1.Basic code concepts

17 January 2024

00:09

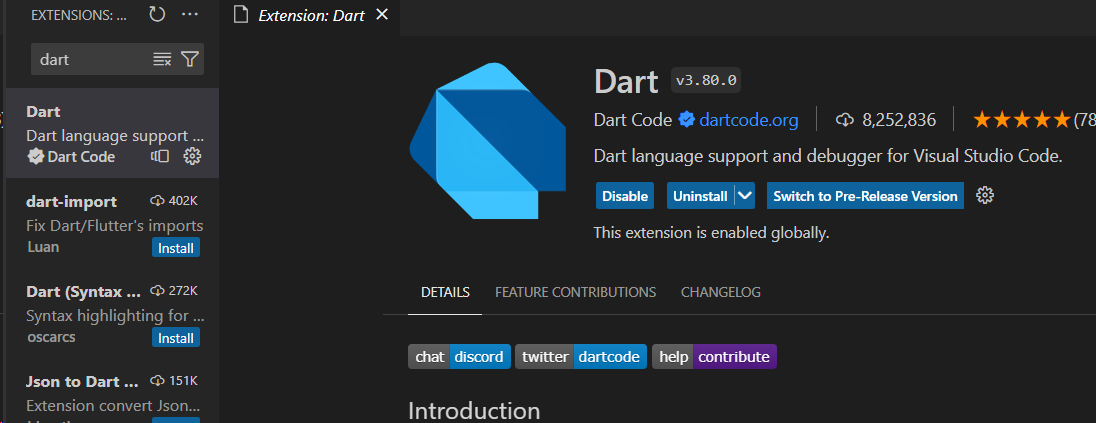
Till now:

What is dart

path till bin in environment variable

dart --version in command prompt

dart extension in vs code



Now moving on to the coding part: making a file named **dart-basics.dart**

Hello World:

(run/debug : it will always be visible on top of main)

void main(){

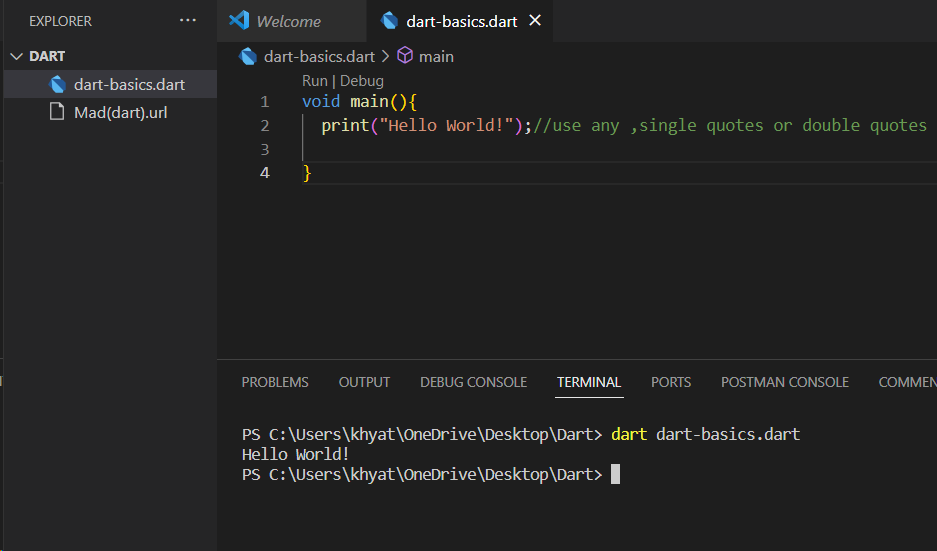
print('Hello world'); //inside single quotes or double

void main(){

print("Hello World!");//use any ,single quotes or double quotes

 }

terminal : dart dart-basics.dart



Variable Declaration and basic printing:

int x=100;

//Normal print

print(x)

//Print value with String

print("The value of the int variable = $x");

$ is interpolation : replace the var with it’s value

Dart String interpolation provides a convenient way to concatenate strings. It follows a simple

syntax where a variable name or an expression inside a literal string gets evaluated and the resultant

value is embedded within the string. This can be achieved using the dollar sign '$'

void main(){

  print("Hello World!");//use any ,single quotes or double quotes

  //variable Declaration

  int i=100;

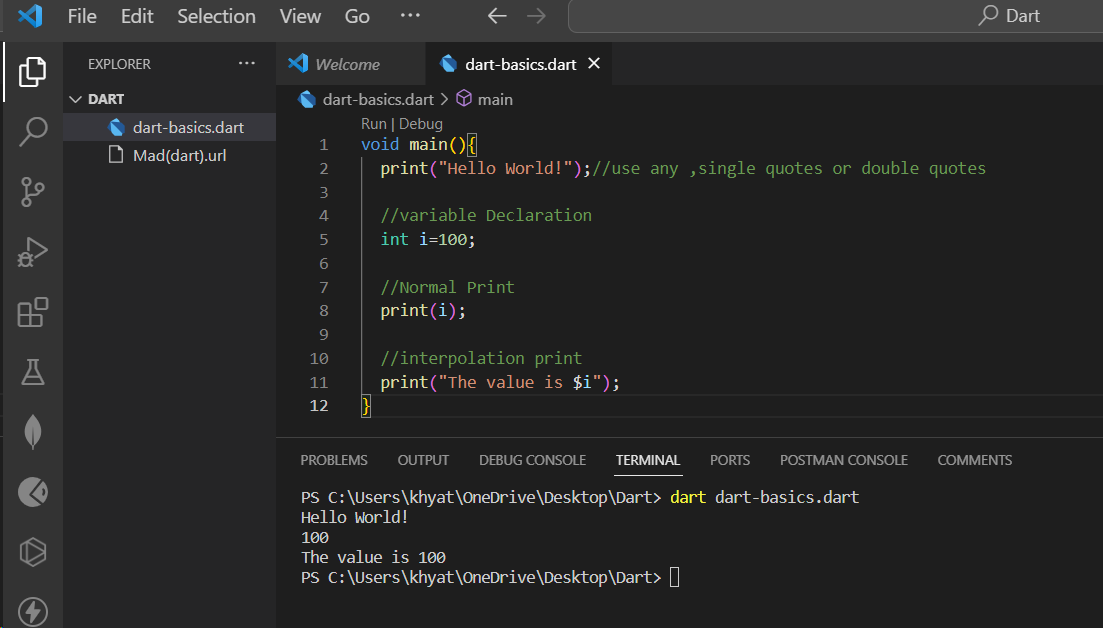
  //Normal Print

  print(i);

  //interpolation print

  print("The value is $i");

}



Print sum of two variables:

void main(){

int x= 10;

int y=20;

print("the sum of $x and $y = ${x+y}");

}

Using $ sign collectively for an expression

void main(){

  print("Hello World!");//use any ,single quotes or double quotes

  //variable Declaration

  int i=100;

  //Normal Print

  print(i);

  //interpolation print

  print("The value is $i");

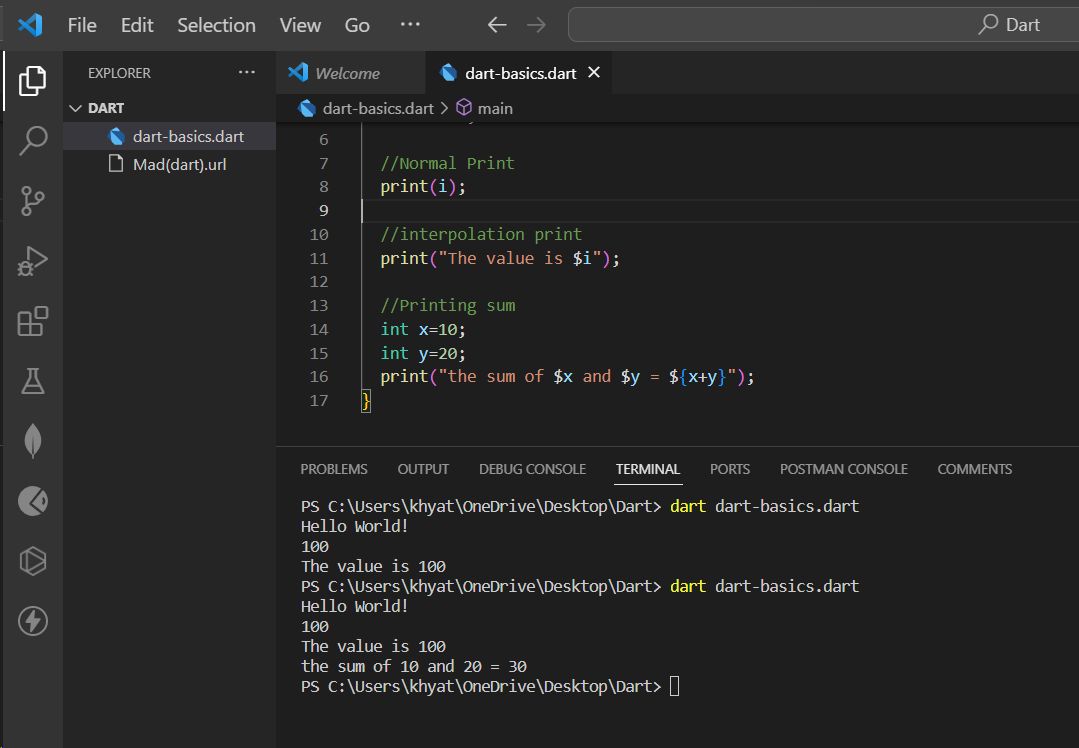
  //Printing sum

  int x=10;

  int y=20;

  print("the sum of $x and $y = ${x+y}");

}



Error Discussion:

int value ;

print(value); compile time error (highlighted while writing): must be initialized first

void main(){

  print("Hello World!");//use any ,single quotes or double quotes

  //variable Declaration

  int i=100;

  //Normal Print

  print(i);

  //interpolation print

  print("The value is $i");

  //Printing sum

  int x=10;

  int y=20;

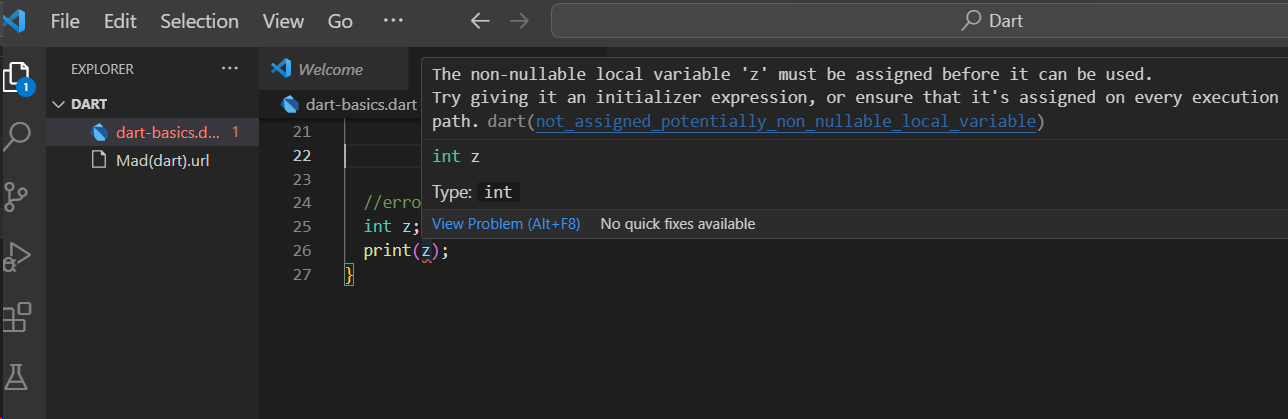
  print("the sum of $x and $y = ${x+y}");

  //error case

  int z;

  print(z);

}



How to resolve it?:

Using ? after int . ? Is a null safety feature. Null safety prevents errors that result from unintentional access of

variables set to null .

For example, if a method expects an integer but receives null , your app causes a runtime error.

This type of error, a null dereference error, can be difficult to debug.

int? z;

print(z);

void main(){

  print("Hello World!");//use any ,single quotes or double quotes

  //variable Declaration

  int i=100;

  //Normal Print

  print(i);

  //interpolation print

  print("The value is $i");

  //Printing sum

  int x=10;

  int y=20;

  print("the sum of $x and $y = ${x+y}");

  //error case

  //int z;

  int? z;

  print(z);

}

Data Types in Dart:

Dart includes data types such as : double , string , Boolean , Null

Dart also includes advanced data types : list , set , maps

Dart multiline string:

Multiline strings are created with triple single or double quotes.

var address = '''Robert Robertson,

1234 NW Bobcat Lane,

St. Robert, MO 65584-5678.''';

print(address);

void main(){

  print("Hello World!");//use any ,single quotes or double quotes

  //variable Declaration

  int i=100;

  //Normal Print

  print(i);

  //interpolation print

  print("The value is $i");

  //Printing sum

  int x=10;

  int y=20;

  print("the sum of $x and $y = ${x+y}");

  //error case

  //int z;

  int? z;

  print(z);

  //dart multiline String

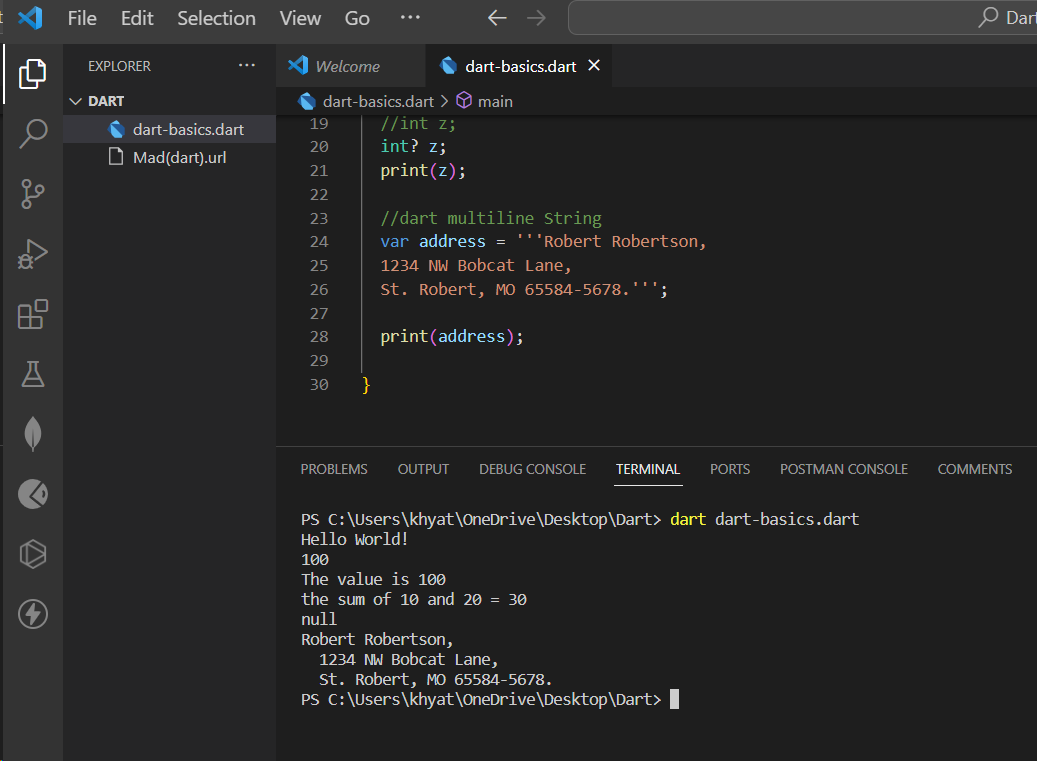
  var address = '''Robert Robertson,

  1234 NW Bobcat Lane,

  St. Robert, MO 65584-5678.''';

  print(address);

}



To know the runtime Datatype:

print(y.runtimeType);

//How to know the data type at runtime

print("The runtime datatype of $y is ${y.runtimeType}");

void main(){

  print("Hello World!");//use any ,single quotes or double quotes

  //variable Declaration

  int i=100;

  //Normal Print

  print(i);

  //interpolation print

  print("The value is $i");

  //Printing sum

  int x=10;

  int y=20;

  print("the sum of $x and $y = ${x+y}");

  //error case

  //int z;

  int? z;

  print(z);

  //dart multiline String

  var address = '''Robert Robertson,

  1234 NW Bobcat Lane,

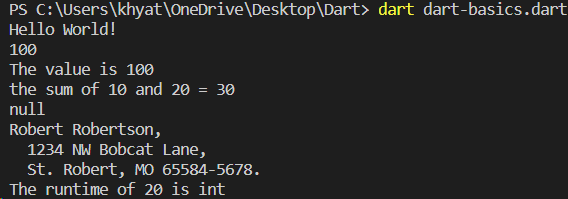
  St. Robert, MO 65584-5678.''';

  print(address);

  //How to know the data type at runtime

  print("The runtime datatype of $y is ${y.runtimeType}");

}



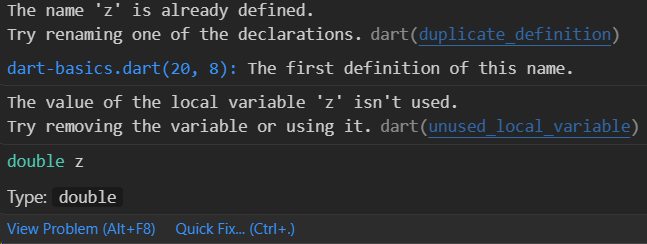
Type inferencing error:

Runtime datatype decided and we can’t change it , error at compile time

var z= 100.34;

print(y.runtimeType);

z=true;



void main() {

  print("Hello World!"); //use any ,single quotes or double quotes

  //variable Declaration

  int i = 100;

  //Normal Print

  print(i);

  //interpolation print

  print("The value is $i");

  //Printing sum

  int x = 10;

  int y = 20;

  print("the sum of $x and $y = ${x + y}");

  //error case

  //int z;

  int? z;

  print(z);

  //dart multiline String

  var address = '''Robert Robertson,

  1234 NW Bobcat Lane,

  St. Robert, MO 65584-5678.''';

  print(address);

  //How to know the data type at runtime

  print("The runtime datatype of $y is ${y.runtimeType}");

  //Type inferencing

  var z = 100.34;

  print(y.runtimeType);

  z = true;

}

Type inferencing in dynamic datatype:

In dynamic datatype, type can be reassigned without getting an error

dynamic c;

c=’John Dae’;

print(c.runtimeType);

c=100;

void main() {

  print("Hello World!"); //use any ,single quotes or double quotes

  //variable Declaration

  int i = 100;

  //Normal Print

  print(i);

  //interpolation print

  print("The value is $i");

  //Printing sum

  int x = 10;

  int y = 20;

  print("the sum of $x and $y = ${x + y}");

  //error case

  //int z;

  int? z;

  print(z);

  //dart multiline String

  var address = '''Robert Robertson,

  1234 NW Bobcat Lane,

  St. Robert, MO 65584-5678.''';

  print(address);

  //How to know the data type at runtime

  print("The runtime datatype of $y is ${y.runtimeType}");

  //Type inferencing error

  var z = 100.34;

  print(y.runtimeType);

  z = true;

  //Type inferencing in dynamic data type

  dynamic c;

  c="John Dae";

  print(c.runtimeType);

  c=100;

}

Predefined Functions for Strings in Dart:

The Dart provides an extensive range of methods. The list of a few essential methods is given below.

Methods        Descriptions

toLowerCase()

It converts all characters of the given string in lowercase.

toUpperCase()

It converts all characters of the given string in uppercase.

trim()

It eliminates all whitespace from the given string.

compareTo()

It compares one string from another.

replaceAll()

It replaces all substring that matches the specified pattern with a given string.

split()

It splits the string at matches of the specified delimiter and returns the list of the substring.

substring()

It returns the substring from start index, inclusive to end index.

toString()

It returns the string representation of the given object.

codeUnitAt()

It returns the 16-bits code unit at the given index.

Console based input in Dart:

To take input from the console you need to import a library, named dart:io from libraries of Dart. About stdin Class: This class allows the user to read data from standard input in both synchronous and asynchronous ways. The method readLineSync() is one of the methods used to take input from the user.

import "dart:io";

void main(){

//Console based i/p:

  print("Enter name");

  String? val= stdin.readLineSync();

  print("Name= $val");

}

Whole code:

import "dart:io";

void main() {

  print("Hello World!"); //use any ,single quotes or double quotes

  //variable Declaration

  int i = 100;

  //Normal Print

  print(i);

  //interpolation print

  print("The value is $i");

  //Printing sum

  int x = 10;

  int y = 20;

  print("the sum of $x and $y = ${x + y}");

  //error case

  //int z;

  int? z;

  print(z);

  //dart multiline String

  var address = '''Robert Robertson,

  1234 NW Bobcat Lane,

  St. Robert, MO 65584-5678.''';

  print(address);

  //How to know the data type at runtime

  print("The runtime datatype of $y is ${y.runtimeType}");

  //Type inferencing error

  // var z = 100.34;

  // print(y.runtimeType);

  // z = true;

  //Type inferencing in dynamic data type

  dynamic c;

  c="John Dae";

  print(c.runtimeType);

  c=100;

  //predefined function in dart

  //Console based i/p:

  print("Enter name");

  String? val= stdin.readLineSync();

  print("Name= $val");

}

Converting Datatype from String to int:

import "dart:io";

void main() {

  print("Hello World!"); //use any ,single quotes or double quotes

  //variable Declaration

  int i = 100;

  //Normal Print

  print(i);

  //interpolation print

  print("The value is $i");

  //Printing sum

  int x = 10;

  int y = 20;

  print("the sum of $x and $y = ${x + y}");

  //error case

  //int z;

  int? z;

  print(z);

  //dart multiline String

  var address = '''Robert Robertson,

  1234 NW Bobcat Lane,

  St. Robert, MO 65584-5678.''';

  print(address);

  //How to know the data type at runtime

  print("The runtime datatype of $y is ${y.runtimeType}");

  //Type inferencing error

  // var z = 100.34;

  // print(y.runtimeType);

  // z = true;

  //Type inferencing in dynamic data type

  dynamic c;

  c="John Dae";

  print(c.runtimeType);

  c=100;

  //predefined function in dart

  //Console based i/p:

  print("Enter name");

  String? val= stdin.readLineSync();

  print("Name= $val");

  //Conversion with every datatype

  print("Enter your roll no.");

  String? val2= stdin.readLineSync();//String value

  int roll =int.parse(val2.toString()); //change to int

  print("Roll number= $roll");

}