String:

A string is a series of characters surrounded by double quotes (""), such as "Hello Swift" and they are represented by the swift data type String.

Initializing a String

There are a number of ways to create a string.

```
// String creation using String literal
var stringA = "Hello, Friends!"
print( stringA )
// String creation using String instance
var stringB = String("Hello, Friends!")
print( stringB )

//Empty String  creation using String literal
var emptyString = ""// Empty String
//String creation using String instance
var stillEmpty = String()
//From boolean
let x = String(true)
// from Double "3.14"
let d = String(3.14)
// from Int "10"
let e = String(10)

//By using the isEmpty property you ca check the string whether is empty or not.
var name = ""
if name.isEmpty {
    print( "string is empty" )
}else {
    print( "string is not empty" )
}
```

Characters:

By passing the String with a for-in loop we can get the individual Character of the String.

By using the count property we can get how many characters are there in a String.

```
var sampleString = "I am From India"
for character in sampleString.characters {
    // to convert a Character into a String use string interpolation
    var characterAsString = "\(character)"
    print(characterAsString)
}
```

String Indices:

Each string value has an associated index type, string.index, which corresponding to the position of each character in the string.

Use the startIndex property to access the position of the first Character of a string.
The endIndex property is the position after the last character in a string.

NOTE: If a string is empty, startIndex and endIndex are equal

```
let sampleString = "Hello Swift!"
sampleString[sampleString.startIndex]
// H

sampleString[sampleString.index(before: sampleString.endIndex)]
// !
sampleString[sampleString.index(after: sampleString.startIndex)]
// e
let index = sampleString.index(sampleString.startIndex, offsetBy: 7)
sampleString[index]
// W
```

Reverse String :

```
let string = "Hello swift"
let characters = string.characters
let reverseCharcters = characters.reversed()
let reversedString = String(reverseCharcters)
print(reversedString) //tfiws olleH
```

String Concatenation:

Combine strings using the concatenation operator(+).

```
let firstName = "Avul Pakir Jainulabdeen"
let lastName = "Abdul Kalam"

let fullName = firstName + lastName
print("\(fullName)") //Avul Pakir Jainulabdeen Abdul Kalam
```

(OR)

By using append method to combine the two strings.

```
let fullName = firstName.appending(lastName)

print("\(fullName)") //Avul Pakir Jainulabdeen Abdul Kalam
```

String Comparision:

You can use the == operator to compare two strings

```
var nameString = "Hello"
var addressString = "World"

if nameString == addressString {
   print( "\(nameString) and \(addressString) are equal" )

}else{
   print( "\(nameString) and \(addressString) are not equal" )

}
```

hasPrefix and hasSuffix:

These are the two properties to check the starting and ending of the string.

```
var sampleString = "Hello World!"

// Check if there is a specific Prefix

if sampleString.hasPrefix("Hello") {
   print("Yes, there is")
}
else {
   print("Not There!")
}


if sampleString.hasSuffix("World!") {
   print("Yes, there is")

}else{
   print("Not There!")

}
```

String Converting to upper/lower case

```
let mixedCaseString = "HelLo WorLD"
let upper = mixedCaseString.uppercased() // "HELLO WORLD"
let lower = mixedCaseString.lowercased() // "hello world"
```

String Trimming:

String Replace:

String Separation:

Generate Unique Identifier(UUID): A UUID is a universally unique identifier, which means if you generate a UUID right now using UUID it's guaranteed to be unique across all device in the world.

```
let uuid = UUID().uuidString
```