SWIFT - VARIABLES

A variable provides us with named storage that our programs can manipulate. Each variable in Swift has a specific type, which determines the size and layout of the variable's memory; the range of values that can be stored within that memory; and the set of operations that can be applied to the variable.

There are following basic types of variable in Swift as explained in last chapter:

- **Int or UInt** - This is is used for whole numbers. More specifically you can use Int32, Int64 to define 32 or 64 bit signed integer where as UInt32 or UInt64 to define 32 or 64 bit unsigned integer variables. For example, 42 and -23.
- **Float** - This is used to represent a 32-bit floating-point number and used for numbers with smaller decimal points. For example 3.14159, 0.1, and -273.158.
- **Double** - This is used to represent a 64-bit floating-point number and used when floating-point values must be very large. For example 3.14159, 0.1, and -273.158.
- **Bool** - This represents a boolean value which is either true or false.
- **String** - This is ordered collection of characters. For example, "Hello, World!"
- **Character** - This is a single-character string literal. For example, "C"

Swift also allows to define various other types of variables, which we will cover in subsequent chapters like **Optional, Array, Dictionaries, Structures,** and **Classes** etc.

Following section will cover how to declare and use various types of variables in Swift programming.

Variable Declaration

A variable declaration means to tell the compiler where and how much to create the storage for the variable. Before you use variables, you must declare them using **var** keyword as follows:

```
var variableName = <initial value>
```

Following is a simple example to show how to declare a variable in Swift:

```
import Cocoa

var varA = 42
println(varA)
```

When we run above program using playground, we get following result.

```
42
```

Type Annotations

You can provide a **type annotation** when you declare a variable, to be clear about the kind of values the variable can store. Following is the syntax:

```
var variableName:<data type> = <optional initial value>
```

Following is a simple example to show how to declare a variable in Swift using Annotation. Here it is important to note that if we are not using type annotation, then it becomes mandatory to provide initial value for the variable, otherwise we can just declare our variable using type annotation.

```
import Cocoa
```

```
var varA = 42
println(varA)

var varB:Float

varB = 3.14159
println(varB)
```

When we run above program using playground, we get following result.

```
42
3.1415901184082
```

Naming Variables

The name of a variable can be composed of letters, digits, and the underscore character. It must begin with either a letter or an underscore. Upper and lowercase letters are distinct because Swift is case-sensitive programming language.

You can use simple or Unicode characters to name your variables. Following are valid examples:

```
import Cocoa

var _var = "Hello, Swift!"
println(_var)

var 你好 = "你好世界"
println(你好)
```

When we run above program using playground, we get following result.

```
Hello, Swift!
你好世界
```

Printing Variables

You can print the current value of a constant or variable with the **println** function. You can interpolate a variable value by wrapping the name in parentheses and escape it with a backslash before the opening parenthesis: Following are valid examples:

```
import Cocoa

var varA = "Godzilla"
var varB = 1000.00

println("Value of \(varA) is more than \(varB) millions")
```

When we run above program using playground, we get following result.

```
Value of Godzilla is more than 1000.0 millions
```