

# How to find SubString in Swift?

## Swift Substring

To find substring of a String in Swift, prepare range using start and end indexes, then use the range on the string.

The syntax to concatenate two strings is:

```
let start =  
str.index(str.startIndex,  
  
let start = str.index(str.startIndex, offsetBy: startPosition)  
let end = str.index(str.endIndex, offsetBy: -endPosition)  
let range = start..  
end  
  
let subStr = str[range]
```

where `startPosition` and `endPosition` are integers that define the bounds of the substring in the main string `str`.

`startPosition` is the index (from starting) in the main string from which substring has to be confirmed.

`endPosition` is the index (from ending) in the main string.

Rest all are keywords or functions or constants. Finally `subStr` holds the substring.

You might be wondering, why all this hassle to find a substring! One possible theory is that Swift Characters being composed of varying number of Unicode codepoints. The actual index has to be uniquely calculated for every string.

### Example – Find Substring of a String in Swift

This example demonstrates how to find the substring of a string. The starting position is taken as `2` and the ending position is taken as `4`.

```
main.swift  
var str = "TutorialKart"
```

```
var str = "TutorialKart"

let start = str.index(str.startIndex, offsetBy: 2)
let end = str.index(str.endIndex, offsetBy: -4)
let range = start..
```

Output

```
torial
```

```
torial
```

## Example – Substring of a String with startPosition 0

In this example, the starting position is `0`, hence the substring starts from the start of main string and ends at the position defined by ending index. Ending index is `4` in this example.

main.swift

```
var str = "TutorialKart"
```

```
var str = "TutorialKart"

let start = str.index(str.startIndex, offsetBy: 0)
let end = str.index(str.endIndex, offsetBy: -4)
let range = start..
```

Output

```
Tutorial
```

```
Tutorial
```

## Example – Substring of a String with endPosition 0

In this example, the ending position is `0`, hence the substring ends at the end of main string. However the

start position is defined by starting index. Starting index is `5` in this example.

main.swift

```
var str = "TutorialKart"
```

```
var str = "TutorialKart"  
  
let start = str.index(str.startIndex, offsetBy: 5)  
let end = str.index(str.endIndex, offsetBy: 0)  
let range = start..  
end  
  
let subStr = str[range]  
  
print( subStr )
```

Output

```
ialKart
```

```
ialKart
```

## Example – Substring of a String with startPosition greater than String length

Sometimes, it can happen that we provide start position or end position exceeding string length.

In this example, we provided starting position greater than the length of the string. Hence the Fatal error with the message: cannot increment beyond endIndex.

main.swift

```
var str = "TutorialKart"
```

```
var str = "TutorialKart"  
  
let start = str.index(str.startIndex, offsetBy: 25)  
let end = str.index(str.endIndex, offsetBy: 0)  
let range = start..  
end  
  
let subStr = str[range]  
  
print( subStr )
```

```
Fatal error: cannot  
increment beyond
```

Fatal error: cannot increment beyond endIndex

Current stack trace:

```
0 libswiftCore.so 0x00007fda520a6bc0 _swift_stdlib_reportFatalError + 171
1 libswiftCore.so 0x00007fda5204cf3f <unavailable> + 4149055
2 libswiftCore.so 0x00007fda51dad96b <unavailable> + 1399147
3 libswiftCore.so 0x00007fda52003c71 <unavailable> + 3849329
4 libswiftCore.so 0x00007fda51dad96b <unavailable> + 1399147
5 libswiftCore.so 0x00007fda51f44840 specialized _fatalErrorMessage(_:_:file:line:flags:) + 96
6 libswiftCore.so 0x00007fda51f816c9 <unavailable> + 3315401
7 libswiftCore.so 0x00007fda51ee4a40 String.index(_:_:offsetBy:) + 79
9 swift 0x0000000000f28eff <unavailable> + 11702015
10 swift 0x0000000000f2d0a2 <unavailable> + 11718818
11 swift 0x00000000004bf4f6 <unavailable> + 783606
12 swift 0x00000000004ae441 <unavailable> + 713793
13 swift 0x00000000004aa3f1 <unavailable> + 697329
14 swift 0x0000000000465424 <unavailable> + 414756
15 libc.so.6 0x00007fda547a1740 __libc_start_main + 240
16 swift 0x0000000000462ce9 <unavailable> + 404713
0 swift 0x00000000003ad7e98
1 swift 0x00000000003ad85d6
2 libpthread.so.0 0x00007fda56077390
3 libswiftCore.so 0x00007fda51f448a0 _T0s18_fatalErrorMessages5NeverOs12StaticStringV_A2E4fileSu4lines6UInt32V5flagstFTf
4 libswiftCore.so 0x00007fda51f816c9
5 libswiftCore.so 0x00007fda51ee4a8f _T0SS5indexSS5IndexVAC_Si8offsetBytF + 79
6 libswiftCore.so 0x00007fda564a4130 _T0SS5indexSS5IndexVAC_Si8offsetBytF + 73135856
7 swift 0x0000000000f28eff
8 swift 0x0000000000f2d0a2
9 swift 0x00000000004bf4f6
10 swift 0x00000000004ae441
11 swift 0x00000000004aa3f1
12 swift 0x0000000000465424
13 libc.so.6 0x00007fda547a1830 __libc_start_main + 240
14 swift 0x0000000000462ce9
Stack dump:
0. Program arguments: /usr/bin/swift -frontend -interpret main.swift -disable-objc-interop -module-name main
timeout: the monitored command dumped core
Illegal instruction
```

## Summary

In this [Swift Tutorial](#), we have concatenated two or more strings using concatenation operator `+`.

‡ Swift Keywords

‡ Swift Comments

‡ Swift If

‡ Swift If-Else

‡ Swift For Loop

‡ Swift While Loop

‡ Swift forEach

‡ Swift Repeat While Loop

‡ Swift Break

‡ Swift Continue

‡ Swift Tuple

‡ Swift Enum

‡ Swift Structure

## Strings

‡ Swift - Substring

‡ Swift - Concatenate Strings

## Arrays

‡ Swift Array Initialization

‡ Swift Print Array

‡ Swift Integer Array

‡ Swift append Integer to Array

‡ Swift String Array

‡ Swift append String to Array

‡ Swift Get Array Size - count

‡ Swift Remove an Element from Array

‡ Swift Append / Concatenate Arrays

‡ Swift Check if an Array is Empty

## Dictionaries

‡ Swift Dictionary

‡ Swift - Create Dictionary

‡ Swift - Create Dictionary using Arrays

‡ Swift - Iterate through Dictionary

‡ Swift - Get Dictionary Size

‡ Swift - Check if Dictionary is Empty

‡ Swift - Add or Append Element to Dictionary

‡ Swift - Get value using key in Dictionary

‡ Swift - Check if a key is present in Dictionary

‡ Swift - Merge Two Dictionaries

‡ Swift - Convert Dictionary into Arrays of Keys and Values

‡ Swift - Print all Keys in a Dictionary

## Sets

‡ Swift Print Set

‡ Swift Get Set Size

‡ Swift Insert Element to Set

‡ Swift Check if Element is present in Set

## File Operations

‡ Swift Read Text File

## Swift Errors [Solved]

‡ Swift error: return from initializer without initializing all stored properties

‡ Swift - struct error: missing argument labels in call