Swift Data Types:

let : used for defining a **constant** (a value that will not change)
var : used for defining a **variable** (a value that might change)

Int : for 32 bit integers
eg :

```
 var age : Int!
 let id = 89
```

Float :
used for 32 bit floating point numbers (number with fraction values)
eg:

```
var amount : Float!
let bill = 68.5
```

Double :
Used for 64 bit floating pointing numbers.
Eg :

```
var total : Double!
let total = 567.57
```

Strings:
Represents series of characters

```
var name : String!
 let lang = “Swift”
```

Booleans:
Used for true or false values
eg:

```
var isLoggedIn : Bool!
let result = true
```

Arrays:
Arrays store a list of values that must be of the same type.
You can create an empty array of a certain type using the following syntax :

```
var array_name = [dataType]()
```

eg:

```
var id = [Int]()
```

Adding values to the array :

```
id.append(5)
id.append(10)
```

Initializing array:

```
let billAmounts: [Double] = [10.25, 21.32, 15.54]

let os_x = ["mojave","sierra", "captain","yosemite"]
```

Retrieving the values:

```
let first =  id[0]
print(first)  -> Prints the first value in the array

let second = id[1]
print(second) - > Prints the second value
```

Iterating Over an Array

You can use **for-in** loop to iterate over the entire set of values in an array as shown in the following example −

```
for item in os_x{
        print(item)
}
```

When the above code is compiled and executed, it produces the following result :

```
mojave
sierra
captain
yosemite
```

Dictionaries:

are used to store key-value pair in unordered manner.
In dictionaries we can store same type of **keys** and same type of **values** with no defined order. The **keys** in dictionaries must be unique and those will act as identifiers for the **values** in dictionaries.

eg:

```
var names : Dictionary<Int,String>
```

Adding elements :

```
names = [1:"Karthik",4:"Muthu"]
```

Adding a value

```
names[24] = “Kumar"
```

Retrieving elements:

```
print(names) -> [4: "Muthu", 1: "Karthik", 24: “Kumar"]

print(names[1]) -> “Karthik”
```