Modern Web Development with Dart

Alan Knight
Google

Smalltalk Industry Conference 2013

The Web

- As an application platform
- We want...
  - Large, extensible widget set
  - Simple, clean APIs
  - Model/View separation
  - Encapsulation
  - Portable
  - Fast startup
  - Performance
  - Good Development Experience

http://www.flickr.com/photos/brettlider/365444758/

Overview

- Making the web better
  - Web Components
  - Dart
- Relevance to Smalltalk

Web Components

- Data binding (Model-Driven Views)
- Custom Elements
- Shadow DOM
- Scoped CSS

==> Widgets for the web

Work in Progress

Model-Driven Views

Template syntax for two-way binding

```
<div>Current time is {{timeString}}</div>

@observable
var timeString;

@observable
palindrome() =>
   currentTime.split("").reversed.join();
```

Custom Elements

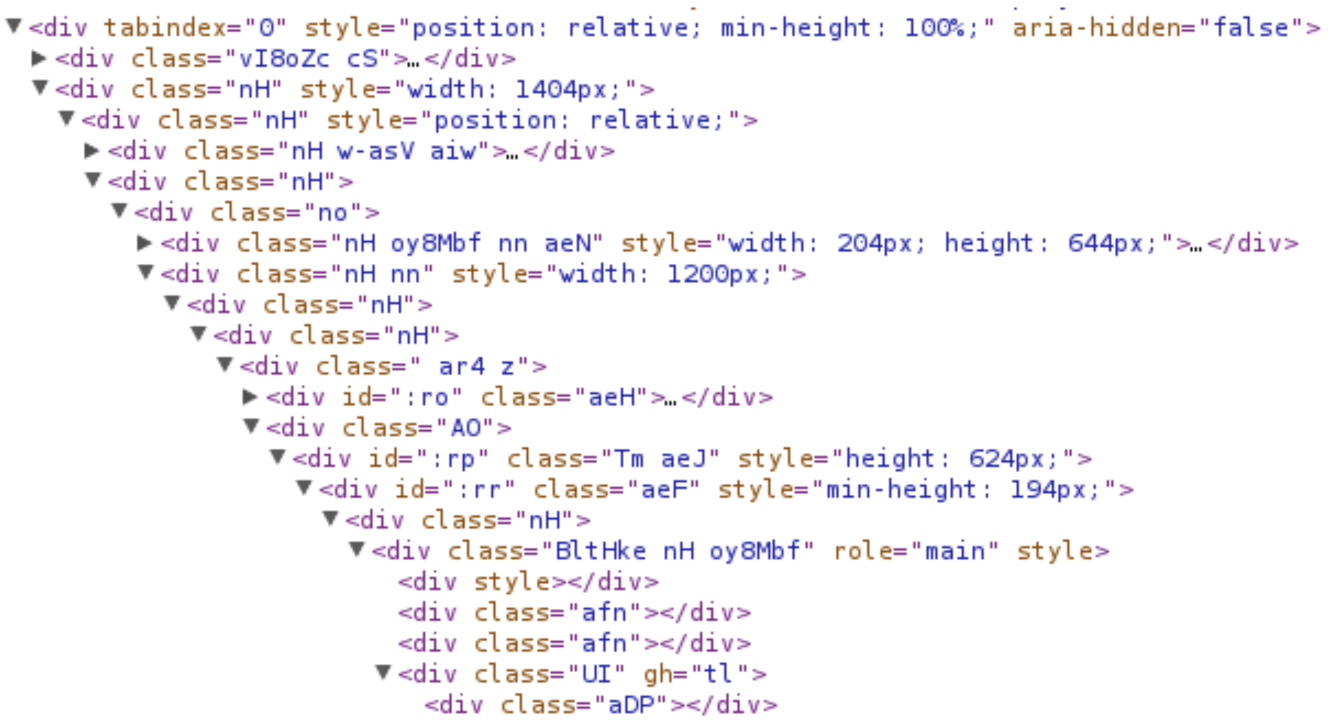
```
<element extends=”div”
  name=”x-converter”
  constructor=”Converter”>

class Converter extends WebComponent {...}

<x-converter ratio=”{{0.5}}”>
```

- Simplifies the HTML we write ...

But in the DOM...



```
▼<div tabindex="0" style="position: relative; min-height: 100%;" aria-hidden="false">
 ▶<div class="vI8oZc cS">…</div>
 ▼<div class="nH" style="width: 1404px;">
  ▼<div class="nH" style="position: relative;">
   ▶<div class="nH w-asV aiw">…</div>
   ▼<div class="nH">
    ▼<div class="no">
     ▶<div class="nH oy8Mbf nn aeN" style="width: 204px; height: 644px;">…</div>
     ▼<div class="nH nn" style="width: 1200px;">
      ▼<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: 624px;">
           ▼<div id=":rr" class="aeF" style="min-height: 194px;">
            ▼<div class="nH">
             ▼<div class="BltHke nH oy8Mbf" role="main" style>
               <div style></div>
               <div class="afn"></div>
               <div class="afn"></div>
              ▼<div class="UI" gh="tl">
                <div class="aDP"></div>
```

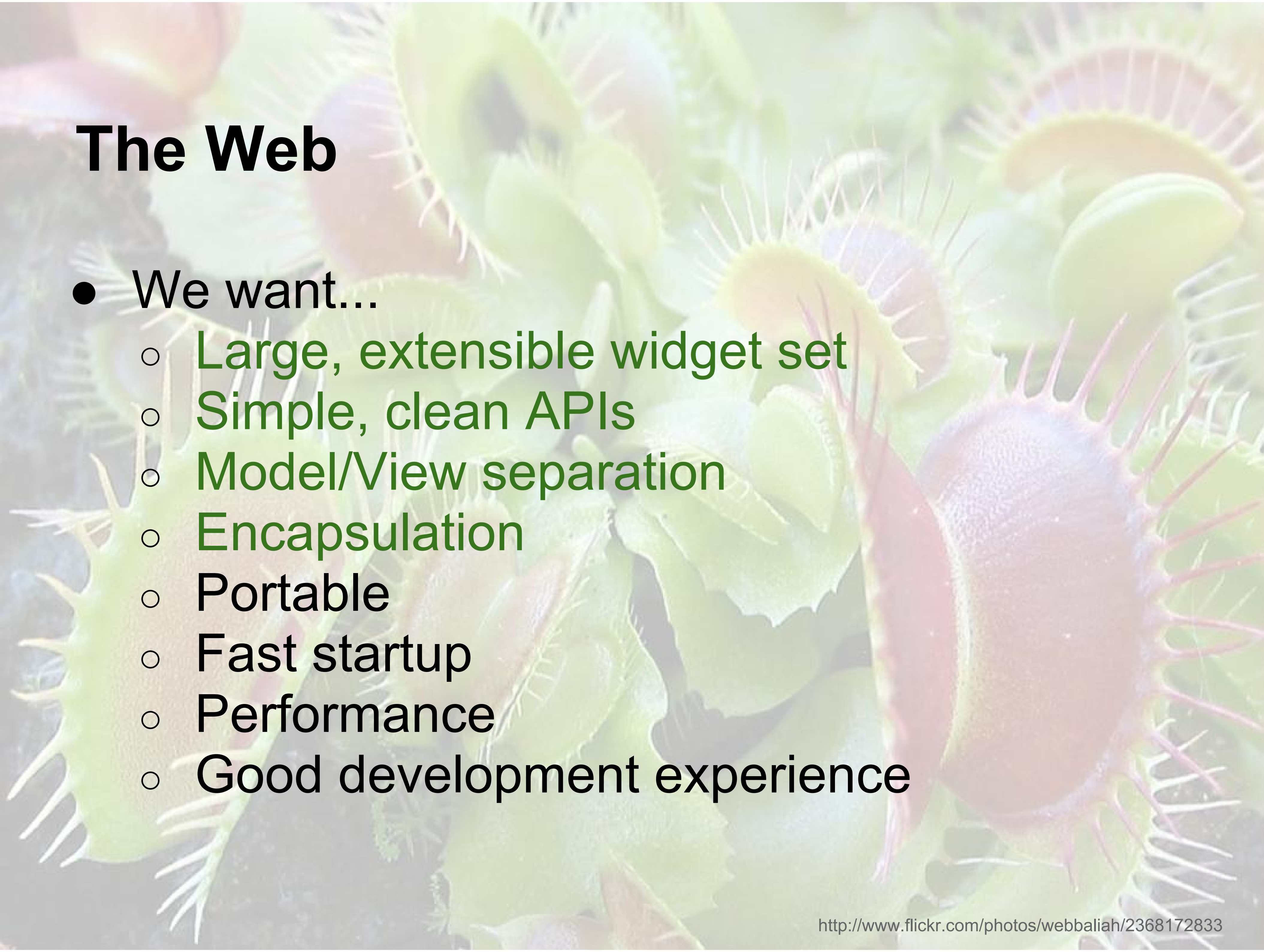
- Back to primitives
- No encapsulation
- Modularity, library usage, security

Encapsulation

- iframes
  - complete isolation
- Shadow DOM
  - Internal structure hidden from other nodes
  - CSS limited in scope
  - Ability to selectively expose structure

Implementations

- Multiple implementations
  - Mozilla: x-tags.org
  - Google: polymer-project.org
  - Dart web-ui
- Parts starting to ship in Chrome, Firefox



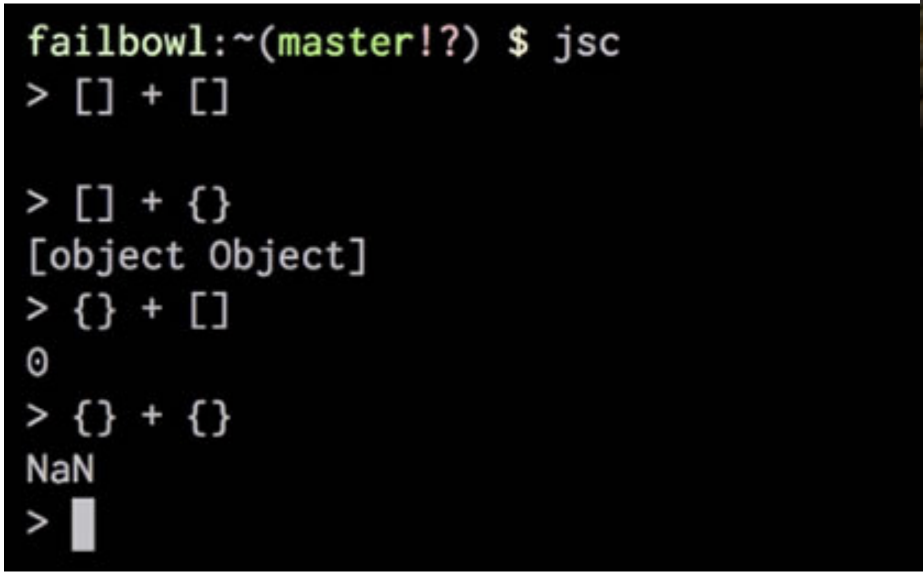
The Web

- We want...
  - Large, extensible widget set
  - Simple, clean APIs
  - Model/View separation
  - Encapsulation
  - Portable
  - Fast startup
  - Performance
  - Good development experience

http://www.flickr.com/photos/webbaliah/2368172833

Javascript (Sort-Of)

- Sort-of Scheme
- Sort-of Smalltalk
- Sort-of Self
- Sort-of Perl
- Sort-of Pascal



```
failbowl:~(master!?) $ jsc
> [] + []

> [] + {}
[object Object]
> {} + []
0
> {} + {}
NaN
> 
```

https://www.destroyallsoftware.com/talks/wat

Web Development Forces

- Download everything every time
- Size really matters
  - Mobile
  - Minification
- Fragmented libraries
- Hard to scale up

Dart

- Google-sponsored, open-source
- Dual-target: Javascript, Dart VM
- Goals
  - Structured, toolable
  - Good development experience
  - Rich standard libraries
  - 10x faster startup
  - Predictable 2x faster execution
  - Avoid fragmentation of the web, interop

Dart example

```
import "dart:math" as math;

class Thing {
  int _a; var b;
  Thing(this._a);
  get a => _a;
  toString() => 'aThing($a)';
  _log() => math.log(a);
  compare(other, {tolerance : 0}) {
    return (_log() - other._log()) <= tolerance;
  }
}
```

Dart Influences

- Syntax: JavaScript and C#
- Object Model: Smalltalk
- Compilation Strategy: Self
- Optional Types: Strongtalk
- Isolates: Erlang

Design constraints

- Familiar to "mainstream" programmers
- Compile to efficient JavaScript

Structured, but Dynamic

- Structured
  - classes
  - libraries
  - no evaluation of strings at runtime
  - reflection separated into "mirrors" library
- Static Types (optional)
  - Documentation
  - Code completion
  - Warnings
  - Zero runtime effect (production mode)
  - unsound
    - List<Shape> foo = new List<Circle>();

What Do You Mean "Types Help"?

- Smalltalk subtly helps avoid some errors

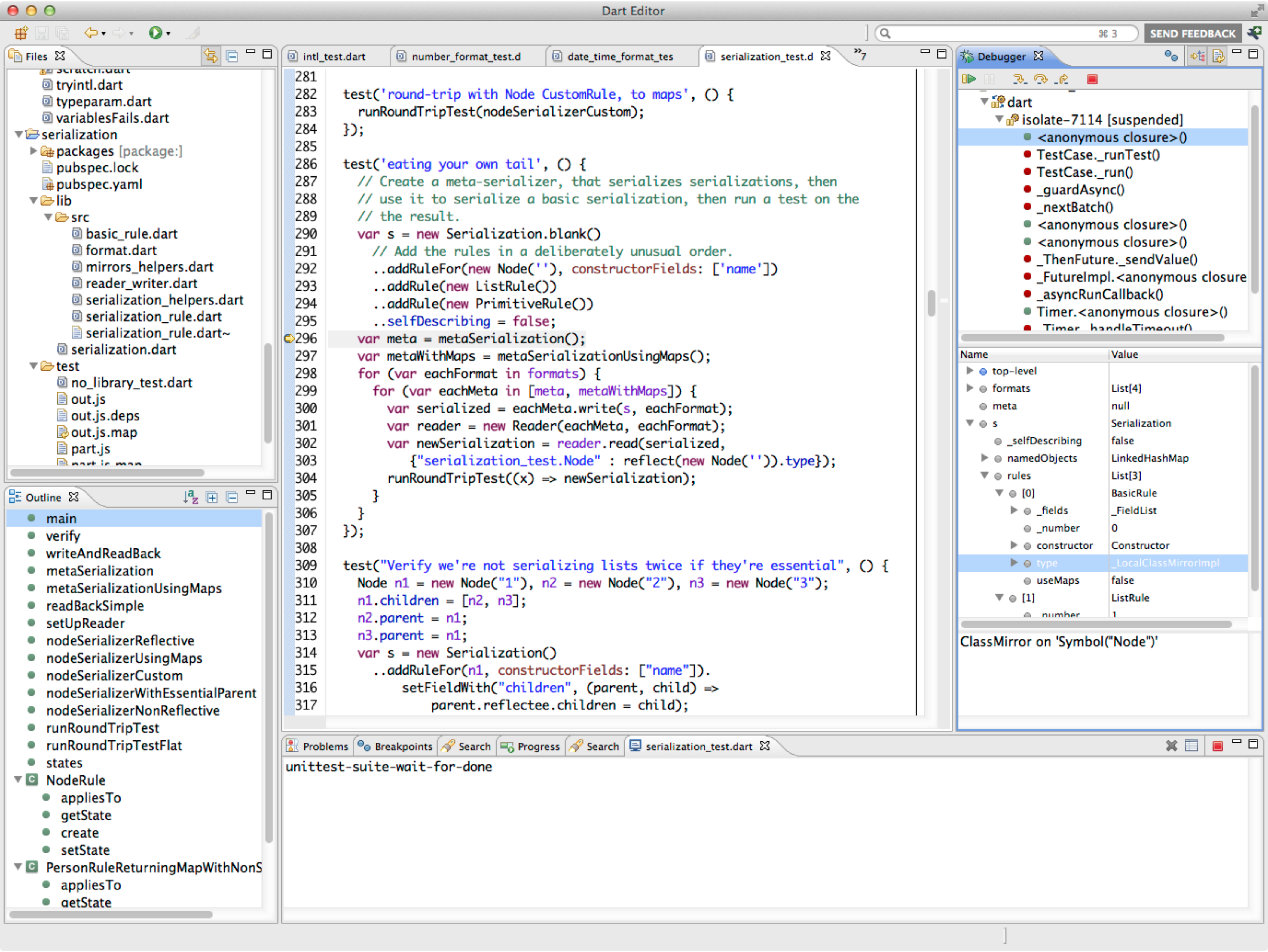
```
  if (anAccount.isOverdue) { ...
```

What if isOverdue is a method?

```
  if (anAccount.isOverdue()) { ...
```

What if isOverdue is a field or getter?

- image.onClick.listen ...



```
test('round-trip with Node CustomRule, to maps', () {
  runRoundTripTest(nodeSerializerCustom);
});

test('eating your own tail', () {
  // Create a meta-serializer, that serializes serializations, then
  // use it to serialize a basic serialization, then run a test on the
  // the result.
  var s = new Serialization.blank()
    // Add the rules in a deliberately unusual order.
    ..addRuleFor(new Node(''), constructorFields: ['name'])
    ..addRule(new ListRule())
    ..addRule(new PrimitiveRule())
    ..selfDescribing = false;
  var meta = metaSerialization();
  var metaWithMaps = metaSerializationUsingMaps();
  for (var eachFormat in formats) {
    for (var eachMeta in [meta, metaWithMaps]) {
      var serialized = eachMeta.write(s, eachFormat);
      var reader = new Reader(eachMeta, eachFormat);
      var newSerialization = reader.read(serialized,
          {"serialization_test.Node" : reflect(new Node('')).type});
      runRoundTripTest((x) => newSerialization);
    }
  }
});

test("Verify we're not serializing lists twice if they're essential", () {
  Node n1 = new Node("1"), n2 = new Node("2"), n3 = new Node("3");
  n1.children = [n2, n3];
  n2.parent = n1;
  n3.parent = n1;
  var s = new Serialization()
    ..addRuleFor(n1, constructorFields: ["name"]).
        setFieldWith("children", (parent, child) =>
            parent.reflectee.children = child);
```

```
dart
  isolate-7114 [suspended]
    <anonymous closure>()
    TestCase._runTest()
    TestCase._run()
    _guardAsync()
    _nextBatch()
    <anonymous closure>()
    <anonymous closure>()
    _ThenFuture._sendValue()
    _FutureImpl.<anonymous closure
    _asyncRunCallback()
    Timer.<anonymous closure>()
    Timer._handleTimeout()
```

Name	Value
top-level	
formats	List[4]
meta	null
s	Serialization
_selfDescribing	false
namedObjects	LinkedHashMap
rules	List[3]
[0]	BasicRule
_fields	_FieldList
_number	0
constructor	Constructor
type	_LocalClassMirrorImpl
useMaps	false
[1]	ListRule
_number	1

ClassMirror on 'Symbol("Node")'

```
unittest-suite-wait-for-done
```

Idiomatic DOM APIs...

DOM: Implemented in C++, specified in IDL

Javascript

```
var arr = document.body.childNodes;
for (var index in arr) {
  // index = 0, 1, 2, 3, length, item
```

Dart

```
List<Node> list= document.body.children;
for(var eachNode in list) {
```

OR

```
list.where((each) => each is MyElement))
  .map((x) => x.thing));
```

...DOM APIs

- Good coverage of APIS
- Sub-libraries rather than prefixes
- Compatibility layer for browser variations
- Named arguments
- JQuery-like mechanisms, performant
- XSS-safe!

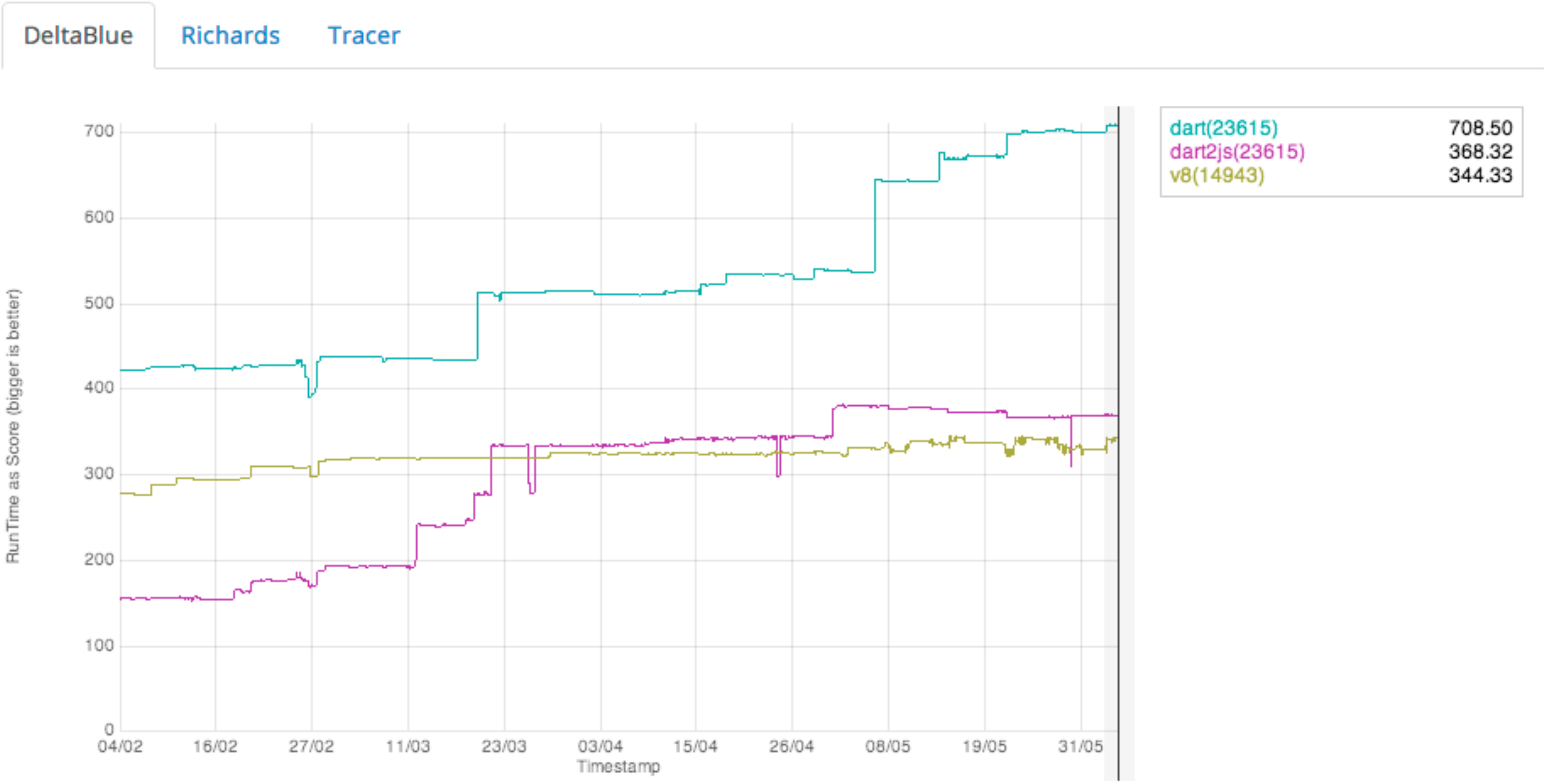
Performance - Dart2js

- Code size, transformations
- Minification
- Tree-shaking
- Dart2dart
- Static types not used
- Requires care with mirrors and other general facilities
  - MirrorSystem.getName(aSymbol)
  - Function.apply

Performance - VM

- Startup
  - snapshots
  - parsing
  - Runtime
- Multiple levels of JIT
  - Inlining
  - OSR
  - Bailouts and de-optimization
  - SIMD

Performance



http://www.dartlang.org/performance/

Things Smalltalkers Will Like

- Optional, unsound types
- Cascades
- Mirrors, MirrorBuilders
- Classes
- Speed
- Collections
- noSuchMethod
- Function.apply
- Symbols
- Named parameters
- Terseness
- Numbers
- Redefining operators
    - +, -, =, call

Mirror examples

```
var p = new Person();
var mirror = reflect(thing);
mirror.setField(const Symbol("name"), "Alan");
mirror.invoke(new Symbol(methodName), []);

Function.apply(p.doSomething, [1, 2, 3]);

var classMirror = reflectClass(Person);
classMirror.newInstance([]);
```

Things Smalltalkers Will Miss

- Class extensions (especially with type tests)
- Non-local returns
- Resumable exceptions
- Class inheritance
- Really easy meta-programming
- become:, thisContext, etc.
- Generality on class operations
  - new MustBeLiteralClassName()
- Coding in the debugger
- DSL-building
- Development image, tool extensibility

"Source code in files. How quaint. How 70's" - Kent Beck

New Things for Smalltalkers

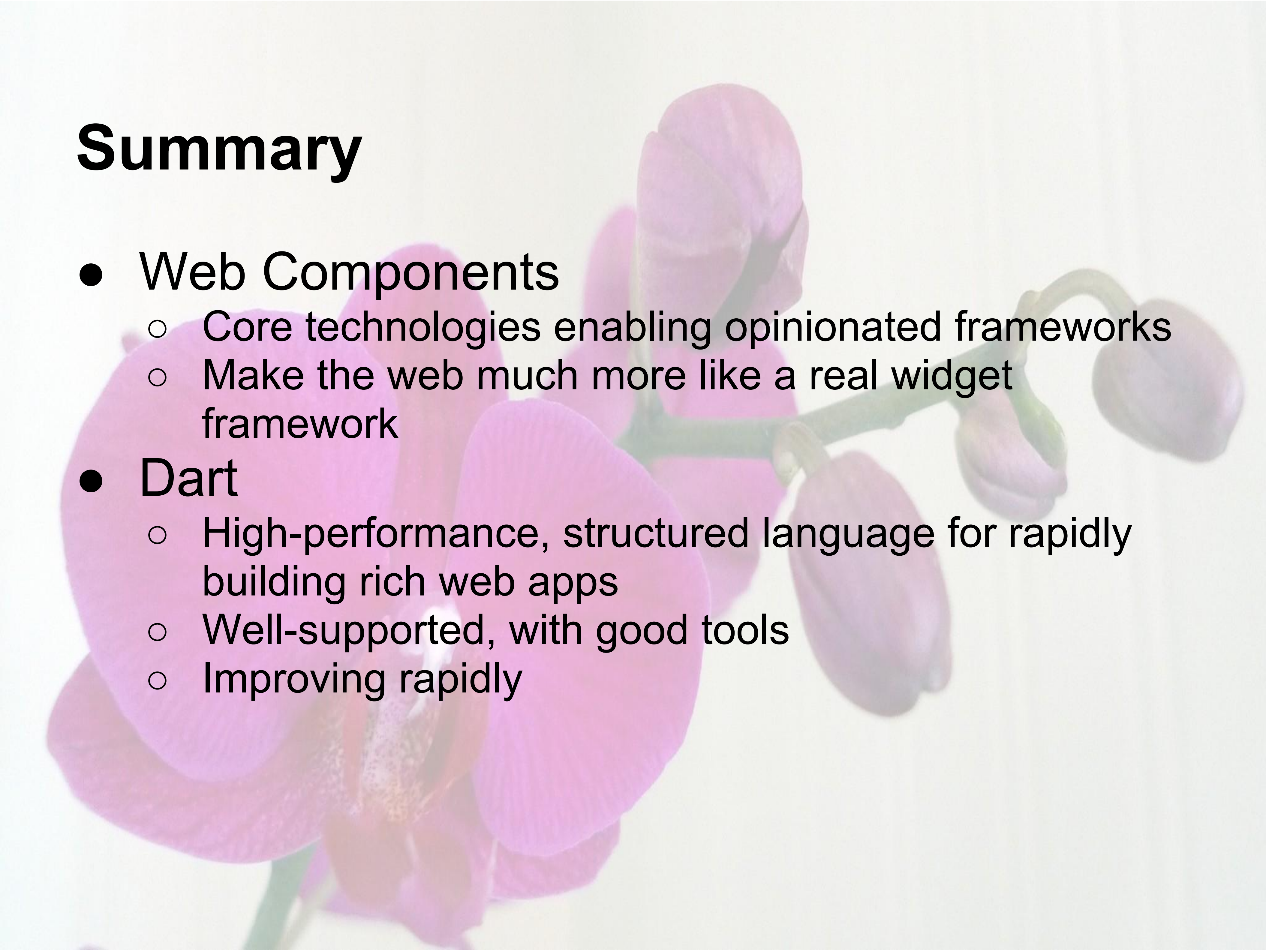
- Standalone functions, closurization
- String interpolation
- Mixins
- Library-based privacy
- Easy tree-shaking
- Source VM, no bytecode
- Async concurrency and isolates

Why should Smalltalkers care?

- Complex web clients
- Congenial language
- Well-supported
- Fast
- Portable
- Runs "natively" on the web
- Compilation target? - NS2Dart

Dart Status

- Not yet 1.0
  - Currently M4, M5 soon
- Targeting "Modern" web browsers
  - Safari 5.1+
  - Not IE8-
- Dart VM standalone
- "Dartium" for development
- In use in real sites
  - glyph3d.com
  - "critical internal projects"



Summary

- Web Components
  - Core technologies enabling opinionated frameworks
  - Make the web much more like a real widget framework
- Dart
  - High-performance, structured language for rapidly building rich web apps
  - Well-supported, with good tools
  - Improving rapidly

Resources

- www.dartlang.org
- try.dartlang.org
- Google I/O sessions
  - Dart: HTML of the future: Cherem, Fortuna
  - Fast Code is Always in Fashion: Bak, Lund
  - Others
- Mailing lists, StackOverflow
- Web Components
  - polymer-project.org
  - x-tags.org