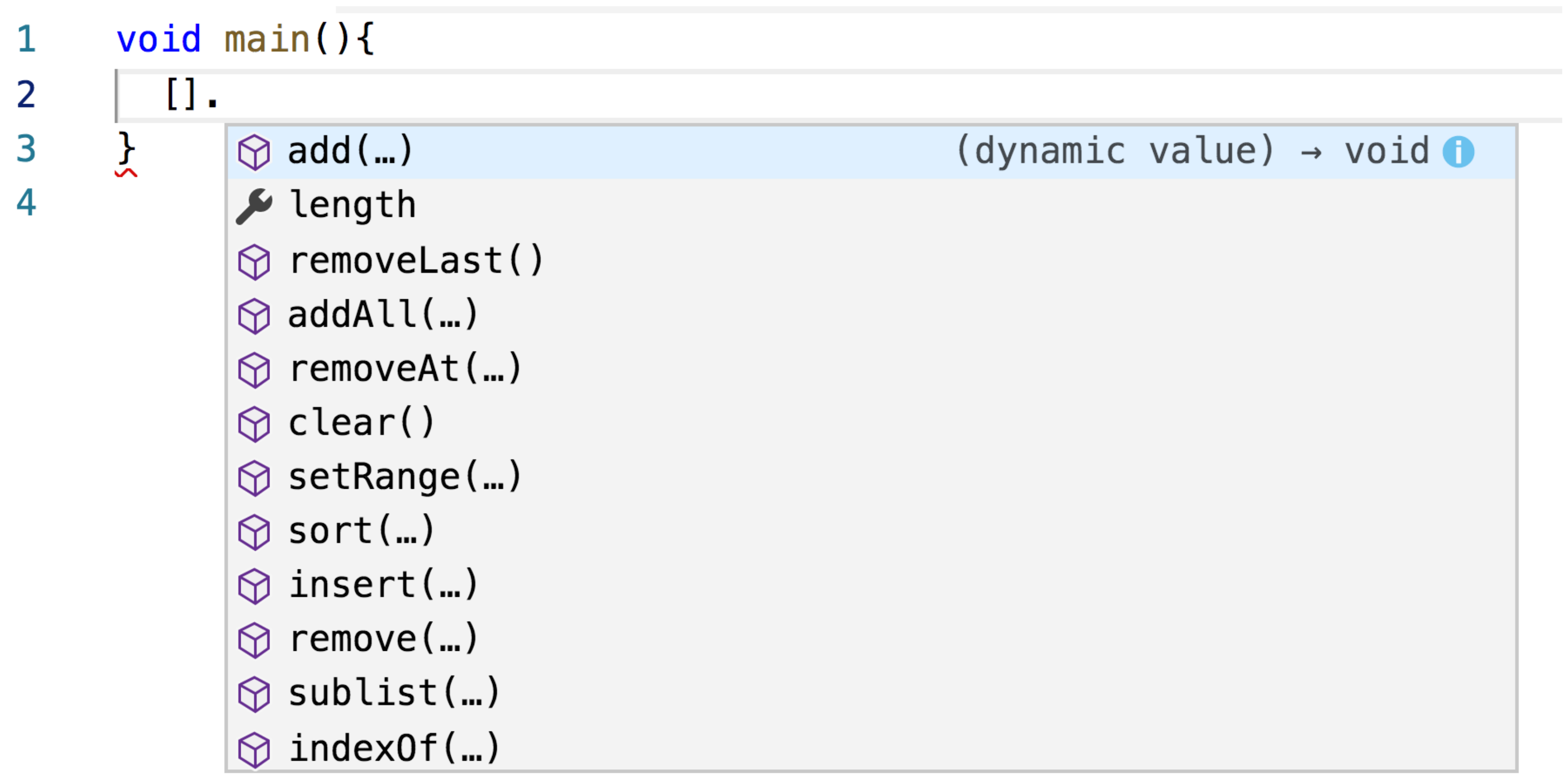
```
import 'dart:collection'
```

Пока еще не Stream

```
Iterable<int> getList() sync* {
  yield 1;
  yield 2;
}

void main(){
  print(getList().join('\n'));
}
```

Пока еще не Stream

```
Iterable<int> getList() sync* {
  yield 1;
  yield 2;
}

void main(){
  print(getList().join('\n'));
}
```



```
void main(){
  [].
}
```

add(…) (dynamic value) → void
length
removeLast()
addAll(…)
removeAt(…)
clear()
setRange(…)
sort(…)
insert(…)
remove(…)
sublist(…)
indexOf(…)

В СТАНДАРТНОЙ БИБЛИОТЕКЕ

Iterable, Map, Set, List

В DART:COLLECTION

DoubleLinkedQueue DoubleLinkedQueueEntry HashMap HashSet HasNextIterator IterableBase IterableMixin LinkedHashMap LinkedHashSet LinkedList LinkedListEntry ListBase ListMixin ListQueue MapBase MapMixin MapView Queue SetBase SetMixin SplayTreeMap SplayTreeSet UnmodifiableListView UnmodifiableMapBase UnmodifiableMapView

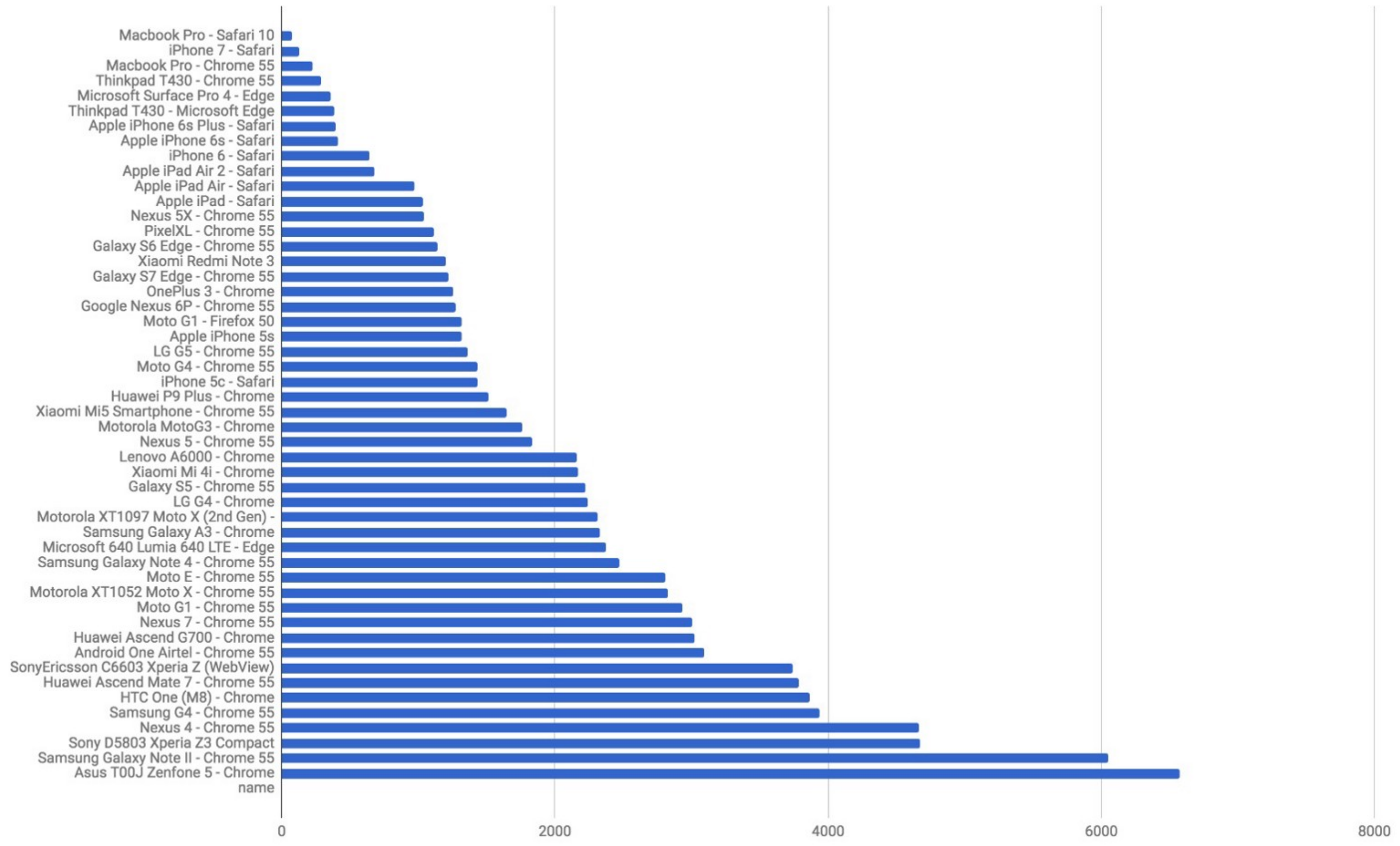
Dart

```
import 'dart:html';

void main() {
  var user = "Jane User";
  document.body.innerHtml = greeter(user);
}
```

Dart

```
import 'dart:html';

void main() {
  var user = "Jane User";
  document.body.innerHtml = greeter(user);
}
```

Размер имеет значение



Macbook Pro - Safari 10
iPhone 7 - Safari
Macbook Pro - Chrome 55
Thinkpad T430 - Chrome 55
Microsoft Surface Pro 4 - Edge
Thinkpad T430 - Microsoft Edge
Apple iPhone 6s Plus - Safari
Apple iPhone 6s - Safari
iPhone 6 - Safari
Apple iPad Air 2 - Safari
Apple iPad Air - Safari
Apple iPad - Safari
Nexus 5X - Chrome 55
PixelXL - Chrome 55
Galaxy S6 Edge - Chrome 55
Xiaomi Redmi Note 3
Galaxy S7 Edge - Chrome 55
OnePlus 3 - Chrome
Google Nexus 6P - Chrome 55
Moto G1 - Firefox 50
Apple iPhone 5s
LG G5 - Chrome 55
Moto G4 - Chrome 55
iPhone 5c - Safari
Huawei P9 Plus - Chrome
Xiaomi Mi5 Smartphone - Chrome 55
Motorola MotoG3 - Chrome
Nexus 5 - Chrome 55
Lenovo A6000 - Chrome
Xiaomi Mi 4i - Chrome
Galaxy S5 - Chrome 55
LG G4 - Chrome
Motorola XT1097 Moto X (2nd Gen) -
Samsung Galaxy A3 - Chrome
Microsoft 640 Lumia 640 LTE - Edge
Samsung Galaxy Note 4 - Chrome 55
Moto E - Chrome 55
Motorola XT1052 Moto X - Chrome 55
Moto G1 - Chrome 55
Nexus 7 - Chrome 55
Huawei Ascend G700 - Chrome
Android One Airtel - Chrome 55
SonyEricsson C6603 Xperia Z (WebView)
Huawei Ascend Mate 7 - Chrome 55
HTC One (M8) - Chrome
Samsung G4 - Chrome 55
Nexus 4 - Chrome 55
Sony D5803 Xperia Z3 Compact
Samsung Galaxy Note II - Chrome 55
Asus T00J Zenfone 5 - Chrome
name
0
2000
4000
6000
8000

```dart
class Car {
  final Button button;
  Car(this.button);
}

abstract class Button {
  void push();
}

class EmergencyButton implements Button {
  void push()=> print('push');
}
```

Помигать аварийкой

```
Future main() async {
  final car = Car(EmergencyButton());
  car.button.push();
}
```

Компилируем

И получаем... 8k в выходном файле

```
(function(){var supportsDirectProtoAccess=function(){var z=
z.prototype={p:{}}
var y=new z()
if(!(y.__proto__&&y.__proto__.p===z.prototype.p))return fal
try{if(typeof navigator!="undefined"&&typeof navigator.use
if(typeof version=="function"&&version.length==0){var x=ve
if(/^\d+\.\d+\.\d+\.\d+$/.test(x))return true}}catch(w){}re
function map(a){a=Object.create(null)
a.x=0
delete a.x
```

Typescript

```
class Car {
    constructor(public button: Button) { }
}

interface Button { push():void; }

class EmergencyButton implements Button {
    push() { console.log('push') };
}

const car = new Car(new EmergencyButton());
car.button.push();
```

```
var Car = /** @class */ (function () {
    function Car(button) { this.button = button; }
    return Car;
}());
var EmergencyButton = /** @class */ (function () {
    function EmergencyButton() {
    }
    EmergencyButton.prototype.push = function () { console.log('push'); };
    ;
    return EmergencyButton;
}());
var car = new Car(new EmergencyButton());
car.button.push();
```

```
...
H.c("push")
...
```

+

SDK

+

Runtime

```dart
import 'package:example/bar.dart' deferred as bar;

Future<void> foo() async {
  await bar.loadLibrary();
  bar.bar();
}
```

Dart2js знает куда положить ваш код

- 60K out.js
- 4.0K out.js_1.part.js
- 4.0K out.js_2.part.js
- 4.0K out.js_3.part.js

Dart не подходит легковесным приложениям

Фреймворки

- Angular 2+
- React
- Vue

Dart на сервере

Node vs Dart

Практически полный паритет в скорости

tiny.cc/dartvsnode

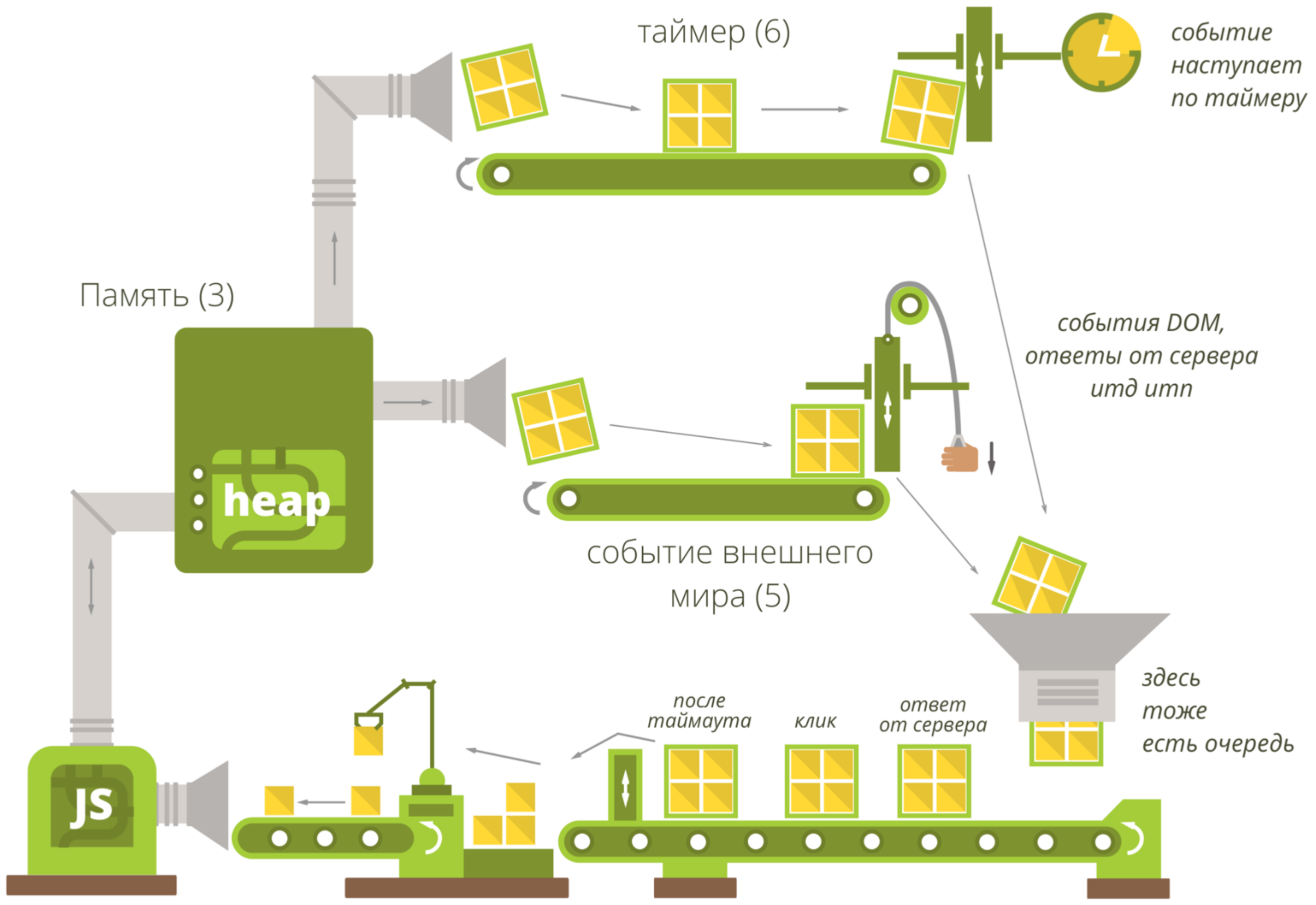
Dart можно скомпилировать для Node node_interop

Завезли немного параллельности

```
Isolate.spawnUri(Uri.parse('isolate.dart'), [], null)
```

```
library isolate;

void main() {
  print('isolate.main()');
  while(true);
}
```

Потеря контекста


таймер (6)
событие наступает по таймеру
Память (3)
heap
события DOM, ответы от сервера итд итп
событие внешнего мира (5)
здесь тоже есть очередь
после таймаута
клик
ответ от сервера
JS

Стек вызовов (1)

Очередь вызовов (2)

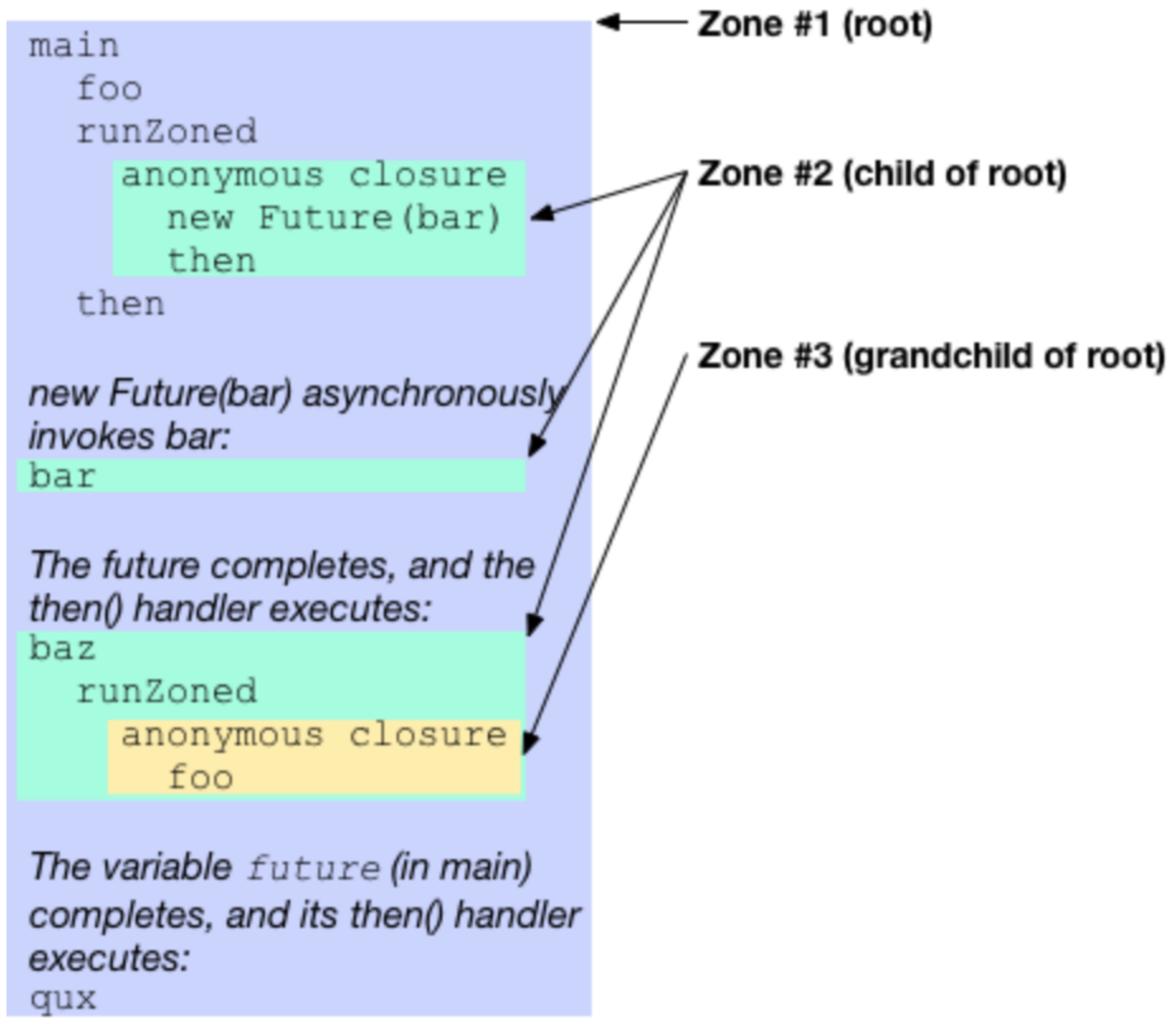
Доклад Алексея Охрименко

www.youtube.com/watch?v=Lrs6puJ4G2Q

Моя статья на Habr

habr.com/company/wrike/blog/302896/

```
import 'dart:async';

main() {
  foo();
  var future;
  runZoned(() {           // Starts a new child zone (zone #2).
    future = new Future(bar).then(baz);
  });
  future.then(qux);
}

foo() => ...foo-body...  // Executed twice (once each in two zones).
bar() => ...bar-body...
baz(x) => runZoned(() => foo()); // New child zone (zone #3).
qux(x) => ...qux-body...
```



```
main
  foo
  runZoned
    anonymous closure
      new Future(bar)
      then
  then
```

new Future(bar) asynchronously invokes bar:

```
bar
```

The future completes, and the then() handler executes:

```
baz
  runZoned
    anonymous closure
      foo
```

The variable `future` *(in main) completes, and its then() handler executes:*

```
qux
```

Зоны

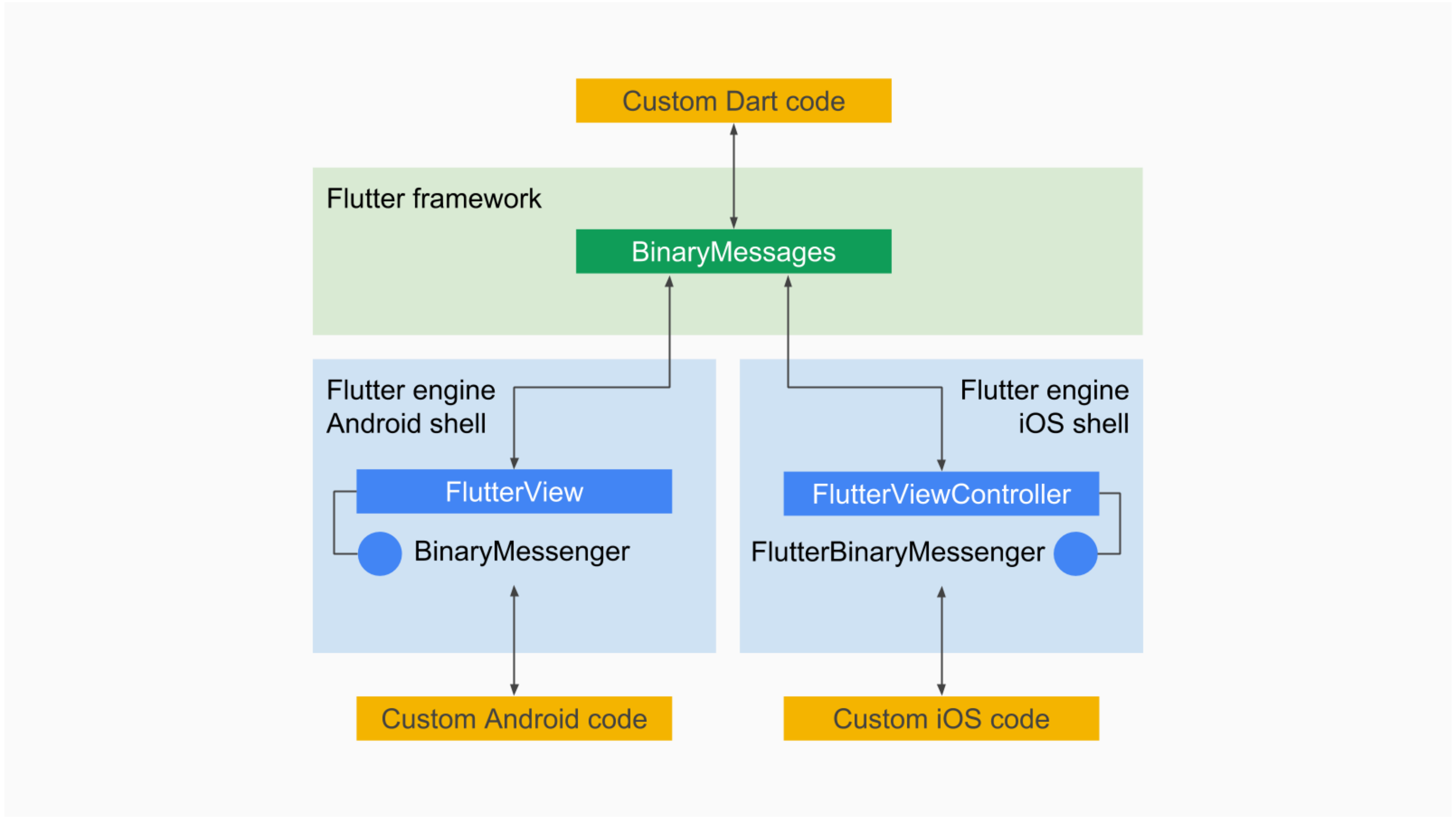
```
runZoned(() {
  Timer.run(() { throw 'Would normally kill the program';
}, onError: (error, stackTrace) {
  print('Uncaught error: $error');
});
```

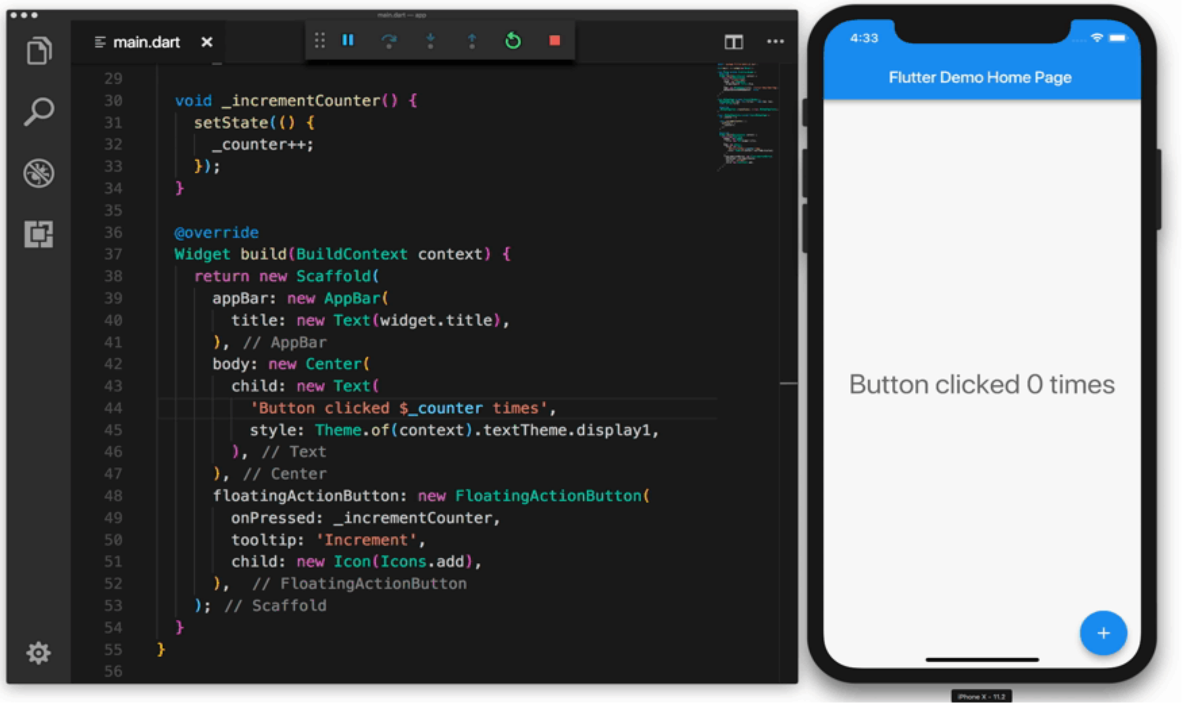
https://habr.com/company/wrike/blog/310422/

Мобильные устройства

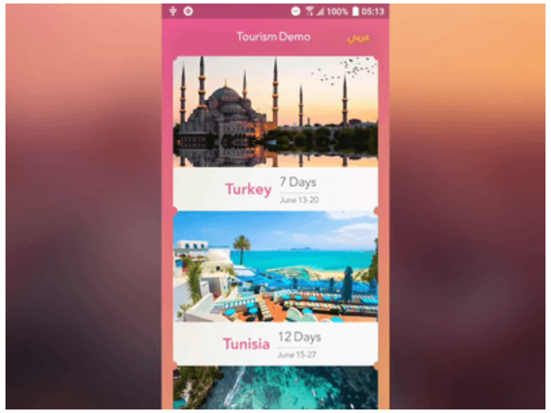
iOS + Android

Flutter


Custom Dart code
Flutter framework
BinaryMessages
Flutter engine
Android shell
FlutterView
BinaryMessenger
Flutter engine
iOS shell
FlutterViewController
FlutterBinaryMessenger
Custom Android code
Custom iOS code



```dart

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      appBar: new AppBar(
        title: new Text(widget.title),
      ), // AppBar
      body: new Center(
        child: new Text(
          'Button clicked $_counter times',
          style: Theme.of(context).textTheme.display1,
        ), // Text
      ), // Center
      floatingActionButton: new FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment',
        child: new Icon(Icons.add),
      ),  // FloatingActionButton
    ); // Scaffold
  }
}

```



https://github.com/bluemix/tourism-demo

Desktop

https://github.com/google/flutter-desktop-embedding

Dart используется

- Frontend
- Server
- Desktop
- Mobile

Dart — самодостаточный язык

Спасибо!

Алексей Золотых

twitter: @zolotyh

e-mail: aazolotyh@gmail.com