

Dart

programming language

A4, gornja i donja margina 2cm, leva i desna 2.5cm

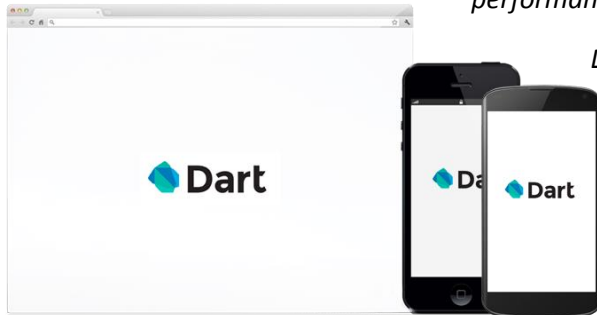
Veličina slova: tekst 11pt, naslovi 16pt i 13pt

Table of Contents

- History 3
 - Standardization 3
- Usage 3
 - Compiled as JavaScript 3
 - In the Dartium Browser 4
 - Stand-Alone 4
- Runtime modes 4
- Compiling to JavaScript 4
- Editors..... 4
 - Chrome Dev Editor 5

Dart language report

Dart is an open-source Web programming language developed by Google. It was unveiled at the GOTO conference in Aarhus, October 10–12, 2011. In order to run in mainstream browsers, Dart relies on a source-to-source compiler to JavaScript. Dart had a mixed reception and the Dart initiative has been criticized by industry leaders for fragmenting the web. According to the project site, Dart was "designed to be easy to write development tools for, well-suited to modern app development, and capable of high-performance implementations."



Dart is a class-based, single inheritance, object-oriented language with C-style syntax. It supports interfaces, abstract classes, reified generics, and optional typing. Static type annotations do not affect the runtime semantics of the code. Instead, the type annotations can provide documentation for tools like static checkers and dynamic run time checks.

General info

1. Paradigm	Object-oriented and Class-based
2. Designed by	Lars Bak and Kasper Lund
3. Developer	Google
4. Appeared in	November 14, 2013
5. Stable release	1.9.1
6. Typing discipline	Optional
7. License	BSD License
8. Filename extensions	.dart
9. Website	www.dartlang.org

History

The project was founded by Lars Bak and Kasper Lund.

Standardization

ECMA has formed technical committee TC52 to work on standardization of Dart, and, inasmuch as Dart can be compiled to standard JavaScript, it works effectively in any modern browser. ECMA approved the first edition of Dart language specification at its 107th General Assembly on July 2014.

Usage

There are three primary ways to run Dart code:

Compiled as JavaScript

When running Dart code in a web browser, the primary intended mechanism is to pre-compile the Dart code into JavaScript using the dart2js compiler. Compiled JavaScript, Dart code is compatible with all major browsers with no specific browser adoption of Dart being required. Through optimization of the compiled JavaScript output to avoid expensive checks and operations, code written in Dart can, in some cases, run faster than equivalent code hand-written using JavaScript idioms.

In the Dartium Browser

The Dart SDK ships with a version of the Chromium web browser modified to include a Dart virtual machine (VM). This browser can run Dart code directly without compilation to JavaScript. It is intended as a development tool for Dart applications, rather than as a general purpose web browser.

Stand-Alone

The Dart SDK also ships with a stand-alone Dart VM, allowing dart code to run in a command-line environment. As the language tools included in the Dart SDK are written primarily in Dart, the stand-alone Dart VM is a critical part of the SDK. These tools include not only the dart2js compiler, but also a package management suite called pub. Dart ships with a complete standard library allowing users to write fully functional system apps, such as custom web servers.

Runtime modes

Dart programs run in one of two modes. In "checked mode", which is not the default mode and must be turned on, dynamic type assertions are enabled. These type assertions can turn on if static types are provided in the code, and can catch some errors when types do not match. For example, if a method is annotated to return a String, but instead returns an integer, the dynamic type assertion will catch this and throw an exception. Running in "checked mode" is recommended for development and testing.

Dart programs run by default in "production mode", which runs with all dynamic type assertions turned off. This is the default mode because it is the fastest way to run a Dart program.

Compiling to JavaScript

dart2js is the current Dart-to-JavaScript compiler from Google, as of March 2009, and is written in Dart. dart2js is intended to implement the full Dart language specification and semantics. It is an evolution of earlier compilers: dartc was the first compiler to generate JavaScript from Dart code but has since been deprecated. Frog was the second Dart-to-JavaScript compiler and was written in Dart. Frog never implemented the full semantics of the language, leading to the development of the dart2js compiler.

On March 28, 2013, the Dart team posted an update on their blog addressing Dart code compiled to JavaScript with the dart2js compiler, stating that it now runs faster than handwritten JavaScript on Chrome's V8 JavaScript engine for the DeltaBlue benchmark.

Editors

On November 18, 2011, Google released Dart Editor, an open-source Dart editor based on Eclipse components, for Mac OS X, Windows, and Linux-based operating systems. The editor supports syntax highlighting, code completion, JavaScript compilation, running both web and server Dart applications, and debugging.

On August 13, 2012, Google announced the release of an Eclipse plugin for doing Dart development.

JetBrains IDEs also support the Dart language. Dart plugin[16] is available for IntelliJ IDEA, PhpStorm and WebStorm. This plugin supports many features such as syntax highlighting, code completion, refactoring, debugging, and more.

Chrome Dev Editor

It has been known since November 2013 that the Chromium team is working on an open source, Chrome App-based development environment with a reusable library of GUI widgets, codenamed Spark, later renamed as Chrome Dev Editor. It is built in Dart, and contains a GUI widgets library powered by Polymer. Developer preview version is available in Chrome Web Store.

$$\left[\frac{\left(x^n - 1 - \frac{7-x}{3+x} \right) * \frac{4}{x^{n+2} + 3x^2}}{\frac{6x^{2n} - 24}{x^{2n+3} + 6x^{n+3} + \sqrt[3]{9x}} * \sqrt{\frac{2x}{3x+1}}} \right] * \left(\frac{x^{n+2}}{(x+2)^2} \right)$$