

# Dart

## An Introduction



Florian Loitsch  
<floitsch@google.com>

October 19, 2011



- 1 Introduction
- 2 How does it look and feel?
- 3 What is this thing about isolates?
- 4 Execution model
- 5 Open-source project
- 6 Conclusion

# Outline

- 1 Introduction
- 2 How does it look and feel?
- 3 What is this thing about isolates?
- 4 Execution model
- 5 Open-source project
- 6 Conclusion

# A Structured Web Programming Language

Dart, a structured web programming language:

- New programming language.
- New programming tools.
- New open source project.

Warning: this is a technology preview.



# Current Web

## Current Web - the good parts:

- JavaScript is flexible and supports incremental development.
- Developing and deploying small applications is easy:
  - Platform independent execution - code runs everywhere.
  - No installation of applications.



# Current Web cont.

## Current Web - The Bad Parts:

- Writing large well-performing applications is hard:
  - Hard to reason about the program structure.
  - Startup performance is often really bad.
- Difficult to document intent and interfaces (lack of types).
- No support for modules, packages, or libraries.
- Comparatively weak tool support.



# Innovation is Essential

Innovation is Essential.

- The Web platform must keep moving fast and innovate.
- There is a clear need for supporting programming in the large.
- JavaScript is a powerful tool but it has many pitfalls.



# Design Goals

## Design Goals for Dart:

- Structured yet flexible programming language for the web.
- Familiar and natural to programmers.
- Appropriate for the full range of devices on the web.
- Tools to support all major modern browsers.





# Outline

- 1 Introduction
- 2 How does it look and feel?**
- 3 What is this thing about isolates?
- 4 Execution model
- 5 Open-source project
- 6 Conclusion

# So What Is Dart?

- Unsurprisingly object-oriented programming language.
- Class-based single inheritance with interfaces.
- Optional static types.
- Proper lexical scoping.
- Single-threaded.
- Familiar syntax.



# First code

## Let's Try Some Dart Code

- Fun with classes, closures, and optional types.
- Easy to experiment with at [try.dartlang.org](http://try.dartlang.org).



# Types

- A Different Type-Checker
- A conventional type-checker
  - ... tries to prove that your program obeys the type system.
  - ... considers it a fatal error if it cannot construct a proof.
- In Dart, you are **innocent until proven guilty**.



# Optional

## Optional static types.

- Static types convey the intent of the programmer:
  - Checkable documentation for code and interfaces.
  - Avoids awkward variable naming schemes.
- Type annotations have no effect on the runtime semantics.



# Types on the Dartboard

- Let's explore a few illustrative examples.



# Outline

- 1 Introduction
- 2 How does it look and feel?
- 3 What is this thing about isolates?**
- 4 Execution model
- 5 Open-source project
- 6 Conclusion

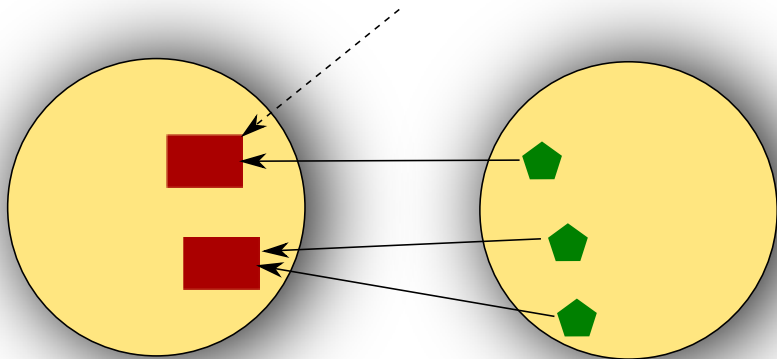
# Isolates

- Isolates are lightweight units of execution:
  - Run in their own address space like processes.
  - Nothing is shared - nothing needs synchronization.
  - All communication takes place via message passing.
- Supports concurrent execution.





# Sending and Receiving Messages



# Ports

- Receive ports accept and enqueue incoming messages.
  - Live inside a specific isolate.
  - Can be created on demand.
- A send port allows sending to a certain receive port.
  - It is an unforgeable, transferable capability.



# Isolates: Example

- Dartboard supports isolates.
- Easy to illustrate the core primitives.

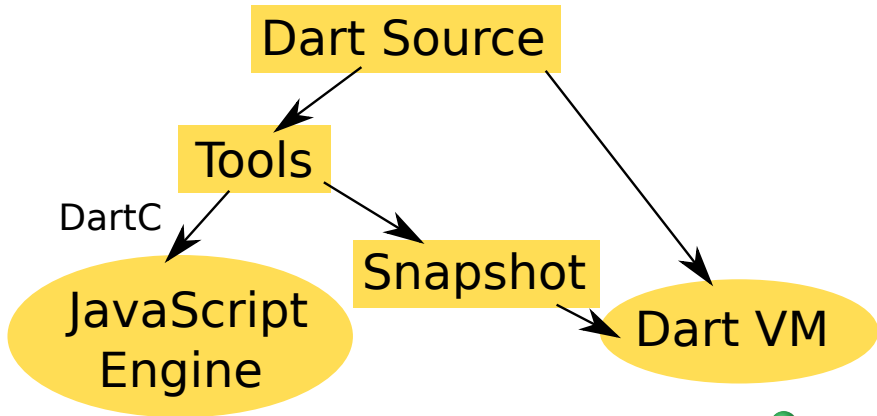
Let's see some code.



# Outline

- 1 Introduction
- 2 How does it look and feel?
- 3 What is this thing about isolates?
- 4 Execution model**
- 5 Open-source project
- 6 Conclusion

# Dart Execution



# Dart Performance

Relative performance compared to JavaScript on V8.

<b>Benchmark</b>	<b>VM</b>	<b>DartC</b>
Mandelbrot	18.2%	88.7%
DeltaBlue	56.6%	52.2%
Richards	56.0%	70.9%
NBody	35.8%	63.6%
BinaryTrees	77.3%	104.3%
Fannkuch	53.8%	22.3%
Meteor	50.3%	42.1%

Details:

- V8 revision 3.5.5.
- DartC used with the `--optimize` flag.



# Dart VM: Snapshotting

Dart VM and snapshotting.

- Process of serializing the heap after loading the application.
- Loading 54173 lines of Dart code takes 640ms.
- Loading same application from a snapshot takes 60ms.
- Startup > 10x faster.



# Web Application in Dart

Sample web application in Dart:

- News reader completely written in Dart.
- Application code: 3210 LOC.
- UP library code: 13200 LOC.
- Animation yields 30 fps.
- Code is part of the open source project.





- Editor for constructing and browsing Dart applications.
- Lightweight editor based on Eclipse components.
- Code is part of the open source project.



# Not Done Yet

Dart is not done:

- Rest arguments and enums?
- Reflection support?
- Pattern matching for easy message decoding like in Erlang?
- ... please give feedback by joining the discussions.



# Outline

- 1 Introduction
- 2 How does it look and feel?
- 3 What is this thing about isolates?
- 4 Execution model
- 5 Open-source project**
- 6 Conclusion

# Open-Source Project

- The Dart web site: <http://dartlang.org>
  - Dart language specification.
  - Dart language tutorial.
- The Dart project <http://dart.googlecode.com>
  - Libraries and code samples.
  - Dart virtual machine.
  - Dart-to-JavaScript compiler



# Outline

- 1 Introduction
- 2 How does it look and feel?
- 3 What is this thing about isolates?
- 4 Execution model
- 5 Open-source project
- 6 Conclusion**

# Dart is a Technology Preview

- Dart: structured programming for the web.
  - Two execution modes: Dart VM or JavaScript engine.
  - Compatible with the current web.
- Please try it out and participate.



Q & A.

