Triple quotes: multi-lined text

Double: decimal

num.parse(x) is used to convert string x to number

int.parse, double.parse are also valid func.

“is” used to check type.

int value=val1>val2? val1:val2;

Map<type,type> is used to store values with keys.

A **List** is an ordered collection of elements where the same element may occur several times at different positions. A **Set** is (usually) an unordered collection of unique elements

[Y=8] is default value of arg in function definition

{y} shows that this is dynamic parameter.

Lambda functions: complete function written in just one line.

void lam1()=> print('First lambda function');

int sum(a,b)=> a+b;

we can store lambda function inside a variable, but write function without return type.

Var x= ()=> print('First lambda function'); or

Void Function() a =()=> print('First lambda function')

HIGHER ORDER FUNCTION:

It is a function that needs another function to pass through it.

Where function:

Used to check for elements that satisfy a certain condition. Used for list elements.

=>firstWhere, we should pass a second element orElse: ()=>0 in case our answer is empty. So it will return 0 from orElse. If we don’t define orElse, it will throught an error.

CLASSES

Preferable to start class name with capital letter

Function can be inside class or outside class, but method is always inside class.

Everything in dart lang is an object

We cannot have multiple inheritance in dart, we can inherit multiple classes using interface or using

Class D **extends** C **with** A,B{}

OR

Interface D implements A,B,C{}

Mixin is type of abstract class. Allows mixing of types. Mixin cannot be instantiated i-e we cannot create an object for mixin.

ENUM:

enum Colors { red, blue, green }

values we define in enum are equal to static const in a class.

GENERIC:

When we define class, we can define a generic data type while will be specified when that class is instantiated.

Class A<T>{}

Var d=A<int>();