Data Types

In this lesson, we will learn about the different data types in Dart.

We'll cover the following

What Are Data Types?

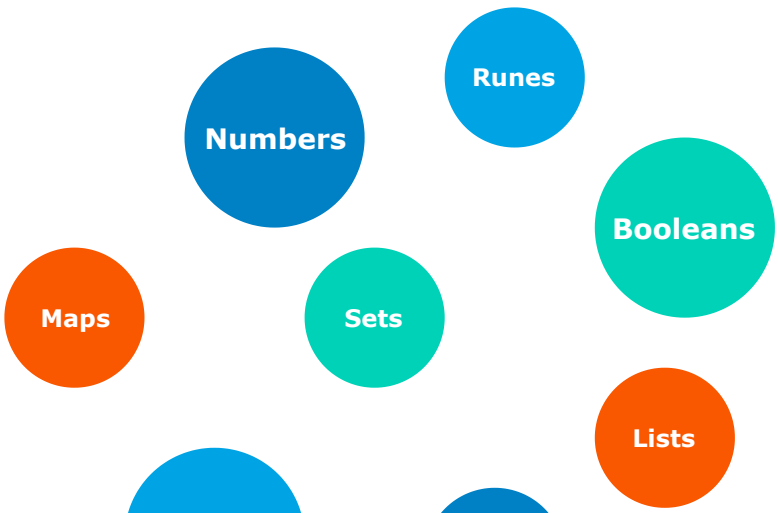
The data type of an item or variable is an attribute that tells us what kind of data that variable can have. This is similar to when we specified which types of items can be stored in our boxes in the previous lesson.

Data types can be found all around us; numbers, alphabets, or characters which are classified based on the similar properties they share.

Dart's Built-In Data Types

The Dart language has special support for the following types:

- Numbers
- Strings
- Booleans
- Lists
- Sets
- Maps
- Runes
- Symbols



The focus of this chapter will be numbers, strings, and booleans. Lists, sets, and maps are topics for another chapter. Runes and symbols are beyond the scope of this course.

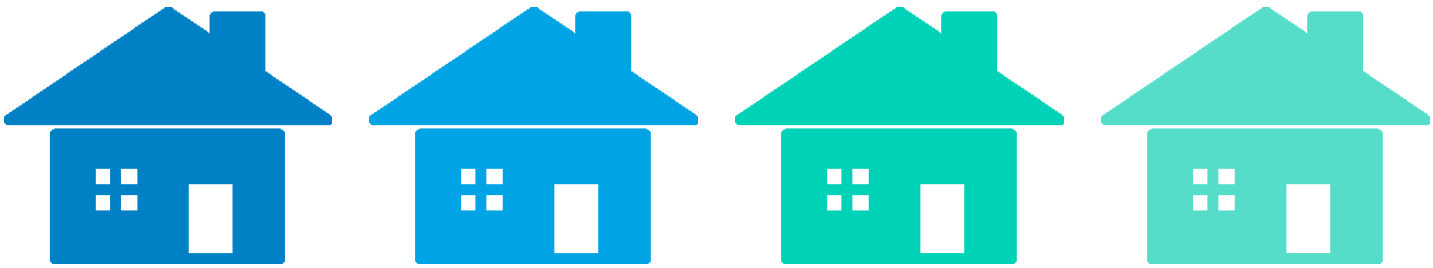
Before we can move on to each data type in detail, there are a couple of things you should familiarize yourself with.

Values and References

Data types can be broadly divided into two categories:

1. **Reference type**
2. **Value type**

The information provided by a value type is the *value* itself. For a reference type, the information it provides is a *reference* to some object, i.e., the *memory address* of where an object is stored. To make this clearer, let's look at this using physical objects.



Imagine you have a piece of paper. You want the paper to hold some information, such as your name. You can write your name on the piece of paper; therefore, the value of the paper is the same as the information it provides. This is an example of *value type*. If someone wants to know your name, all they have to do is read the paper.

Now imagine, you want the paper to hold your house. That's not physically possible, so you write the address to your house, hence the value of the paper is a reference to the required information. This is an example of *reference type*. If someone wants to go to your house, they will first have to read the address on the paper and then get directions to your house.

In the same way, a reference type holds the memory address location of the value, while value types hold the value themselves.

Data Types Are Objects

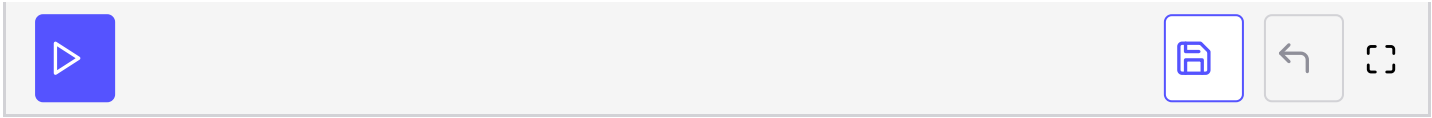
In most languages, primitive data types are value types, but in Dart, **all** data types are objects. This means that even primitive data types are reference types. Therefore, we can say that in Dart, variables specifically store references and are referring to objects.

Default Value

Uninitialized variables have an initial value of `null`. Even variables with numeric types are initially `null` because numbers—like everything else in Dart—are objects. `null` simply means that the variable is not referencing an object; it's not referencing anything.

In the code snippet below, we are creating a variable `notInitialized` without initializing it. When we try to print the value of `notInitialized`, we get `null` as the output.

```
main() {
    int notInitialized;
    print(notInitialized);
}
```



Literals

We will be using the word *literal* throughout the course, so let's take a closer look at what a literal actually is.

A literal is defined as taking anything in its most usual and basic sense. Mapping this onto computer programming, literals are fixed values appearing directly as they do in the source code. For example, "Hello World", **5**, and 'A' are all literals.

Let's take a closer look at numbers in the next lesson.