



# Productivity, Performance. Pick Two.

Building the modern web with Dart.

Seth Ladd, Developer Advocate, +Seth Ladd

---

#dartlang

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



#dartlang

# Dart is Open Source

- BSD-style license
- [dart.googlecode.com](http://dart.googlecode.com)
- [github.com/dart-lang](http://github.com/dart-lang)
- [Contributing guide](#)



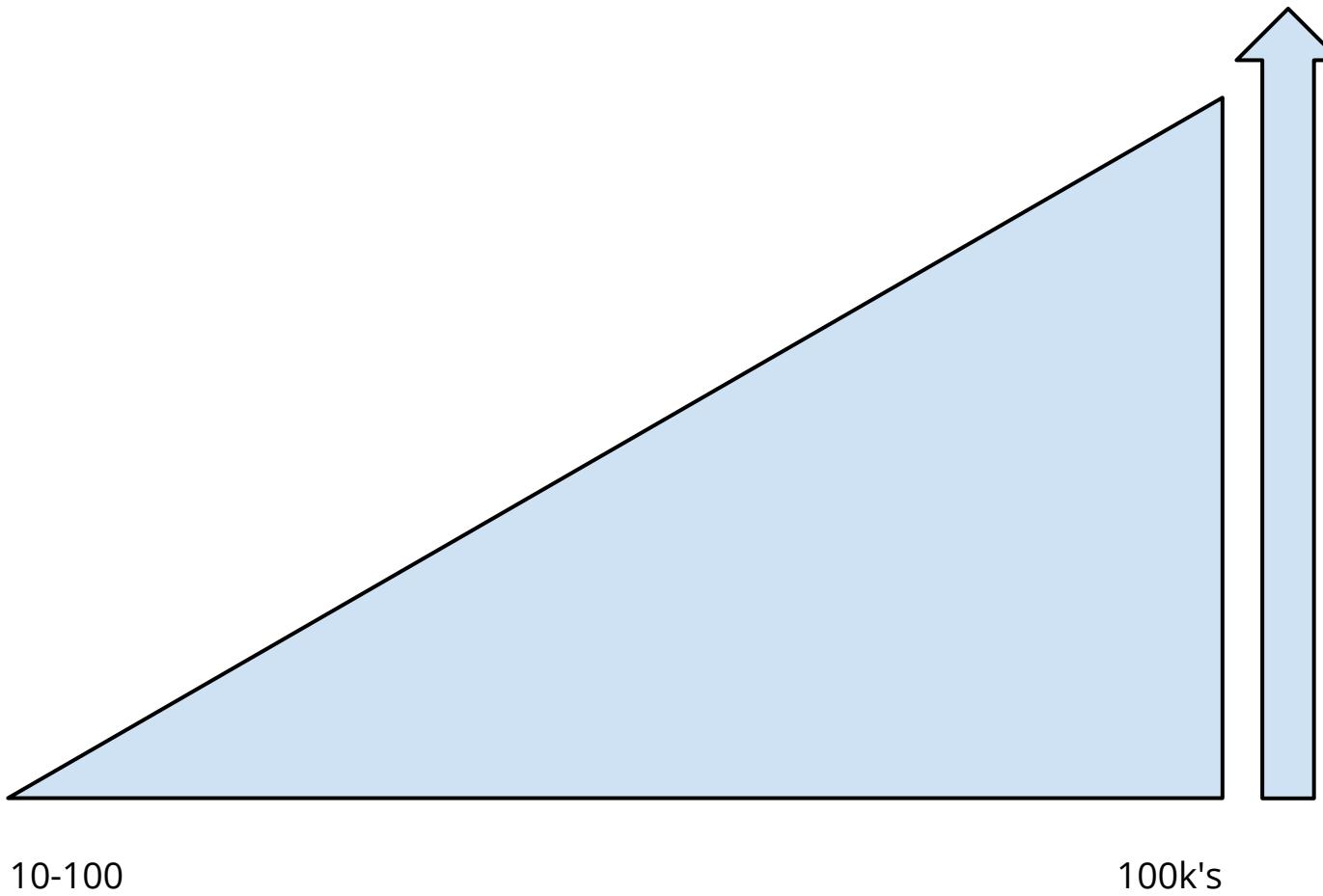
open source  
initiative

# Dart is for modern web apps

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



#dartlang



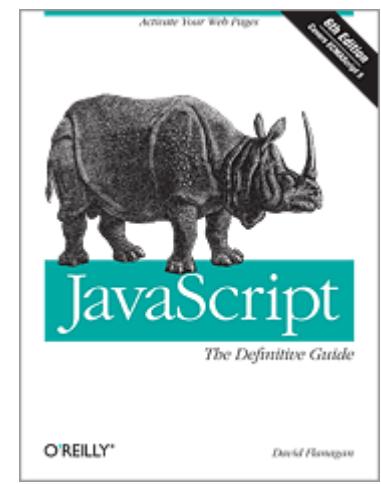
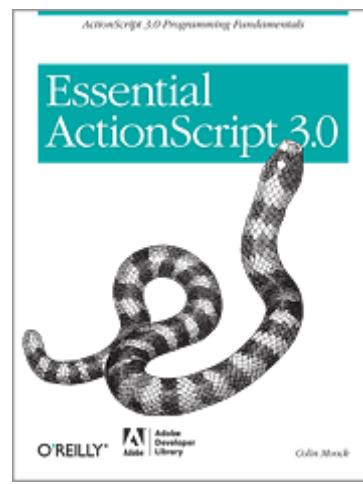
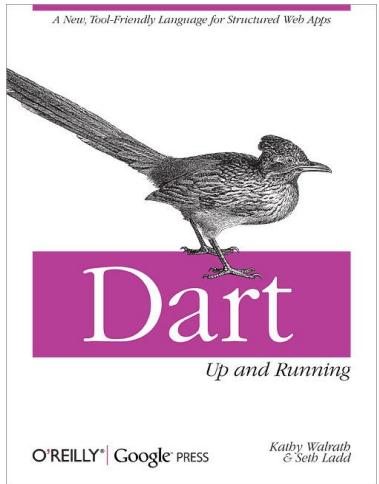
# Scale Lines of Code



#dartlang

(demo)

# Familiar



#dartlang

# Java you know and love

```
public class Money {  
    private int amount;  
  
    public Money(double amount) { this.amount = Math.floor(amount * 100); }  
    public Money(int amount) { this.amount = amount; }  
    public Money(String amount) { this(Double.parseDouble(amount)); }  
  
    public Money add(Money other) { return new Money(this.amount + other.amount); }  
    public Money add(Money other, boolean chargeTax) {  
        return new Money(add(other).amount * 0.93);  
    }  
  
    public int getAmount() { return amount; }  
}
```

Java



#dartlang

# Dart you'll grow to love

```
class Money {  
    final int amount;  
  
    static final Money ZERO = new Money(0);  
  
    Money(this.amount);  
  
    Money.fromDouble(double amount) : this((amount*100).toInt());  
  
    Money add(Money other, {bool chargeTax: false}) {  
        var newAmount = (this.amount + other.amount) * (chargeTax ? 0.93 : 1.0);  
        return new Money.fromDouble(newAmount);  
    }  
  
    Money operator +(Money other) => add(other);  
    String toString() => amount.toString();  
}
```

Dart

# Dart you'll grow to love

```
class Money {  
    final int amount;  
  
    static final Money ZERO = new Money(0);  
  
    Money(this.amount);  
    Money.fromDouble(double amount) : this((amount*100).toInt());  
  
    Money add(Money other, {bool chargeTax: false}) {  
        var newAmount = (this.amount + other.amount) * (chargeTax ? 0.93 : 1.0);  
        return new Money.fromDouble(newAmount);  
    }  
  
    Money operator +(Money other) => add(other);  
    String toString() => amount.toString();  
}
```

Dart

# Dart you'll grow to love

```
class Money {  
    final int amount;  
  
    static final Money ZERO = new Money(0);  
  
    Money(this.amount);  
    Money.fromDouble(double amount) : this((amount*100).toInt());  
  
    Money add(Money other, {bool chargeTax: false}) {  
        var newAmount = (this.amount + other.amount) * (chargeTax ? 0.93 : 1.0);  
        return new Money.fromDouble(newAmount);  
    }  
  
    Money operator +(Money other) => add(other);  
    String toString() => amount.toString();  
}
```

Dart

# Dart you'll grow to love

```
class Money {  
    final int amount;  
  
    static final Money ZERO = new Money(0);  
  
    Money(this.amount);  
  
    Money.fromDouble(double amount) : this((amount*100).toInt());  
  
    Money add(Money other, {bool chargeTax: false}) {  
        var newAmount = (this.amount + other.amount) * (chargeTax ? 0.93 : 1.0);  
        return new Money.fromDouble(newAmount);  
    }  
  
    Money operator +(Money other) => add(other);  
  
    String toString() => amount.toString();  
}
```

Dart

# Dart you'll grow to love

```
class Money {  
    final int amount;  
  
    static final Money ZERO = new Money(0);  
  
    Money(this.amount);  
  
    Money.fromDouble(double amount) : this((amount*100).toInt());  
  
    Money add(Money other, {bool chargeTax: false}) {  
        var newAmount = (this.amount + other.amount) * (chargeTax ? 0.93 : 1.0);  
        return new Money.fromDouble(newAmount);  
    }  
  
    Money operator +(Money other) => add(other);  
  
    String toString() => amount.toString();  
}
```

Dart

# Dart you'll grow to love

```
class Money {  
    final int amount;  
  
    static final Money ZERO = new Money(0);  
  
    Money(this.amount);  
    Money.fromDouble(double amount) : this((amount*100).toInt());  
  
    Money add(Money other, {bool chargeTax: false}) {  
        var newAmount = (this.amount + other.amount) * (chargeTax ? 0.93 : 1.0);  
        return new Money.fromDouble(newAmount);  
    }  
  
    Money operator +(Money other) => add(other);  
    String toString() => amount.toString();  
}
```

Dart

# Dart you'll grow to love

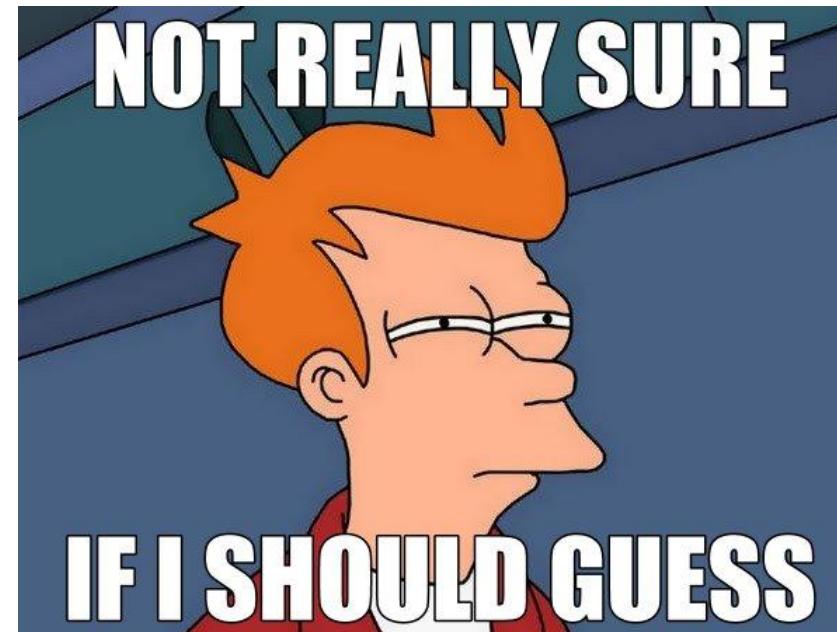
```
class Money {  
    final int amount;  
  
    static final Money ZERO = new Money(0);  
  
    Money(this.amount);  
    Money.fromDouble(double amount) : this((amount*100).toInt());  
  
    Money add(Money other, {bool chargeTax: false}) {  
        var newAmount = (this.amount + other.amount) * (chargeTax ? 0.93 : 1.0);  
        return new Money.fromDouble(newAmount);  
    }  
  
    Money operator +(Money other) => add(other);  
    String toString() => amount.toString();  
}
```

Dart

# Not entirely clear

```
recalculate(origin, offset, estimate) {  
    ...  
}
```

Without optional types



#dartlang

# Easier to understand

```
recalculate(origin, offset, estimate) {  
  ...  
}
```

Without optional types

```
num recalculate(Point origin, num offset,  
                {bool estimate: false}) {  
  ...  
}
```

With optional types

# Lexical this. WYSIWYG.

```
class AwesomeButton {  
  int awesomeDial;  
  ButtonElement elem;  
  
  AwesomeButton(this.elem) {  
    elem.on.click.add((e) => this.crankTheAwesome());  
  }  
  
crankTheAwesome() {  
  awesomeDial = 11;  
}  
}
```

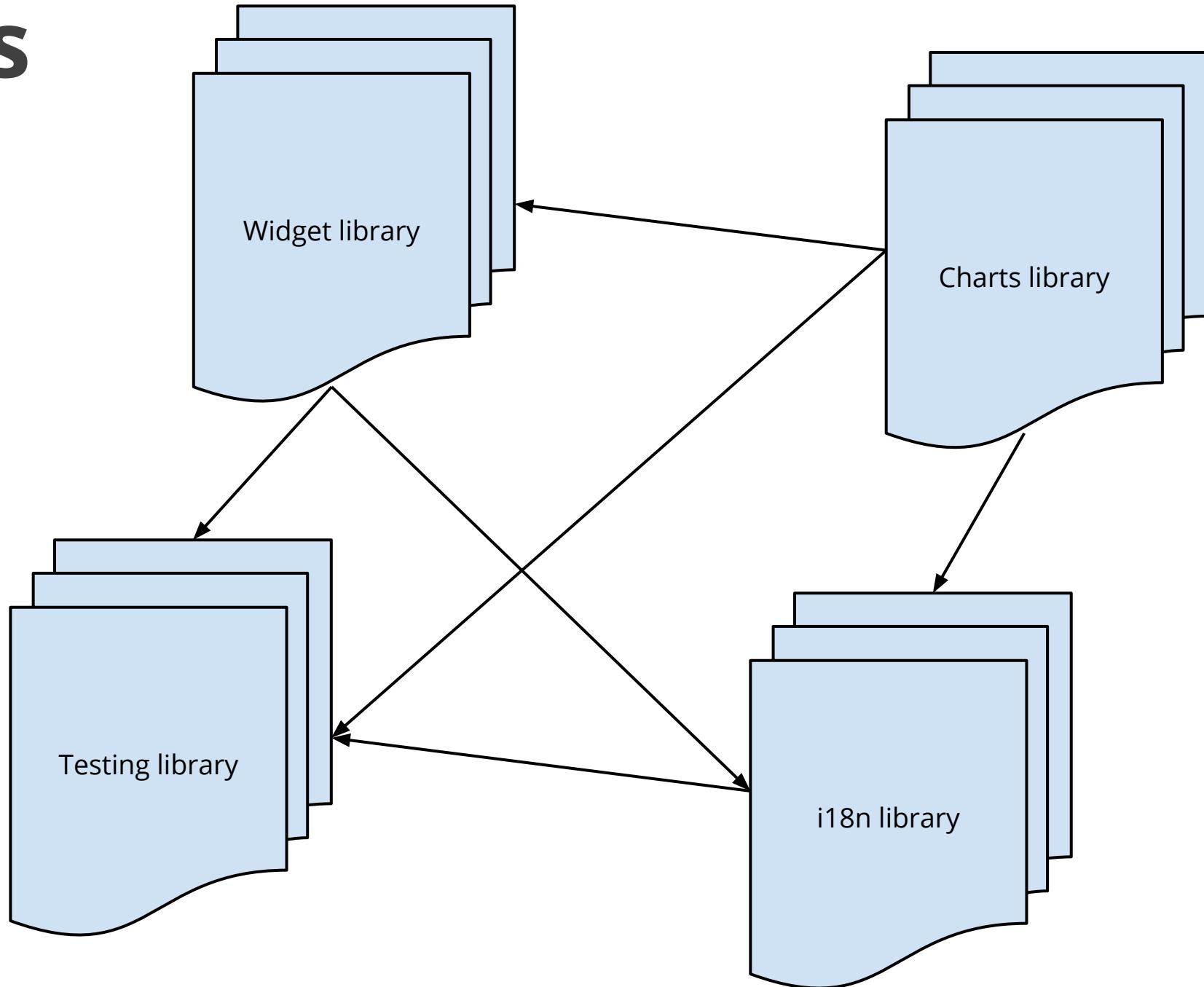
Dart

# Lexical this. WYSIWYG.

```
class AwesomeButton {  
  int awesomeDial;  
  ButtonElement elem;  
  
  AwesomeButton(this.elem) {  
    elem.on.click.add((e) => this.crankTheAwesome());  
  }  
  
crankTheAwesome() {  
  awesomeDial = 11;  
}  
}
```

Dart

# Libraries

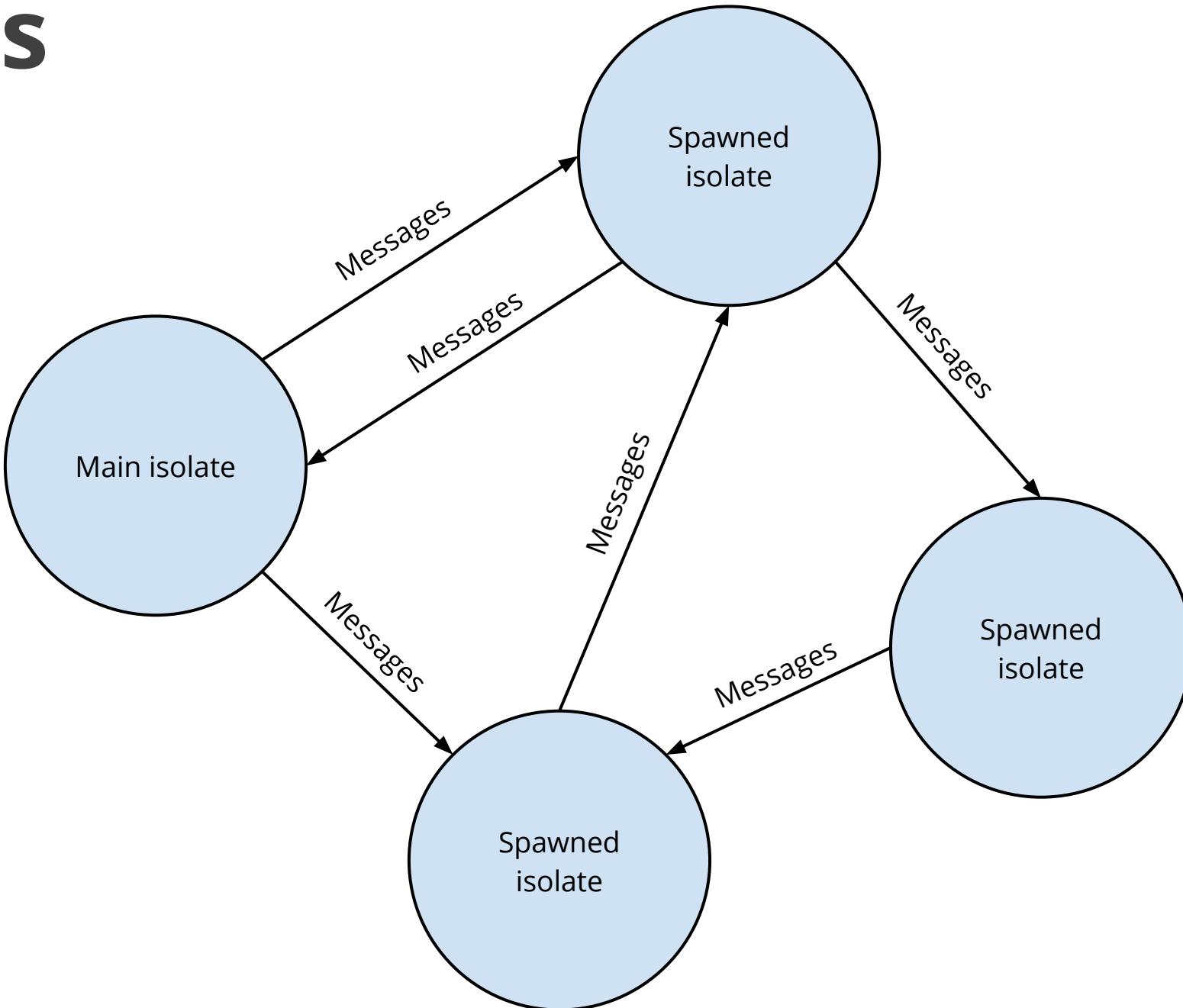


# Dart's bundled libraries

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



# Isolates



# Before isolates...

## Main Isolate (HTTP Server)

1. Receive request
2. Handle request
3. Log req to disk
4. GOTO 1

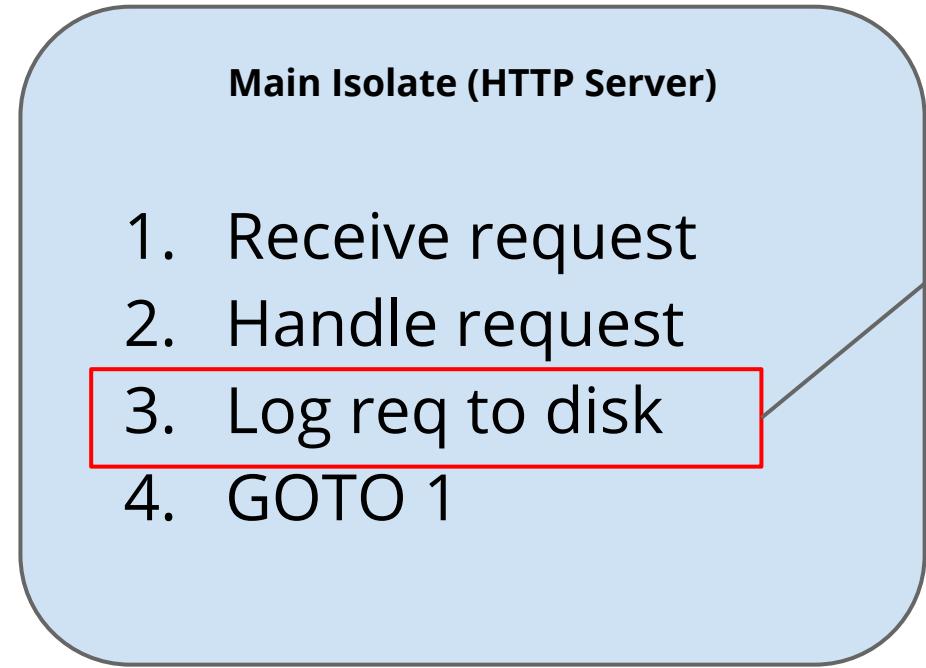
184 microsec  
total



#dartlang

disclaimer: totally unoptimized code

# Before isolates...



184 microsec  
total

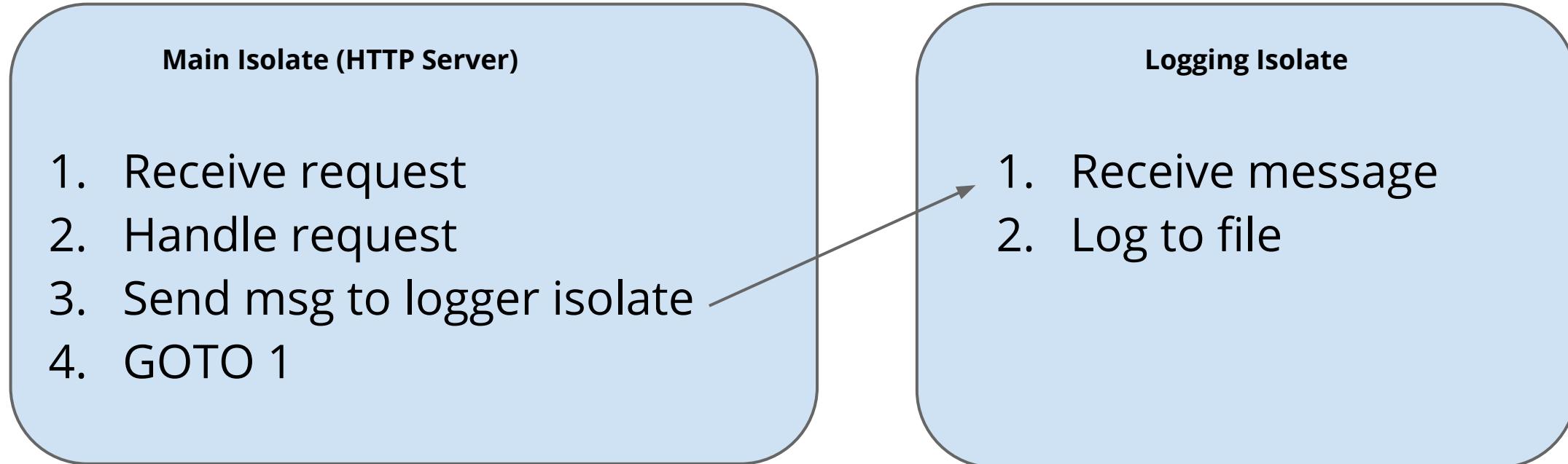
164  
microsec  
89% of  
total



#dartlang

disclaimer: totally unoptimized code

# Logging with isolates



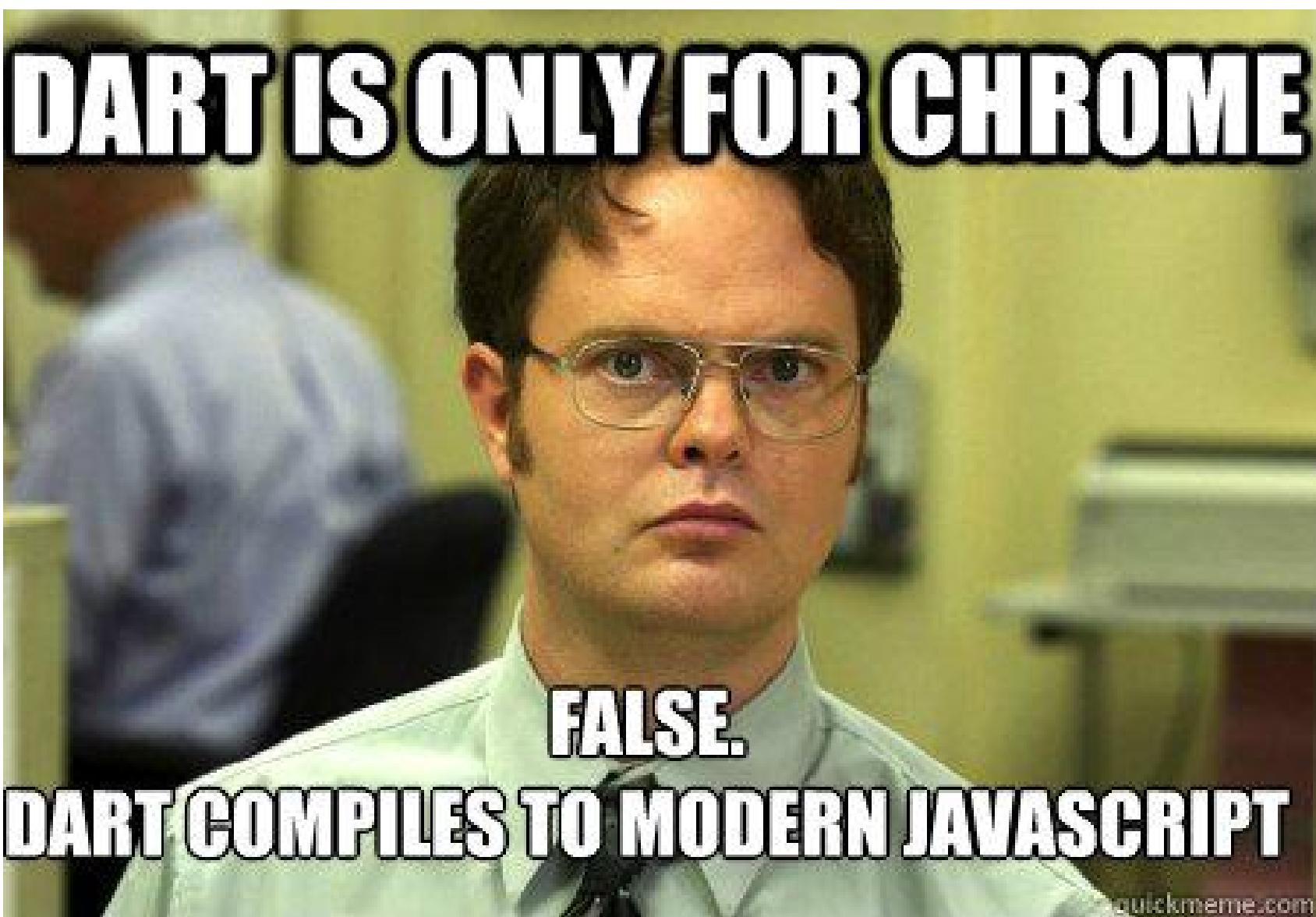
*41us total*

*164us to log to disk*



#dartlang

disclaimer: totally unoptimized code



#dartlang

quickmeme.com

```
import 'dart:html';

void main() {
  var button = new ButtonElement();

  button.text = 'Click me';
  button.classes.add('important');

  button.on.click.add((e) => window.alert('Clicked!!'));

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

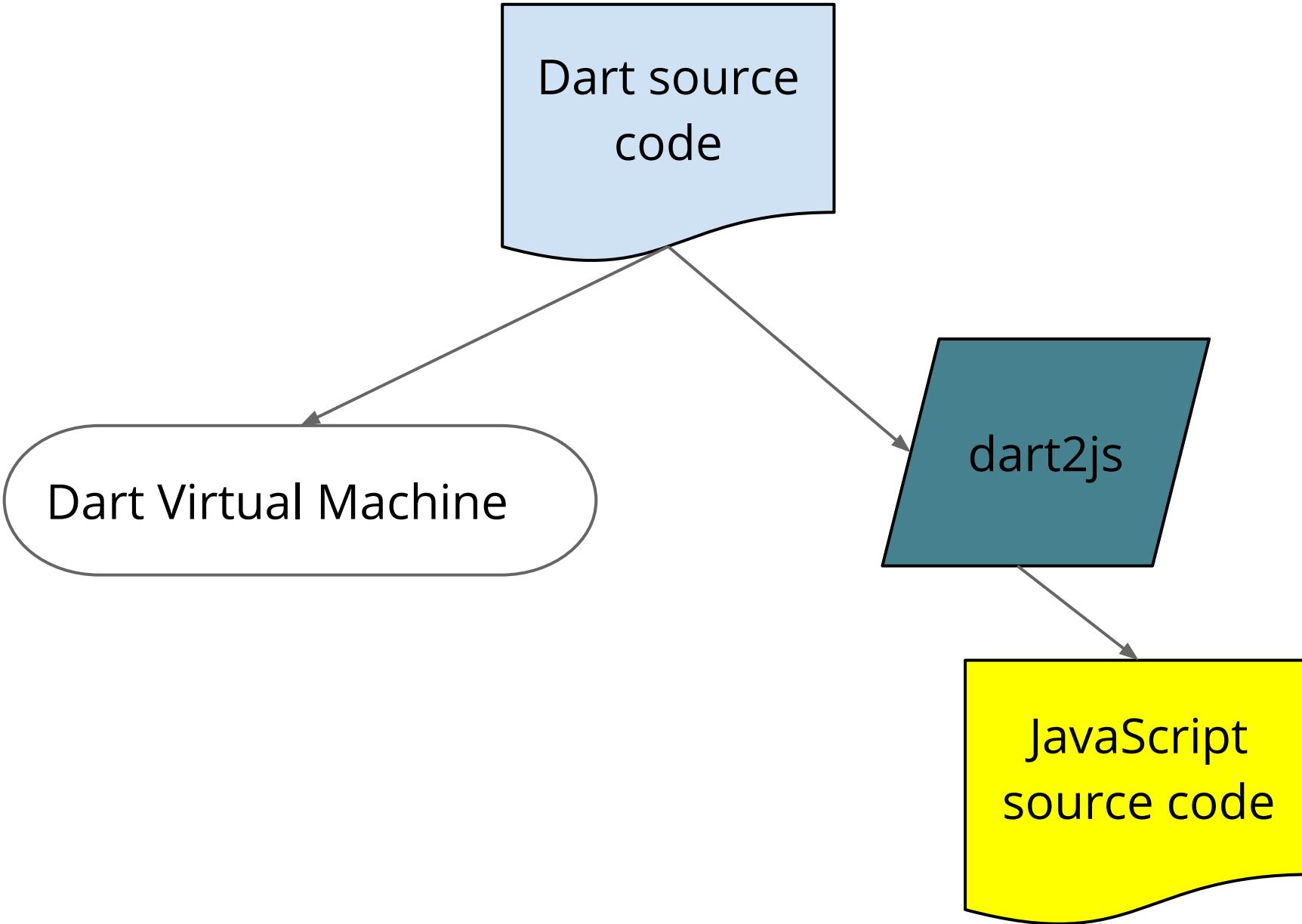


```
import 'dart:html';

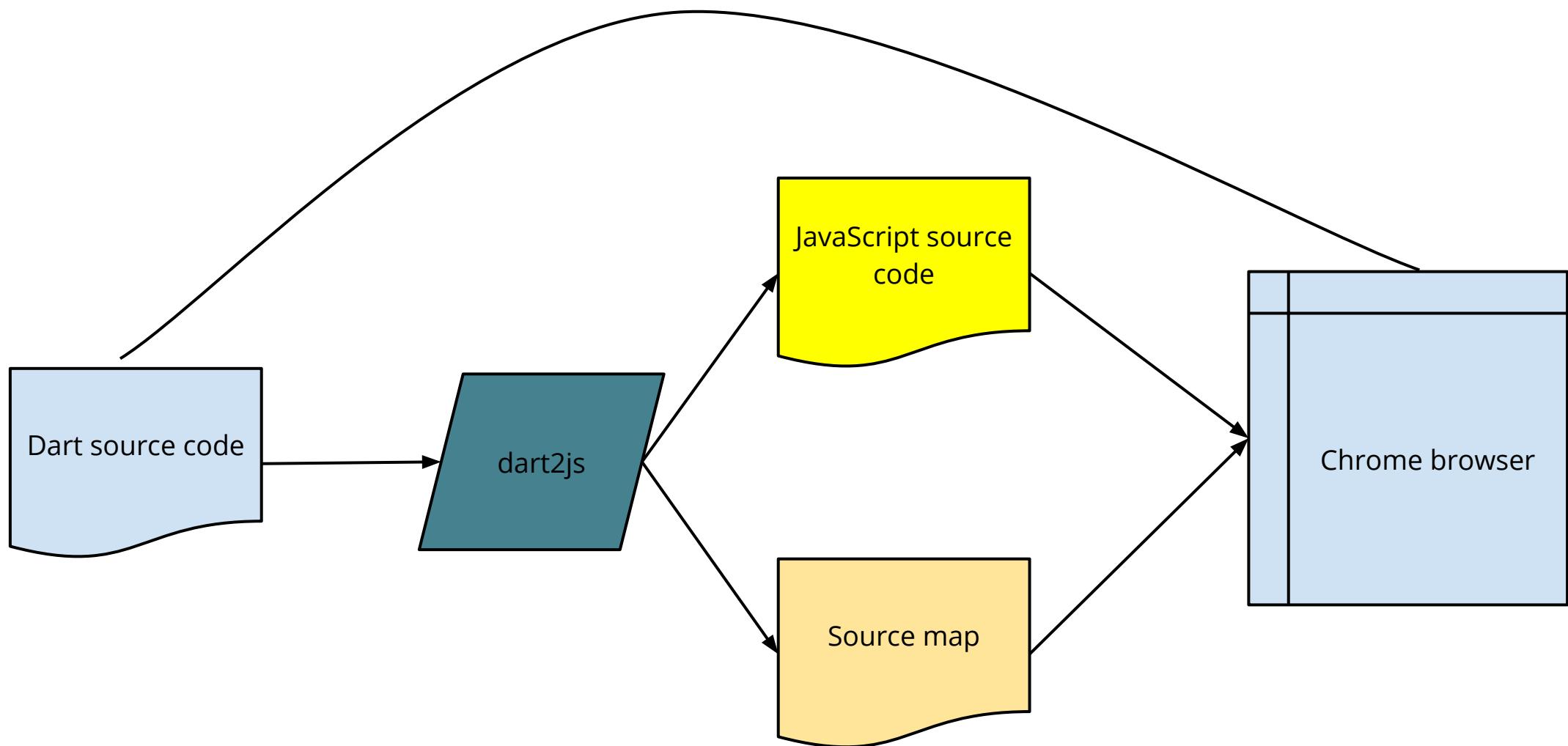
void main() {
  var button = new ButtonElement()
    ..text = 'Click me'
    ..classes.add('important')
    ..on.click.add((e) => window.alert('Clicked!!!'));

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



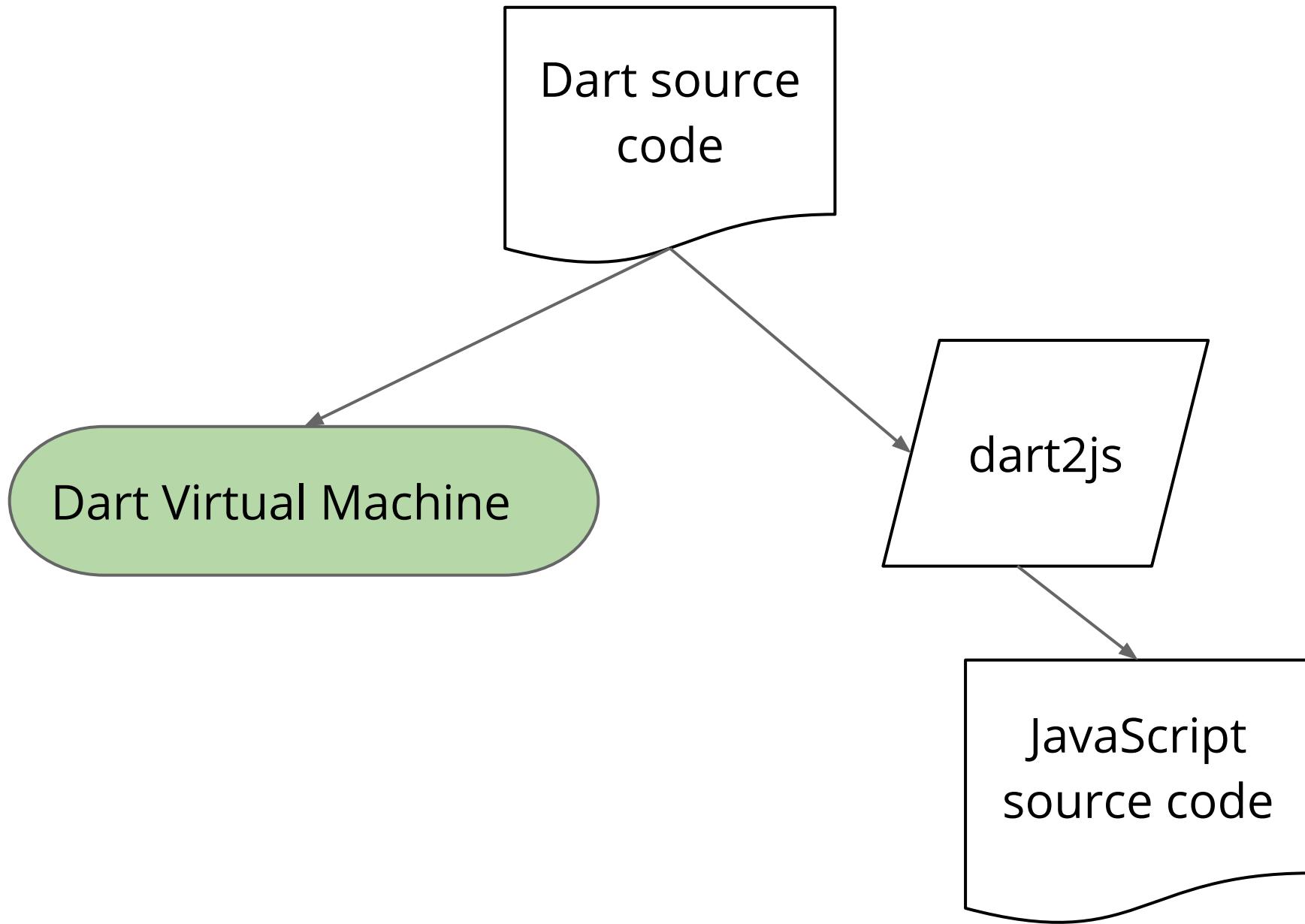


#dartlang

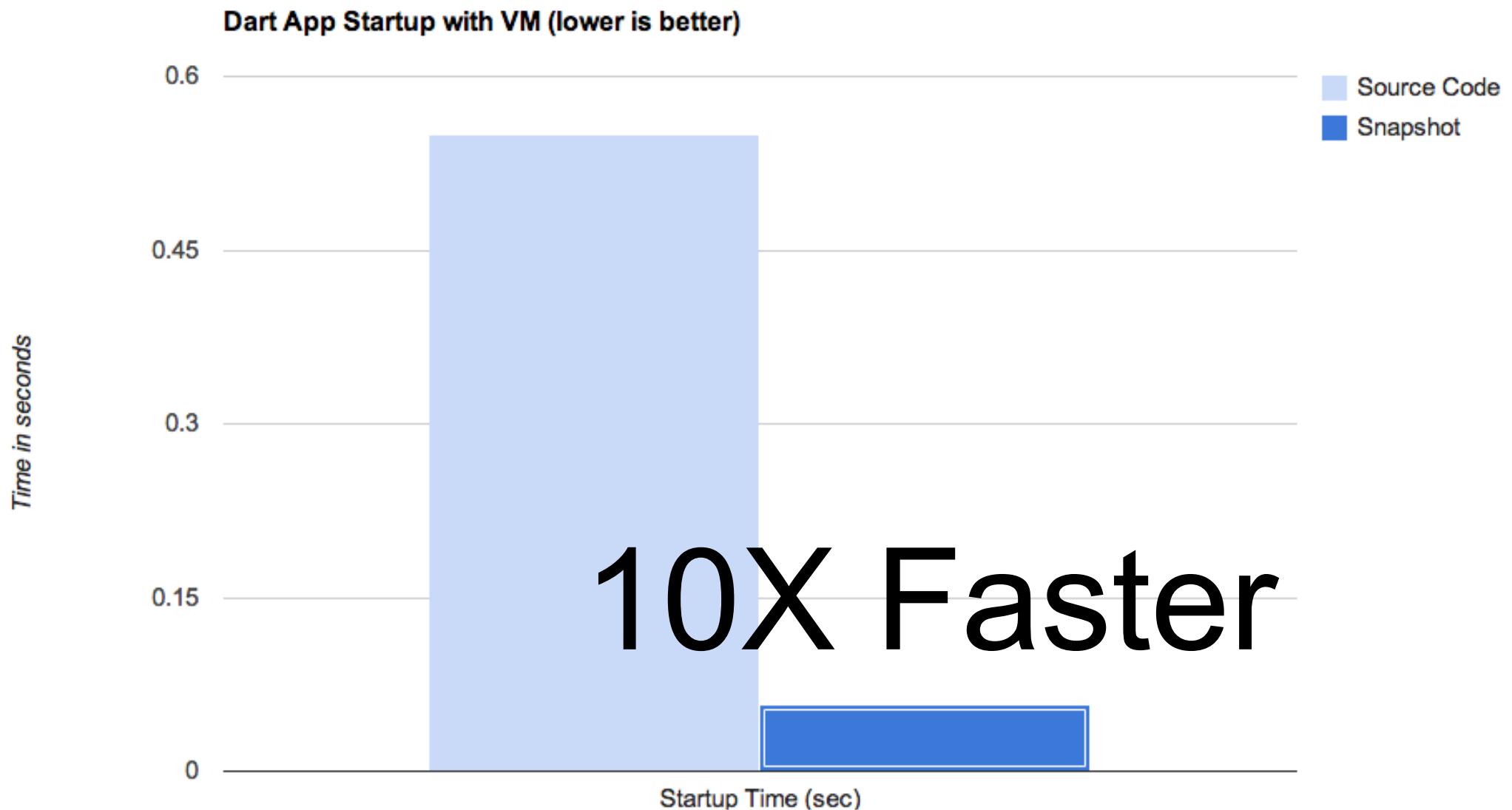


#dartlang

(demo clock)

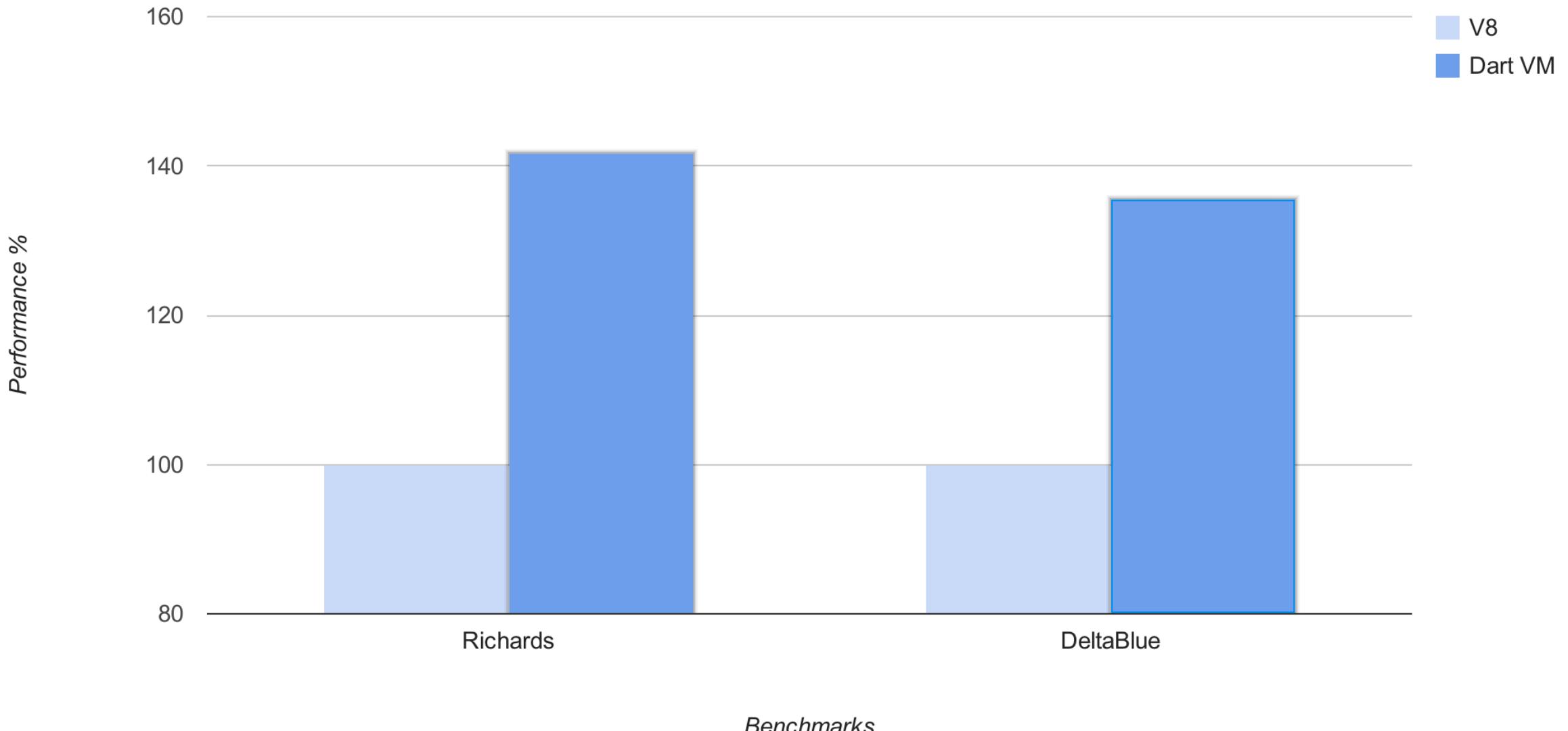


#dartlang

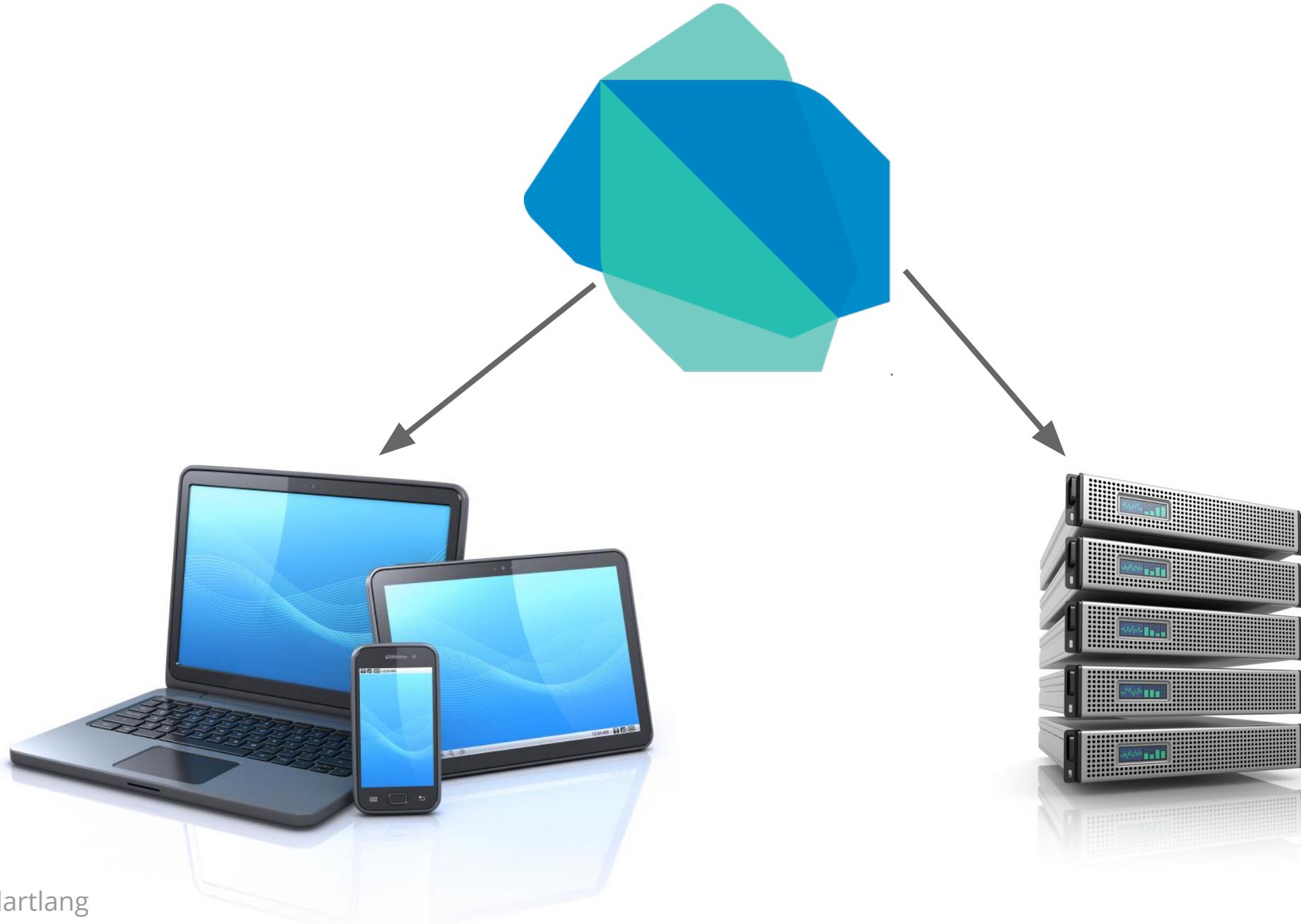


#dartlang

## Relative performance between V8 and Dart VM (higher is better)

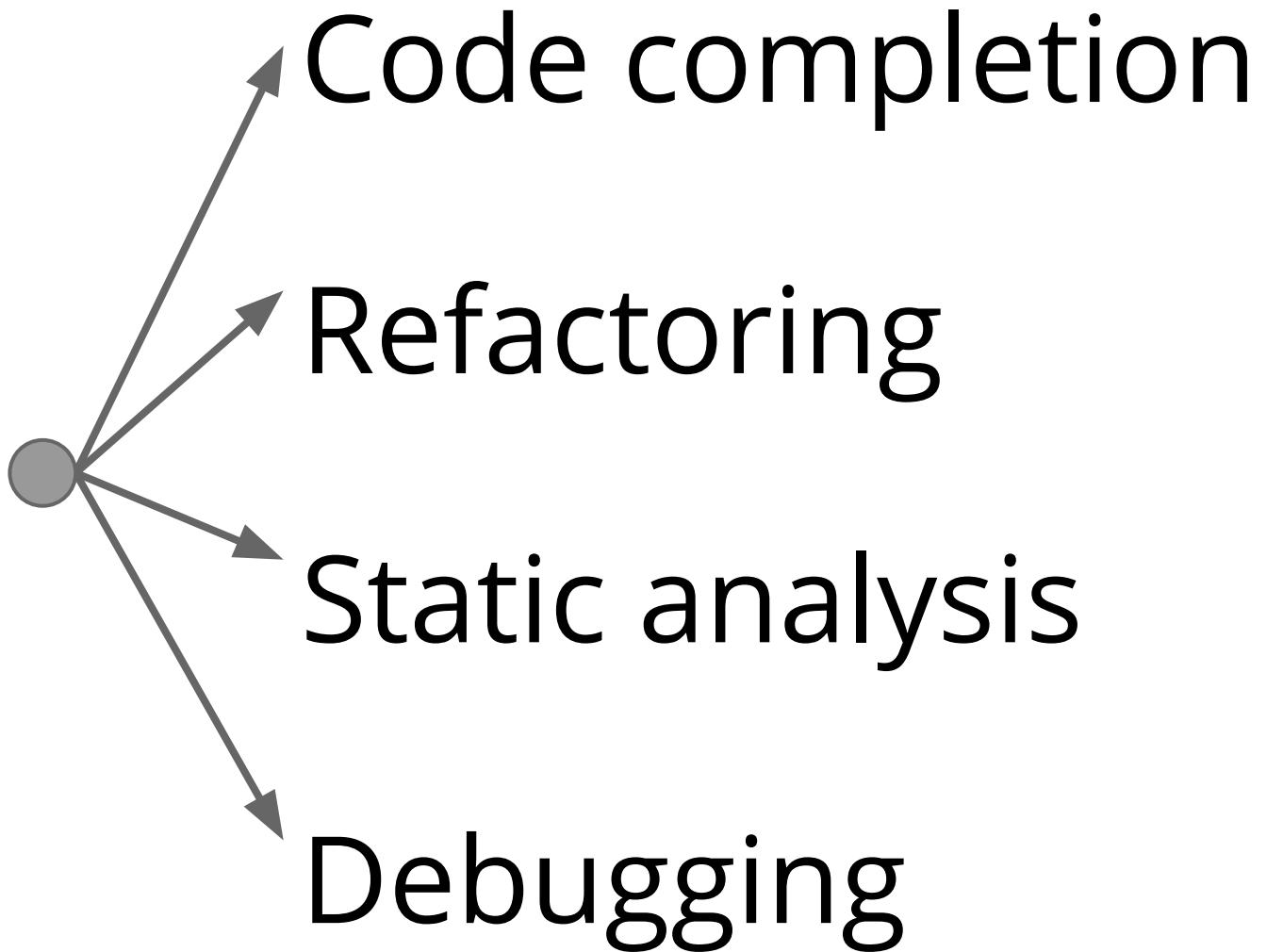


#dartlang



#dartlang

(demo)



#dartlang

(demo)

# Dart also supported in:



IntelliJIDEA  
WebStorm

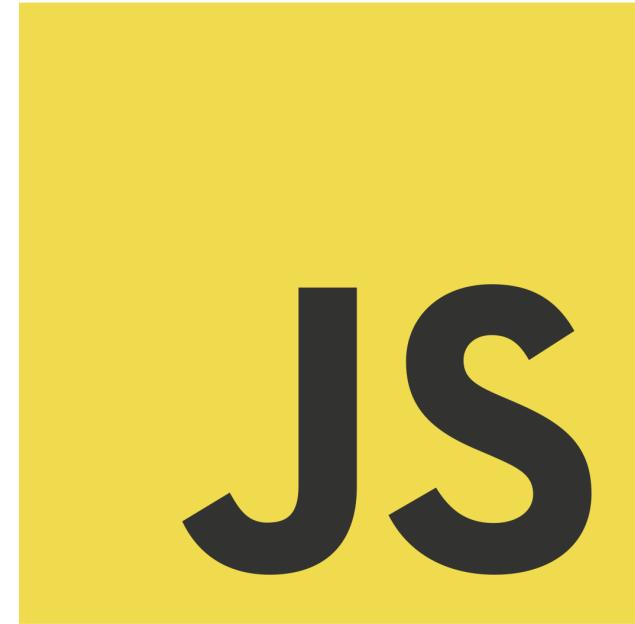
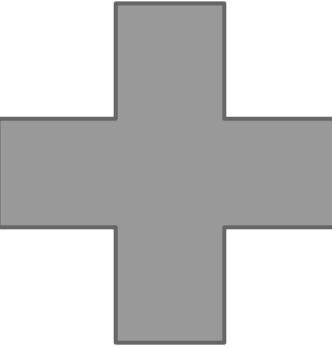


#dartlang



Use and share code with Pub

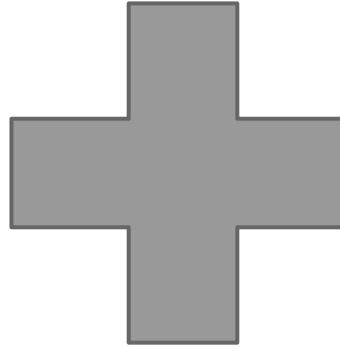
# Together at last!



#dartlang

(demo)

# Together at last!



WEB COMPONENTS



#dartlang



WEB COMPONENTS

# Declarative Renaissance



#dartlang

(demo)

# You decide:

```
<div tabindex="0" style="position: relative; min-height: 100%;">
  <div class="vI8oZc cS">...</div>
  <div class="nH" style="width: 1440px;">
    <div class="nH" style="position: relative;">
      <div class="nH w-asV aiw">...</div>
      <div class="nH">
        <div class="no">
          <div style="width: 220px; height: 662px;" class="nH oy8Mbf nn aeN">...</div>
          <div class="nH nn" style="width: 1220px;">
            <div class="nH">
              <div class="nH">
                <div class="ar4 z">
                  <div id=":ro" class="aeH">...</div>
                <div class="AO">
                  <div id=":rp" class="Tm aeJ" style="height: 642px;">
                    <div id=":rr" class="aef" style="min-height: 216px;">
                      <div class="nH">
                        <div class="BltHke nH oy8Mbf" style role="main">
                          <div></div>
                          <div class="afn"></div>
                          <div class="afn"></div>
                        <div class="UI" gh="tl">
                          <div class="aDP"></div>
                        <div class="ae4" style="border: 1px solid #ccc; padding: 5px; width: fit-content; margin: auto;">
                          <div>...</div>
                        <div class="Cp">
                          <div>
                            <table cellpadding="0" id=":nl" class="F cf zt">
                              <colgroup>...</colgroup>
                              <tbody>
                                <tr class="zA zE" id=":1v5">...</tr>
                                <tr class="zA zE" id=":33s">...</tr>
                                <tr class="zA zE" id=":373">
                                  <td class="PF xY"></td>
                                  <td id=":374" class="oZ-x3 xY aid" style="background-color: #e0f2e0; border: 1px solid #ccc; padding: 2px; font-size: small; color: inherit; text-decoration: none; cursor: pointer;">...</td>
                                  <td class="apU xY">...</td>
                                  <td class="WA xY">...</td>
                                  <td class="yX xY ">...</td>
                                <td id=":379" tabindex="0" role="link" class="xY">
                                  <div class="xS">
                                    <div class="xT">
                                      <div class="y6">
                                        <span id=":37b">...</span>
                                        <span class="y2">...</span>
                                      </div>
                                    </div>
                                  </td>
                                </tr>
                              </tbody>
                            </table>
                          </div>
                        </div>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

```
<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



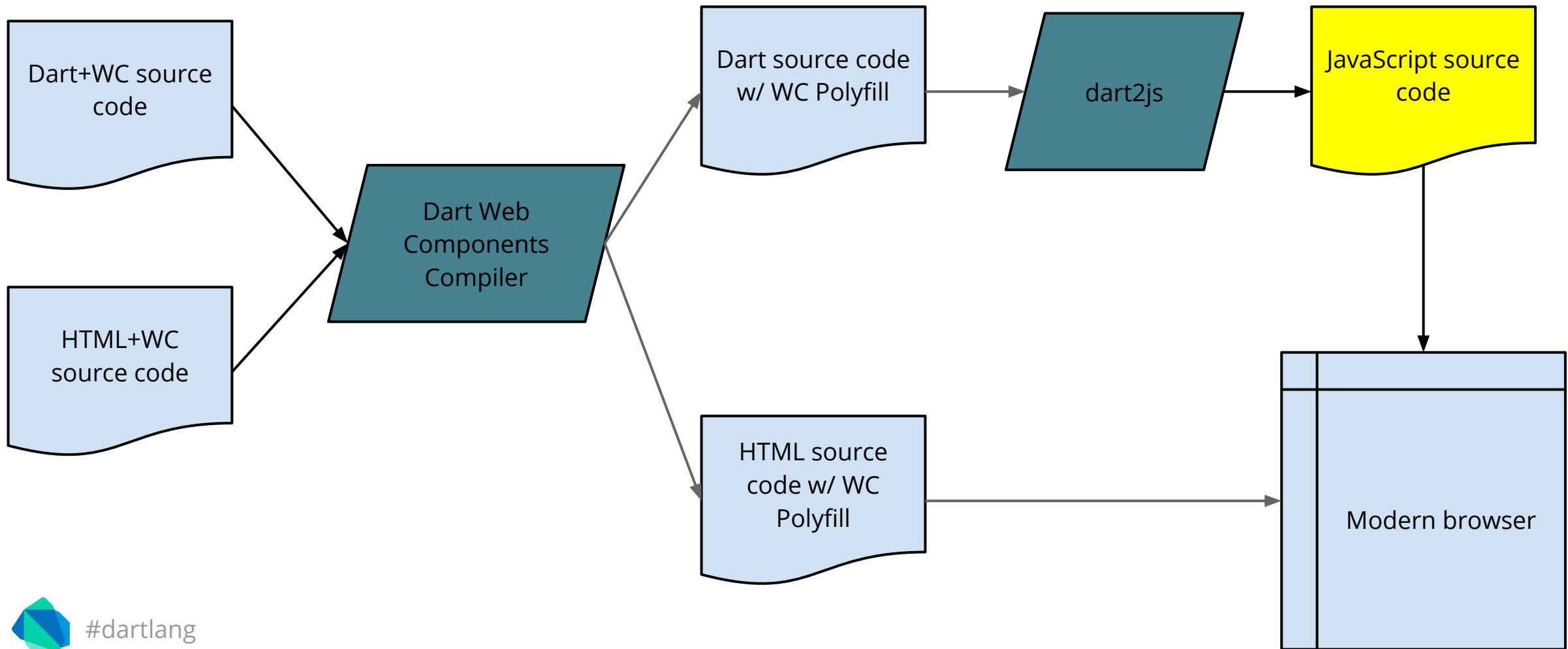
- Data binding
- Event handing
- Loops and conditionals
- Nested object binding
- Custom tags



#dartlang

(demo)

# Don't sweat it, this is automatic.



#dartlang

# Batteries included



- Language
- Libraries
- Editor
- VM
- JavaScript Compiler
- Package manager
- Web Components
- Browser integration



#dartlang



- Mixins
- Better async (??)
- Library improvements
- dart2js size and speed
- More reflection
- More stabilization
- M2... etc



#dartlang

# Visit dartlang.org

The screenshot shows the Dartlang.org homepage. At the top, there's a navigation bar with links for Docs, Tools, Resources, Development, News, and social sharing options (+1 and Tweet). The main header features the DART logo and a video player titled "Introducing Dart" showing a man speaking. Below the video are four calls-to-action: "Download now", "Circle +Dart", "Subscribe", and "Follow @dart\_lang". The page is divided into three main sections: "FAMILIAR" (describing Dart as a class-based, object-oriented language), "PRODUCTIVE" (describing the integrated development and debugging experience provided by the Dart Editor and SDK), and "UBIQUITOUS" (describing Dart's ability to be compiled to JavaScript for web apps and run on the server). A code snippet at the bottom illustrates Dart syntax.

- Tutorials
- Articles
- Downloads
- Videos and presentations
- Videocast series
- How to participate

## ✓ FAMILIAR

Dart is a **class-based, object-oriented language** with lexical scoping, closures, and optional static typing. Dart helps you build structured modern web apps and is **easy to learn** for a wide range of developers. [Learn more](#)

```
import 'dart:html';
```

## ⚙ PRODUCTIVE

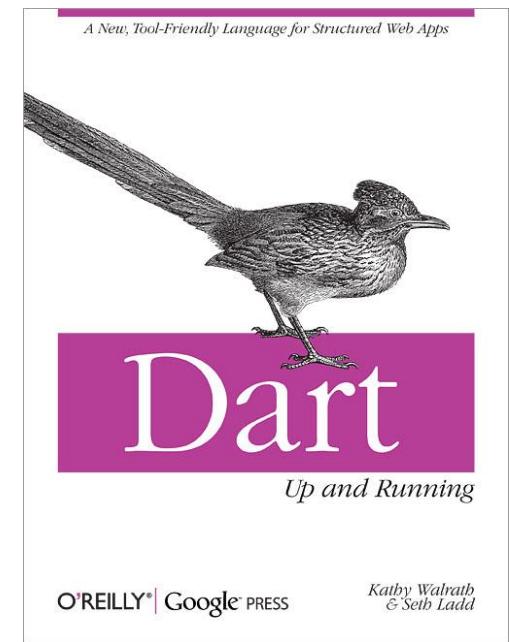
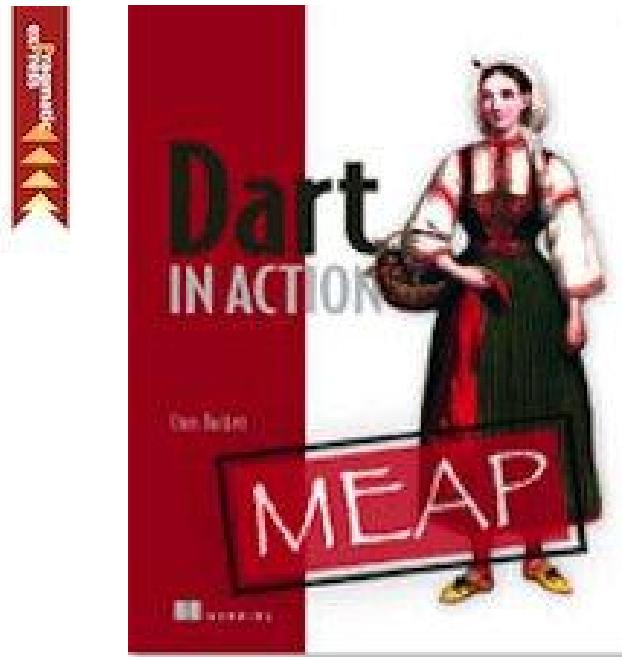
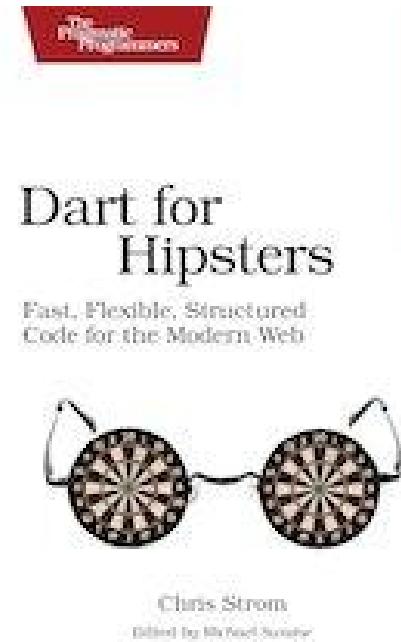
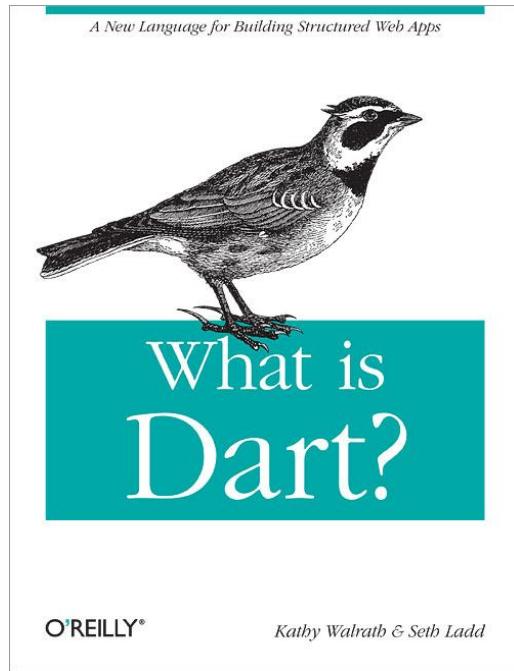
Dart Editor and the SDK provide an **integrated development and debugging** experience. The editor supports refactoring, breakpoints, code completion, code navigation, and more. The SDK contains the stand-alone virtual machine, a package manager, and Chromium with an embedded Dart VM. [Learn more](#)

## 🌐 UBIQUITOUS

Dart can be **compiled to JavaScript**, so you can use it for web apps in all modern desktop and mobile browsers. Our JavaScript compiler generates minimal code thanks to tree-shaking. Dart apps can also **run on the server**, in a stand-alone Dart VM.



# Read moar Dart!!1



#dartlang

# Coupons!

Use code: **WCYAZ**

**40%** off print

**50%** off ebook



#dartlang

*A New, Tool-Friendly Language for Structured Web Apps*



**Dart**  
*Up and Running*

O'REILLY® | Google™ PRESS

Kathy Walrath  
& Seth Ladd

## Dart Editor Feedback



Help improve Dart Editor by telling us what you think.

Please tell us what you think!

[misc@dartlang.org](mailto:misc@dartlang.org)

[google.com/+dartlang](https://google.com/+dartlang)

[Stack Overflow \(tag: dart\)](#)

[dartbug.com](http://dartbug.com)

Or, use Send Feedback in the editor.

Thank you!

Send additional editor data

[show data that will be sent](#)

Cancel

Send Feedback

# Thank You!

Try Dart today at [dartlang.org](https://dartlang.org)

@sethladd  
+Seth Ladd



#dartlang

# 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



#dartlang



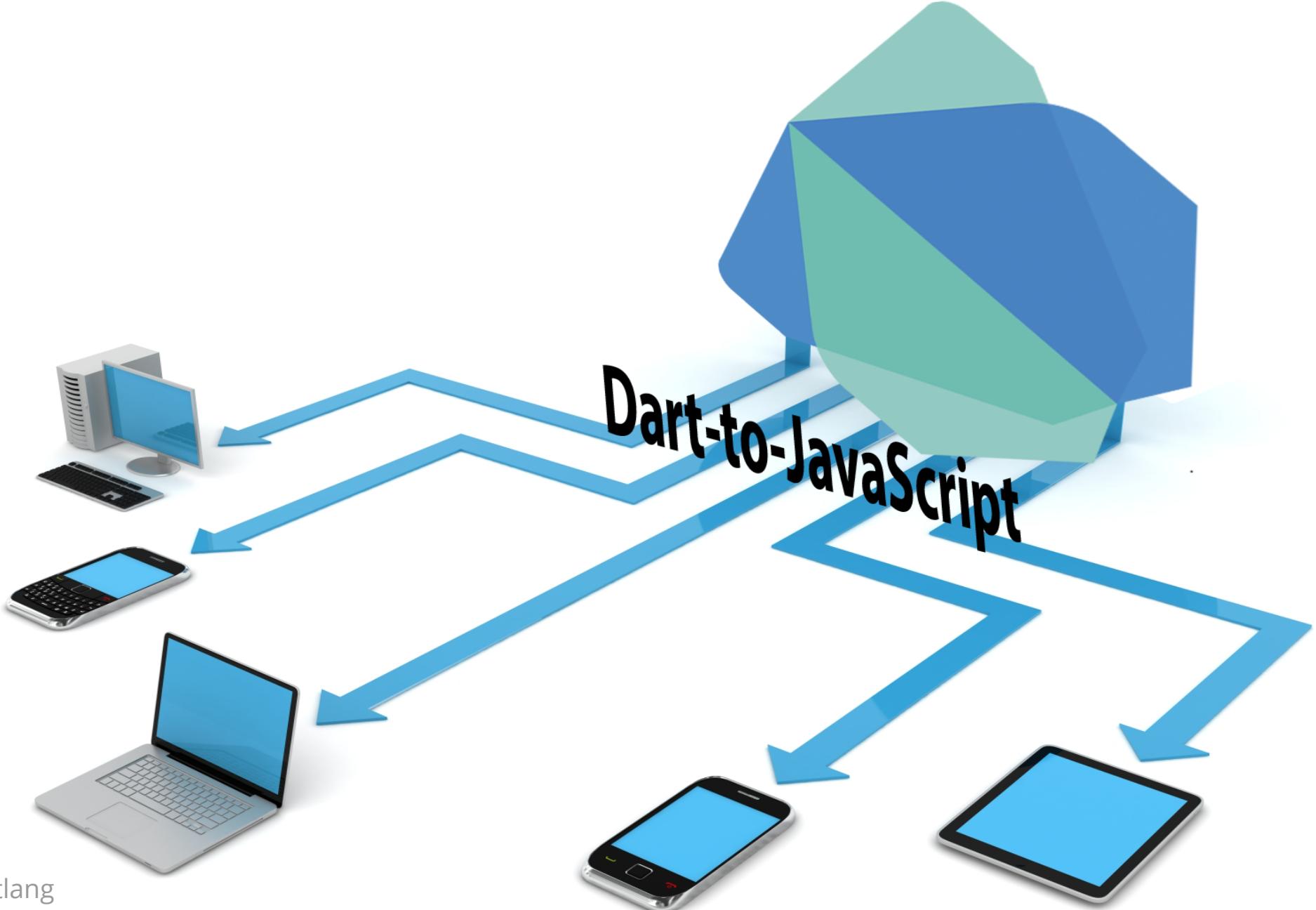
DART



On the train to 1.0!



#dartlang



#dartlang

(demo)