

Built in Data Types

In **Dart language**, there are the types of values that can be represented and manipulated in a programming language.

Data Types in Dart

The data type classification is as given below:

Data Type	Keyword	Description
Number	int, double, num, BigInt	Numbers in Dart are used to represent numeric literals
Strings	String	Strings represent a sequence of characters
Booleans	bool	It represents Boolean values true and false
Lists	List	It is an ordered group of objects
Maps	Map	It represents a set of values as key-value pair

1. Number

The number in Dart Programming is the data type that is used to hold the numeric value. Dart numbers can be classified as:

- **int:** data type is used to represent whole numbers (64-bit Max).
- **double:** data type is used to represent 64-bit precise floating-point numbers.
- **num:** type is an inherited data type of the int and double types.

Below is the implementation of Numbers in Dart:

```
// Number Data Type
```

```
void main() {  
  
    // declare an integer  
    int num1 = 2;  
  
    // declare a double value  
    double num2 = 1.5;  
  
    // print the values  
    print(num1);  
    print(num2);  
    var a1 = num.parse("1");  
    var b1 = num.parse("2.34");  
  
    var c1 = a1+b1;  
    print("Product = ${c1}");  
}
```

Output:

```
2  
1.5  
Product = 3.34
```

2. String

It used to represent a sequence of characters. It is a sequence of UTF-16 code units. The keyword string is used to represent string literals. String values are embedded in either single or double-quotes.

Syntax:

```
String str_name;
```

Below is the implementation of String data type in Dart:

Dart

```
// Dart Program to demonstrate

void main () {
  // Declaration of string typ
  String str = 'Coding is ';
  String str1 = 'Fun';

  print (string);
  print (str + str1);
}
```

Output:

```
Coding is Fun
```

3. Boolean

It represents Boolean values true and false. The keyword bool is used to represent a Boolean literal in DART.

Syntax:

```
bool var_name;
```

```
// Dart Program to demonstrate
// Boolean Data Type

void main() {
  String str = 'Coding is ';
  String str1 = 'Fun';

  bool val = (str==str1);
```

```
    print (val);  
}
```

Output:

```
false
```

4. List

List data type is similar to arrays in other programming languages. A list is used to represent a collection of objects. It is an ordered group of objects.

Declaration of List

There are multiple methods to declare List in Dart as mentioned below:

1. Variable Size List

```
int var_name1 = [];  
  
// Alternative for the above declaration  
List<int> var_name2;
```

2. Fixed Size List

Fixed Size doesn't mean that we can't change the size of List, but we have predefined that the List has this much elements while declaring.

```
List<int> var_name1 = new List<int>.fixed(size,0);  
  
List<int> var_name2 = new List<int>.generate(size, function/expressions);
```

Below is the implementation of List in Dart:

```
void main()  
{  
    List<String> fig = new List<String>.filled(3,"default");  
}
```

```

    gfg[0] = 'Flutter';
    gfg[1] = 'is';
    gfg[2] = 'Good';

    print(fig);
    print(fig[0]);
}

```

Output:

```

[Flutter , is , Good]
Flutter

```

Declare Map in Dart

While Declaring Map there can be only two cases one where declared Map is empty and another where declared Map contains elements in it. Both Cases are mentioned below:

1. Declaring Empty Map

```

// Method 1
Map? map_name;

// Method 2
Map<key_datatype , value_datatype>? map_name;

// Method 3
var map_name = new Map();

```

2. Declaring Map with Elements inside it.

```

// Method 1
Map x={
    key1 : value1;
    key2 : value2;
};

```

```
// Method 2
Map<key_datatype , value_datatype> map_name{
    key1 : value1;
    key2 : value2;
};

// Method 3
var map_name{
    key1 : value1;
    key2 : value2;
};
```

Below is the implementation of Map in Dart:

```
void main() {
    Map gfg = new Map();
    gfg['First'] = 'Flutter';
    gfg['Second'] = 'is';
    gfg['Third'] = 'Good';
    print(gfg);
}
```

Output:

```
{First: FLutter, Second: is, Third: Good}
```