

# Swift Tuple – Initialize, Access – Examples

## Swift Tuples

Tuples are used to group values of different datatypes under a single variable name.

Following is a simple example of Swift Tuple.

```
var player1 = (45,  
"Rohit")
```

```
var player1 = (45, "Rohit")
```

The tuple player1 has two values. The first is integer and the second is string: different datatypes.

You can have a mix of any datatypes in a single tuple.

## Accessing a Tuple

You can access a tuple using the dot operator and index.

```
var player1 = (45,  
"Rohit")
```

```
var player1 = (45, "Rohit")  
var jersyNumber = player1.0  
var name = player1.1
```

## Named Tuple

You can also have names for the values inside tuple. And access those values using names.

```
var player2 =  
(jersyNumber: 15,
```

```
var player2 = (jersyNumber: 15, name: "Kohli")  
var jersyNumber = player2.jersyNumber  
var name = player2.name
```

## Edit Tuple

You can edit values at value level inside a tuple.

```
var player1 = (1,  
"Rohit")
```

```
var player1 = (1, "Rohit")  
player1.0 = 2
```

## Example – Swift Tuple Initialization & Access Values

This example demonstrates how to initialize and access values of a Tuple in Swift.

main.swift

```
//tuple initialization  
var player1 = (45,
```

```
//tuple initialization  
var player1 = (45, "Rohit")  
//access values of tuple  
print( player1 )  
print( player1.0 )  
print( player1.1 )  
  
/* named values in tuple */  
//tuple initialization  
var player2 = (jerseyNumber: 15, name: "Kohli")  
//access values inside tuple using names  
print( player2 )  
print( player2.jerseyNumber )  
print( player2.name )
```

Output

```
(45, "Rohit")  
45
```

```
(45, "Rohit")  
45  
Rohit  
(jerseyNumber: 15, name: "Kohli")  
15  
Kohli
```

## Example – Modify Values inside Tuple

In this example, we will modify the values inside tuple. We will edit the first value or jerseyNumber for the player tuples.

main.swift

```
//tuple initialization
var player1 = (1
```

```
//tuple initialization
var player1 = (1, "Rohit")
player1.0 = 2
print( player1 )

/* named values in tuple */
var player2 = (jerseyNumber: 15, name: "Kohli")
player2.jerseyNumber = 27
print( player2 )
```

Output

```
(2, "Rohit")
(jerseyNumber: 27
```

```
(2, "Rohit")
(jerseyNumber: 27, name: "Kohli")
```

## Summary

In this [Swift Tutorial](#), we have learned to initialize, access and modify Swift Tuples with examples.

### Swift Tutorial

- [Swift Tutorial](#)
- [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 File

↳ 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