**Returning multiple values using class and object concept**

Typically, you can create a class to encapsulate the returned values and then use an instance of that class to return and access the values. Here’s how you can do it

|  |
| --- |
| **class** MultipleValues {      final **int** value1;    final **int** value2;      MultipleValues(**this**.value1, **this**.value2);  }    **class** DifferentDataTypes {    final **int** integer;    final String stringValue;      DifferentDataTypes(**this**.integer, **this**.stringValue);  }    **void** main() {    final multipleValues = returnMultipleValues();    print(multipleValues.value1);    print(multipleValues.value2);      final differentDataTypes = returnDifferentDatatypes();    print(differentDataTypes.integer);    print(differentDataTypes.stringValue);  }    // Function returning multiple values using a custom class  MultipleValues returnMultipleValues() {  **return** MultipleValues(1, 2);  }    // Function returning multiple values with  // different data types using a custom class  DifferentDataTypes returnDifferentDatatypes() {  **return** DifferentDataTypes(2, "This is a string");  } |

**Returning multiple values using List**

The same functionality of returning and accessing multiple values can be implemented using lists instead of a custom class. Here’s the code

|  |
| --- |
| **void** main() {    final List<**int**> values1 = returnMultipleValues();    print(values1[0]);    print(values1[1]);      final List<dynamic> values2 = returnDifferentDatatypes();    print(values2[0]);    print(values2[1]);  }    // Function returning multiple  // values as a List of integers  List<**int**> returnMultipleValues() {  **return** [1, 2];  }    // Function returning multiple values  // as a List of different data types  List<dynamic> returnDifferentDatatypes() {  **return** [2, "This is a string"];  } |

**Returning multiple values using Records : Dart 3**

Programming just got simpler, and precision became effortless, thanks to Dart 3’s Records.

|  |
| --- |
| **void** main() {    final (value1,value2)=returnMultipleValues();    print(value1);    print(value2);    final (integer,string\_value)=returnDifferentDatatypes();    print(integer);    print(string\_value);  }    // function  (**int**,**int**) returnMultipleValues(){  **return**(1,2);  }    // Function returning values having  // different datatypes in a single line  (**int**,String)  returnDifferentDatatypes(){  **return**(2,"This is a string");  } |