Dart

For ChromeConf, 2012/03

Seth Ladd, Developer Advocate, Dart

#dartlang

Cuddle time

Dart

```
class Hug {
  final num strength;
  Hug(this.strength);
  Hug.bear() : strength = 100;

  Hug operator +(Hug other) {
    return new Hug(strength + other.strength);
  }

  void patBack({int hands: 1}) {
    // ...
  }

  String toString() => "Embraceometer reads $strength";
}
```

Cuddle time

Dart

```
class Hug {
  final num strength;
  Hug(this.strength);
  Hug.bear() : strength = 100;

  Hug operator +(Hug other) {
    return new Hug(strength + other.strength);
  }

  void patBack({int hands: 1}) {
    // ...
  }

  String toString() => "Embraceometer reads $strength";
}
```

Cuddle time

Dart

```
class Hug {
  final num strength;
  Hug(this.strength);
  Hug.bear() : strength = 100;

  Hug operator +(Hug other) {
    return new Hug(strength + other.strength);
  }

  void patBack({int hands: 1}) {
    // ...
  }

  String toString() => "Embraceometer reads $strength";
}
```

Cuddle time

Dart

```
class Hug {
  final num strength;
  Hug(this.strength);
  Hug.bear() : strength = 100;

  Hug operator +(Hug other) {
    return new Hug(strength + other.strength);
  }

  void patBack({int hands: 1}) {
    // ...
  }

  String toString() => "Embraceometer reads $strength";
}
```

Cuddle time

Dart

```
class Hug {
  final num strength;
  Hug(this.strength);
  Hug.bear() : strength = 100;

  Hug operator +(Hug other) {
    return new Hug(strength + other.strength);
  }

  void patBack({int hands: 1}) {
    // ...
  }

  String toString() => "Embraceometer reads $strength";
}
```

Cuddle time

Dart

```
class Hug {
  final num strength;
  Hug(this.strength);
  Hug.bear() : strength = 100;

  Hug operator +(Hug other) {
    return new Hug(strength + other.strength);
  }

  void patBack({int hands: 1}) {
    // ...
  }

  String toString() => "Embraceometer reads $strength";
}
```

Cuddle time

Dart

```
class Hug {
  final num strength;
  Hug(this.strength);
  Hug.bear() : strength = 100;

  Hug operator +(Hug other) {
    return new Hug(strength + other.strength);
  }

  void patBack({int hands: 1}) {
    // ...
  }

  String toString() => "Embraceometer reads $strength";
}
```

Cuddle time

Dart

```
main() {
  var small = new Hug(2);
  var big = new Hug.bear();

  var together = small + big;

  print(together); // Embraceometer reads 102
}
```

Mixins

Dart

```
abstract class Persistence {
  void save(String filename) {
    print('saving the object as ${toJson()}
');
  }

  void load(String filename) {
    print('loading from $filename');
  }

  Object toJson();
}
```

Dart

```
class Ninja extends Object with Persistence
{
  Map toJson() {
    return {'throwing_stars': true};
  }
}

main() {
  var snakeEyes = new Ninja();
  snakeEyes.save('file.txt');
}
```

#dartlang

Lexical this. WYSIWYG.

Dart

```
class AwesomeButton {
  int awesomeDial;
  ButtonElement elem;

  AwesomeButton(this.elem) {
    elem.onClick.listen((e) => crankTheAwesome());
  }

  crankTheAwesome() {
    awesomeDial = 11;
  }
}
```

Lexical this. WYSIWYG.

Dart

```
class AwesomeButton {
  int awesomeDial;
  ButtonElement elem;

  AwesomeButton(this.elem) {
    elem.onClick.listen((e) => crankTheAwesome());
  }

  crankTheAwesome() {
    awesomeDial = 11;
  }
}
```

Fields evolve with getters/setters

Dart

```
class Car {
 bool isEngineRunning;
}

var roadster = new Car();
roadster.isEngineRunning = true;
```

#dartlang

Fields evolve with getters/setters

Dart

```
class Car {
 Engine engine;

 bool get isEngineRunning => engine.isRunning;

 void set isEngineRunning(bool isRunning) {
   engine.isRunning = isRunning;
 }
}

var roadster = new Car();
roadster.isEngineRunning = true;  // same as before!
```

Futures

Dart

```
Future lengthyComp();

// Consume a Future
Future doLengthyComputation() {
  return lengthyComp().then((value) => print(value))
                      .catchError((e) => print(e));
}

// Produce a Future
Future queryDb(String sql) {
  var completer = new Completer();
  dbConnection.query(sql,
    (results) => completer.complete(results),
    (error) => completer.completeError(error));
  return completer.future;
}
```

Futures

Dart

```
Future costlyQuery();
Future expensiveSearch();
Future lengthyCalc();

// Chaining multiple Futures
costlyQuery()
    .then((results) => expensiveSearch(results))
    .then((results) => lengthyCalc(results))
    .then((results) => print(results))
    .catchError(print);
```

Dart

```
// Or...
costlyQuery()
    .then(expensiveSearch)
    .then(lengthyCalc)
    .then(print)
    .catchError(print);
```

Streams

Dart

```
Stream<Hug> hugs();

hugs().listen((hug) => embrace(hug));

// Or
hugs()
    .where((hug) => !hug.isBearHug)
    .map((hug) => new Hug.bear(hug))
    .listen(embrace,
      onError: (e) => handleError(e),
      onDone: () => smile());
```

Streams

Dart

```
element.onClick.listen(handleClick);

HttpServer.bind('127.0.0.1', port)
    .then((HttpServer server) {
      print('listening for connections on $port');

      server.listen((HttpRequest request) {
        fileHandler.onRequest(request);
      });
    },
    onError: (error) => print("Error starting HTTP server: $error"));
```

#dartlang

Cleaned up DOM

Dart

```
import 'dart:html';

void main() {
  var button = new ButtonElement();
  button.id = 'activate';
  button.text = 'Click me';
  button.classes.add('important');
  button.onClick.listen((e) => window.alert('Clicked!!'));

  document.body.children.add(button);
}
```

Method Cascades

Dart

```
import 'dart:html';

void main() {
  var button = new ButtonElement()
    ..id = 'activate'
    ..text = 'Click me'
    ..classes.add('important')
    ..onClick.listen((e) => window.alert('Clicked!!'));

  document.body.children.add(button);
}
```

#dartlang

Traditional async APIs

JavaScript

```
 // Traditionally:
navigator.getMedia = ( navigator.getUserMedia ||
                       navigator.webkitGetUserMedia ||
                       navigator.mozGetUserMedia ||
                       navigator.msGetUserMedia);

navigator.getMedia (
   // constraints
   {video: true, audio: true},
   // successCallback
   function(localMediaStream) {
      var video = document.querySelector('video');
      video.src = window.URL.createObjectURL(localMediaStream);
      video.onloadedmetadata = function(e) {
         // Do something with the video here.
      };
   },
   // errorCallback
   function(err) {
    console.log("The following error occured: " + err);
   }
);
```

Future-based APIs

Dart

```
// With Dart:
window.navigator.getUserMedia(audio:true, video: true)
  .then((stream) {
    var video = new VideoElement()
      ..autoplay = true
      ..src = Url.createObjectUrl(stream)
      ..onLoadedMetadata.listen((e) => /* ... */);
    document.body.append(video);
  })
  .catchError(reportIssue);
```

#dartlang

Traditional XHR

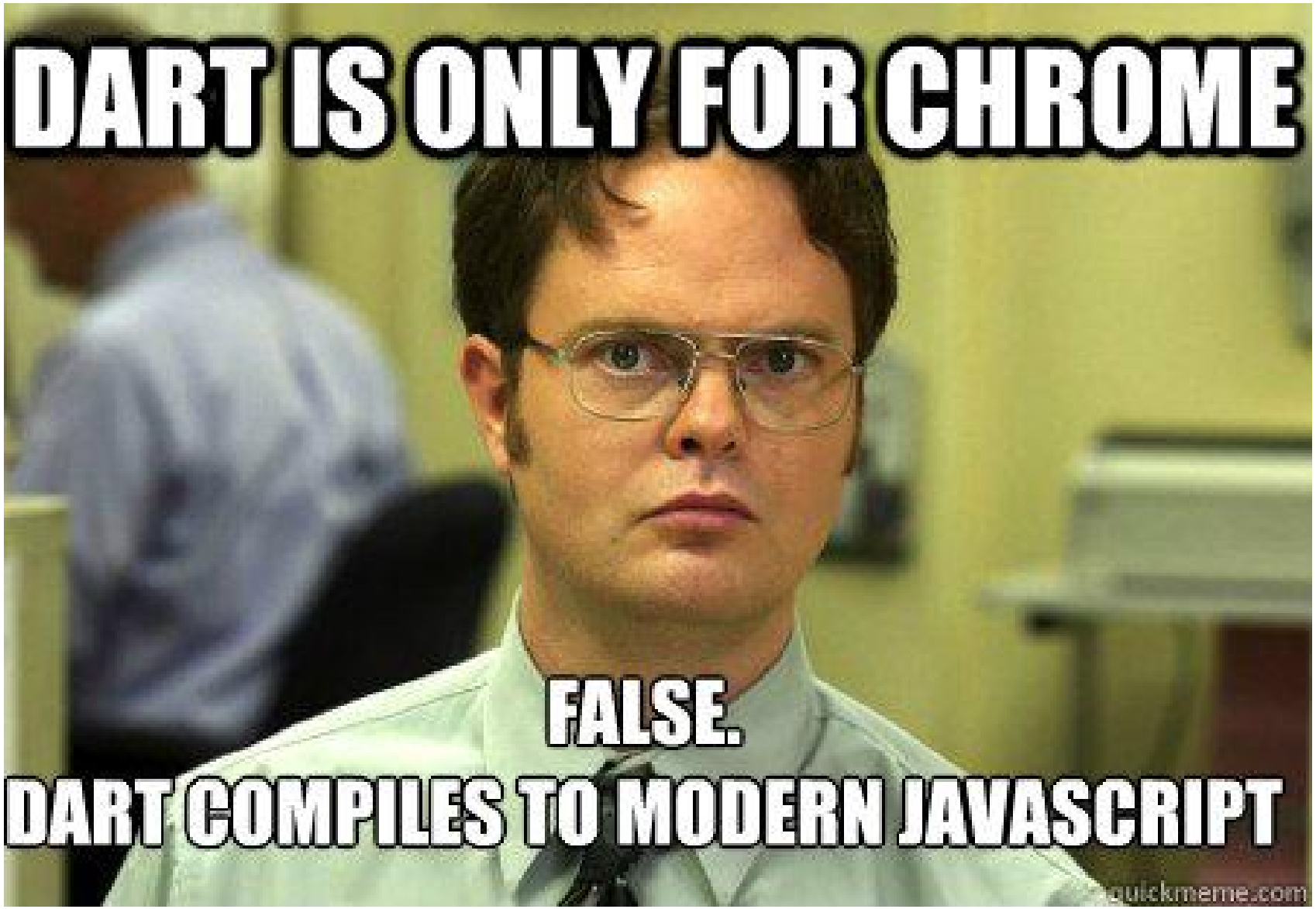
JavaScript

```
var req = new XMLHttpRequest();
req.onreadystatechange=function() {
  if (req.readyState == 4) {
    if (req.status == 200) {
      var elem = document.getElementById("myDiv");
      elem.innerHTML = req.responseText;
    } else {
      console.log(req.status);
    }
}
req.open("GET", "data.txt", true);
req.send();
```

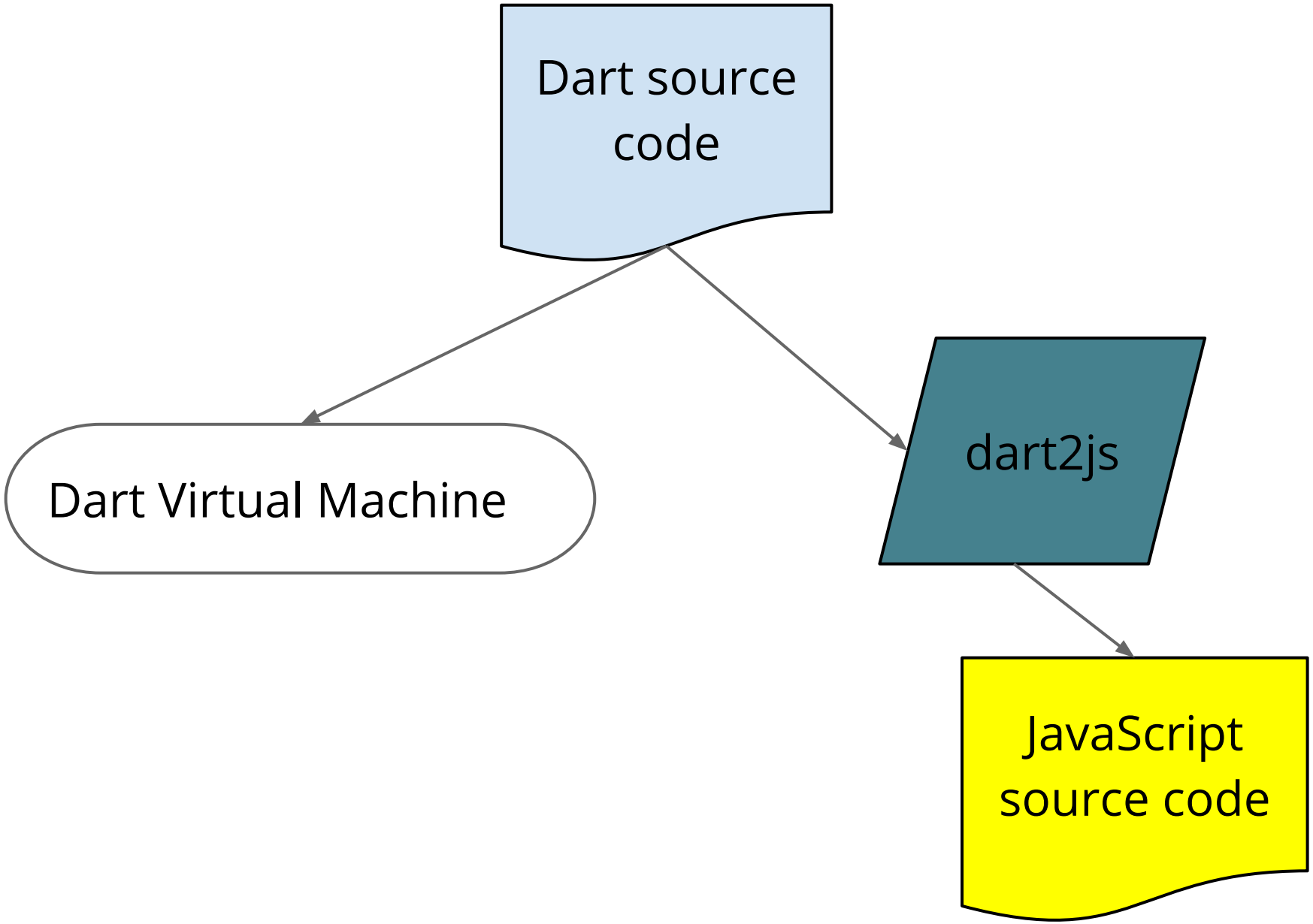
Future-based (X)HR

JavaScript

```
HttpRequest.getString("data.txt")
  .then((String result) {
    var elem = query("#myDiv");
    elem.text = result;
  })
  .catchError(print);
```


DART IS ONLY FOR CHROME
FALSE.
DART COMPILES TO MODERN JAVASCRIPT
quickmeme.com

#dartlang


Dart source code
Dart Virtual Machine
dart2js
JavaScript source code

(demo)

Generated JS size

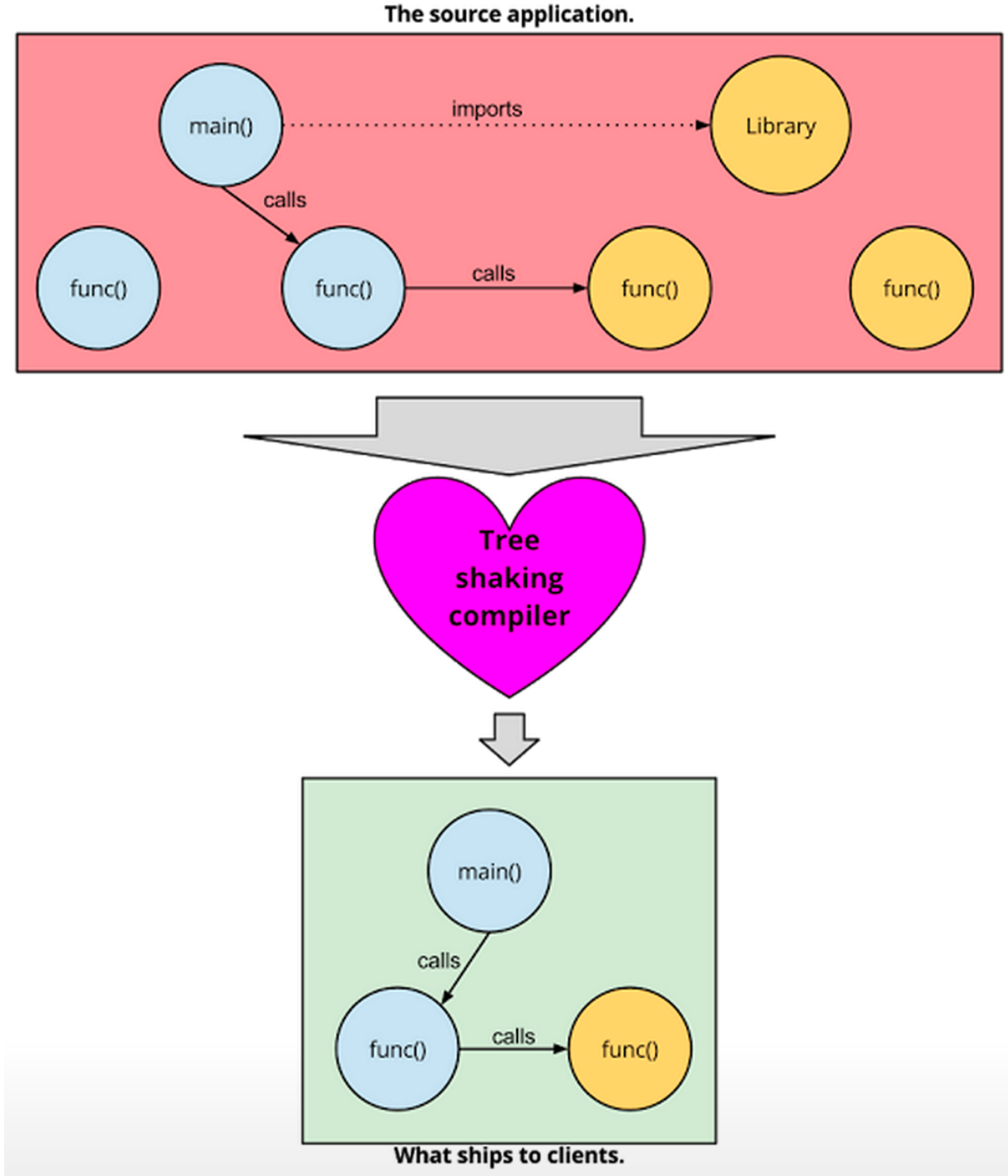
- JS: 31115
- JS + gzip: 7175
- JS + minification: 15644
- JS + minification + gzip: 5184

Dart

```
import 'dart:html';

main() {
  var elem = query('#foo');
  elem.text = 'hello, world';
}
```

#dartlang


The source application.
main()
imports
Library
calls
func()
func()
calls
func()
func()
Tree shaking compiler
main()
calls
func()
calls
func()
What ships to clients.

#dartlang

Dart-JavaScript Interop

JavaScript

```
js.scoped(() {
  var maps = js.context.google.maps;

  // Allocate a new JS Map
  var myOptions = js.map({
    'zoom': 9,
    'mapTypeId': maps.MapTypeId.ROADMAP,
    'center': new js.Proxy(maps.LatLng, 47.6097, -122.3331)
  });

  var map = new js.Proxy(maps.Map, query('#map_canvas'), myOptions);
```

Web UI is a polyfill for Web Components, *and*

- Dynamic data-driven views, live two-way data binding
- Observable objects and expressions
- Fast developer cycles
- *Compiles to vanilla JavaScript and HTML*

Data Binding

HTML

```
<p>MDV is {{ superlative }}</p>

<button id="change-it" on-click="changeIt()">Change</button>
```

DART

```
import 'package:web_ui/web_ui.dart';

@observable
String superlative = 'awesome';
int i = 0;

List<String> alternatives = const <String>['wicked cool', 'sweet', 'fantastic'];

changeIt() => superlative = alternatives[i++ % alternatives.length];

main() { }
```

#dartlang

Conditionals

HTML

```
<select name="language" bind-value="langChoice">
  <option value="">-- Choose one --</option>
  <option value="js">JavaScript</option>
  <option value="java">Java</option>
</select>

<div template if="langChoice != null && !langChoice.isEmpty">
  <h3>{{ langChoice }}</h3>

  <code><pre>{{ languageExamples[langChoice] }}</pre></code>
</div>
```

DART

```
Map languageExamples = {
  'js': 'function Person(firstName, lastName) { ... };',
  'java': 'public class Person { ... }'
};

String langChoice = '';

main() { }
```

#dartlang

Iteration

HTML

```
<ul template iterate="feature in features">
  <li>
    <label>
      <input type="checkbox" value="{{ feature }}"
             on-click="addToFavorites($event)">
      {{ feature }}
    </label>
  </li>
</ul>
```

DART

```
import 'dart:html';

List<String> features = const <String>['lexical scope', 'closures', 'getters and setters', ...];
List<String> userFavorites = new List<String>();

addToFavorites(Event e) {
  InputElement checkbox = e.target;
  var fav = checkbox.value;

  // add or remove from userFavorites
}

main() { }
```

Custom Elements - The page

HTML

```
<input type="text" id="new-todo">

<button on-click="createNewTodo()">
  Create New Todo
</button>

<ul template id="todos" iterate="todo in todoItems">
  <li>
    <x-todo-item bind-todo="todo"></x-todo-item>
  </li>
</ul>
```

#dartlang

Custom Elements - <element>

HTML

```
<!DOCTYPE html>
<html>
  <body>
    <element name="x-todo-item" constructor="TodoItemComponent"
             apply-author-styles="" extends="div">
      <template>
        <label class="{{ completeClass }}">
          <input type="checkbox" on-change="toggle()">
          <span>{{ todo.actionItem }}</span>
        </label>
      </template>
      <script type="application/dart" src="TodoItemComponent.dart"></script>
    </element>
  </body>
</html>
```

#dartlang

Custom Elements - The code

DART

```dart
import 'package:web_ui/web_ui.dart';
import 'models.dart';

class TodoItemComponent extends WebComponent {
  TodoItem todo;

  toggle() => todo.toggle();

  String get completeClass => todo.complete ? 'completed' : '';
}
```

DART

```dart
library models;

class TodoItem {
  String actionItem;
  bool complete = false;

  TodoItem(this.actionItem);

  toggle() => complete = !complete;
}
```

#dartlang

Dart's bundled libraries

- Core lib
- HTML
- Crypto
- JSON
- Mirrors
- UTF
- Unit test and mocks
- Math
- Logging
- URI
- I18N
- *More*

Dart's community libraries

- MongoDB
- Redis
- BOT
- vector_math
- game_loop
- gl_enum
- dbcrypt
- UUID
- Stream web server
- Widgets
- OAuth2 client
- *Many more*

Active community involvement at pub.dartlang.org

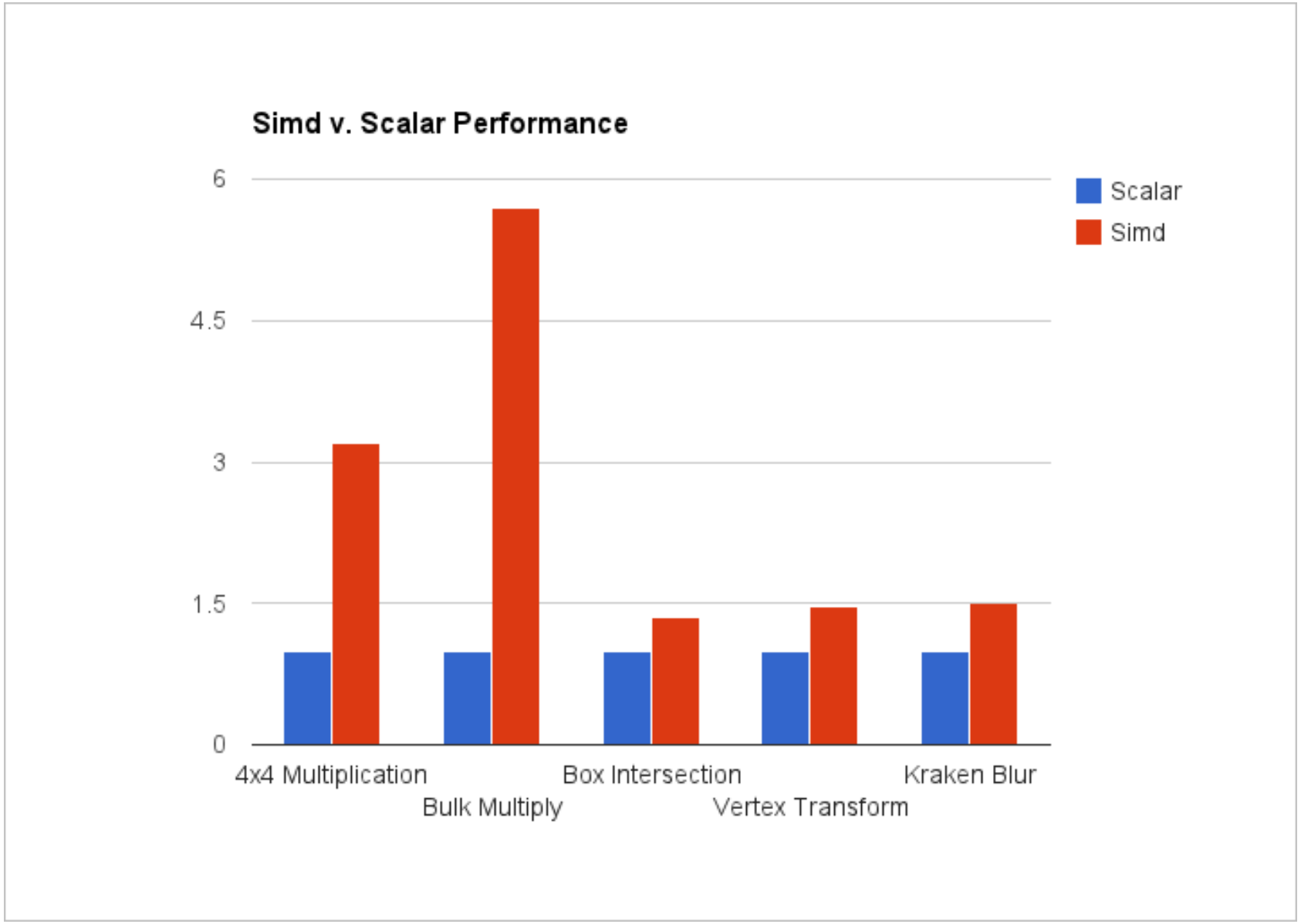
SIMD in Dart

- New types
  - Float32x4
  - Float32x4List
  - Uint32x4

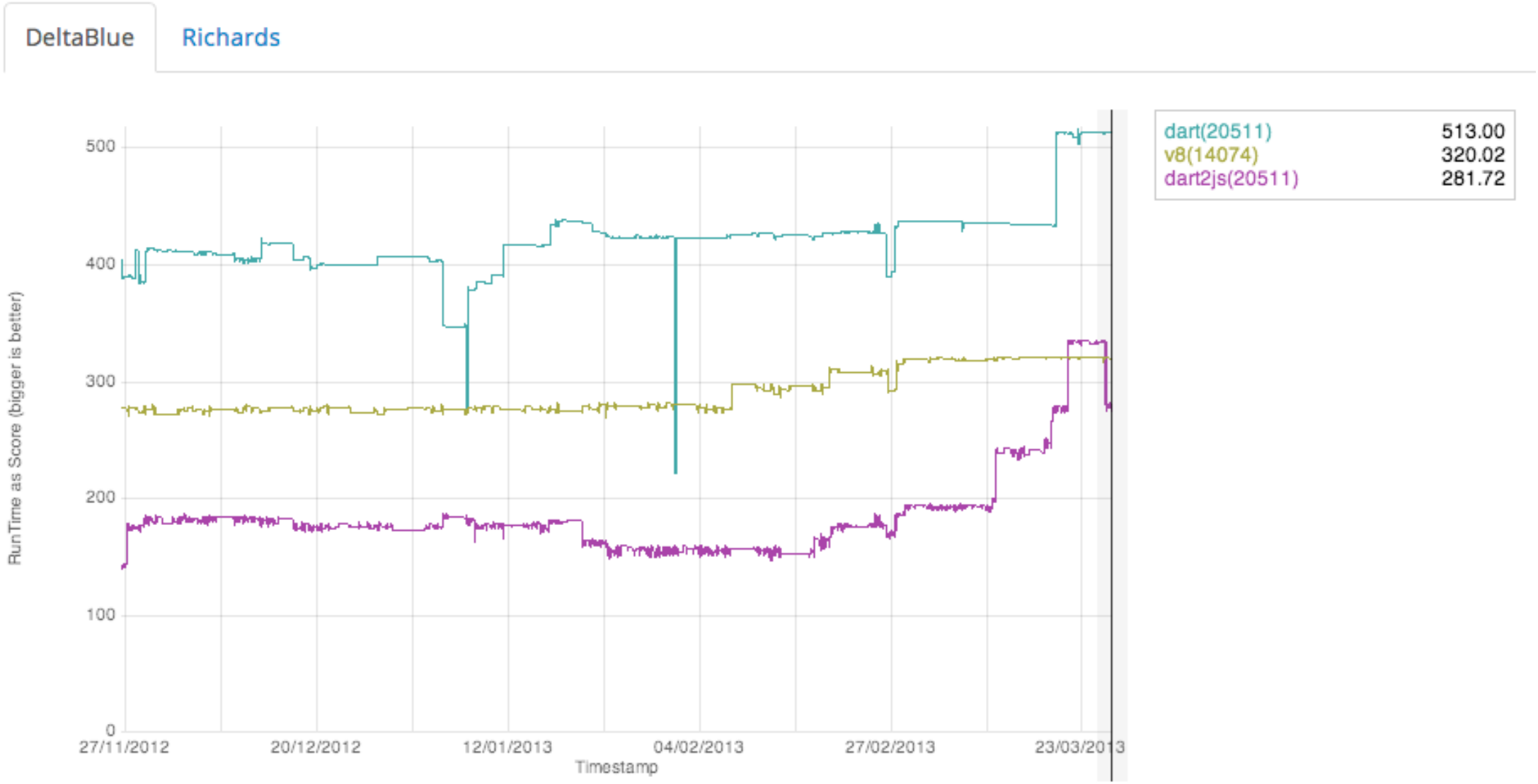
4 IEEE-754 32-bit Floating Point Numbers
List of Float32x4
4 Unsigned 32-bit Integer Numbers

- Composable operations
  - Arithmetic
  - Logical
  - Comparisons
  - Reordering (shuffling)

SIMD Benchmarks



Benchmark tracking



go/lem

dartlang.org/performance

Partners

Google fiber

Green Tea

Adobe

Google3 Integration

- Blaze
- TAP
- Code Search, Grok
- Dart Editor, Eclipse Plugin
- Dartium
- "Instant" Edit/Refresh Cycle
- Protocol Buffers
- Open-Source Apiary Client

Coming Soon...

- Testacular / Karma
- Official Apiary Client

Further Out...

- VM in Chrome
- Borg
- AppEngine

Topics not covered

- Dart on the server
- Pub
- Testing
- Isolates
- Other editors
- Core libraries
- Generics
- Mirrors (aka reflection)
- Many other language features
- *Lots more…*

Plans for the future

- Launch 1.0 this year
- VM integration with Chrome
- App Engine support
- Lang features like enums, await (??)
- UI Toolkit support


DART
LIFTOFF
HACKATHON
APRIL 3RD - 4TH
04/03/13-
04/04/13
MOUNTAIN VIEW, CA
<FIRST CLASS> AIRCRAFT
FIRST CLASS
2013
MEMBER
Book Now!
GO/LIFTOFF

Thank You!

Find us at ***dartlang.org***

#dartlang

Resources

Internal:

- go/dart
- dart-google3@
- dart-discuss@
- unittest codelab

External:

- Web UI Intro
- Web UI Spec
- Style guide
- Idiomatic Dart
- Improving the DOM
- Dart Language Tour
- All articles
- dartlang.org

Dart is more ambitious

- Tree shaking
- Getters and setters (though I presume TypeScript will get those eventually)
- Operator overloading
- Real block scope, no hoisting, no IIFEs
- A native VM
- Sane equality semantics
- No weird implicit conversion craziness
- Lexically bound this everywhere
- Mixins
- Annotations
- An import system
- User-defined subscript operators
- Generics, with reification
- Mirrors
- Better collection classes
- A cleaner DOM API

Code completion

Refactoring

Static analysis

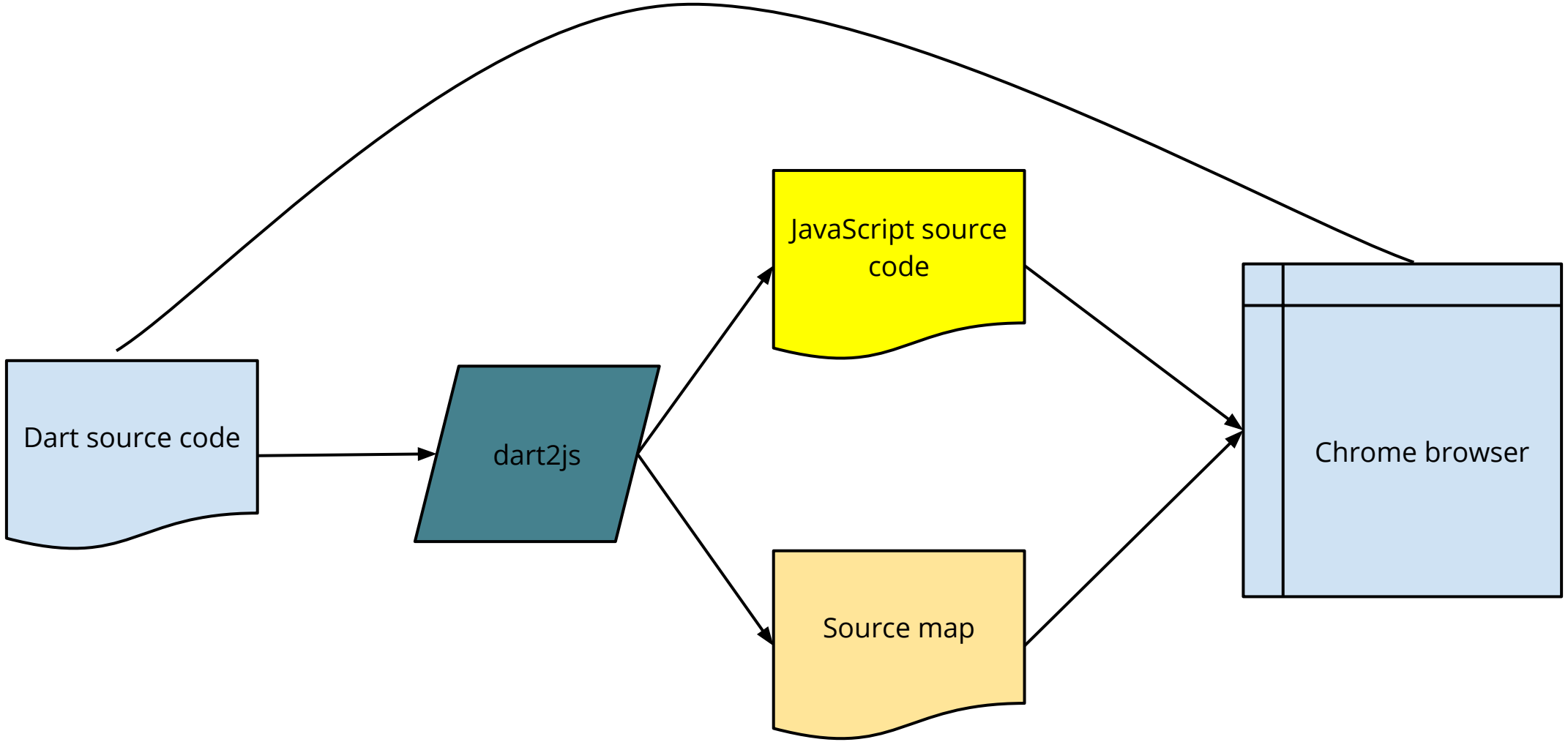
Debugging

(demo)

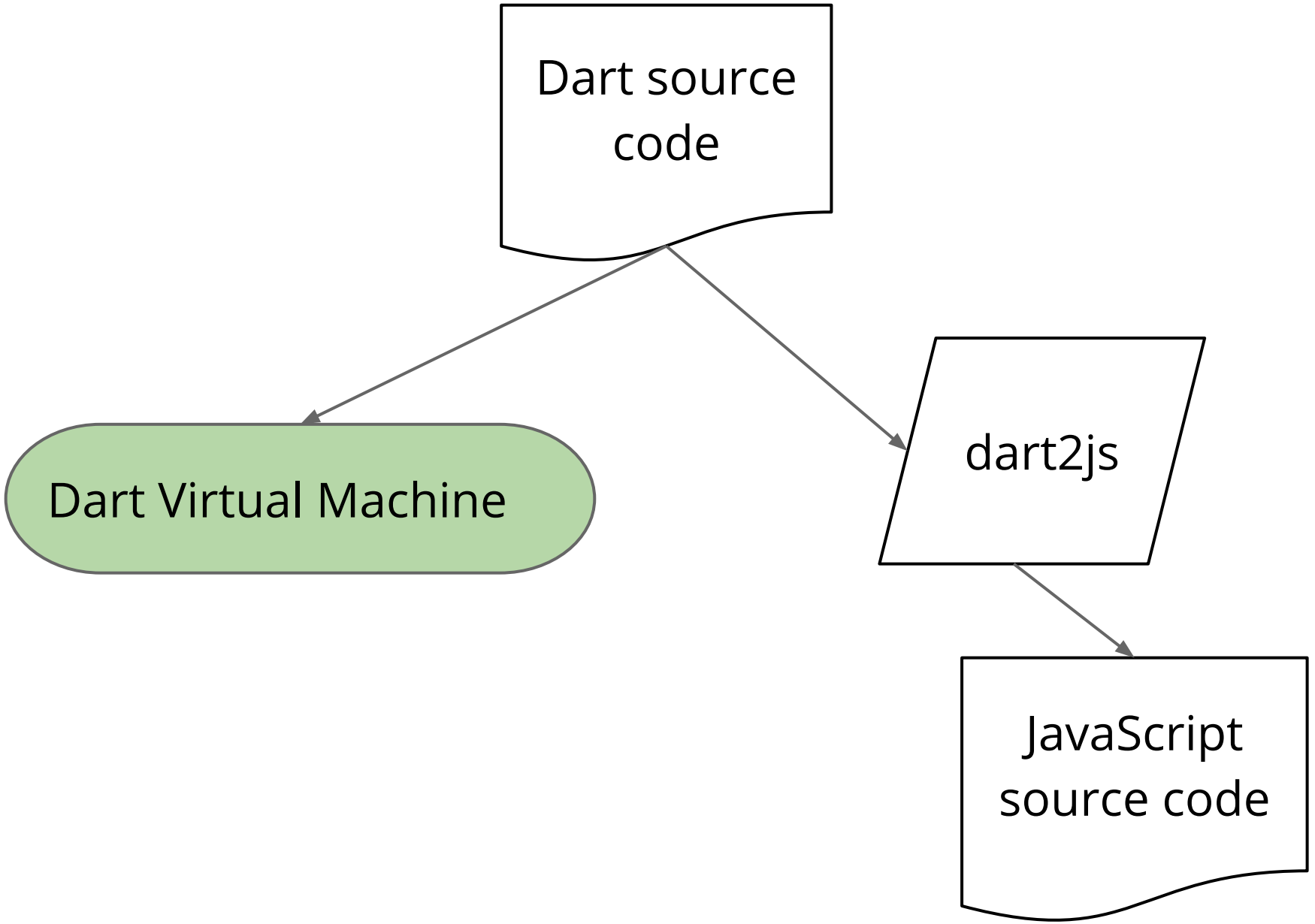
Dart also supported in:

IntelliJIDEA

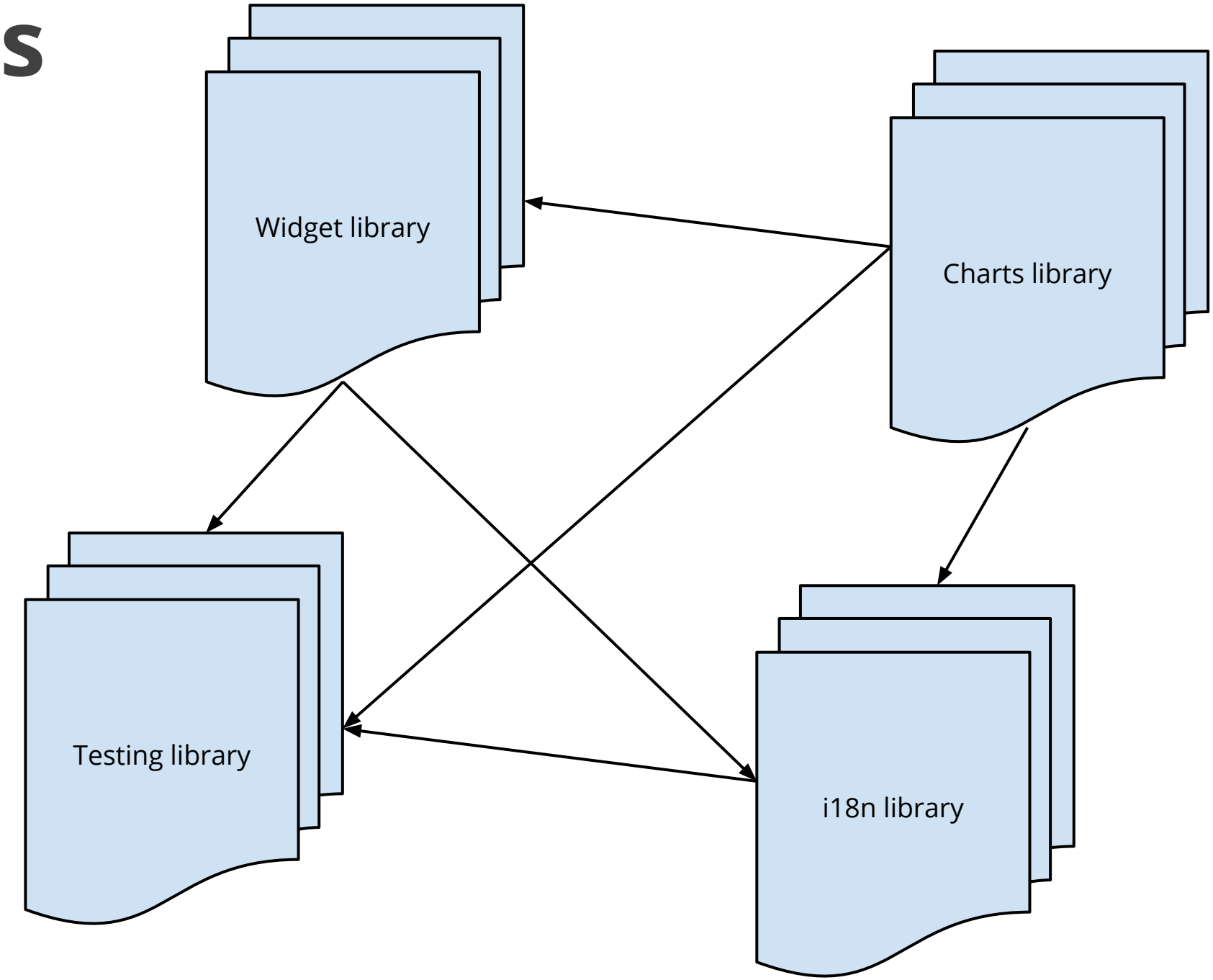
WebStorm


Dart source code
dart2js
JavaScript source code
Source map
Chrome browser

(demo clock)

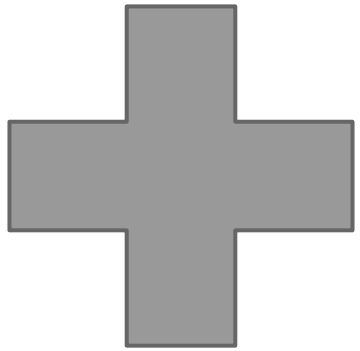

Dart source code
Dart Virtual Machine
dart2js
JavaScript source code

Libraries


Widget library
Charts library
Testing library
i18n library

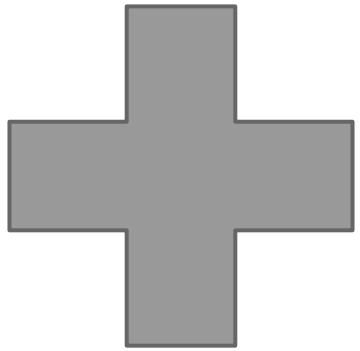
Use and share code with Pub

Together at last!



(demo)

Together at last!



Declarative Renaissance

(demo)

You decide:

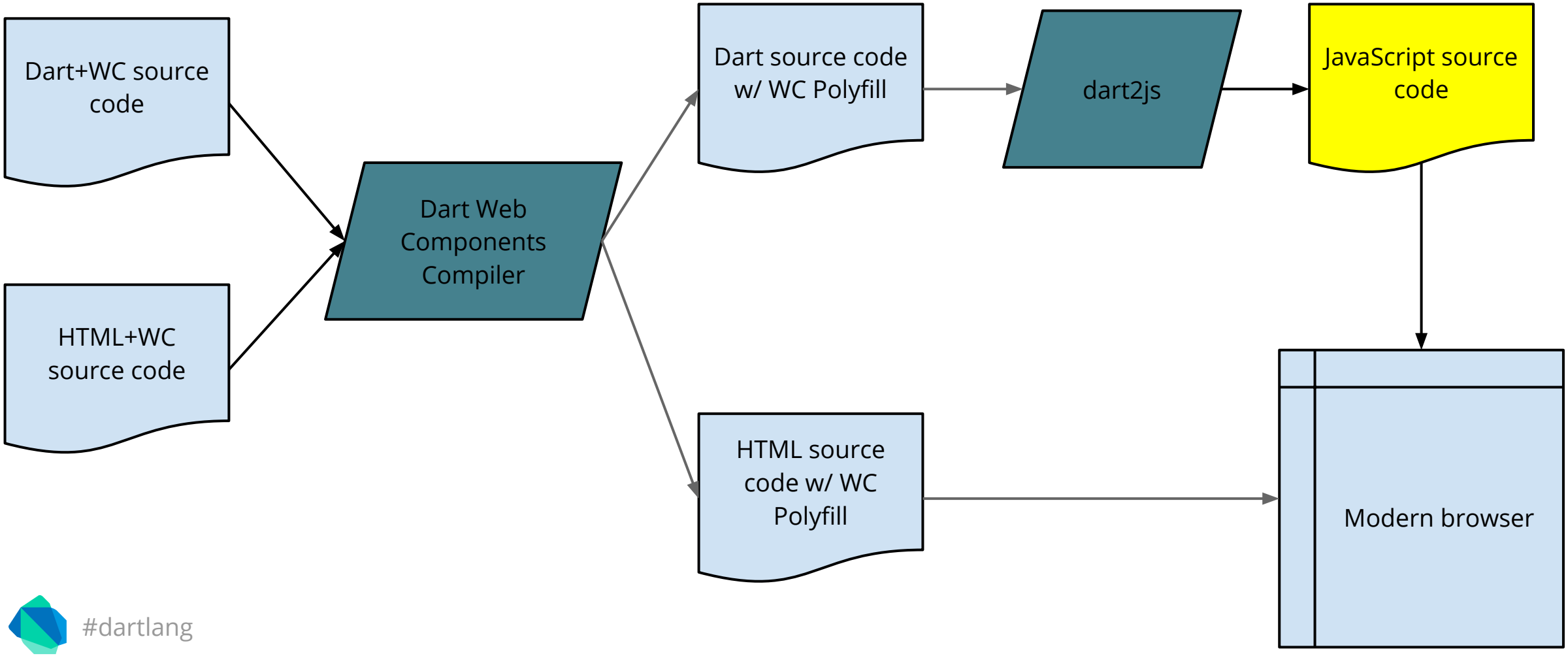

```
<messages>
  <message>
    <subject>
      Please fill out the TPS report
    </subject>
    <sent>2012-10-03</sent>
    <summary>
      I'm going to have to ask you to come in...
    </summary>
  </message>
  <message>
    <subject>
      Reminder: fill out that TPS report!
    </subject>
    <sent>2012-10-04</sent>
    <summary>
      It's been 24 hours...
    </summary>
  </message>
  ...
</messages>
```

Demos

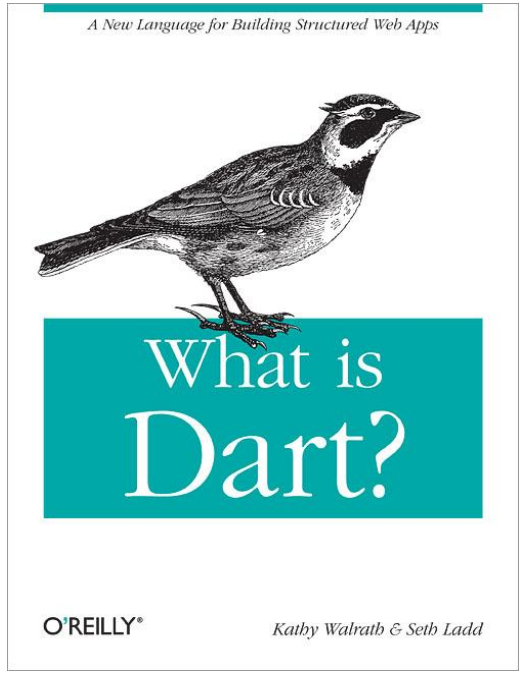
- Data binding
- Event handing
- Loops and conditionals
- Nested object binding
- Custom tags

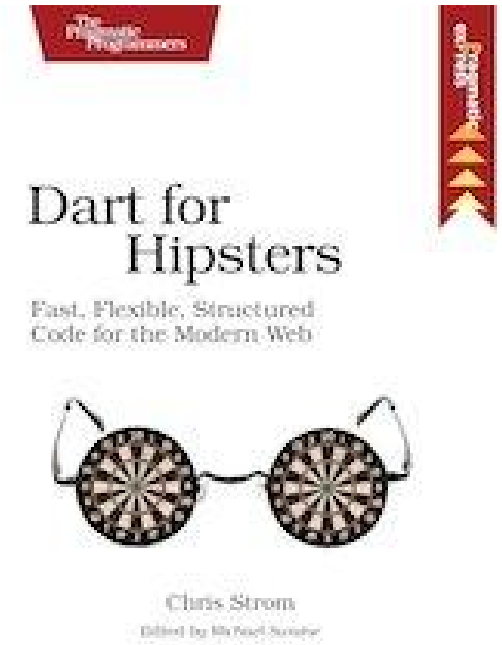
(demo)

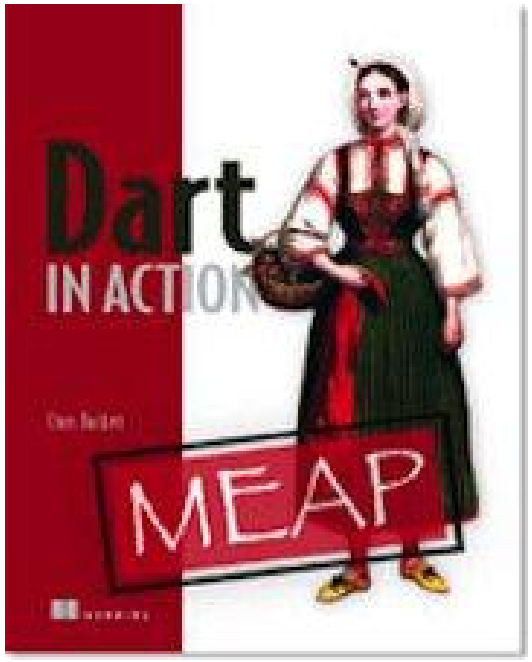
Don't sweat it, this is automatic.

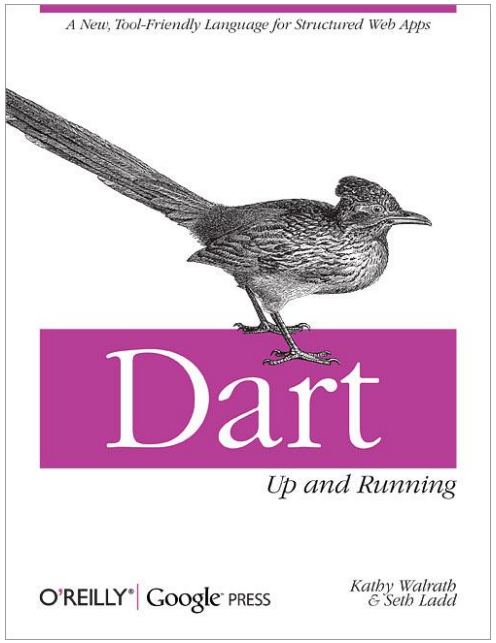


Read moar Dart!!1









#dartlang