

Dart



Dec 2011

Dart: Structured Web Programming

- New language
- New tools
- New libraries

- Open source
 - Announced in early October 2011
- Being developed on dart.googlecode.com

- **Currently a TECHNOLOGY PREVIEW**



Agenda

- Motivation
- Language
- Isolates
- DOM
- Code samples
- Demos
- Timeline



Current Web: the Good Parts

- Small-to-medium apps easy to develop
- Platform independent
- No application installation
- Supports incremental development
- Platform improving fast
- Everywhere
 - And getting more modern:
 - ~50% users on IE9/FF7/Chrome/Safari



Current Web: the Bad Parts

- 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



Innovation Is Essential

- We believe Dart fills a vacuum
- The competition is NOT JavaScript
- ... but fragmented mobile platforms



What Is Dart?

- A simple, unsurprising OO language
- Class based, single inheritance with interfaces
- Optional static typing
- Real lexical scoping
- Single threaded
- Familiar syntax



A Different Type-Checker

- Conventional type-checker is a lobotomized theorem prover.
 - Tries to prove program obeys type system
 - If checker can't construct proof, it considers program invalid:
 - "GUILTY UNTIL PROVEN INNOCENT"
 - Dart is different:
 - "INNOCENT UNTIL PROVEN GUILTY"



Optional Static Types

- Static checker provides warnings
 - Tuned to be unobtrusive
 - Type annotations have no effect except ...
 - During development, you can check dynamic types against declarations



Dart Types at Runtime

- Developers may check types at runtime

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

- By default, type annotations have:
 - No effect
 - No cost



Isolates

- Inspired by Erlang, Dart has isolates
- Lightweight units of execution
 - Each isolate conceptually a process
 - Nothing shared
 - All communication via message passing
- Isolates support concurrent execution



Isolates

- In the browser
 - *Lightweight isolates* on UI thread
 - *Heavyweight isolates* map to their own OS thread
 - When compiled to JS, web workers
- On the server
 - Imagine `node.dart` using isolates for concurrency



Isolates

- Many potential uses for isolates
 - Isolation of 3rd-party code
 - Security
 - JavaScript interop
 - Uniform model for:
 - Client-server
 - Intra-client

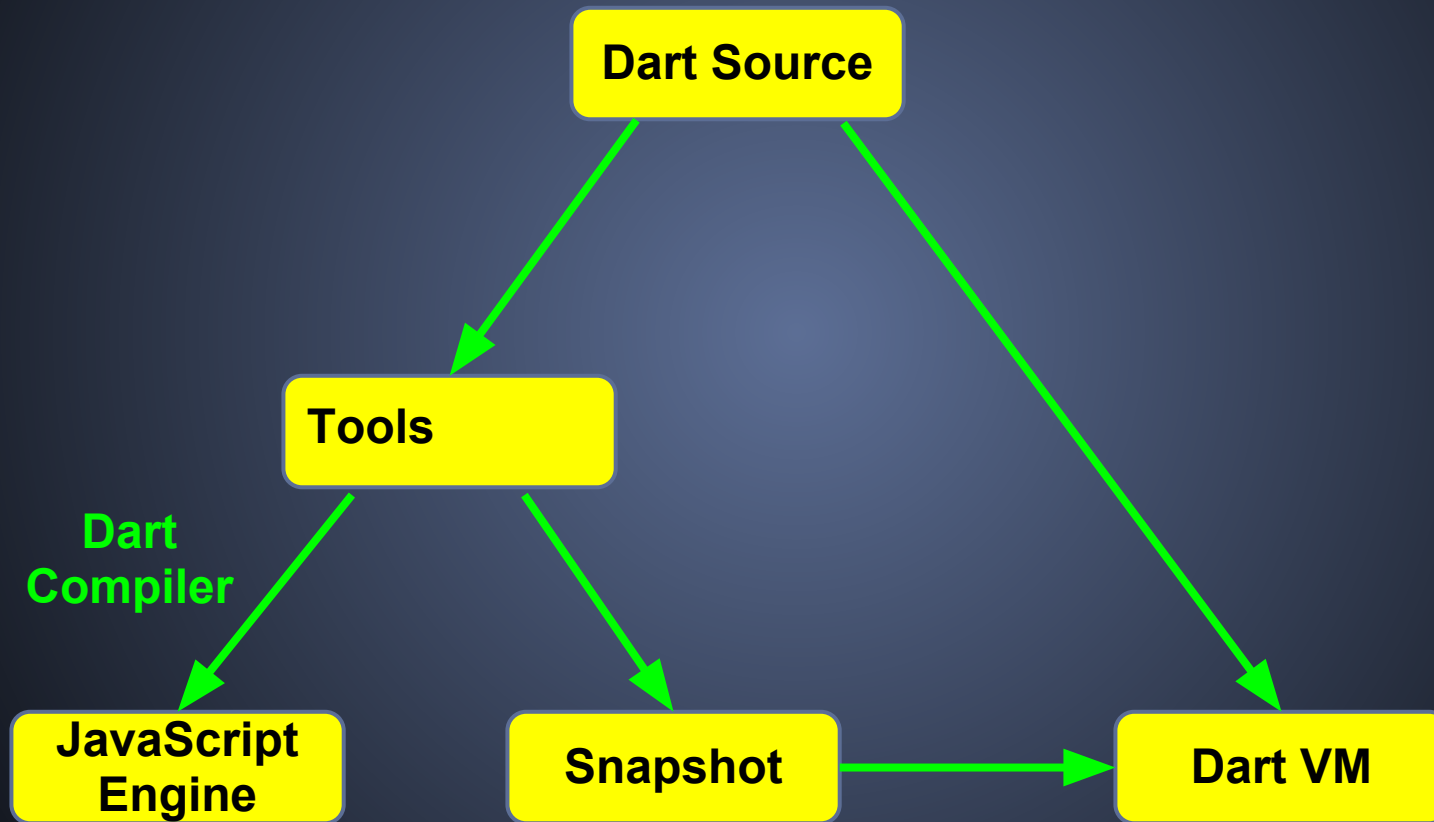


DOM Overhaul

- Dart DOM loosely based on HTML5 DOM
- Lots of cleanup done
 - Cruft gone - XML legacy
- DOM much more Dart-like
 - Support type-checking
 - Structure - lists are Dart lists
 - Terse



Dart Execution



Dart Performance

Performance Relative to JavaScript on V8

Benchmark	VM	Dart->JS Compiler
Mandelbrot	18.1%	101%
DeltaBlue	60.5%	85%
Richards	49.9%	79.9%
NBody	37.5%	83.2%
BinaryTrees	70.3%	99.9%
Fannkuch	58.4%	78.9%
Meteor	48.2%	99.4%

Details: dart bleeding_edge frog compiler and VM measured at r1331, V8 version 3.4.14.35



Snapshotting in the Dart VM

- Snapshot
 - Heap serialized after application loaded
- Performance:
 - Un-snapshotted, 54000 lines of dart -> 640ms
 - Same application w/ snapshot -> 60ms
 - 10x faster startup
- Requires VM



Sample web application

- [Newsreader](#) completely written in Dart
- Size:
 - App - 3210 lines
 - UI Library - 13200 lines
 - Animation: 30 fps
 - Compile time: 2.1s
 - Resulting JS: 539KB
- Available in open-source project



Dart Editor

- Editor for constructing and browsing Dart apps
- Lightweight, based on Eclipse components
- Available in open-source (and as a [prebuilt binary](#))



Dart Is Not Done

- Reflection support?
- Rest arguments?
- enum?
- Pattern matching?
- How to integrate with browsers?



Dart Open Source Project

- Dart Web Site: www.dartlang.org
 - Dart language spec
 - Dart tutorials
 - Some prebuilt binaries
- Dart Project: dart.googlecode.com
 - Library and code samples
 - Dart virtual machine
 - Dart core libraries
 - Dart -> JavaScript compiler



Dart is a Technology Preview

- A programming language for the web
 - Two execution modes - compiled and on VM
 - Compatible with today's web
- Please try it and give us feedback!



Fun code examples

- Generics
- Classes have interfaces
- Simple point
- Minimal
- Twin prime finding with streams
- Improving the DOM API
- Lexical scoping

Dart to JavaScript

<https://gist.github.com/1385015>