



in Action

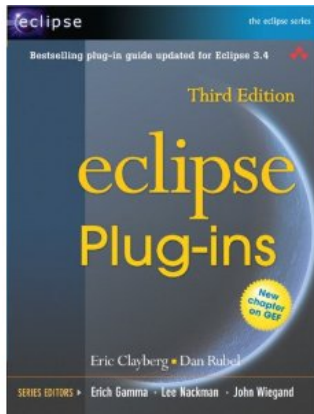
An introduction to the
Dart language and tools



Who am I



- Technical Lead for Dart Editor at Google
- Former CTO at Instantiations
- Co-author of Eclipse Plug-ins and Eclipse GEF





Overview

- Motivation
- Language
- Tools
- Performance
- Demo

Special thanks to Seth Ladd, Florian Loitsch, Gilad Bracha, Steve Messick, Brian Wilkerson, Alan Knight, and Eric Clayberg, for slides and ideas



Web Programming

- Small apps are easy
- Platform independent
- No installation
- Platform improving fast
- Everywhere... and getting more modern

~50% of users on IE9/FF7/Chrome/Safari



Why create Dart?

- Developing large applications is hard
 - Hard to find program structure
 - No static types
 - No support for libraries
 - Weak tool support
 - Slow startup
- Lots of cruft after 15 years



Our goal...

**Help app developers
write complex, high
fidelity client apps
for the modern web**



Dart is ...

- Structured Web Programming
 - New language
 - New tools
 - New libraries
- Open source as of early October 2011
- Available at <http://dartlang.org>



The Dart Language

- Object oriented language for the web
- Optional types
- Libraries, Isolates
- Real lexical scoping
- Single threaded



Hello World

```
#import('dart:html');

void main() {
  new Hello().doStuff();
}

class Hello {
  void doStuff() {
    var message = "Hello World";
    document.querySelector('#status').innerHTML = message;
  }
}
```

more at <http://dartlang.org/language-tour/>



Hello World

```
#import('dart:html');
```

```
void main() {  
  new Hello().doStuff();  
}
```

```
class Hello {  
  void doStuff() {  
    var message = "Hello World";  
    document.query('#status').innerHTML = message;  
  }  
}
```

Libraries

A red arrow originates from the word "Libraries" and points to the `#import('dart:html');` line in the code above.



Hello World

```
#import('dart:html');
```

```
void main() {  
  new Hello().doStuff();  
}
```

← **Functions**

```
class Hello {  
  void doStuff() {  
    var message = "Hello World";  
    document.query('#status').innerHTML = message;  
  }  
}
```

more at <http://dartlang.org/language-tour/>



Hello World

```
#import('dart:html');
```

```
void main() {  
  new Hello().doStuff();  
}
```

```
class Hello {  
  void doStuff() {  
    var message = "Hello World";  
    document.query('#status').innerHTML = message;  
  }  
}
```

Classes





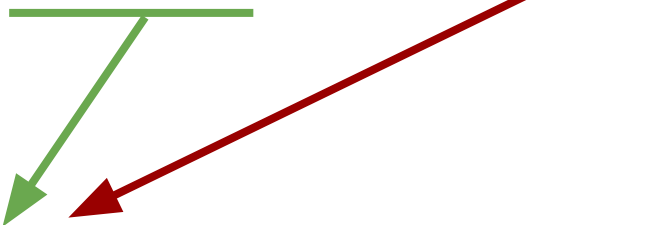
Hello World

```
#import('dart:html');
```

```
void main() {  
  new Hello().doStuff();  
}
```

```
class Hello {  
  void doStuff() {  
    var message = "Hello World";  
    document.query('#status').innerHTML = message;  
  }  
}
```

Methods



more at <http://dartlang.org/language-tour/>



Hello World

```
#import('dart:html');
```

```
void main() {  
  new Hello().doStuff();  
}
```

```
class Hello {  
  void doStuff() {  
    var message = "Hello World";  
    document.query('#status').innerHTML = message;  
  }  
}
```

Optional Types

A red arrow originates from the text "Optional Types" and points to the "var" keyword in the code snippet above, specifically highlighting the line "var message = 'Hello World';".

more at <http://dartlang.org/language-tour/>



Hello World

```
#import('dart:html');
```

```
void main() {  
  new Hello().doStuff();  
}
```

```
class Hello {  
  void doStuff() {  
    String message = "Hello World";  
    document.query('#status').innerHTML = message;  
  }  
}
```

Optional Types

A red arrow originates from the text "Optional Types" and points diagonally down and to the left, ending at the "String" type in the line "String message = 'Hello World';" within the class definition.

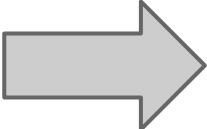


Optional Static Types

- Low friction mechanism for communicating intent to machines and other developers
- Easily scale up from prototype (untyped) to production (typed)
- Increases your productivity via Dart Editor and other tools

Types At Runtime

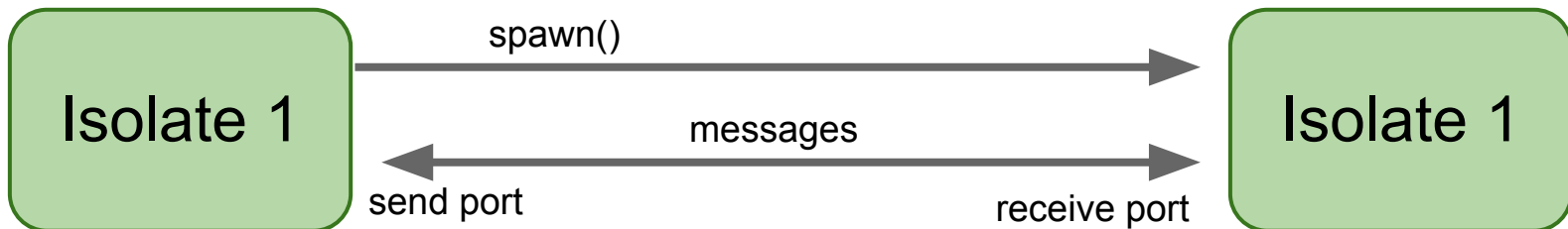
- Developers may check types at runtime...

$T\ x = o$  `assert(o === null || o is T)`

- By default, types have
 - No effect
 - No runtime cost

Isolates

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





Isolates

- Can be ...
 - Lightweight on UI thread
 - Heavyweight on their own thread
- Uses ...
 - Isolate 3rd party code
 - JavaScript interop
 - Client / Server communication



Dart Board

<http://www.dartlang.org/>

Get started

Read a [technical overview](#), take a [language tour](#), or download [Dart Editor](#). Or play with Dart code right here in your browser.

```
Fibonacci
```

```
▶ Checked Mode
```

```
1 int fib(int n) {
2   if (n <= 1) return n;
3   return fib(n - 1) + fib(n - 2);
4 }
5
6 main() {
7   print('fib(20) = ${fib(20)}');
8 }
```



Dart Board

Select template

Get started

Read a [technical overview](#), take a [language tour](#), or download [Dart Editor](#). Or play with Dart code right here in your browser.

Fibonacci

```
1 int fib(int n) {
2   if (n <= 1) return n;
3   return fib(n - 1) + fib(n - 2);
4 }
5
6 main() {
7   print('fib(20) = ${fib(20)}');
8 }
```



Dart Board

Type stuff

Get started

Read a [technical overview](#), take a [language tour](#), or download [Dart Editor](#). Or play with Dart code right here in your browser.

```
Fibonacci
```

```
1 int fib(int n) {
2   if (n <= 1) return n;
3   return fib(n - 1) + fib(n - 2);
4 }
5
6 main() {
7   print('fib(20) = ${fib(20)}');
8 }
```

A red arrow originates from the text "Type stuff" and points to the word "int" in the first line of the code block, highlighting the type annotation.




Dart Board

Run program

Get started

Read a [technical overview](#), take a [language tour](#), or download [Dart Editor](#). Or play with Dart code right here in your browser.

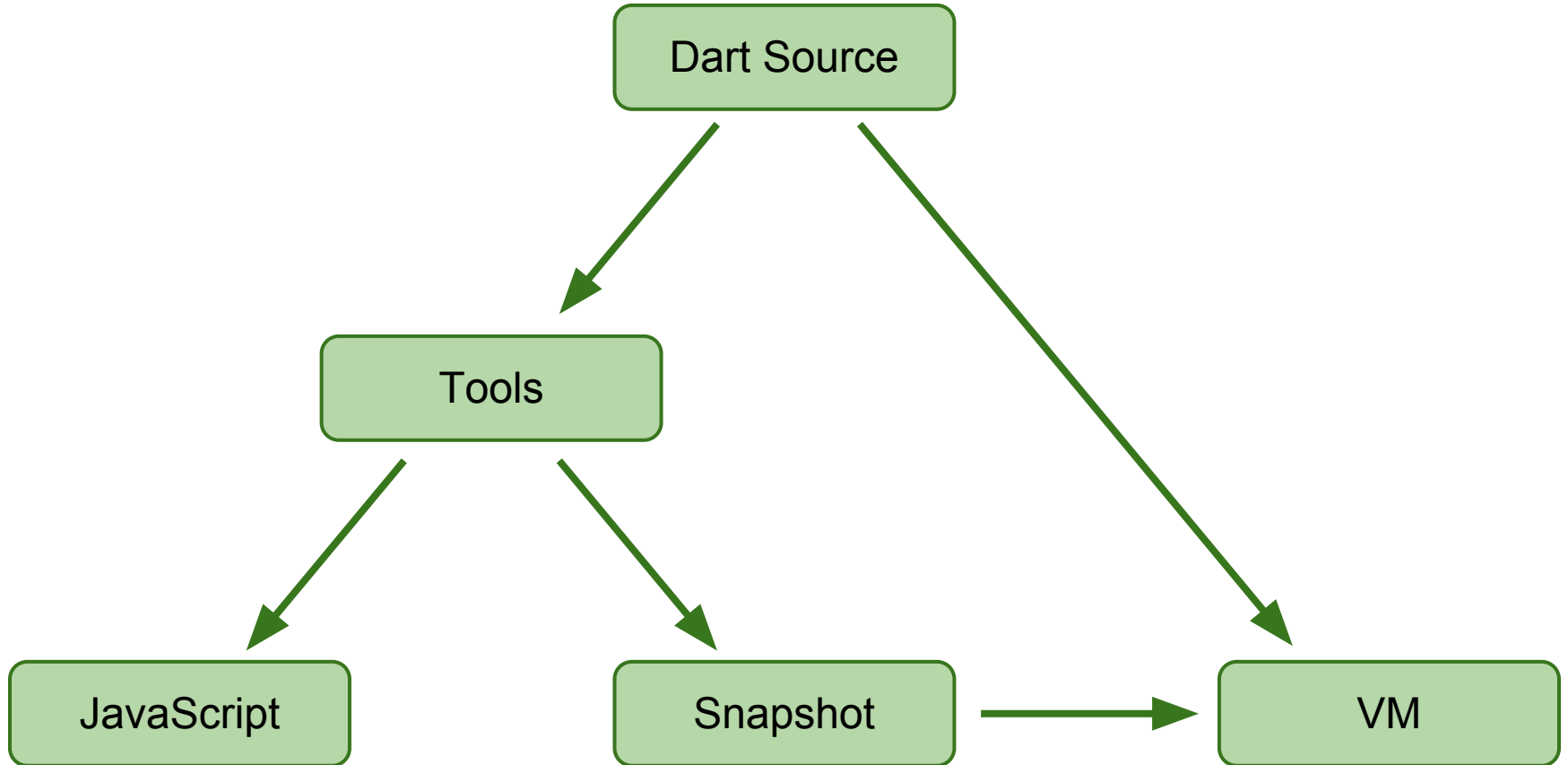
Fibonacci

 Checked Mode

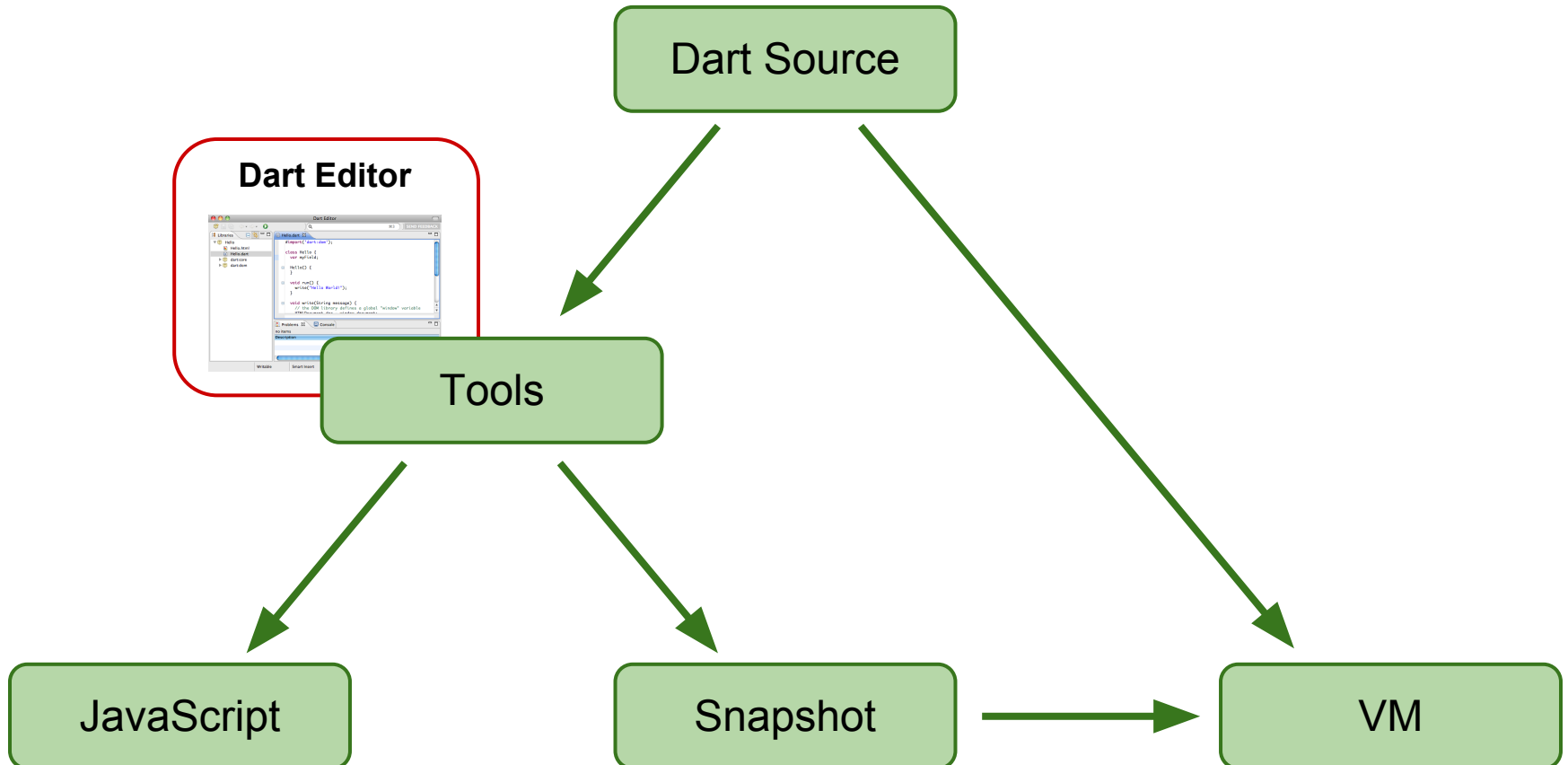
```
1 int fib(int n) {
2   if (n <= 1) return n;
3   return fib(n - 1) + fib(n - 2);
4 }
5
6 main() {
7   print('fib(20) = ${fib(20)}');
8 }
```



Tools



Tools





Dart Editor Goals

- Easy to understand
- Introduce programmers to Dart
- Increase productivity
 - Code completion, etc
- Fast
- Open Source and pre-built binary
 - Available at <http://dartlang.org>



Dart Editor Users

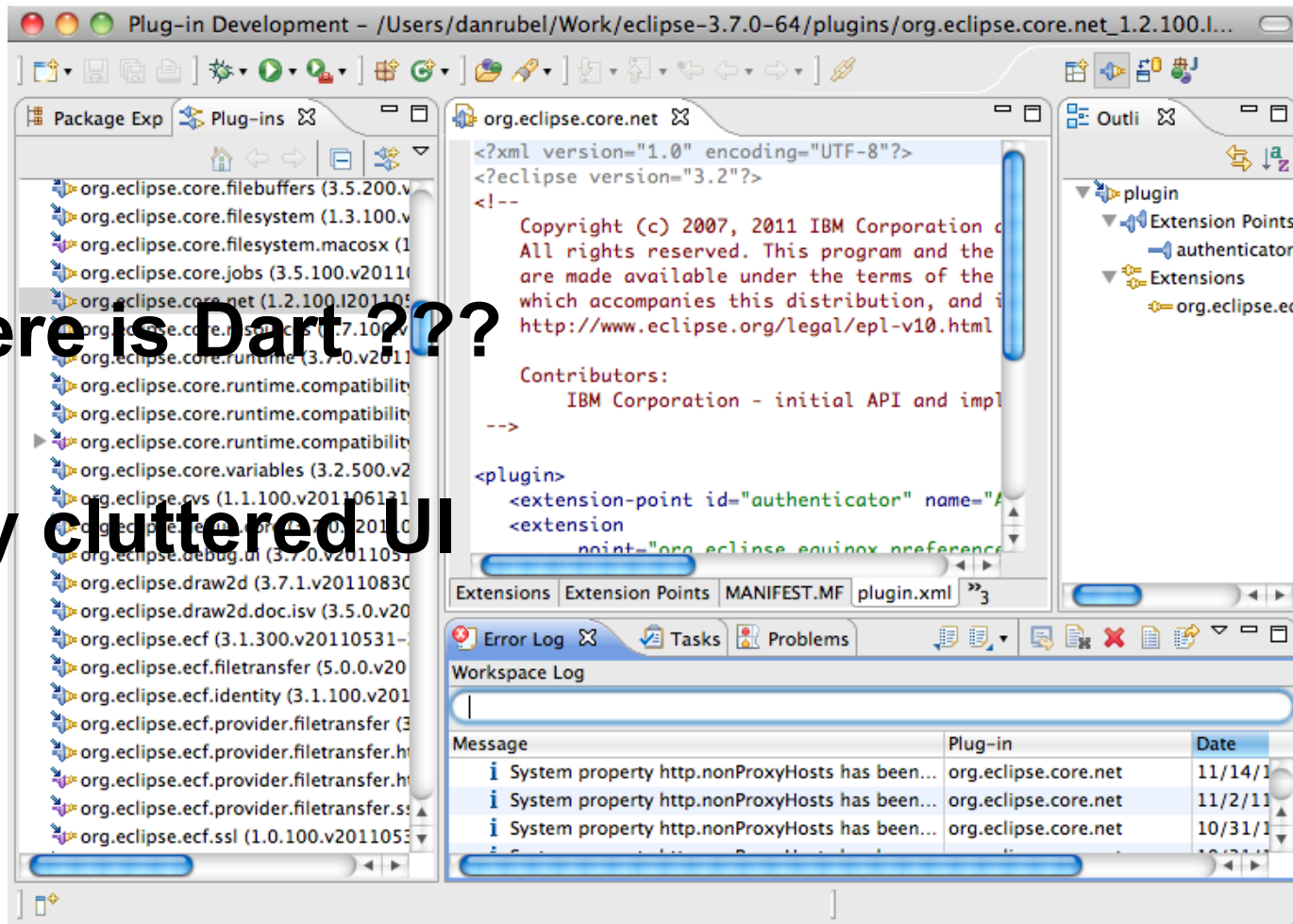
- Web programmers of varying backgrounds
 - Many languages - HTML, JS, Python, Java
 - Wide range of programming experience
- Primarily not Eclipse users



Dart Editor ... Before

Where is Dart ???

Very cluttered UI





Dart Editor Strategy

Narrow the scope

Focus on doing a few things well

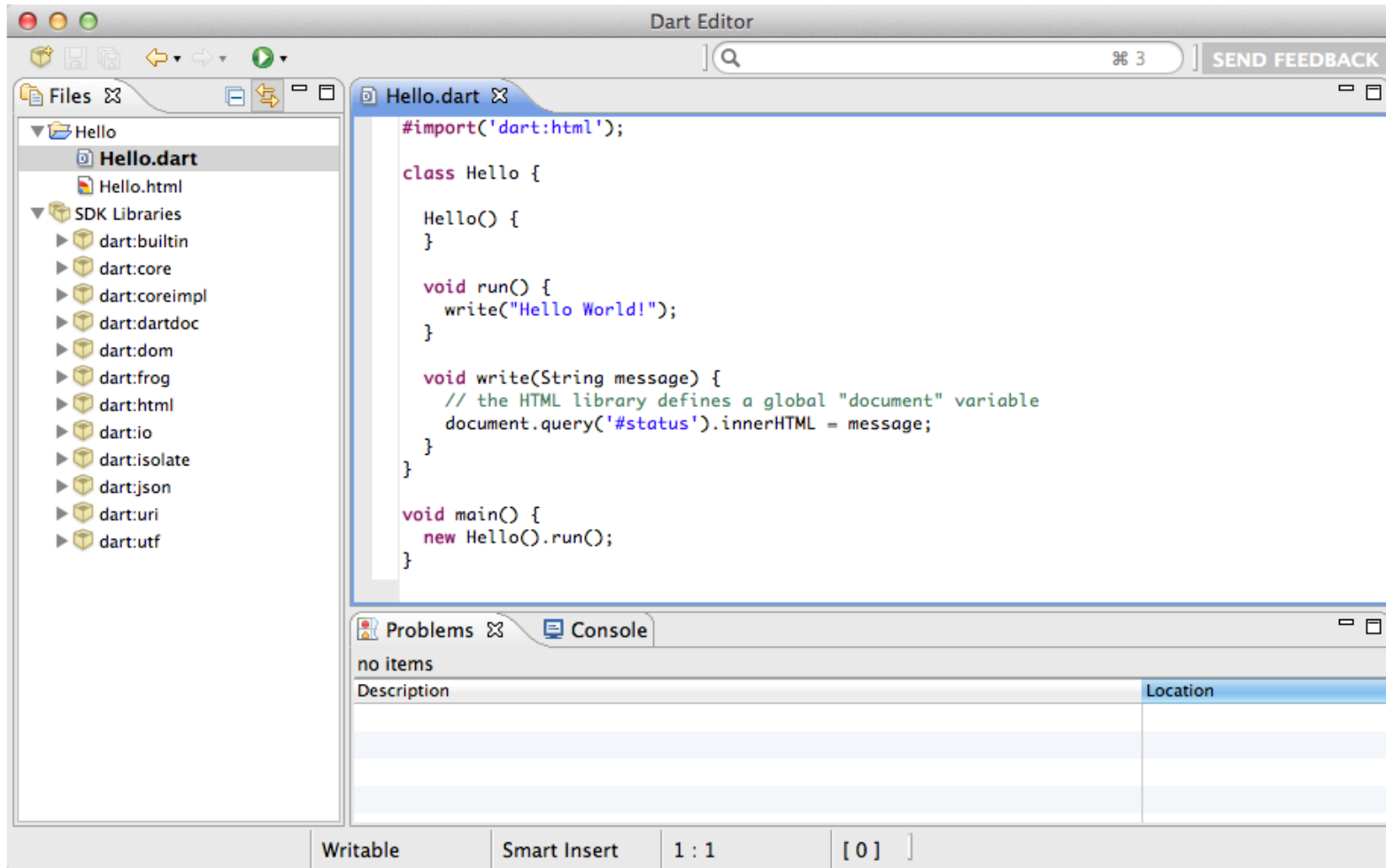
Minimalist UI

Make it easy to understand

Reduce decision making



Dart Editor ... Now



Simple and Clean UI



Dart Editor Strategy

- 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



Critical Performance Areas

- Background analysis (errors / warnings)
- Background indexing
- Code completion
- Dart to JavaScript compiler



Metrics

First RCP build

65 MB

170 plugins

20s startup

Current build

37 MB

69 plugins

4s startup



Dart Is Not Done

- Reflection?
- Rest arguments?
- enum?
- Pattern matching?
- More browser integration?



Getting Involved

<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