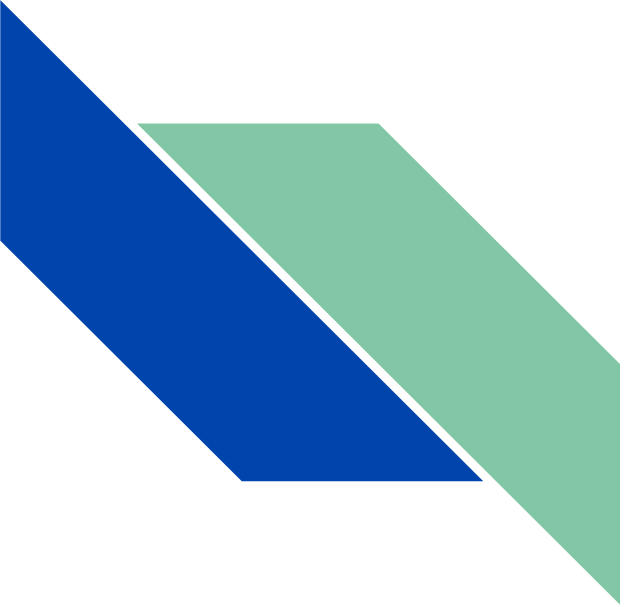

Evolução da linguagem Dart

Luca Kurotaki

Versionamento de Dart

- Um único Dart SDK (kit de desenvolvimento de software para Dart) pode suportar várias versões da linguagem Dart;
- O compilador identifica a versão adequada para o código, e interpreta o código de acordo com a versão determinada;
- O versionamento é importante para casos em que o Dart introduz funcionalidades incompatíveis em uma nova versão (ex: null safety).

Numeração de versionamento de Dart

- As versões da linguagem Dart são identificadas por um número principal (major), seguido de indicador menor (minor);
- Os dois primeiros números de uma versão do Dart SDK fazem referência à versão da linguagem Dart. (Ex: a versão de Dart mais recente que o Dart SDK 2.16.1 suporta é Dart 2.16);
- Uma versão de Dart SDK suporta todas as versões anteriores da linguagem Dart que possuam o mesmo número principal. (Ex: Dart SDK 2.16.1 suporta Dart 2.15, 2.14, 2.13..., 2.1, e 2.0);

Numeração de versionamento de Dart

- Sempre que é lançada uma versão de Dart SDK com um novo identificador, surge, então, uma nova versão da linguagem Dart;
- Na prática, a maioria das versões da linguagem Dart são muito parecidas e compatíveis com suas versões anteriores. (Ex: a versão 2.16 é essencialmente idêntica à versão 2.15);
- Uma versão remendada (patch) de Dart SDK não pode introduzir novas funcionalidades.

Seleção da versão de linguagem para a biblioteca

(Per-library language version selection)

- Por padrão, todos os arquivos Dart de um mesmo pacote usa a mesma versão da linguagem;
- Caso necessário empregar uma versão mais antiga, pode-se utilizar a expressão “// @dart = <major>.<minor>”.

  Ex: //@dart = 1.14

Seleção da versão de linguagem para a biblioteca

(Per-library language version selection)

- Nesses casos, a expressão deve ser escrita utilizando //, e não outras formas de comentários como /// e /*;
- Caso queira escrever outros comentários, deve ser feita antes de empregar a expressão “@dart...”.

As diferenças de cada versão da linguagem Dart

- Dart 2.0
  - Novo sistema de tipos;
  - Nas versões anteriores, muitos erros envolvendo tipagem eram identificados somente no momento da execução.
- Dart 2.1
  - Conversão de int para double;
  - Ex: double num = 10.0 -> double num = 10

As diferenças de cada versão da linguagem Dart

- Dart 2.2
  - Introdução do tipo Set;
  - Ex: var dias = {‘segunda’, terça’, ‘quarta’} -> const Set<String> dias = {‘segunda’, ‘terça’, ‘quarta’}
- Dart 2.3
  - Adição de três operadores;
  - spread operator (...);
  - var lista = [1, 2, 3]; var lista2 = [0, ...lista] //0,1,2,3

As diferenças de cada versão da linguagem Dart

- Dart 2.3
  - collection if
  - Ex: var colecao = [‘a’, ‘b’, ‘c’, if(condicao) ‘z’];
  - collection for
  - var lista = [1,2,3]; var listaFor = [0, for(var i in lista) i];
- Dart 2.5
  - Adição da biblioteca dart:ffi

As diferenças de cada versão da linguagem Dart

- Dart 2.6
  - Adição da ferramenta dart2native;
  - A funcionalidade foi incluída no comando dart compile.
- Dart 2.7
  - Adição de extension methods;
  - extension, desde então, passa a estender funcionalidades para qualquer tipo.

As diferenças de cada versão da linguagem Dart

- Dart 2.8
  - Não adicionou novas funcionalidades;
  - Implementou mudanças que adiantaram a usabilidade e performance do null safety, que viria a ser lançado em versão mais adiante;
  - Melhorias em pub tool e pub outdated.
- Dart 2.9
  - Não adicionou novas funcionalidades;

As diferenças de cada versão da linguagem Dart

- Dart 2.10
  - Não adicionou novas funcionalidades;
  - Melhorias em dart tool.
- Dart 2.12
  - Introdução de null safety;
  - O tratamento dos valores null ficam facilitados e mais seguros.
  - Ex: int? valorNulo = null;

As diferenças de cada versão da linguagem Dart

- Dart 2.13
  - Extensão de type aliases (typedef);
  - Antes, só funcionava para function;
  - Passou a englobar todos os tipos;
  - Ex: typedef IntList = List<int>; IntList lista = [1,2,3];
  - Também implementou melhorias em dart:ffi.

As diferenças de cada versão da linguagem Dart

- Dart 2.14
  - Adicionou o operador triple-shift (>>>);
  - Ex: assert((-value >> 4) == -0x03);
  - Ex: assert((-value >>> 4) > 0);
  - Também removeu algumas restrições nos argumentos de tipo (type).

As diferenças de cada versão da linguagem Dart

- Dart 2.15
  - Melhorias nos ponteiros de função chamados tear-offs;
- Dart 2.16
  - Não incluiu novas funcionalidades.

Programação para Dispositivos Móveis