Dan Rubel
Eric Clayberg

Overview

- Dart Project
- Dart Language
- Client-side Dart
- Server-side Dart
- Developing Dart Editor
- Questions

http://www.dartlang.org

Project

http://www.dartlang.org

Web Apps

- The web is everywhere
- Developing small apps is easy
- No installation required
- Supports incremental development
- Open platform

... but we need to innovate

- Productivity
- Scalability
- Speed

http://www.dartlang.org

Why?

What is wrong with JavaScript?

- Lack of structure
- Very difficult to find dependencies
- All numbers are floats
- Monkey patching
- Keep on truckin' mentality
- Libraries are files you concatenate

... which makes it ...

- Difficult to optimize performance
- Difficult to maintain large code bases
- Difficult to assert correctness

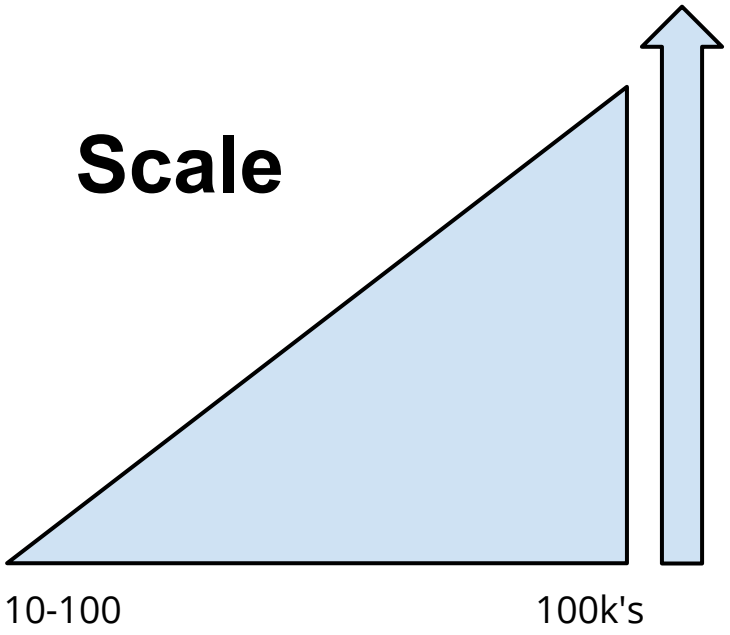
Goal

“Dart helps developers from all platforms build complex, high performance client apps for the modern web.”

VM Speed

x2 → **x10**

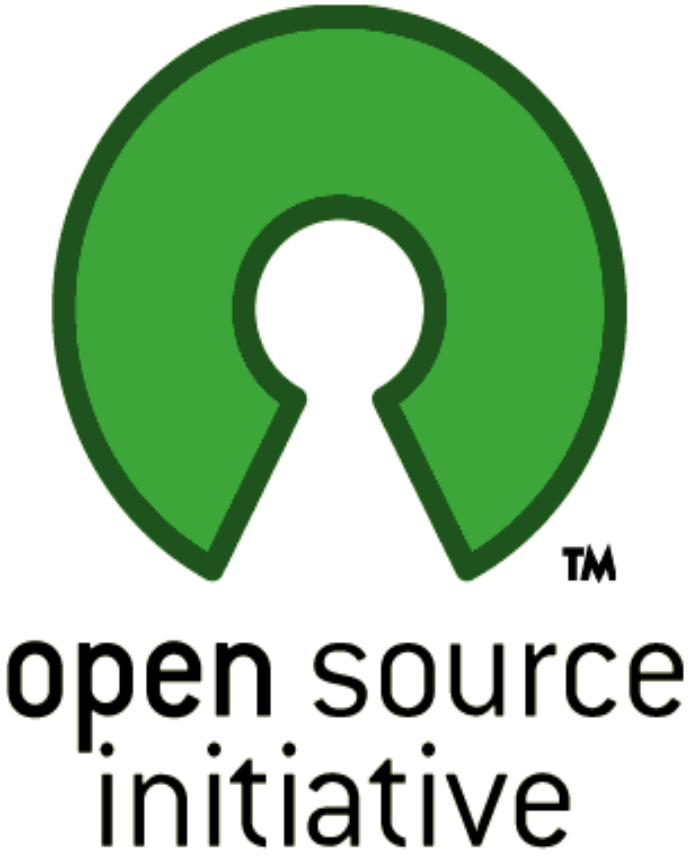
Scale



What is Dart?

Dart is open source

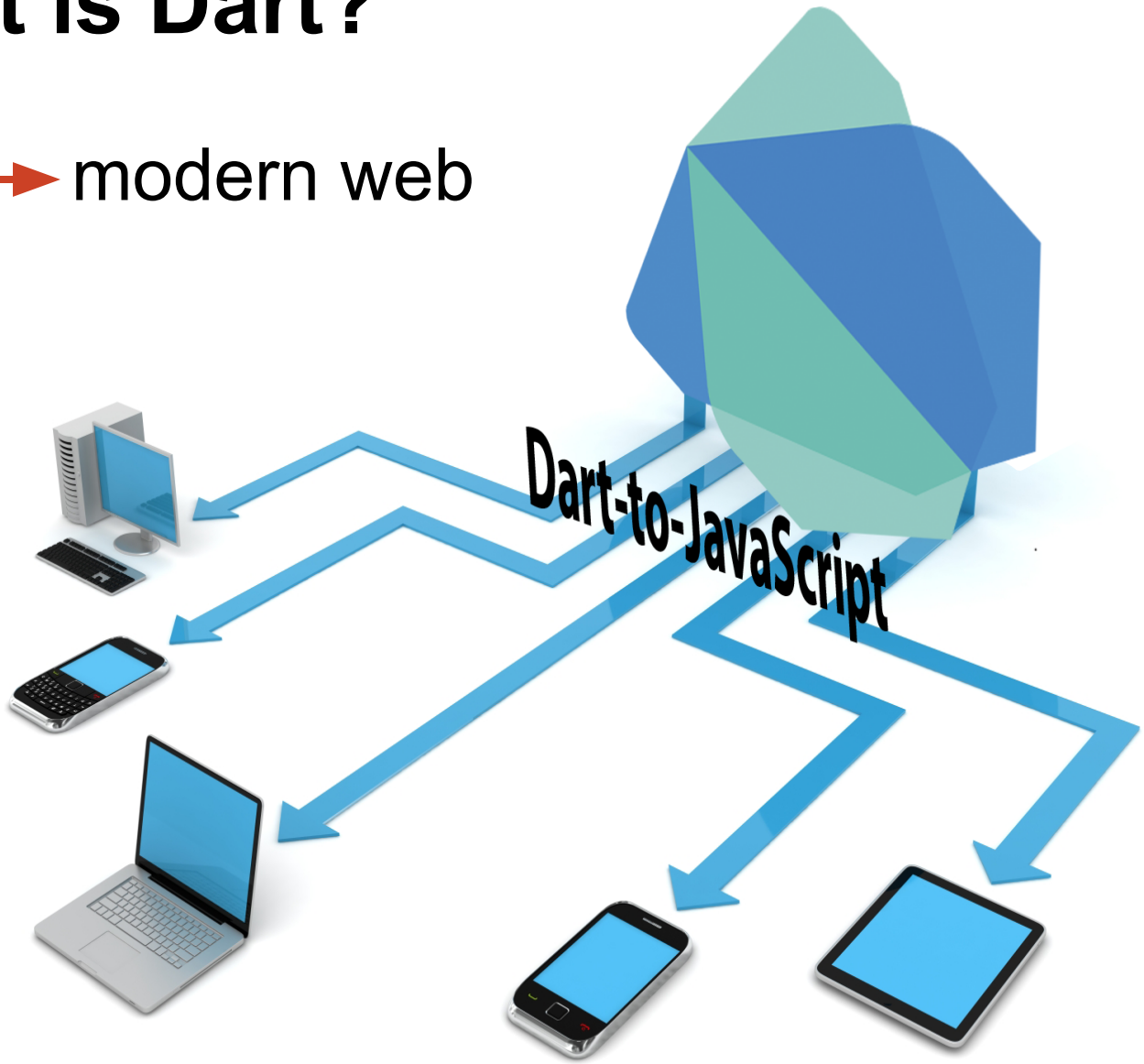
- BSD-style license
- dart.googlecode.com
- GitHub mirror
- Contributing guide



http://www.dartlang.org

What is Dart?

Dart → modern web



What is Dart?

- Rich client apps
- Server apps
- Offline-capable
- 60fps
- ES5+
- HTML5

http://www.dartlang.org

What is Dart?

Language
Libraries (core, math, html, isolate, io, json, uri, utf, crypto, ...)
Editor
Virtual machine
Compiler to JavaScript
Browser integration
Package manager
Web Components integration
Community

"Batteries Included"

http://www.dartlang.org

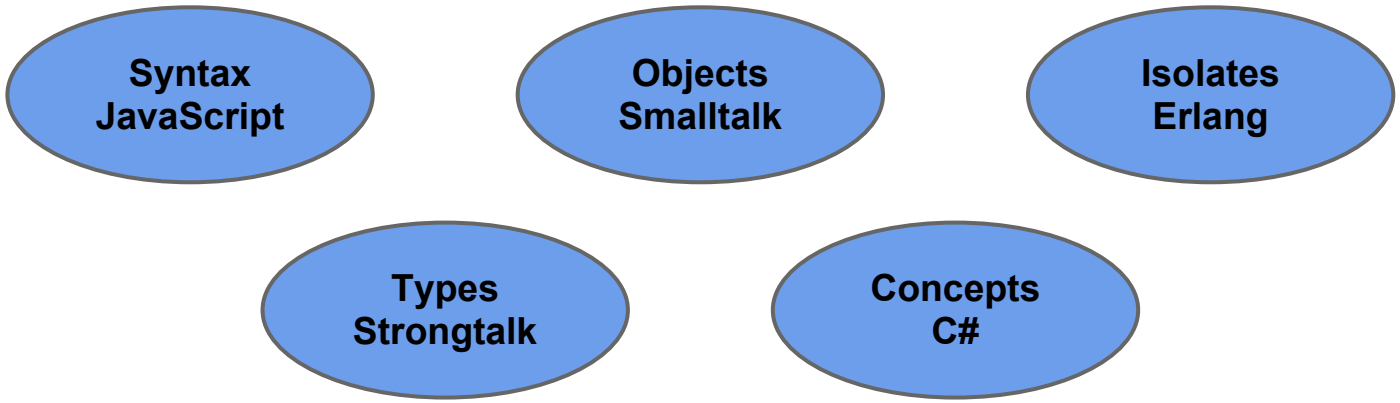
Language

http://www.dartlang.org

Language Goals

- Unsurprising simple and familiar
- Class-based single inheritance
- Proper lexical scoping
- Single-threaded with isolates
- Optional static types

Dart =



http://www.dartlang.org

Optional Static Types

Static types ...

- Communicate developer intent
- Used by tools to assert program correct
- Not used at runtime unless specified

You can mix typed and untyped code ...

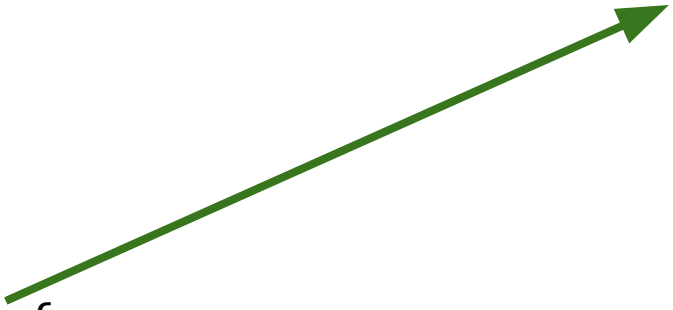
```
var p = new Point(2, 3);
int x = 4;
print (p + new Point(x, 5));
```

Implicit Interfaces

```
class Printer {
  void print(String message) {
    // print the message
  }
}

class MockPrinter implements Printer {
  void print(String message) {
    // validate method was called
  }
}
```

Mixins

```
class PrettyPrinter extends ASTTool with Counter
{
 ...
}

class Counter {
      int count = 0;
      int get increment ==> ++count;
      int get decrement => --count;
}
```

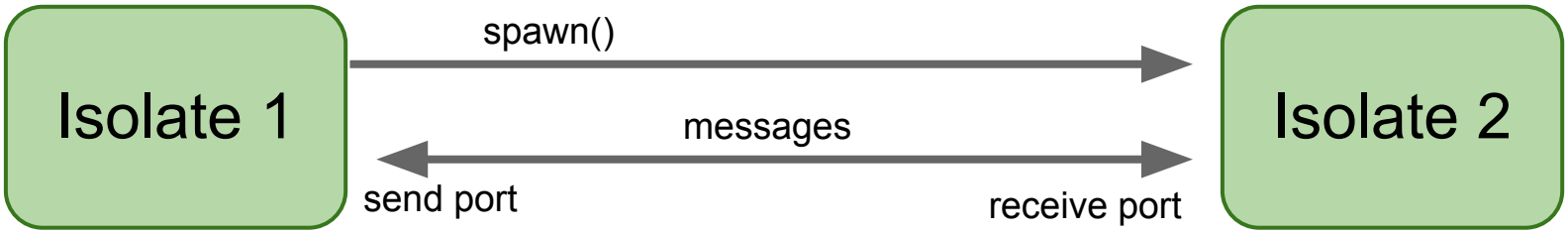


... available early 2013

http://www.dartlang.org

Isolates

- Inspired by Erlang
- Lightweight units of execution
  - Each isolate conceptually a process
  - Nothing shared
  - All communication via message passing
- Support concurrent execution



Client-sid

http://www.dartlang.org

Hello World

```
import 'dart:html';

void main() {
  new Hello().welcome("World");
}


class Hello {
  void welcome(String name) {
    var message = "Hello $name";
    document.query('#status')
        ..text = message
        ..on.click.add(handler);
  }
}
```

http://www.dartlang.org

Hello World

```
import 'dart:html';

void main() {
  new Hello().welcome("World");
}


class Hello {
  void welcome(String name) {
    var message = "Hello $name";
    document.query('#status')
        ..text = message
        ..on.click.add(handler);
  }
}
```

Libraries

http://www.dartlang.org

Hello World

```
import 'dart:html';

void main() {
  new Hello().welcome("World");
}

class Hello {
  void welcome(String name) {
    var message = "Hello $name";
    document.query('#status')
        ..text = message
        ..on.click.add(handler);
  }
}
```

Functions

http://www.dartlang.org

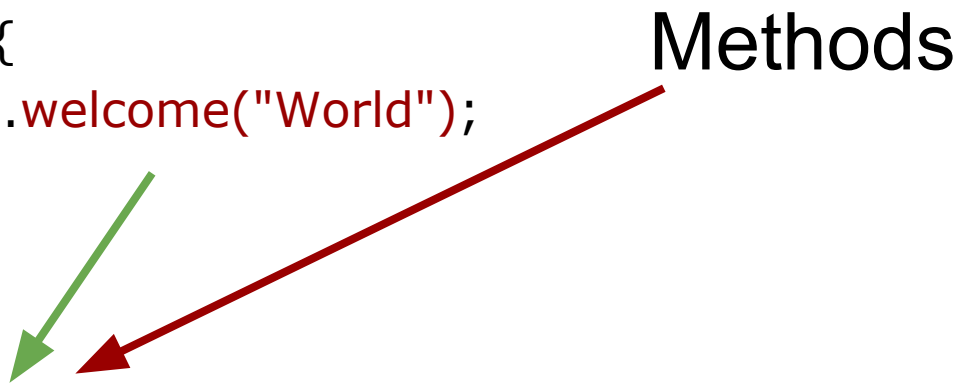
Hello World

```
import 'dart:html';

void main() {
  new Hello().welcome("World");
}

class Hello {
  void welcome(String name) {
    var message = "Hello $name";
    document.query('#status')
        ..text = message
        ..on.click.add(handler);
  }
}
```

http://www.dartlang.org

Hello World

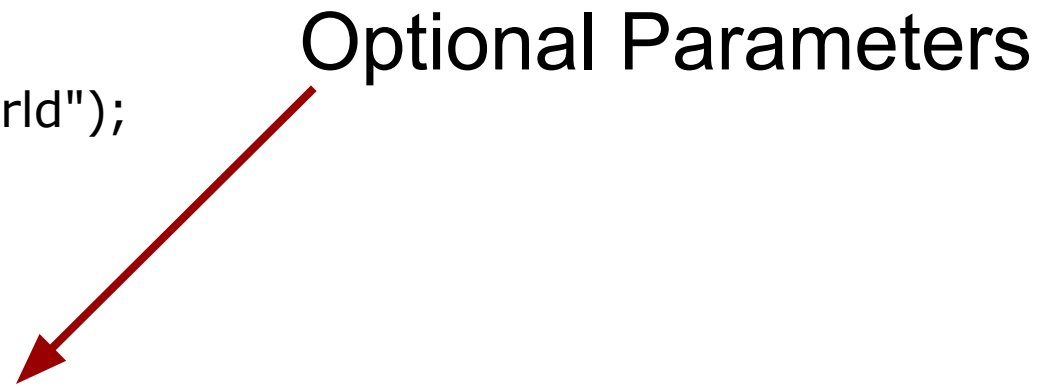
```
import 'dart:html';

void main() {
  new Hello().welcome("World");
}

class Hello {
  void welcome(String name) {
    var message = "Hello $name";
    document.query('#status')
        ..text = message
        ..on.click.add(handler);
  }
}
```



http://www.dartlang.org

Hello World

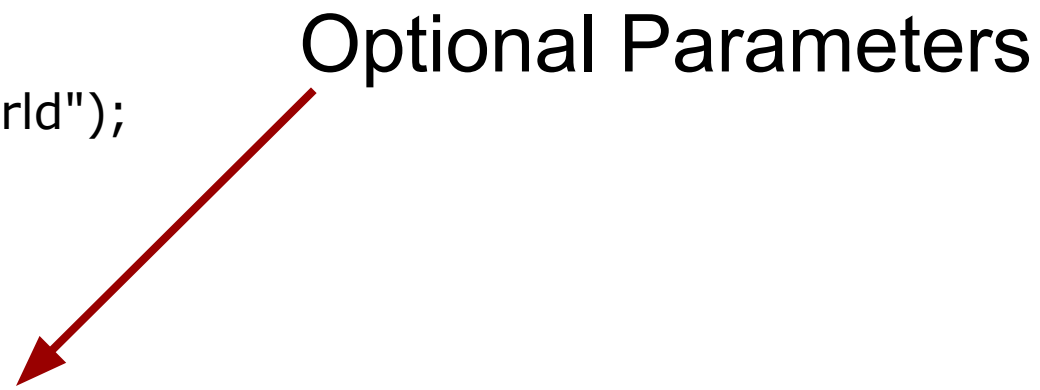
```
import 'dart:html';

void main() {
  new Hello().welcome("World");
}

class Hello {
  void welcome(String name) {
    var message = "Hello $name";
    document.query('#status')
        ..text = message
        ..on.click.add(handler);
  }
}
```



http://www.dartlang.org

Hello World

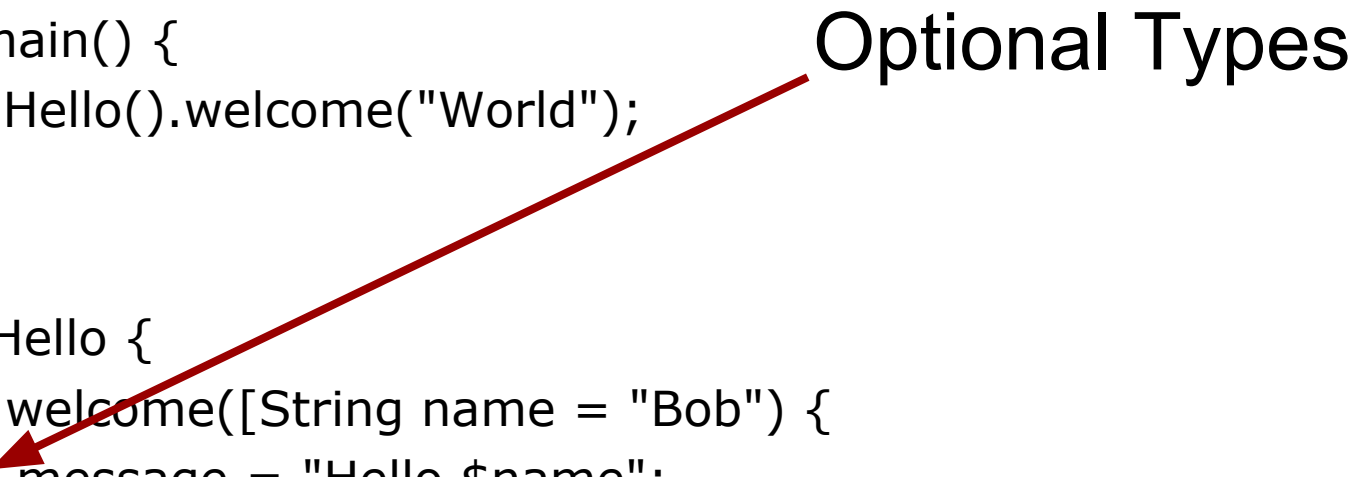
```
import 'dart:html';

void main() {
  new Hello().welcome("World");
}


class Hello {
  void welcome([String name = "Bob"]) {
    var message = "Hello $name";
    document.query('#status')
        ..text = message
        ..on.click.add(handler);
  }
}
```



http://www.dartlang.org

Hello World

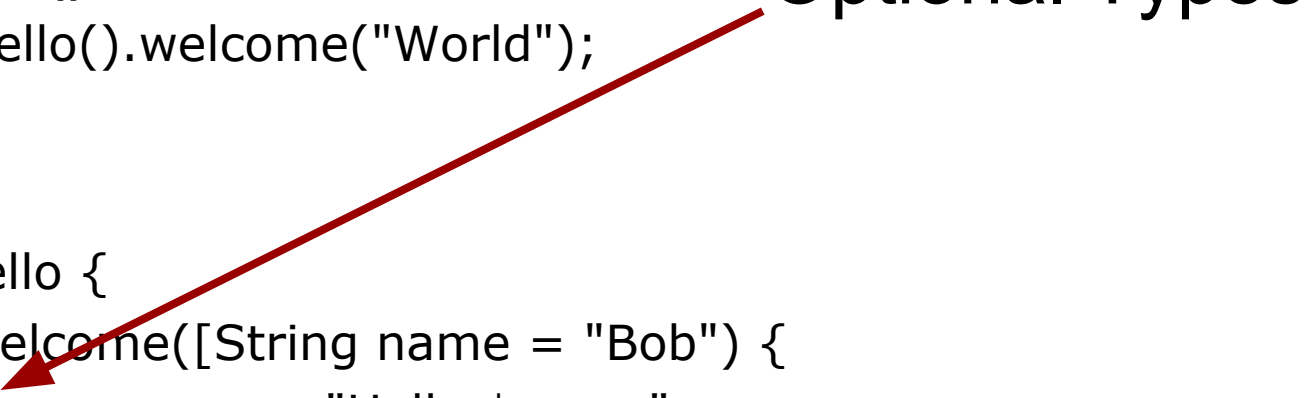
```
import 'dart:html';

void main() {
  new Hello().welcome("World");
}


class Hello {
  void welcome([String name = "Bob") {
    var message = "Hello $name";
    document.query('#status')
        ..text = message
        ..on.click.add(handler);
  }
}
```



http://www.dartlang.org

Hello World

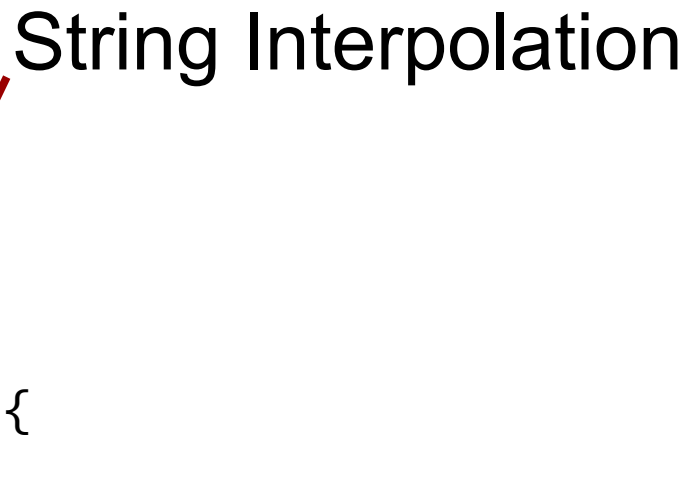
```
import 'dart:html';

void main() {
  new Hello().welcome("World");
}


class Hello {
  void welcome([String name = "Bob") {
    String message = "Hello $name";
    document.query('#status')
        ..text = message
        ..on.click.add(handler);
  }
}
```



http://www.dartlang.org

Hello World

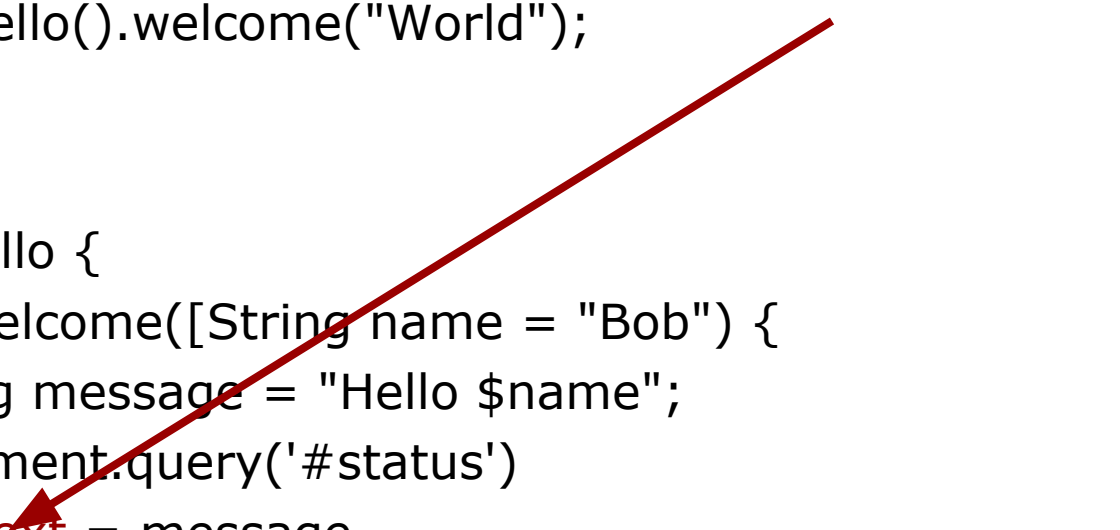
```
import 'dart:html';

void main() {
  new Hello().welcome("World");
}


class Hello {
  void welcome([String name = "Bob") {
    String message = "Hello $name";
    document.query('#status')
        ..text = message
        ..on.click.add(handler);
  }
}
```



http://www.dartlang.org

Hello World

```
import 'dart:html';

void main() {
  new Hello().welcome("World");
}


class Hello {
  void welcome([String name = "Bob") {
    String message = "Hello $name";
    document.query('#status')
        ..text = message
        ..on.click.add(handler);
  }
}
```

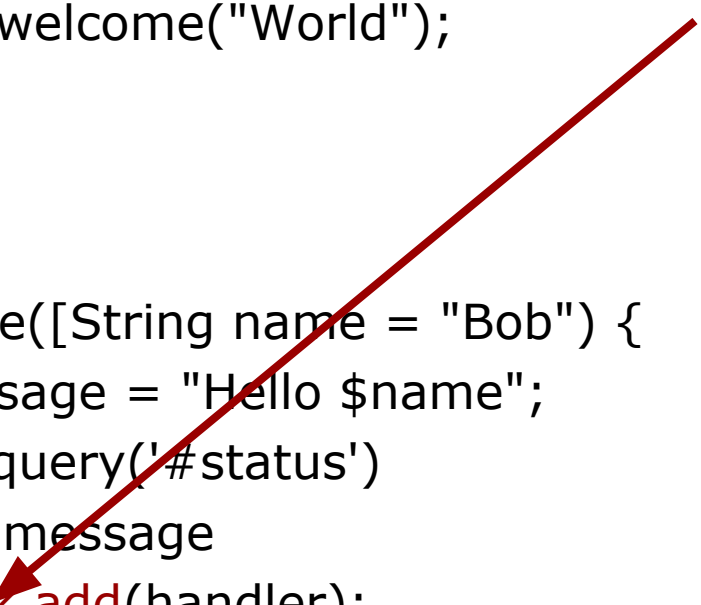


http://www.dartlang.org

Hello World

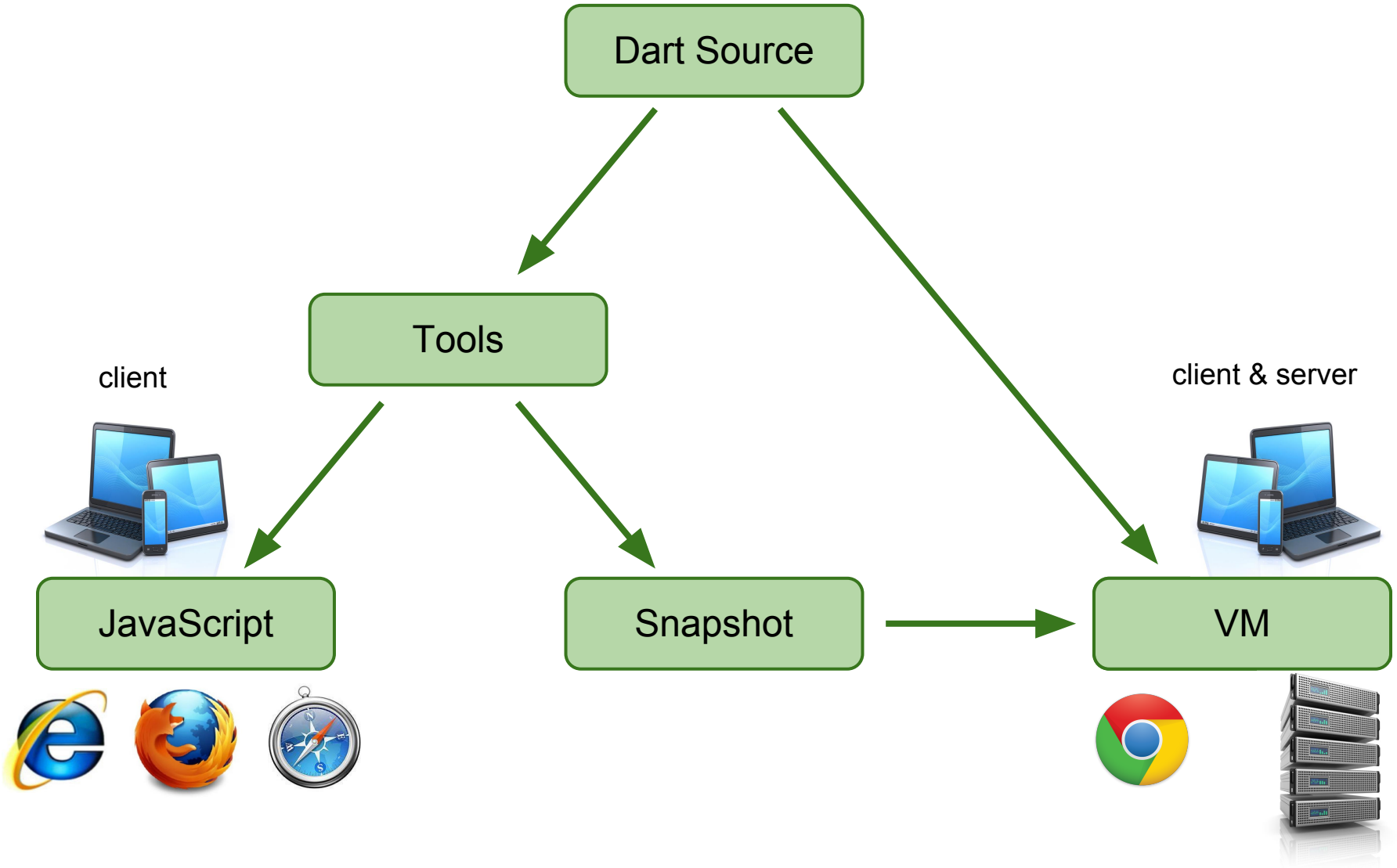
```
import 'dart:html';

void main() {
  new Hello().welcome("World");
}


class Hello {
  void welcome([String name = "Bob") {
    String message = "Hello $name";
    document.query('#status')
        ..text = message
        ..on.click.add(handler);
  }
}
```

Simple DOM API

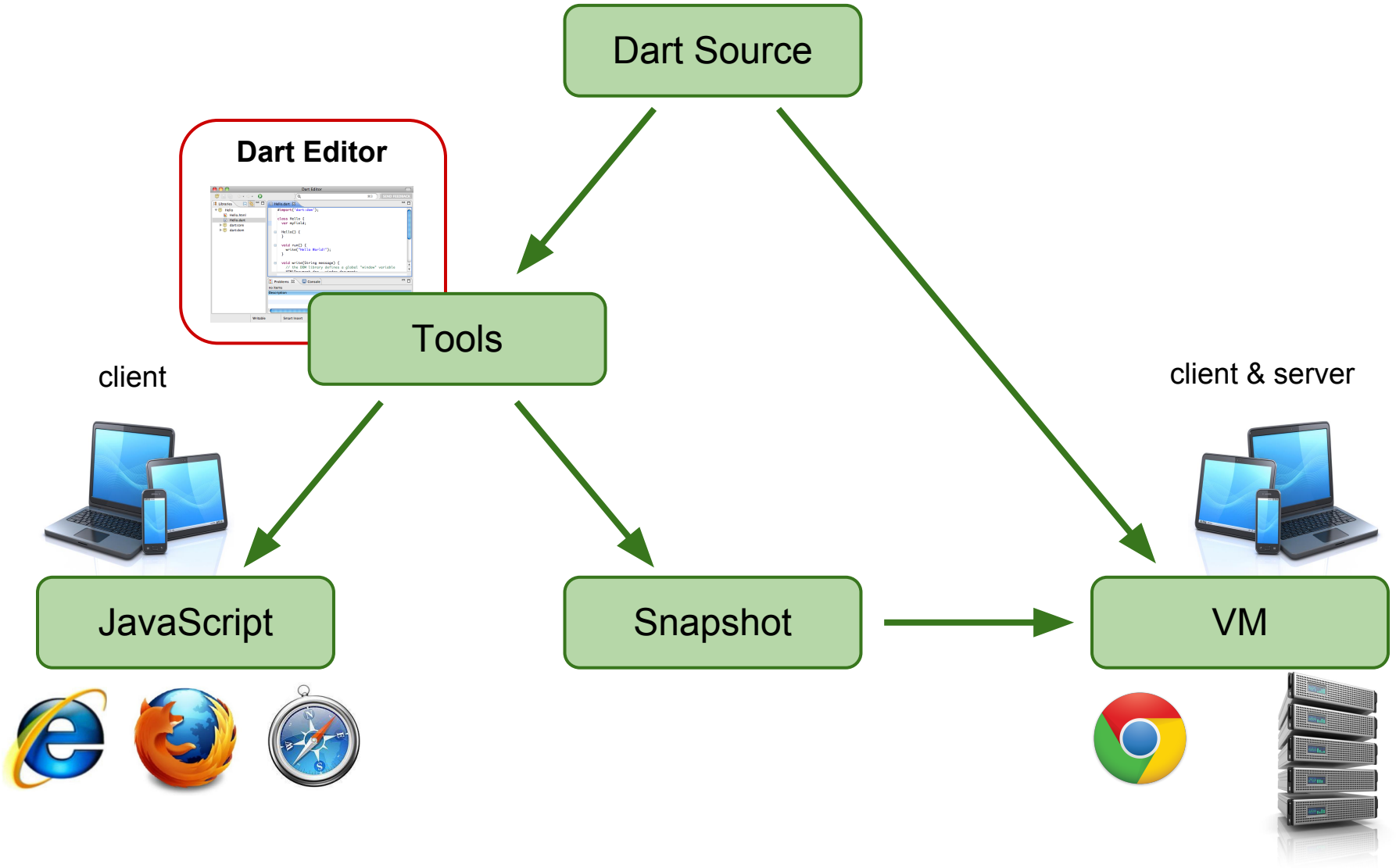


http://www.dartlang.org

Tools

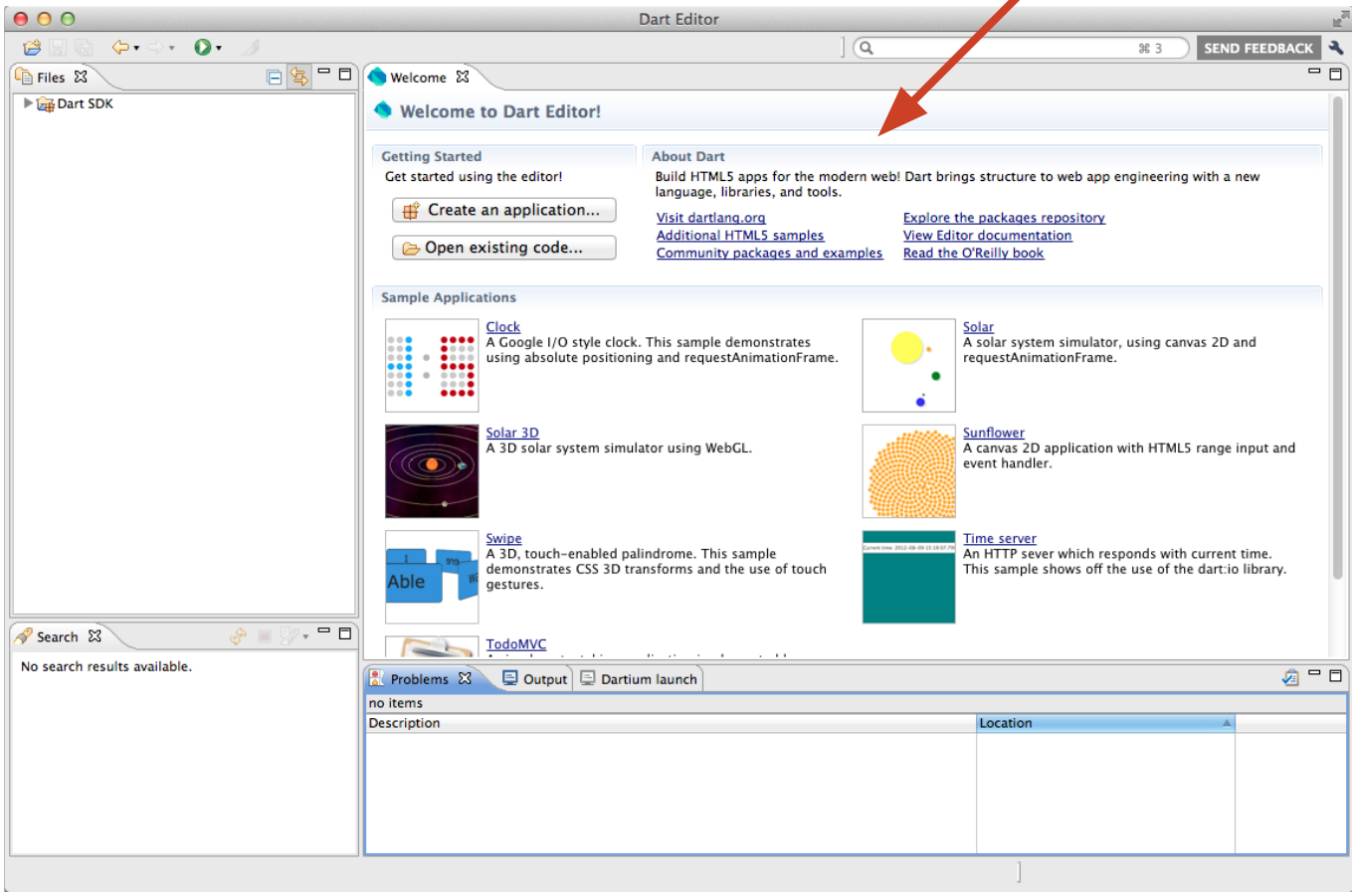

Dart Source
Tools
client
client & server
JavaScript
Snapshot
VM

Tools

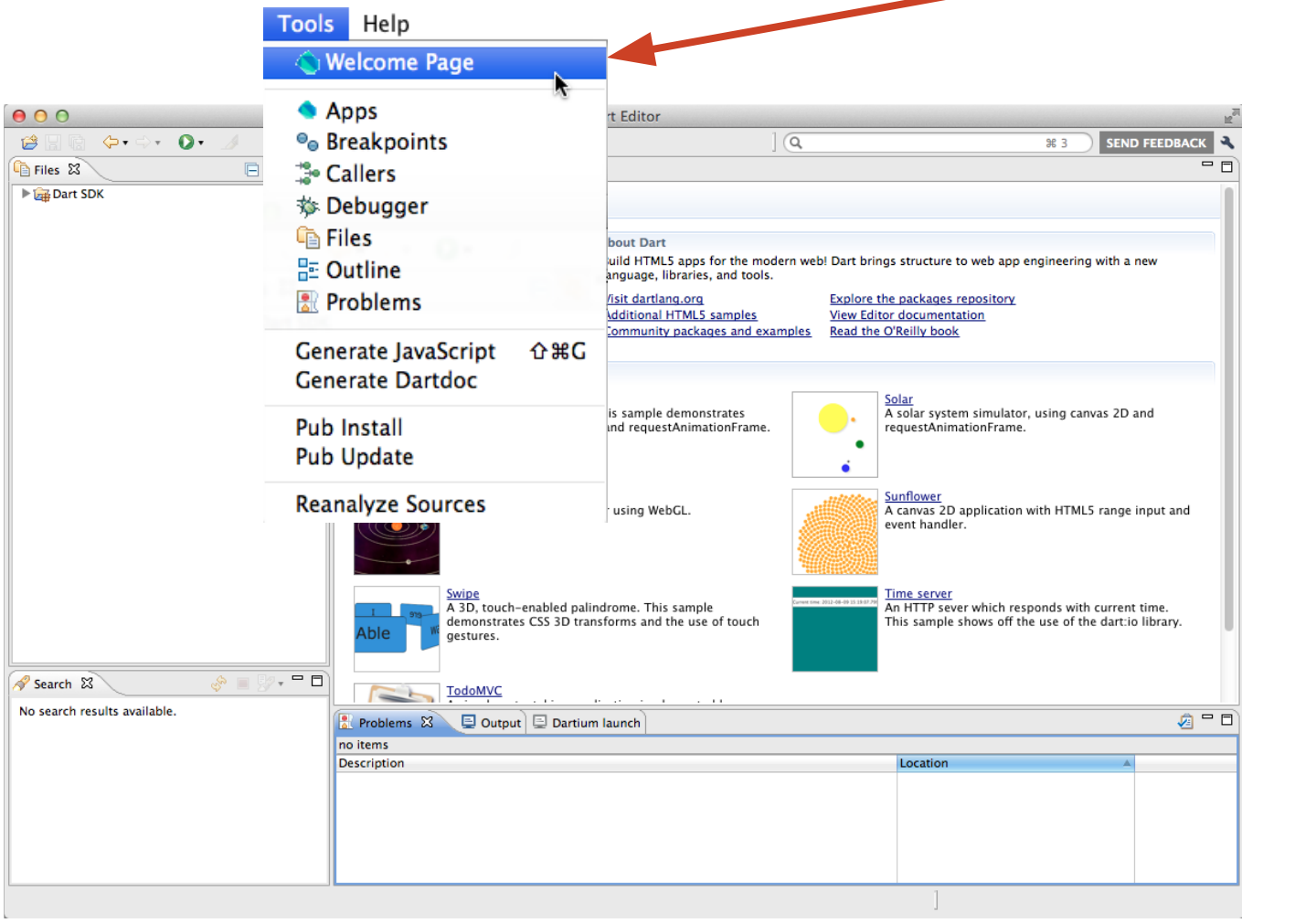

Dart Source
Dart Editor
Tools
client
client & server
JavaScript
Snapshot
VM

Dart Editor

Welcome Page

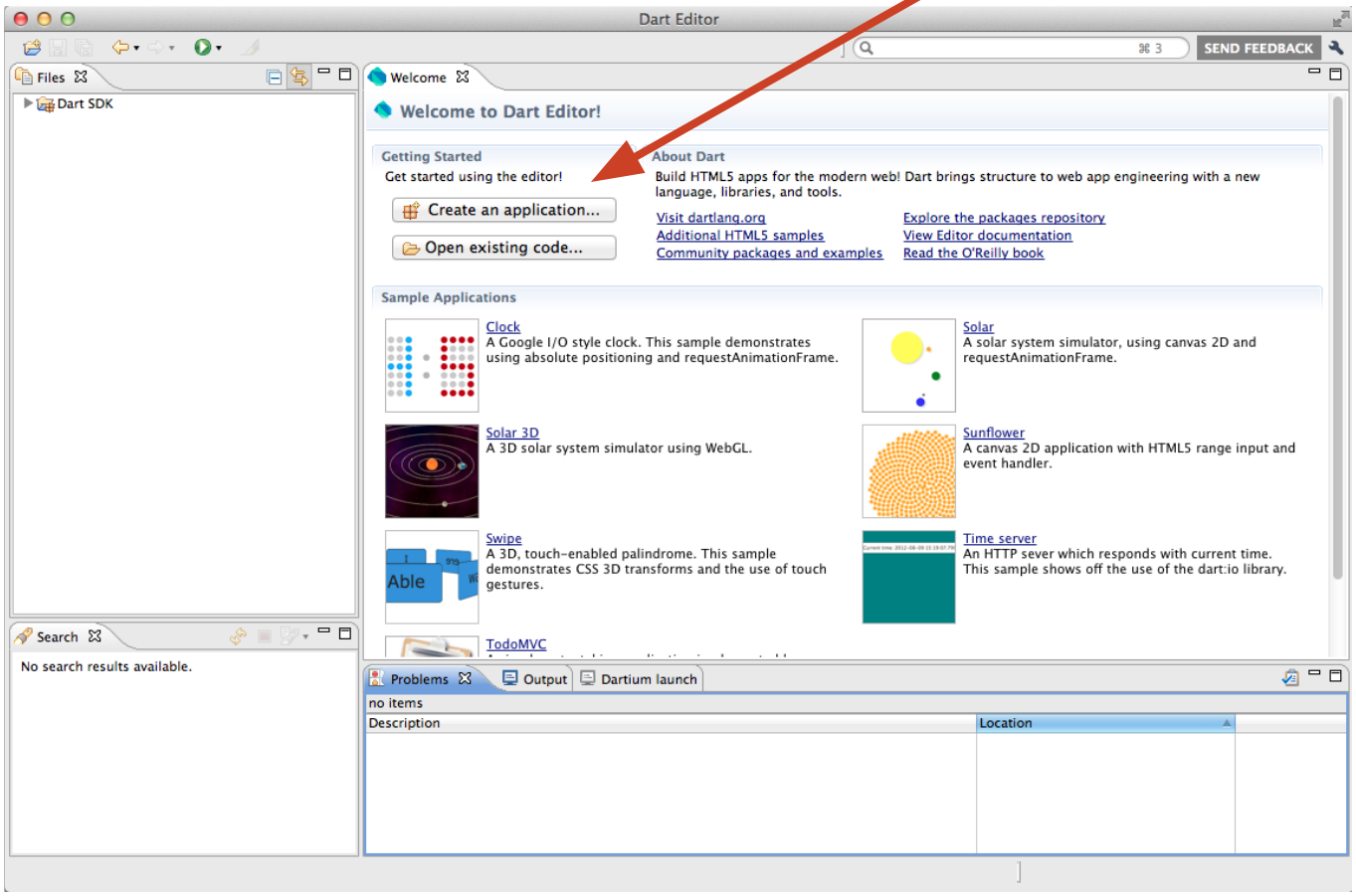


Dart Editor


Welcome Page
Tools
Help
Welcome Page
Apps
Breakpoints
Callers
Debugger
Files
Outline
Problems
Generate JavaScript ⇧⌘G
Generate Dartdoc
Pub Install
Pub Update
Reanalyze Sources
Files
Dart SDK
Search
No search results available.
SEND FEEDBACK
Solar
A solar system simulator, using canvas 2D and requestAnimationFrame.
Sunflower
A canvas 2D application with HTML5 range input and event handler.
Swipe
A 3D, touch-enabled palindrome. This sample demonstrates CSS 3D transforms and the use of touch gestures.
Time server
An HTTP sever which responds with current time. This sample shows off the use of the dart:io library.
TodoMVC
Problems
Output
Dartium launch
no items
Description
Location

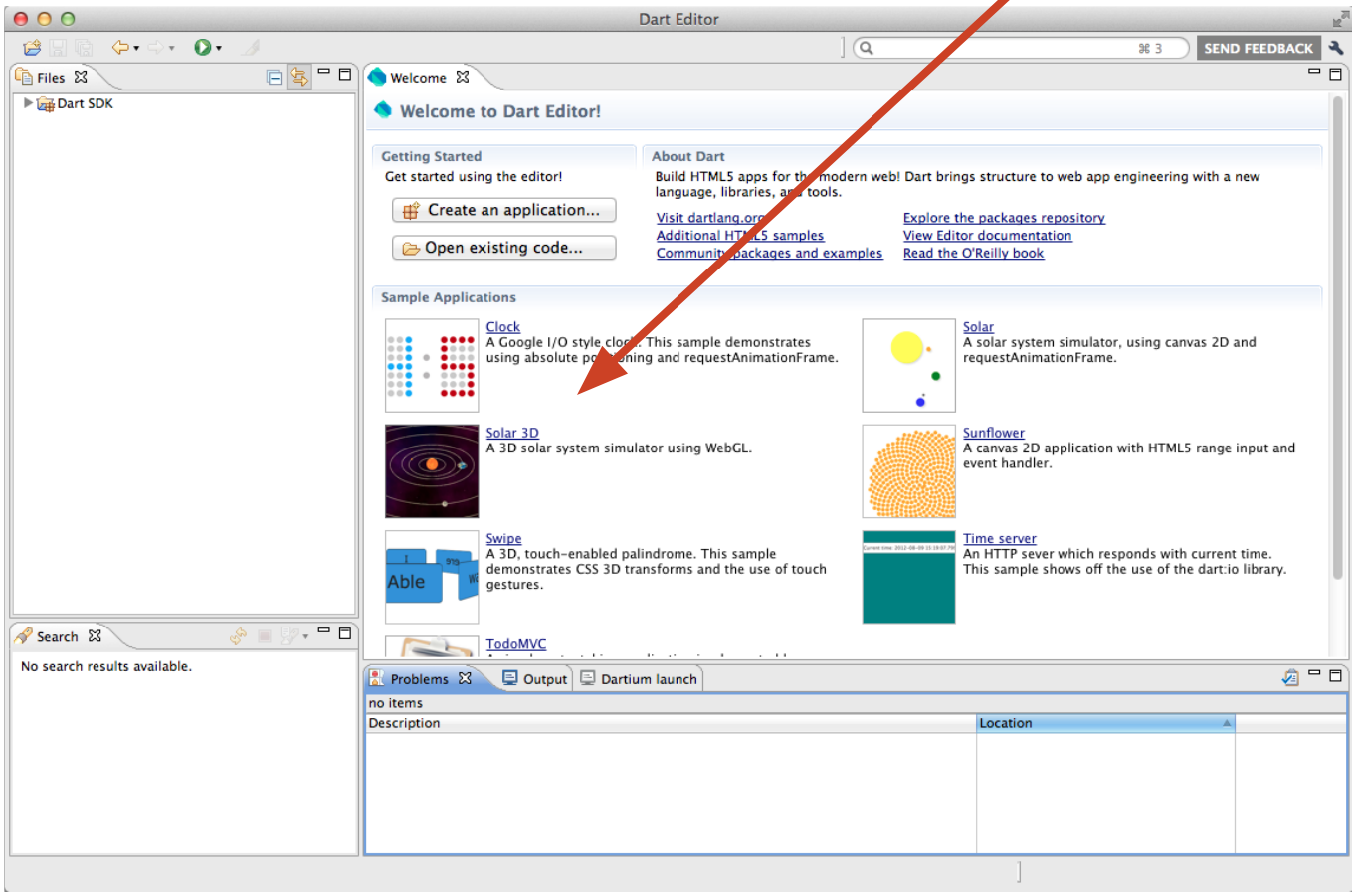
Dart Editor

Create application



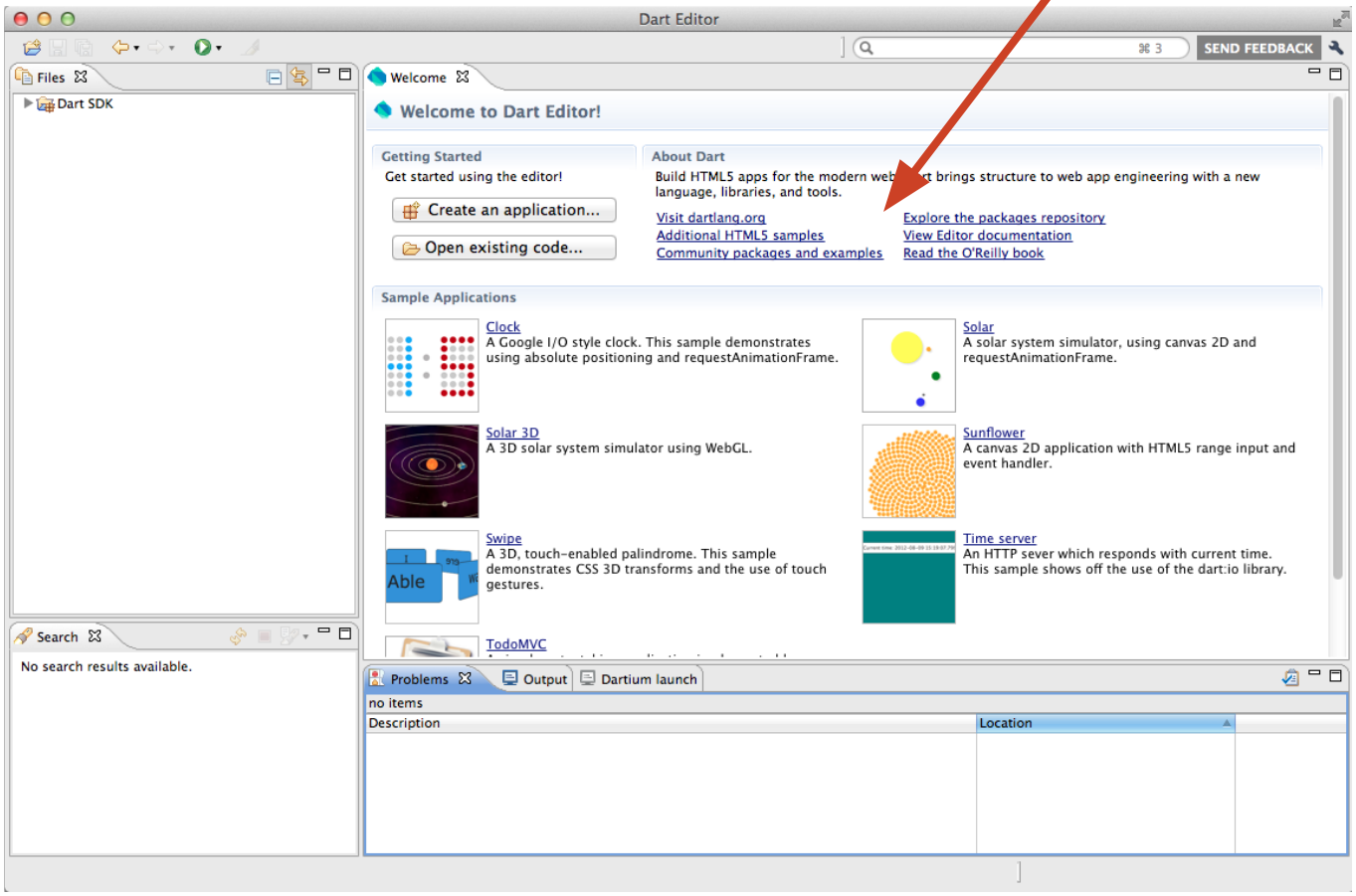
Dart Editor

Load Sample Code

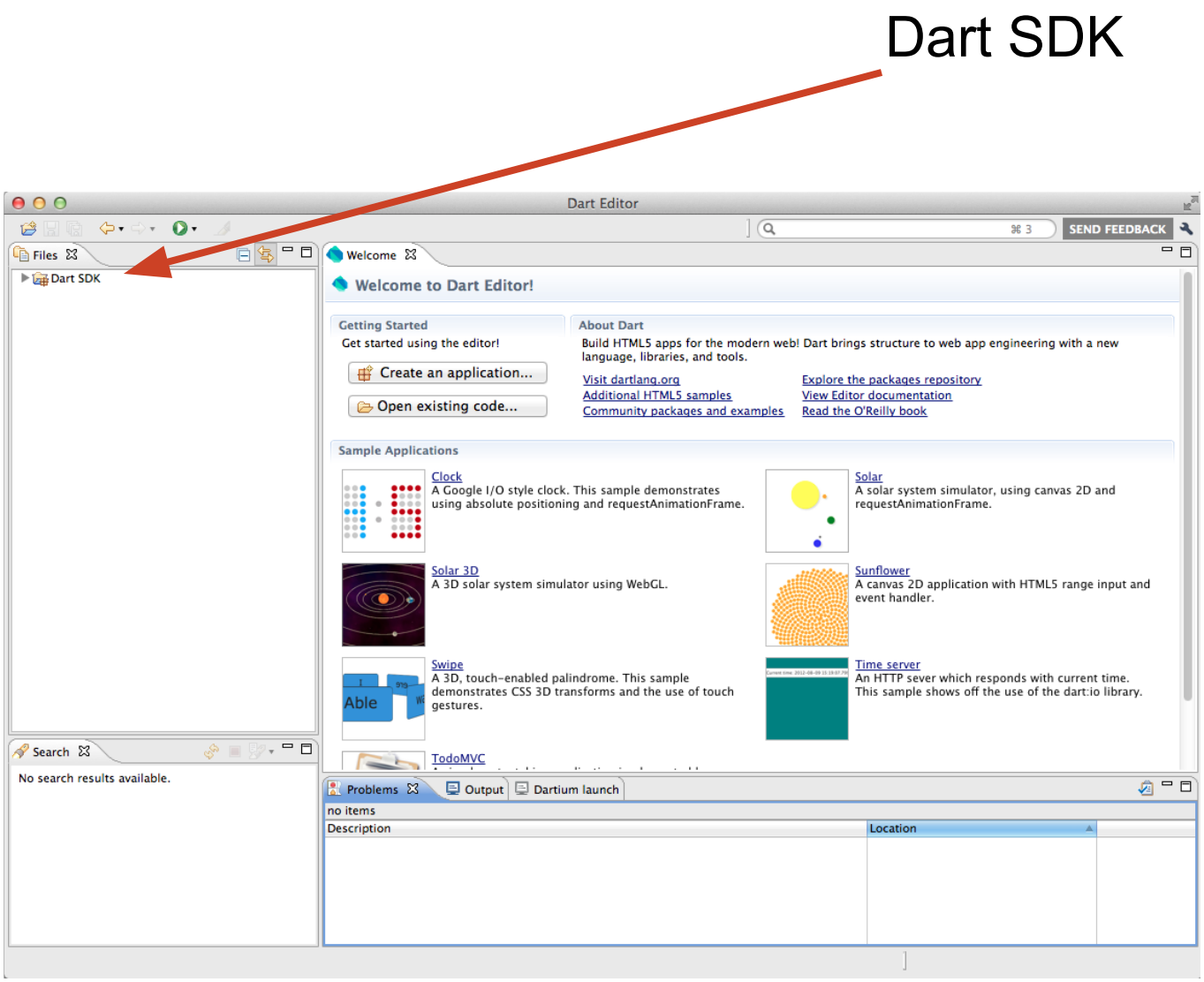


Dart Editor

More Information

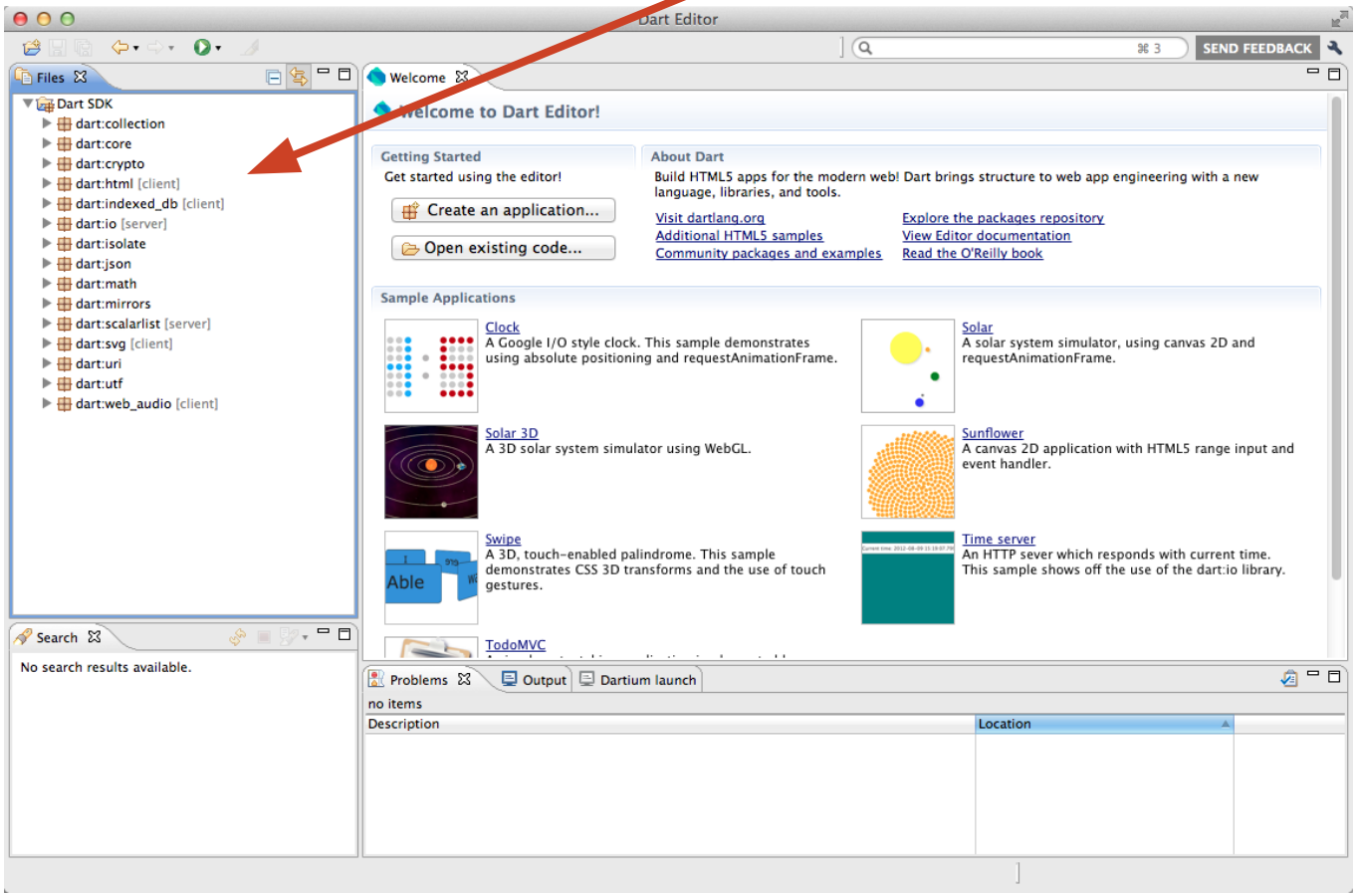


Dart Editor


Dart SDK
Dart Editor
SEND FEEDBACK
Files
Dart SDK
Welcome
Welcome to Dart Editor!
Getting Started
Get started using the editor!
Create an application...
Open existing code...
About Dart
Build HTML5 apps for the modern web! Dart brings structure to web app engineering with a new language, libraries, and tools.
Visit dartlang.org
Additional HTML5 samples
Community packages and examples
Explore the packages repository
View Editor documentation
Read the O'Reilly book
Sample Applications
Clock
A Google I/O style clock. This sample demonstrates using absolute positioning and requestAnimationFrame.
Solar
A solar system simulator, using canvas 2D and requestAnimationFrame.
Solar 3D
A 3D solar system simulator using WebGL.
Sunflower
A canvas 2D application with HTML5 range input and event handler.
Swipe
A 3D, touch-enabled palindrome. This sample demonstrates CSS 3D transforms and the use of touch gestures.
Able
Time server
An HTTP sever which responds with current time. This sample shows off the use of the dart:io library.
TodoMVC
Search
No search results available.
Problems
Output
Dartium launch
no items
Description
Location

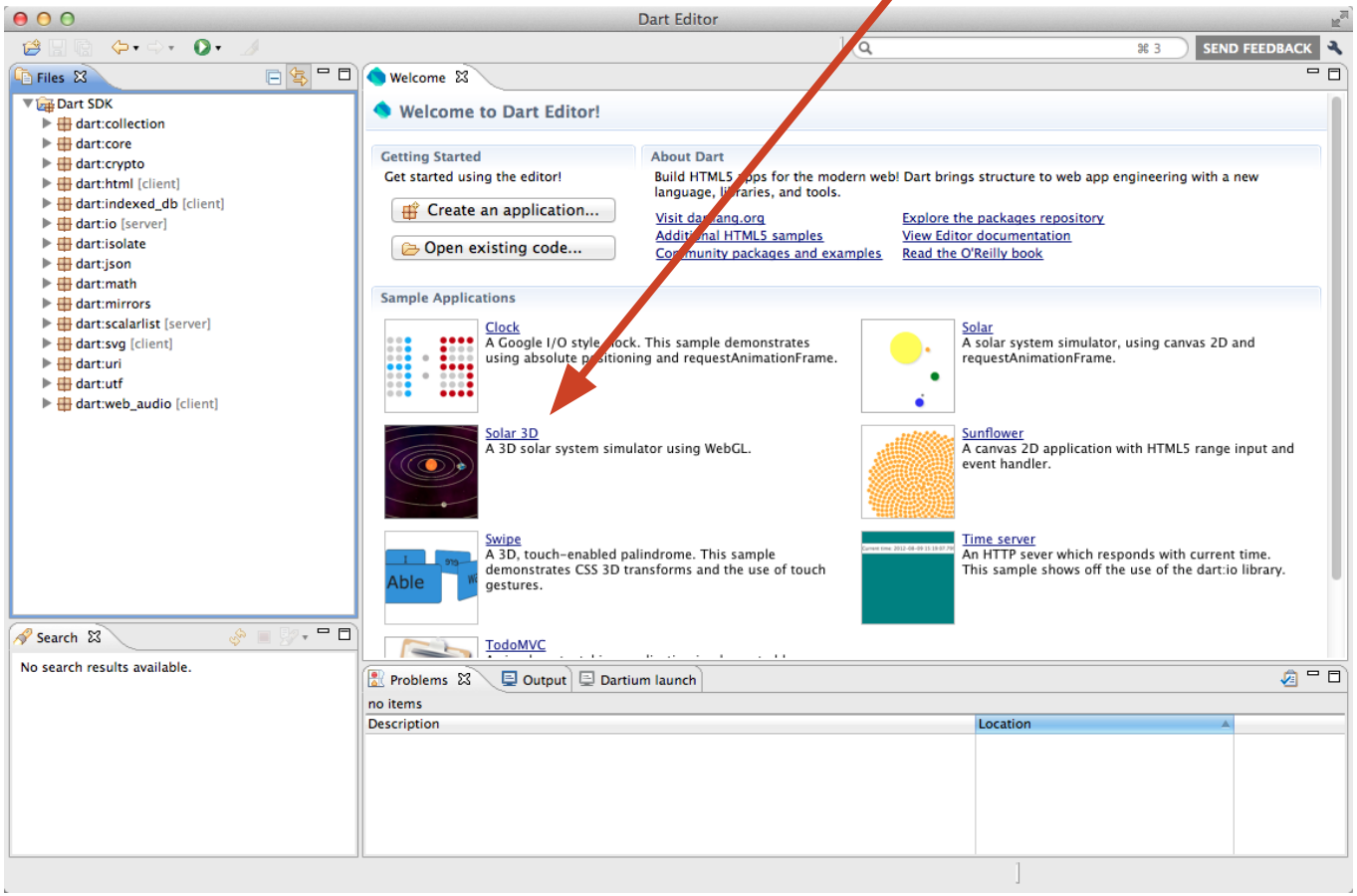
Dart Editor

Dart SDK Libraries



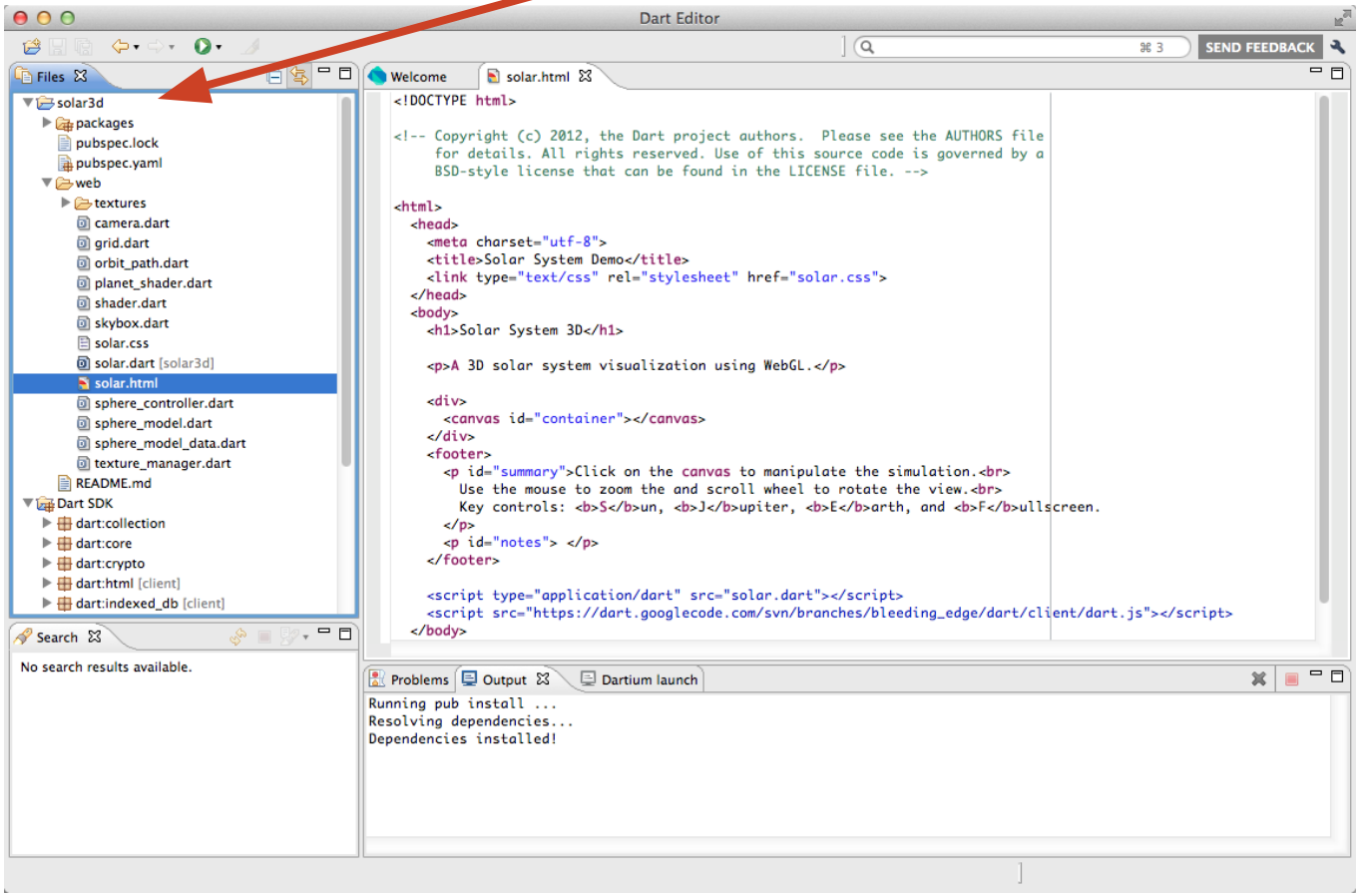
Dart Editor

Load Solar 3D Sample



Dart Editor

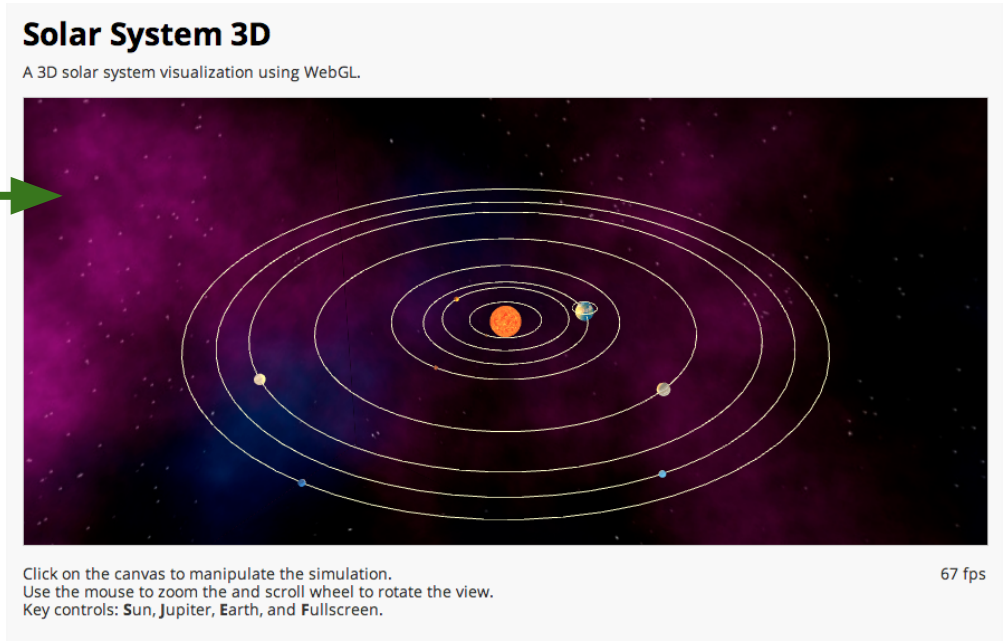
Solar 3D Sample



Demo

Solar 3D

- Dartium
- JavaScript
- Debugging

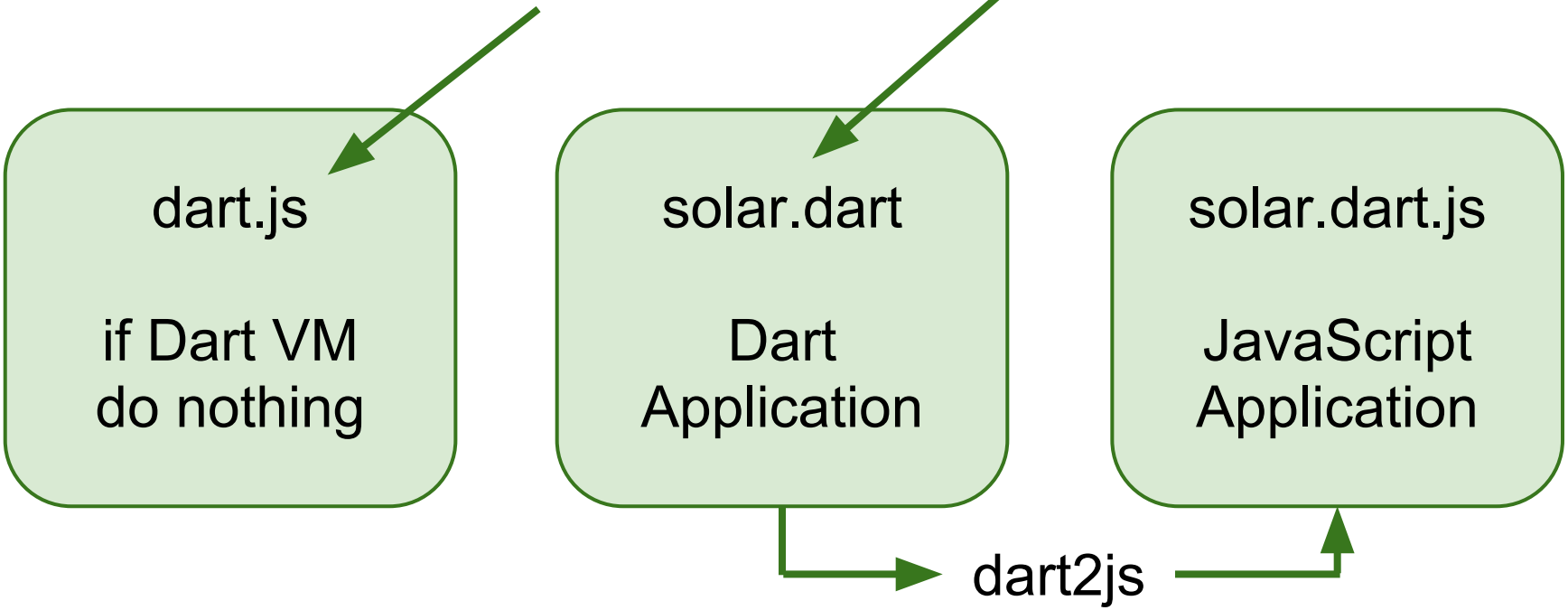


Source → dart2js

HTML + Dart

Dartium

```
<html>
  <body>
    <script type="application/dart" src="solar.dart"/>
    <script type=".../dart.js"/>
```



HTML + Dart

Others

```
<html>
  <body>
    <script type="application/dart" src="solar.dart.js"/>
    <script type=".../dart.js"/>
```

dart.js

if no Dart VM
rewrite DOM

solar.dart

Dart
Application

solar.dart.js

JavaScript
Application

dart2js

dart2js

Converts Dart source to JavaScript

- Targets ES5+ (modern browsers)
- Tree shaking and dead code elimination
- Written in Dart

In progress:

- Smaller JavaScript output
- Performance improvements

Pub - package manager

Declaration

- pubspec.yaml defines the package

Operations

- pub install installs dependencies
- pub update updates dependencies

See packages on **http://pub.dartlang.org**

Pub - pubspec.yaml

```
name: my_app
description: some application
version: 1.2.7
author: Bob <bob@smith.org>
homepage: http://www.smith.org/...
dependencies:
   one_package: any
   another_package: "1.2.1"
   web_components: ">=0.2.8+4 <0.2.9"
```

Pub - layout

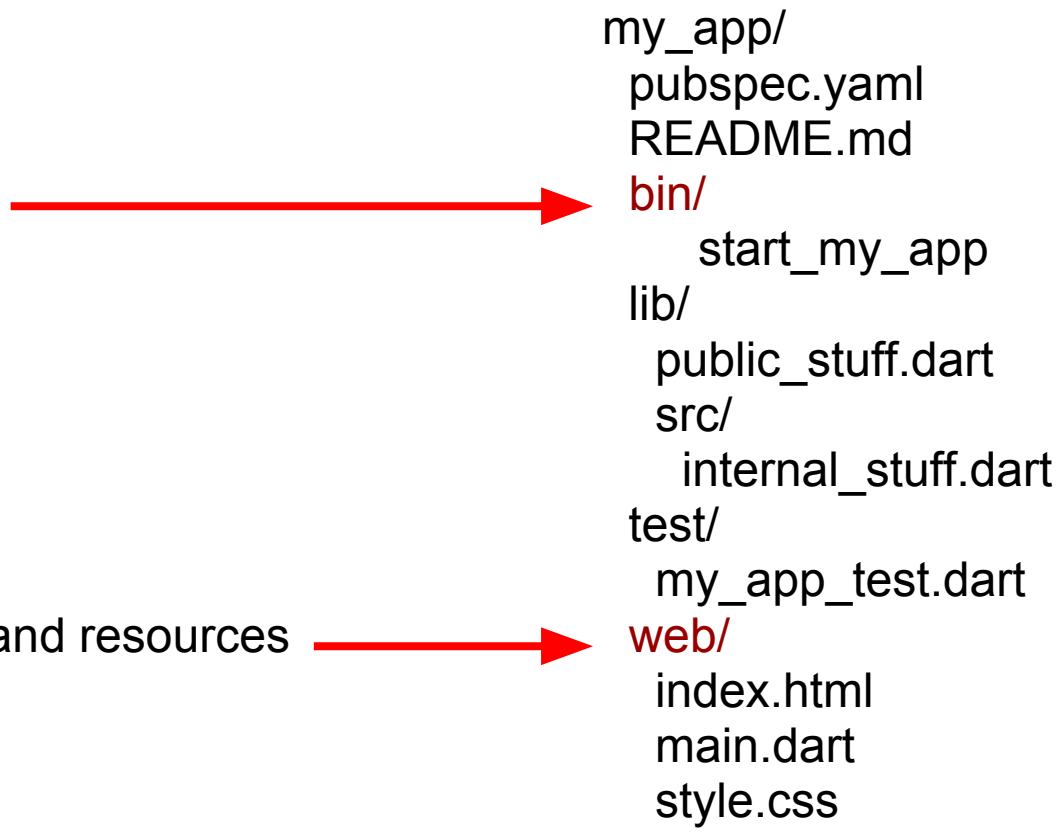
package declaration

```
my_app/
  pubspec.yaml
  README.md
  bin/
      start_my_app
  lib/
    public_stuff.dart
    src/
      internal_stuff.dart
  test/
    my_app_test.dart
  web/
    index.html
    main.dart
    style.css
```

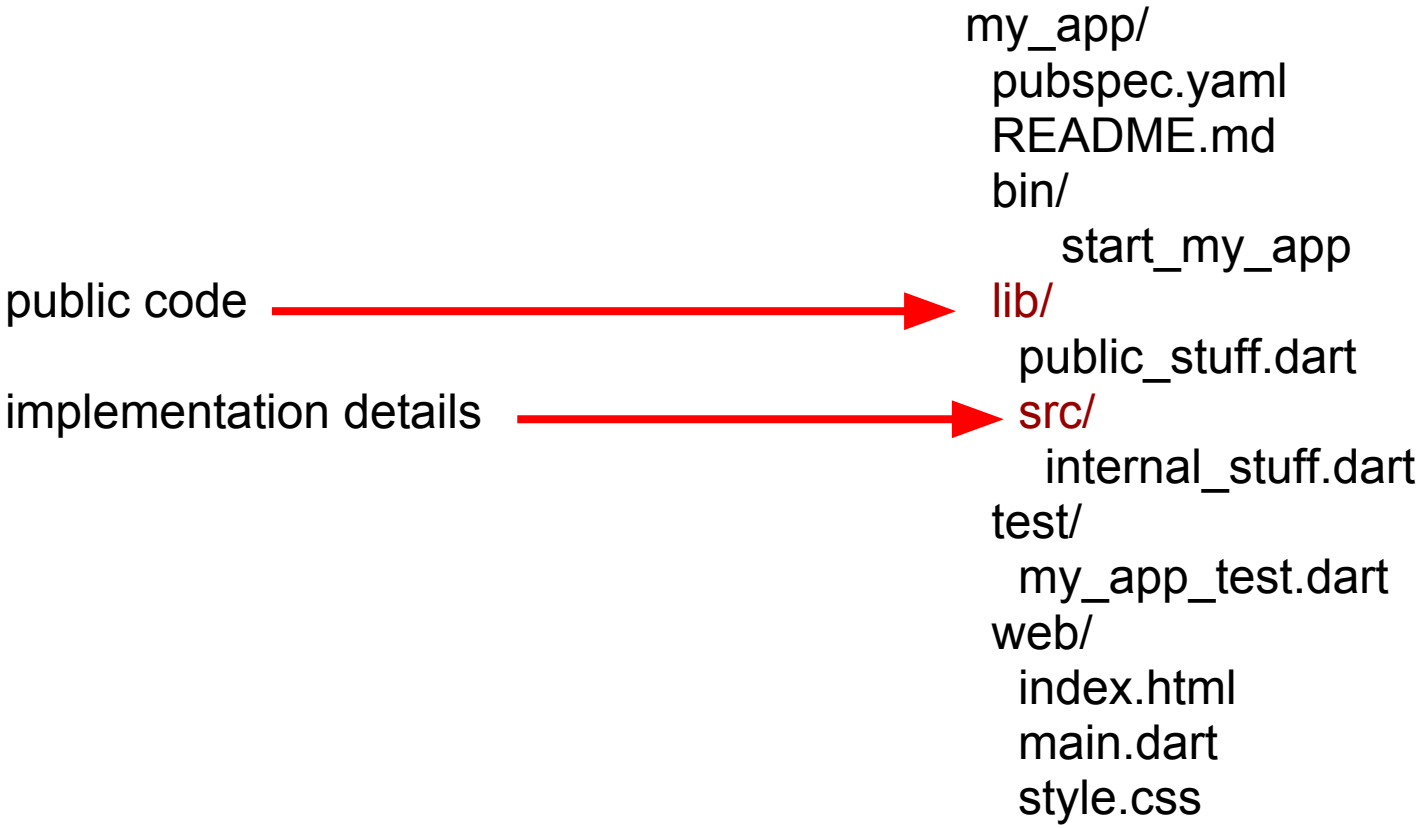
Pub - layout

server side code

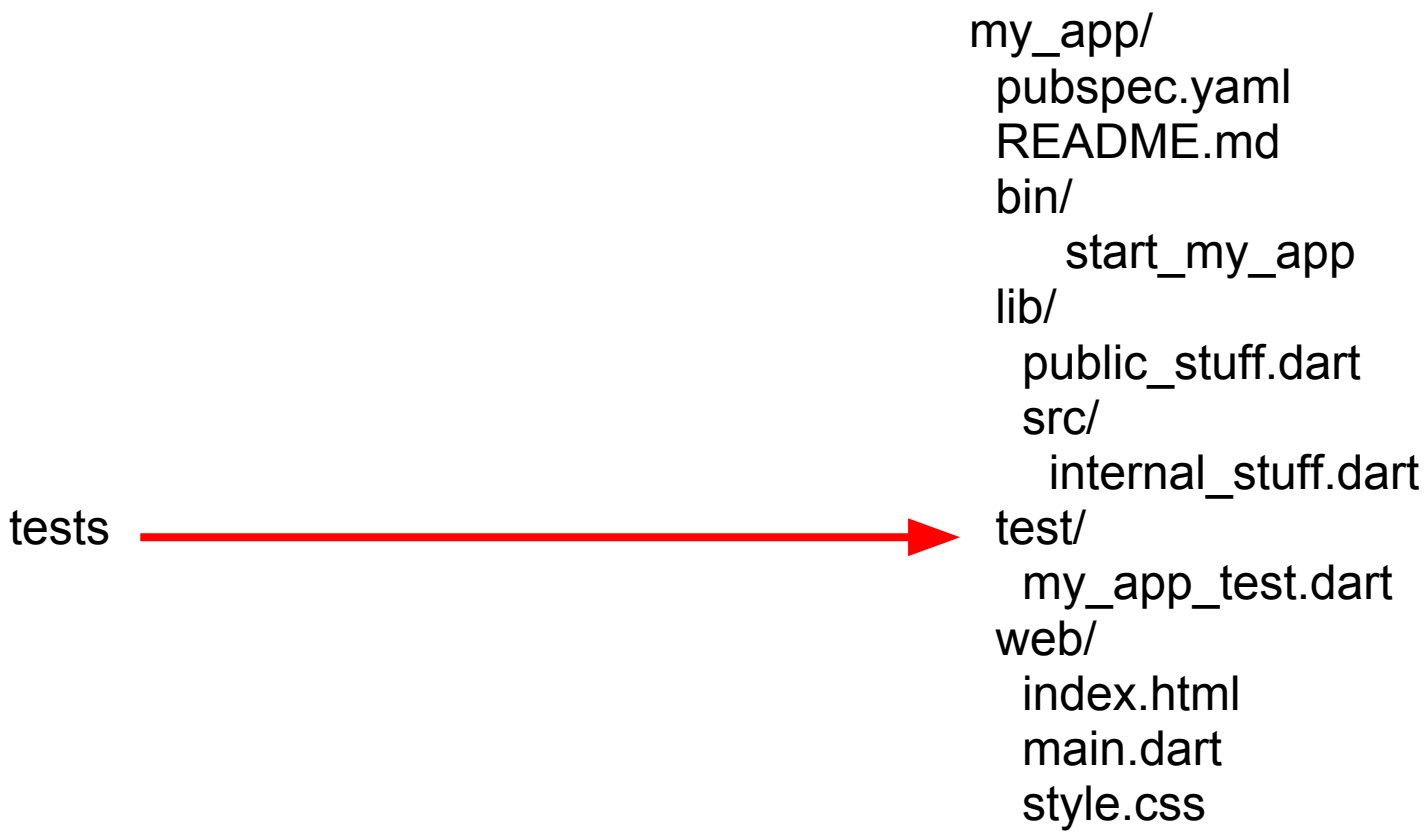
client side code and resources



Pub - layout


public code
implementation details
my_app/
pubspec.yaml
README.md
bin/
start_my_app
lib/
public_stuff.dart
src/
internal_stuff.dart
test/
my_app_test.dart
web/
index.html
main.dart
style.css

Pub - layout


my_app/
pubspec.yaml
README.md
bin/
start_my_app
lib/
public_stuff.dart
src/
internal_stuff.dart
test/
my_app_test.dart
web/
index.html
main.dart
style.css
tests

Server-sid 

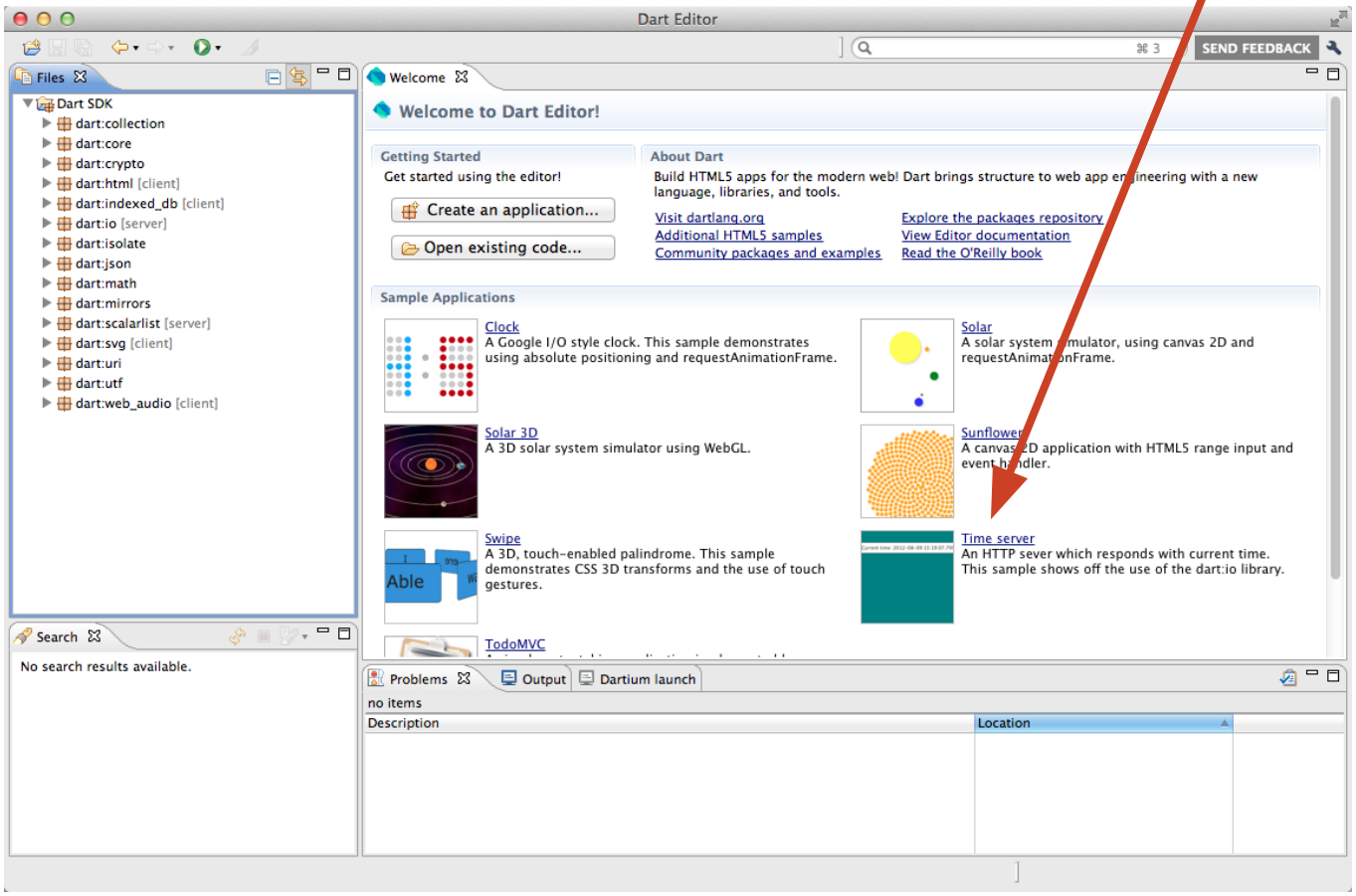
http://www.dartlang.org

Dart on the server

- File system
- Sockets
- HTTP server and client
- Web sockets server and client
- Async or Future style
- Share code on client and server

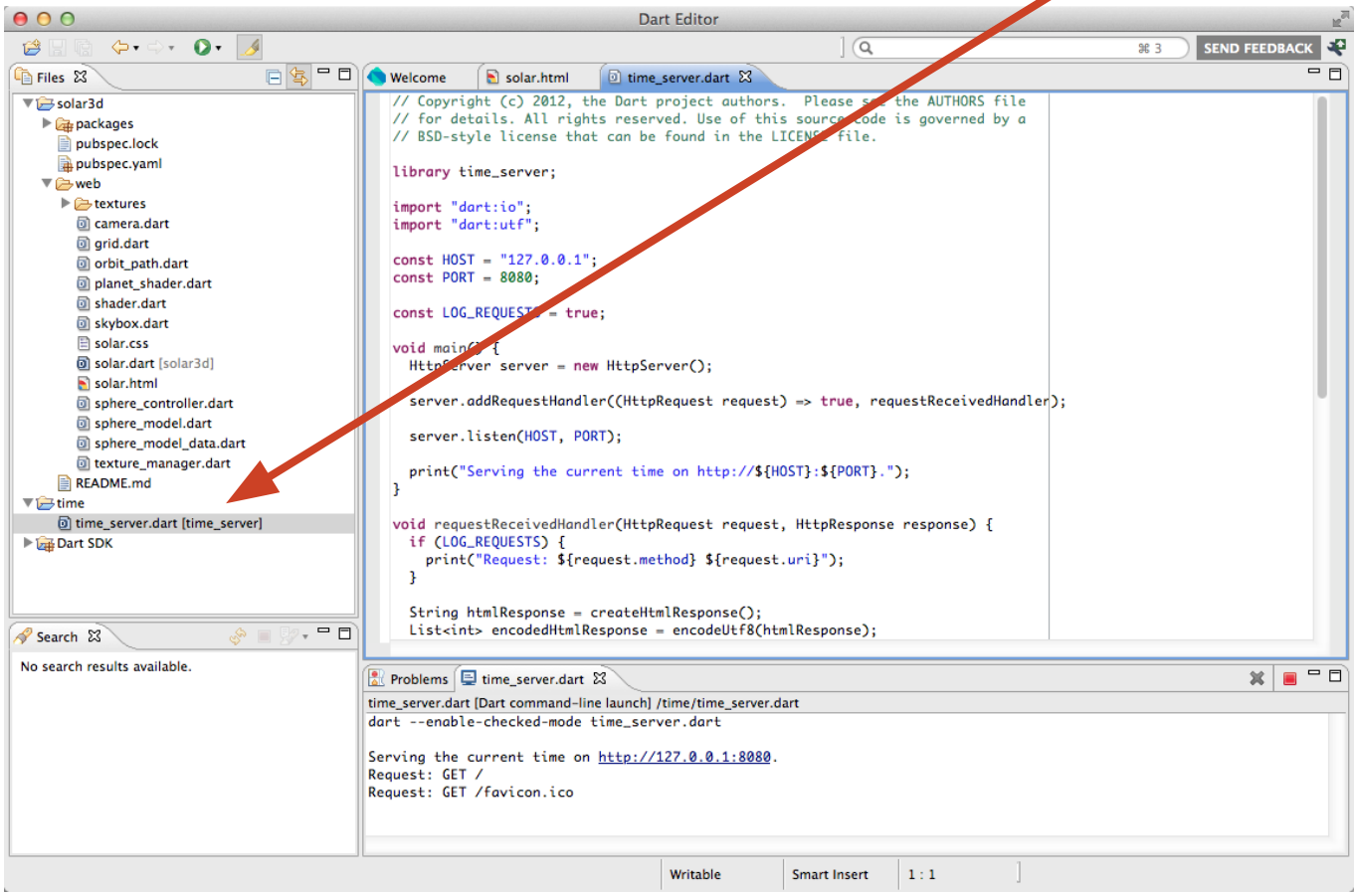
Dart on the server

Time Server Sample

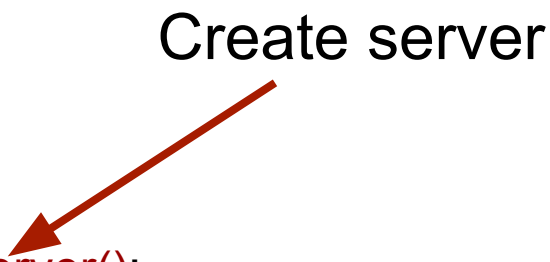


Dart on the server

Time Server Sample



Dart on the server




```
void main() {
  HttpServer server = new HttpServer();
  server
      ..addRequestHandler((HttpRequest request) => true, handler)
      ..listen(HOST, PORT);
  print("Serving the current time on http://${HOST}:${PORT}.");
}
void handler(HttpRequest request, HttpResponse response) {
  // process request ... send response
}
```

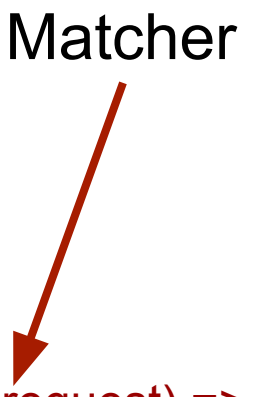
http://www.dartlang.org

Dart on the server

Add multiple handlers

```
void main() {
  HttpServer server = new HttpServer();
  server
      ..addRequestHandler((HttpRequest request) => true, handler)
      ..listen(HOST, PORT);
  print("Serving the current time on http://${HOST}:${PORT}.");
}
void handler(HttpRequest request, HttpResponse response) {
  // process request ... send response
}
```

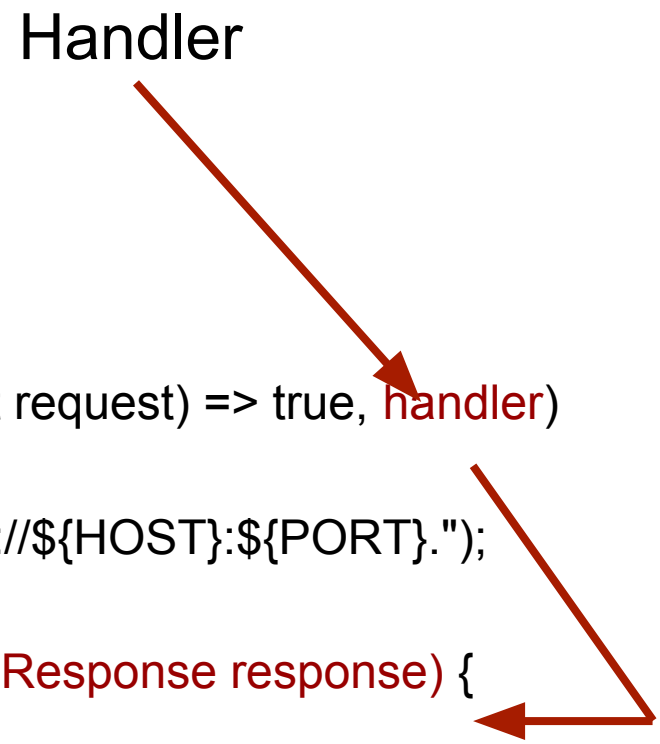
Dart on the server




```
void main() {
  HttpServer server = new HttpServer();
  server
      ..addRequestHandler((HttpRequest request) => true, handler)
      ..listen(HOST, PORT);
  print("Serving the current time on http://${HOST}:${PORT}.");
}
void handler(HttpRequest request, HttpResponse response) {
  // process request ... send response
}
```

http://www.dartlang.org

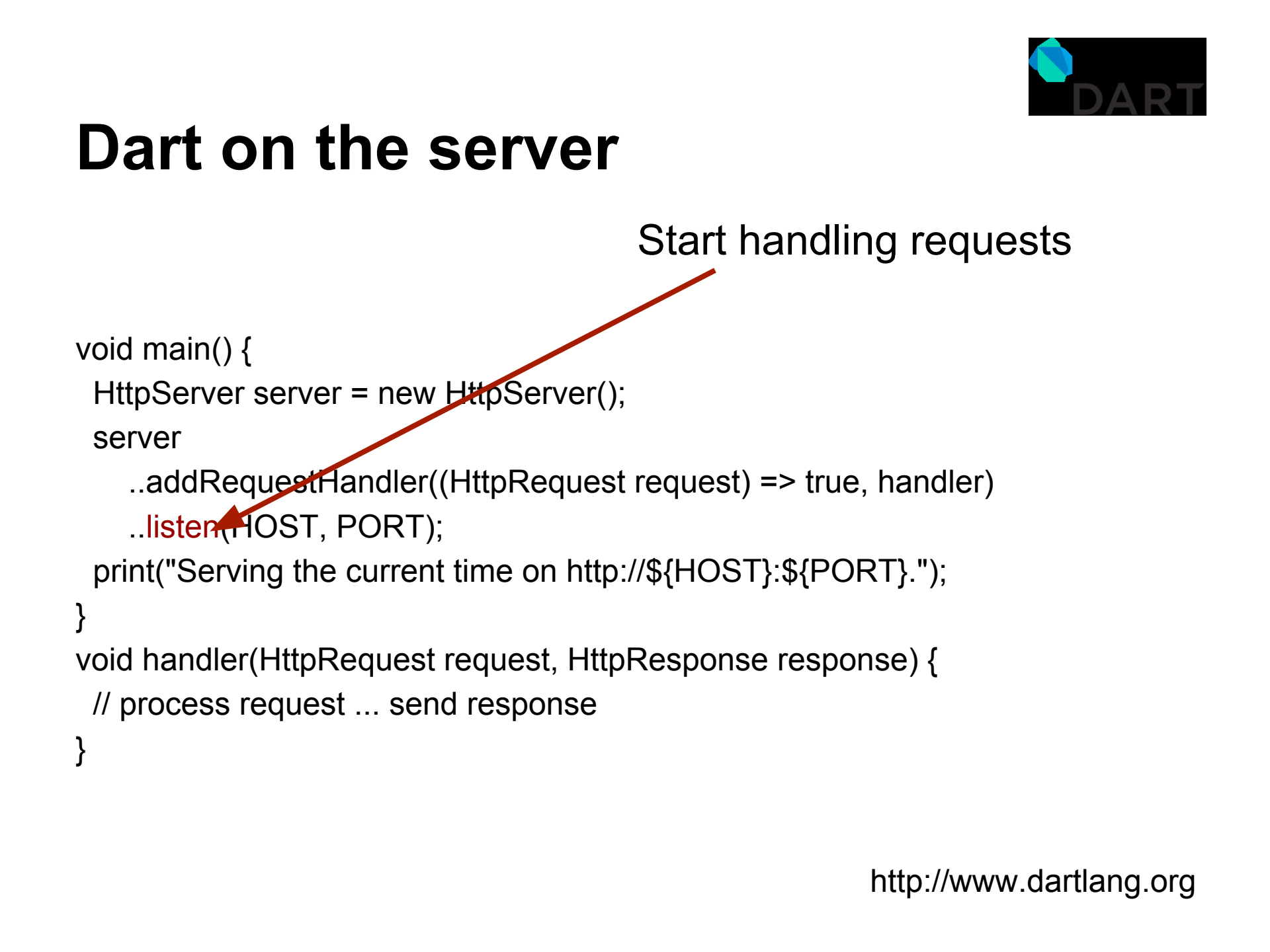
Dart on the server




```
void main() {
  HttpServer server = new HttpServer();
  server
      ..addRequestHandler((HttpRequest request) => true, handler)
      ..listen(HOST, PORT);
  print("Serving the current time on http://${HOST}:${PORT}.");
}
void handler(HttpRequest request, HttpResponse response) {
  // process request ... send response
}
```

http://www.dartlang.org

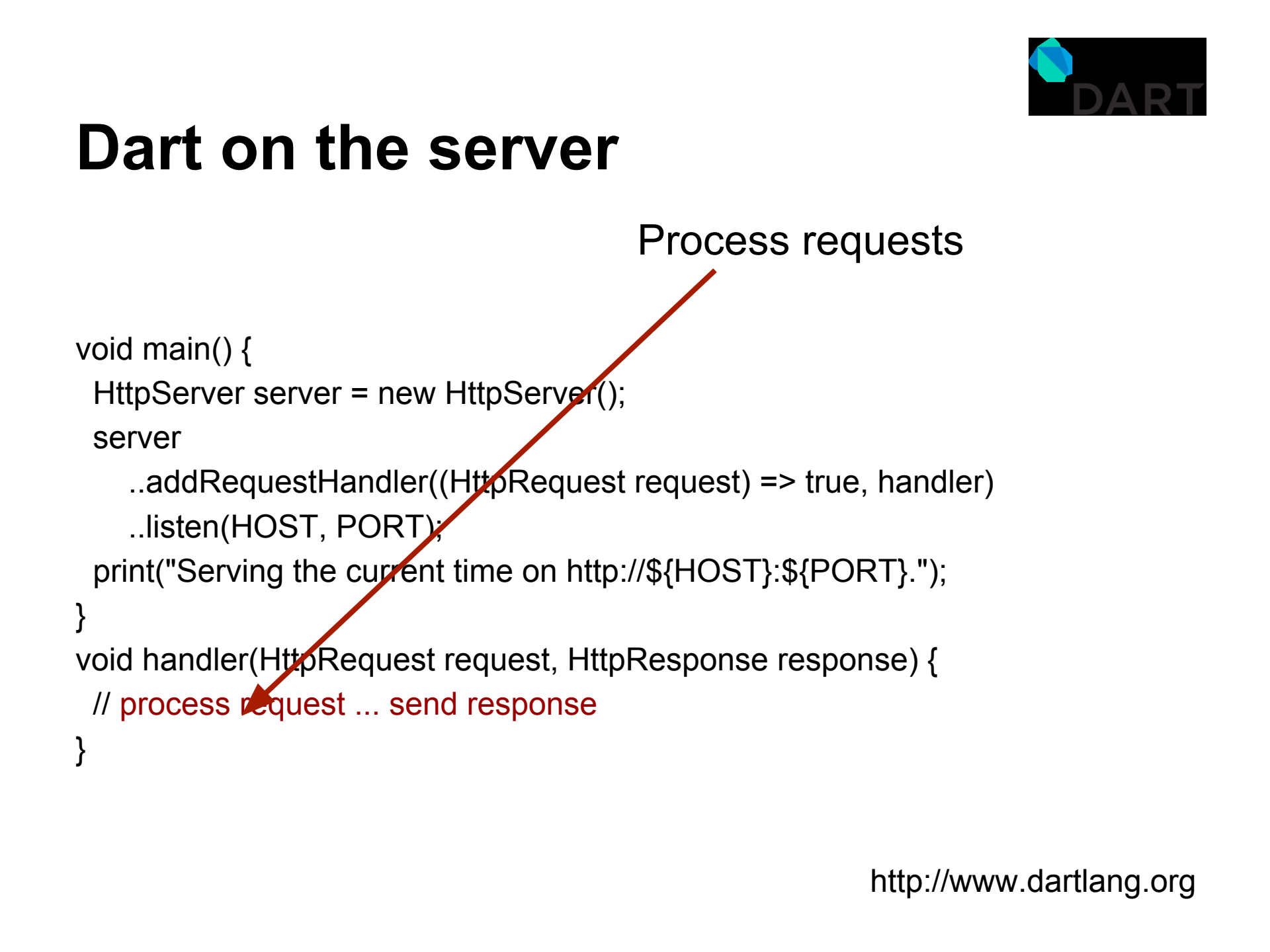
Dart on the server




```
void main() {
  HttpServer server = new HttpServer();
  server
      ..addRequestHandler((HttpRequest request) => true, handler)
      ..listen(HOST, PORT);
  print("Serving the current time on http://${HOST}:${PORT}.");
}
void handler(HttpRequest request, HttpResponse response) {
  // process request ... send response
}
```

http://www.dartlang.org

Dart on the server




```
void main() {
  HttpServer server = new HttpServer();
  server
      ..addRequestHandler((HttpRequest request) => true, handler)
      ..listen(HOST, PORT);
  print("Serving the current time on http://${HOST}:${PORT}.");
}
void handler(HttpRequest request, HttpResponse response) {
  // process request ... send response
}
```

http://www.dartlang.org

Developing Dart Editor

http://www.dartlang.org

Dart Editor - Users

- Web programmers of varying backgrounds
  - Many languages - HTML, JS, Python, Java
  - Wide range of programming experience

- Primarily **not** Eclipse users

Dart Editor - Goals

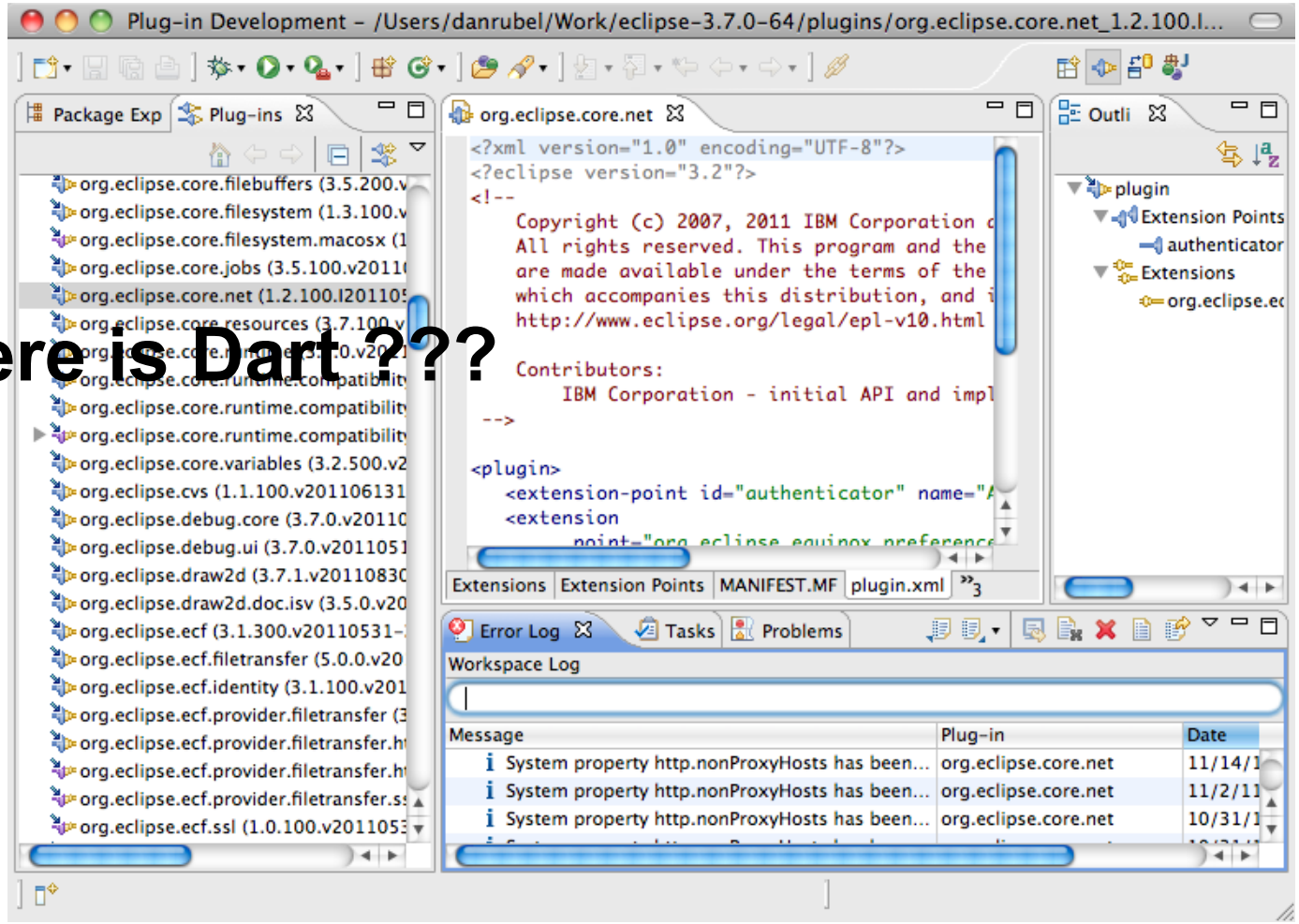
- Easy on-ramp to learn Dart
- Simplified UI

but also...

- Power tools
  - refactoring
  - quick fixes/assists
  - code completion
  - semantic search
  - ...and more

http://www.dartlang.org

Dart Editor - Before



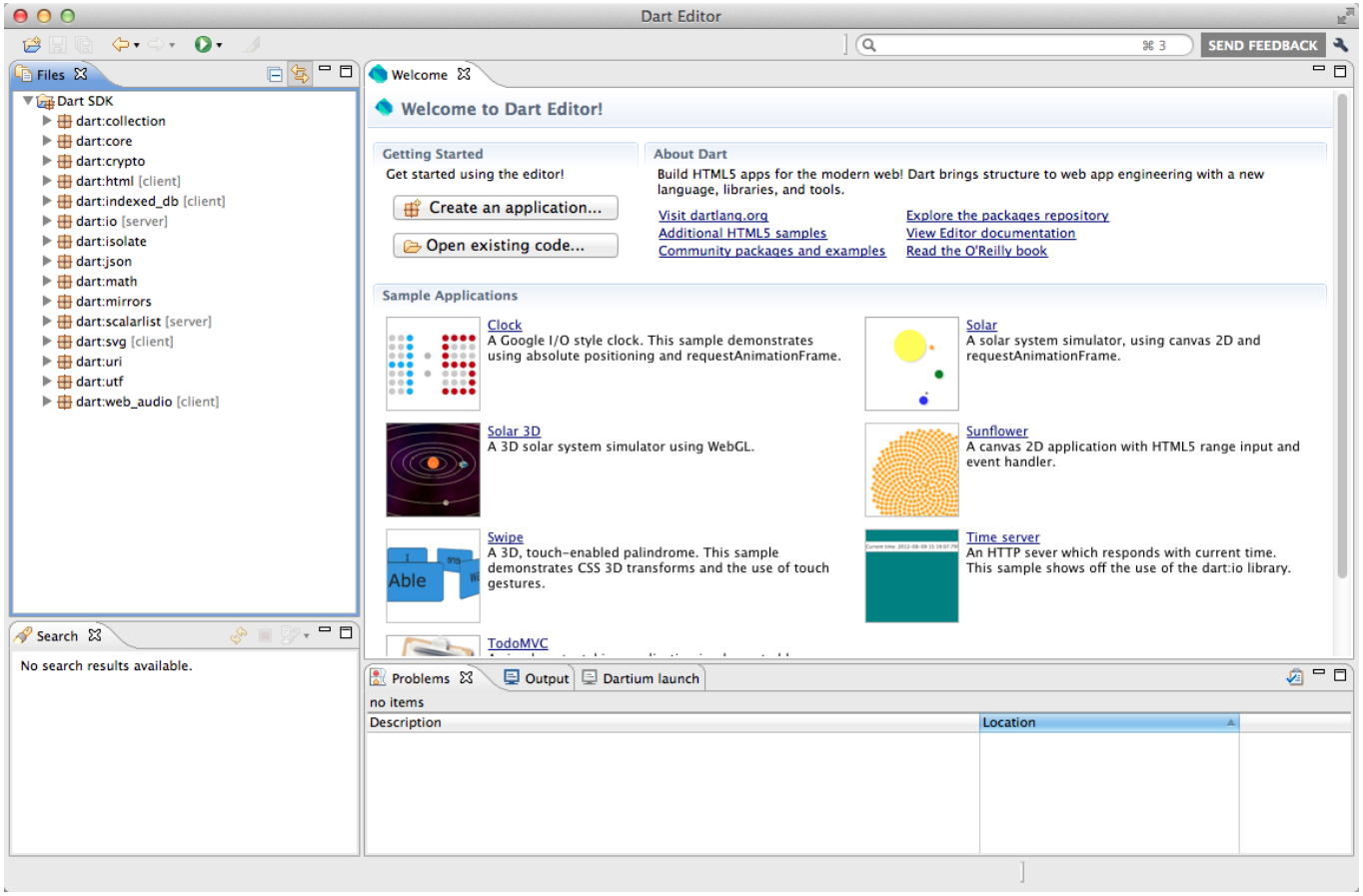
Where is Dart ???

Dart Editor - Strategy

- Narrow the scope
  - Focus on doing a few things well

- Minimalist UI
  - Make it easy to understand
  - Reduce decision making

Dart Editor - Now



Simple and Clean UI

How?

- Single perspective
- Remove unnecessary plugins
- Redefine entire menu bar
- Use "activities" to suppress UI elements
- Key binding schema

Start-up Performance

- Remove unused plugins
  - Modify plugins to remove dependencies
- Defer work until after UI appears
  - Early startup extension point
  - Display.asyncExec(...)
- Optimize load order
  - Record class load order
  - Reorder classes in plugin jar files

Application Performance

- Profile and optimize the code
  - Identify hotspots with VM profiler
  - Rewrite or eliminate slow code
- Defer work to background tasks

Performance-critical Areas

- Background indexing
- Code completion - how to make it fast
- Compilation time via incremental compilation

Metrics

First RCP build

65 MB

170 plugins

20s startup

Current build

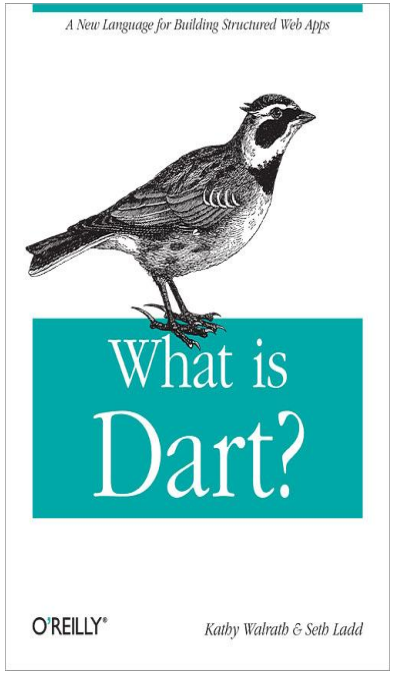
39 MB

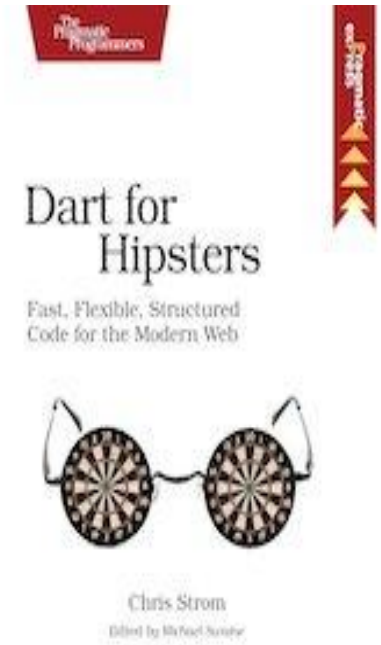
75 plugins

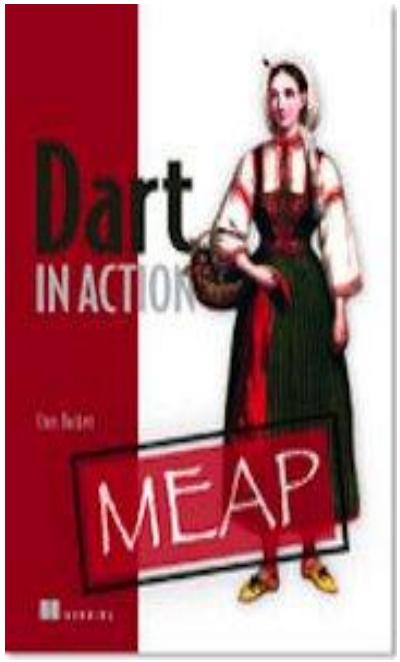
5s startup

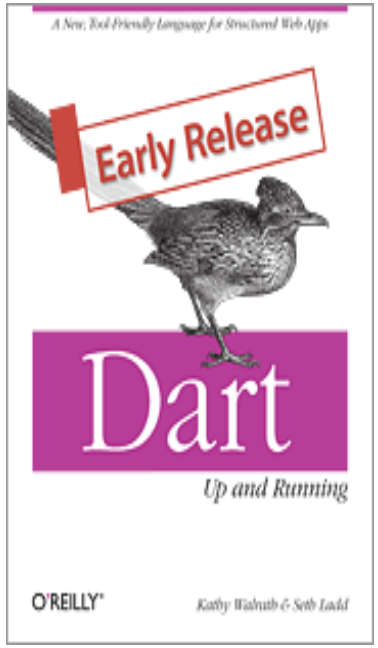
http://www.dartlang.org

Books









http://www.dartlang.org

Questions?

More information....

- https://**dartlang.org**
  - Introduction, language spec, articles
  - Download Dart Editor
- https://**code.google.com/p/dart/**
  - Source code to editor, compiler, and virtual machine
  - See the wiki for instructions

Community

- G+: **google.com/+dartlang**
- Mailing list: **misc@dartlang.org**
- Stack Overflow: **Tag dart**
- Twitter: **@dart_lang**
- Hashtag: **#dartlang**
- Blogs: **http://dartosphere.org**
- IRC: **#dart**