Session 2

Variables and Data Types

Session Overview

- Define data types.
- Outline the major data types available in Dart.
- Explain how and where data types are used.
- Explain lists, maps, and arrays in Dart.
- Define variables in Dart.
- Outline the types of variables and their usage.

Understanding Data Types in Dart

Variables are utilized in programming to hold values used in a program.

A data type is a categorization that determines which type of value a variable can hold.

Mathematical, logical, and relational operations can be utilized on the created variable.

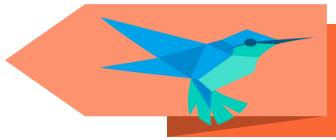
In clearer terms, a data type defines the kind of operations that can be performed upon variables.

Data Type	Keyword
Numbers	`int, double, num`
Boolean	`bool`
Strings	`String`
List	`List`
Map	`Map`
Constants	`const`
Dynamic	`dynamic`
Var	`var`

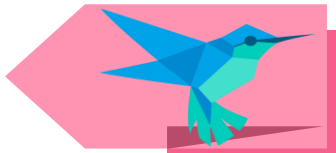
Numbers

The number data type holds a numeric value.

It is categorized into two types: integer and double.



Integer represent a whole number and is declared using the `int` keyword.



Double represents a 64-bit floating number and is declared using the `double` keyword.

Example for int:

```
int price  = 450;
```

Example for double:

```
double total  = 899.90;
```

String and Boolean

String

- The string data type represents the concatenation of several characters.
- The variable value will be encased either in single or double quotes.
- The `String` keyword has to be used to declare a string variable.

```
String name = "Alexander";
```

Boolean

- The boolean data type can either hold a value of true or false.
- Boolean data is mostly used in decision-making statements.
- The `bool` keyword has to be used to declare a boolean variable.

```
bool value = false;
```

List

- The list data type is utilized to represent a list of entities using a single variable.
- The list can only hold an ordered group of items.

Fixed Length List	Variable Length List
This kind of list cannot be changed or modified at runtime.	This specific list is one that can be altered or modified at runtime.

`var marks = List.filled(2,0);`
`marks[0] = 72;`
`marks[1] = 63;`
	`var marks = [10,20];`
	`marks.add(72);`
	`marks.add(63);`

Map

- The map data type is utilized to represent data in key and value form.
- The item stored in this particular type can only be accessed using the associated key provided for the map object.
- The key-value pairs are stored inside curly braces {} and separated from other key-pairs using commas.
- `HashMap`, `LinkedHashMap`, and `SplayTreeMap` are the commonly used map types.

Dynamic and Constants

`dynamic`

The `dynamic` keyword, as the name indicates, is dynamic in nature and chooses not to be initialized with a value at declaration time.

```
dynamic value =
100.05;
```

`const`

The `const` keyword in Dart behaves exactly like the `final` keyword.

```
const pi = 3.14;
```

`final`

The `final` keyword is used to hardcode the values of the variable and it cannot be altered in future.

```
final name =
'Felix';
```

Arrays

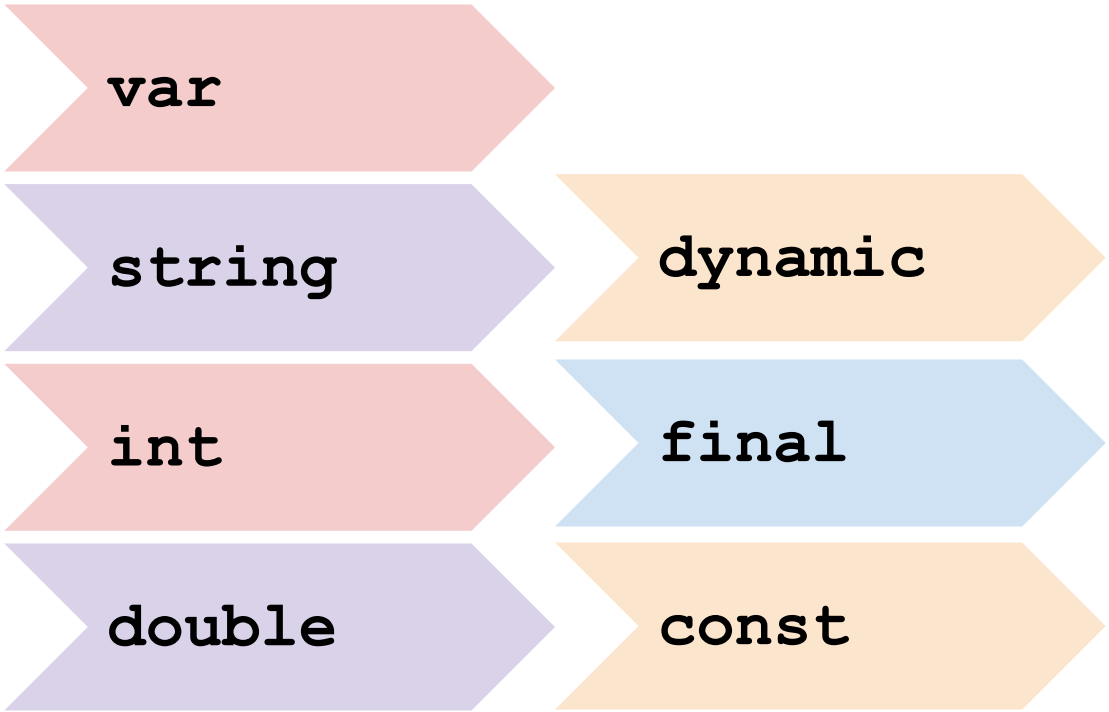
- An array is one of the most commonly used data types to represent a collection of data in Dart.
- Arrays are represented using lists in Dart.
- A list is a collection of elements of the same data type.
- An element in a list can be accessed using its index number.

Identifier	Index	Value
my_array	0	10
	1	14

Variables in Dart

- Upon creation of a variable in Dart, some space is allotted for that particular variable by the Dart compiler.

Keywords for declaring variables



Creating a Variable Using the `var` Keyword

- Developers can use the `var` keyword for assigning variables in Dart.
- The Dart compiler will infer the type of data assigned based on the value given with the `var` keyword.

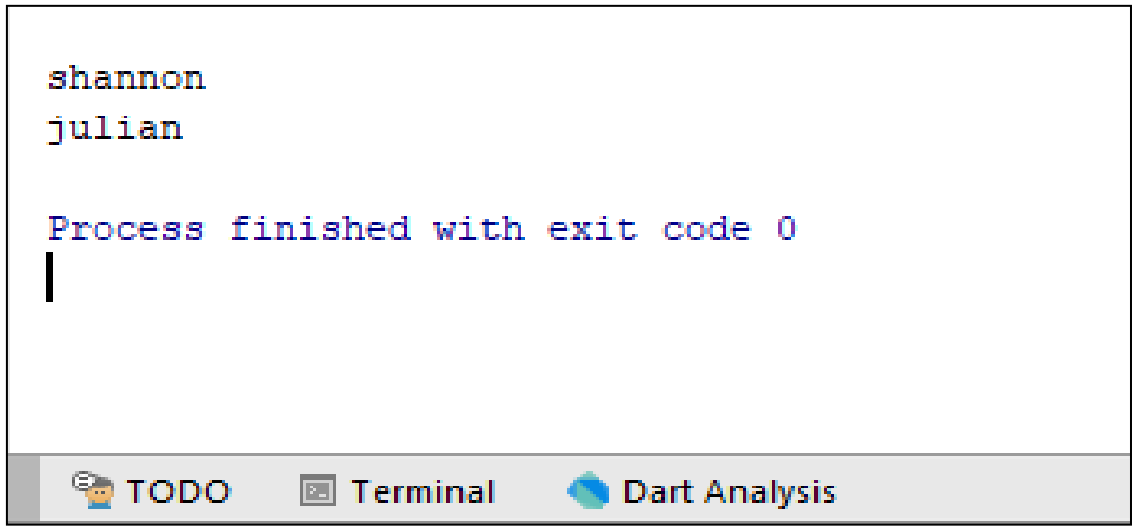
```
void main(){
  var name = &#39;dan&#39;;
  var number1 = 10;
  var number2 = 5.5;
  print(name);
  print(number1);
  print(number2);
}
```

```
void main(){
  var tocheckerror =
      'alexa';
  tocheckerror = 5;
}
```

Creating a Variable Using the `String` Keyword

- The `String` keyword is utilized for assigning a string data type to a variable.
- A string represents a chain of individual characters.
- It is practical to declare the variable as a string and then assign a certain value accordingly.

```
void main(){
  String name = 'shannon';
  print(name);
  name = 'julian';
  print(name);
}
```



Creating a Variable Using the `int` Keyword

- Variables declared using the `int` keyword represent a non-decimal number that can be either a positive or a negative number.
- It is feasible to declare a variable as an int and assign the respective value to it.

```
void main(){
  int num1 = 5;
  print(num1);
  num1 = 10;
  print(num1);
}
```

```
5
10

Process finished with exit code 0
```

TODO Terminal Dart Analysis

Creating a Variable Using the `dynamic` Keyword

- Variables declared using the `dynamic` keyword do not have any predefined data type.
- The assigned value for a dynamic variable can be modified or altered at any juncture.

```
void main(){
  dynamic name = 'rubel';
  dynamic number1 = 10;
  dynamic number2 = 5.5;
  print(name);
  print(number1);
  print(number2);
}
```

```
void main(){
  dynamic changingdatatype =
5.5;
  print(changingdatatype);
  changingdatatype = 'jimmy';
  print(changingdatatype);
}
```

Creating a Variable Using the `final` Keyword

- The `final` keyword is utilized for the creation of immutable objects in Dart.
- The variable with the `final` keyword is utilized when the user wants to always store the same value and does not want it to change.
- An important fact to keep in mind is that the value assigned to a final variable cannot be modified once it is assigned.

```
void main(){
    final name = David;
    print(name);
}
```

```
void main(){
    final name = David;
    name = 'sam';
}
```

Creating a Variable Using the `const` Keyword

- The `const` keyword is utilized to let the user know that the value stored in the variable is constant and cannot change during program execution.
- If the value for the said variable is known at compile-time, then `const` can be used over `final`.
- The only difference between the `final` and `const` keyword is that final is a runtime-constant, which in turn means that its value can be assigned at runtime instead of compile-time.

```
void main(){
    const name = 'Sylvia';
    print(name);
}
```

```
void main(){
    const name = 'Sylvia';
    name = 'sam';
    print(name);
}
```

Summary

- A data type in Dart determines the value a specific variable can hold. This enables any mathematical, logical, or relational operation to be performed on the variable while avoiding errors in code.
- When a variable is declared in Dart, memory is assigned for that variable. The value of that variable is stored in this memory location and can be modified through execution.
- The data type for any variable dictates the memory size, space allocation, and the value the particular variable tends to hold.
- The type annotation in Dart helps to add a data type before the name while declaring a variable.
- The prefix or data type warrants the fact that the variable holds the assigned data type.