Kotlin Variables and Data Types

A variable is a location in memory (storage area) that holds data.
There are 2 types of variables in Kotlin:

- Immutable variable: Once assigned a value, it cannot be changed or assigned a different value.
- Mutable variables: Can be assigned different values from time to time

Immutable variables are declared using the `val` keyword e.g:

```
val name = "Mary"
```

If you try to reassign an immutable variable e.g for name above:

```
name = "Anna"
```

the program will not compile. It will show an error `val cannot be reassigned`

Mutable variables are declared using the `var` keyword e.g:

```
var wheels = 4
```

In summary:

- A variable is a memory location that holds data
- A variable has a name
- A variable has a type. This is the data type of the value given to a variable. Once assigned a given type, a variable cannot accept a value of a different type.
- A mutable variable `var` can be reassigned different values. An immutable variable `val`cannot be reassigned once given a value.

Naming Variables

A Kotlin variable can have any name so long as it is not a keyword. The name of the variable also needs to be descriptive of the data it holds.
Here are some conventions used in naming variables:

- A variable name should begin with a small letter
- A variable name must not begin with a number
- A variable name can contain any combination of numbers and letters e.g `year2020Revenue, counter2Max, counter1, annualCost`
- Camel case is preferred over underscores for long variable names e.g use `agentNumber` instead of `agent_number`

Kotlin Basic Data Types

Kotlin variables are statically typed, meaning once assigned a value a a variable can only hold values of the same type as the first declared type.

The built-in types in Kotlin can be categorized as:

- Numbers
- Characters
- Booleans
- Arrays

Numbers

There are 6 numeric types in Kotlin:

- Byte - Can hold numeric values from -127 to 128. Used to save memory when we are certain that a variable's value will fall strictly between -127 to 128. e.g:

```
var age: Byte = 20
```

- Short - Can hold values between -32768 and 32767. Used to save memory when we the variable’s value falls within range. e.g:

```
var students: Short = 4507
```

- Int - Used to hold values between -2,147,483,648 and 2,147,483,647. e.g:

```
var balance: Int = 87432563
```

- Long - The `Long` data type can hold values from -2^63 to 2^63 -1. e.g:

```
var worldPopulation: Long = 7000000000L
```

- Float - Used to hold 32bit decimal numbers with maximum 6-7 decimal digits e.g:

```
var worldPopulation: Float = 20.324
```

- Double - Used to hold 64bit decimals with maximum 15-16 decimal digits e.g:

```
var worldPopulation: Double = 42979.829792792728
```

Characters:

Characters in Kotlin are represented by the String type. e.g:

```
var name: String = "Jane Nafula"
val text: String = 'PB782BU56G Confirmed. You have received Ksh100 from
Ronald Ngala'
```

Strings must be enclosed in quotes. These can be single or double matching quotes.

Booleans

The `Boolean` data type has two possible values, either `true` or `false`. They are used in making decisions. An example of a boolean variable is:

```
var daytime: Boolean = true
val receivedParcel: Boolean = false
```

Arrays

An array is a container that holds many values. e.g an array of 10 integers or 2 Strings etc. Arrays could also hold values of different types. Examples:

```
var names = arrayOf("Jane", "Anne", "Mary", "Asha")
var ages = arrayOf(19, 20, 19, 21)
var data = arrayOf("MAXIMUM", 234, 32.5, false)
```

Inferred types:

You may declare a variable explicitly stating its type e.g

```
val height: Float = 5.5
```

or you may declare the same variable without stating its type like

```
val height = 5.5
```

Both variable declarations are correct and safe. In the second style the compiler will infer that the height variable is a float from its assigned value.