Contents

[Urls 2](#_Toc526620437)

[Environment 2](#_Toc526620438)

[Dart language 2](#_Toc526620439)

[Strings 3](#_Toc526620440)

[Boolean 3](#_Toc526620441)

[Lists 3](#_Toc526620442)

[Map 3](#_Toc526620443)

[Symbol 3](#_Toc526620444)

[Enumerators 3](#_Toc526620445)

[Functions 4](#_Toc526620446)

[Class 4](#_Toc526620447)

[Collections 6](#_Toc526620448)

[Exceptions 6](#_Toc526620449)

[Typedef 6](#_Toc526620450)

[Library 7](#_Toc526620451)

[Concurrency 7](#_Toc526620452)

[Html 7](#_Toc526620453)

# Urls

//tutorial I followed

<https://www.tutorialspoint.com/dart_programming/>

// Better tutorial by manufacturer of dart

https://www.dartlang.org/guides/language/language-tour

start:05-10-2018

end: 06-10-2018

# Environment

* Online dart using dartPad at <https://dartpad.dartlang.org/>
* Download setup and run setup and set path to sdk/bin
* On command line ‘dart’ works

You can download Intellij community edition. Add dart plugin and new dart project can be created. I found Intellij best for dart.

# Dart language

* Each statement is ended by ; .
* Dart is optionally typed language. If type of a variable is not given it is treated as dynamic type. Dynamic or var keyword can also be used to infer dynamic type.
* Everything in dart is an object including number, string and all data types. Hence variables in dart only keep reference to value rather than keeping value itself.
* Dart support numbers, string, Boolean, lists and maps data types. Int, double are data types.
* There are final and const variable types whose values cannot be changed.
* Dart uses operators and expressions like javascript
* It uses for…in, while, if then else etc. as in other languages
* parse(‘1’) gives output as 1
* Since numbers are object there are properties and methods for num:

Properties: isnan, isFinite, isNegative, sign etc.

Methods: abs, ceiling, floor etc.

* Package manager in dart is pub like npm in JavaScript. The package metadata is defined in pubsec.yaml. You can do ‘pub get’, ‘pub upgrade’, ‘pub build’ (to build web app). Sample pubsec.yaml file:

name: TestApp

version: 0.0.1

description: A simple console application.

#dependencies:

# foo\_bar: '>=1.0.0 <2.0.0'

dependencies: https://mail.google.com/mail/u/0/images/cleardot.gif

xml:

* To install a package in dart, write the package in pubspec.yaml and then do pub get.
* Dart uses underscore ‘\_’ to denote private methods and properties.
* Dart is single threaded programming language. It uses Future objects to implement async. For async / future you use dart:async library.

## Strings

Has single or double quotes. Multi line strings have triple quotes.

int n=1+1;

String str1 = "The sum of 1 and 1 is ${n}";

String has also properties and methods like that of objects.

## Boolean

bool var\_name = false

## Lists

Dart represents array in form of list. It is ordered group of objects. Lists can be growable or fixed length.

Fixed length list

var list\_name = new List(initial\_size)

Growable list

var list\_name = new List()

var list\_name = [val1,val2,val3]

## Map

Map is key / value pair. This is like json object in JavaScript.

var identifier = { key1:value1, key2:value2 [,…..,key\_n:value\_n] }

var identifier = new Map()

map\_name[key] = value

## Symbol

This is a powerful concept. More like reflection. At run time you can check if a library contains a class, a class contains a property or method, or iterate over all properties or methods.

## Enumerators

enum Status {

none,

running,

stopped,

paused

}

## Functions

Lambda functions and optional parameters are allowed. Optional parameters have ? at end.

Interfaces

Interface have no separate syntax. It is same syntax as that of class. You can implement an interface using ‘implements’ clause.

## Class

class class\_name {

<fields>

<getters/setters>

<constructors>

<functions>

}

Dart can have multiple constructors or named constructor.

class Car {

Car() {

print("Non-parameterized constructor invoked");

}

Car.namedConst(String engine) {

print("The engine is : ${engine}");

}

}

Class supports this keyword. Class inheritance through extends keyword.

Method overriding of parent class is supported.

void main() {

Child c = new Child();

c.m1(12);

}

class Parent {

void m1(int a){ print("value of a ${a}");}

}

class Child extends Parent {

@override

void m1(int b) {

print("value of b ${b}");

}

}

Dart supports static methods and properties of class. Super keyword can be used in child class.

void main() {

Child c = new Child();

c.m1(12);

}

class Parent {

String msg = "message variable from the parent class";

void m1(int a){ print("value of a ${a}");}

}

class Child extends Parent {

@override

void m1(int b) {

print("value of b ${b}");

super.m1(13);

print("${super.msg}") ;

}

}

Cascade operator (..) in object

class Student {

void test\_method() {

print("This is a test method");

}

void test\_method1() {

print("This is a test method1");

}

}

void main() {

new Student()

..test\_method()

..test\_method1();

}

## Collections

Dart has List, map, Set and Queue collections. Every collection has iterator property. Collections are heterogenous. Using generics the collections can be made homogenous. This is also called type-safe. This is used same way as c#.

## Exceptions

Exception is run time problem when program aborts. Every exception in Dart is subtype of predefined type Exception.

There is try with on / catch block. On block specifies the exception name. catch block need not specify the exception name, the exception details is available at run time. Both the on and catch block can be used together with same try block.

main() {

int x = 12;

int y = 0;

int res;

try {

res = x ~/ y;

}

on IntegerDivisionByZeroException {

print('Cannot divide by zero');

}

catch(e) {

print(e);

}

}

Throw and finally are also available. You can also define your own exception using “extends Exception”

## Typedef

This is a pointer to a function. In typedef variable you can store functions and later execute the function. Define a typedef variable with signature of function which does not include the return type. Typedefs can be used as parameters to a function.

## Library

A Dart library comprises of a set of classes, constants, functions, typedefs, properties, and exceptions. Library can be imported with import keyword. Format for import is “import ‘uri’ ”. The uri has scheme. The inbuilt dart libraries have scheme dart:. So import ‘dart:math’, import ‘dart:io’ etc.

Other libraries can use file system path like ‘dir/mylib’

Libraries provided by package manager like pub uses the package; scheme like:

Import ‘package:lib1/libfile.dart’;

To import partial library parts:

import 'package: lib1/lib1.dart' show foo, bar;

// Import only foo and bar.

import 'package: mylib/mylib.dart' hide foo;

// Import all names except foo

To create custom library

library library\_name

// library contents go here

## Concurrency

Parallel operations are performed in Dart (single threaded) through isolate by using library dart:isolate.

## Html

Dart library ‘dart:html’ using you can manipulate DOM in a web application not in console one.