

## 1 Scope

This Ecma standard specifies the syntax and semantics of the Dart programming language. It does not specify the APIs of the Dart libraries except where those library elements are essential to the correct functioning of the language itself (e.g., the existence of class `Object` with methods such as `noSuchMethod`, `runtimeType`).

## 2 Conformance

A conforming implementation of the Dart programming language must provide and support all the APIs (libraries, types, functions, getters, setters, whether top-level, static, instance or local) mandated in this specification.

A conforming implementation is permitted to provide additional APIs, but not additional syntax, except for experimental features in support of null-aware cascades that are likely to be introduced in the next revision of this specification.

## 3 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

1. The Unicode Standard, Version 5.0, as amended by Unicode 5.1.0, or successor.
2. Dart API Reference, <https://api.dartlang.org/>

## 4 Terms and Definitions

Terms and definitions used in this specification are given in the body of the specification proper. Such terms are highlighted in italics when they are introduced, e.g., ‘we use the term *verbosity* to refer to the property of excess verbiage’, and add a marker in the margin. ◇

## 5 Notation

We distinguish between normative and non-normative text. Normative text defines the rules of Dart. It is given in this font. At this time, non-normative text includes:

Rationale Discussion of the motivation for language design decisions appears in italics. *Distinguishing normative from non-normative helps clarify what part of the text is binding and what part is merely expository.*