**Swift Tuple**

In Swift, a tuple is a group of different values. And, each value inside a tuple can be of different data types.

Suppose we need to store information about the name and price of a product, we can create a tuple with a value to store name (string) and another value to store price (float)

**Create A Tuple**

In Swift, we use the parenthesis () to store elements of a tuple. For example,

var product = ("MacBook", 1099.99)

Here, product is a tuple with a string value Macbook and integer value **1099.99**.

**Access Tuple Elements**

Like an [array](https://www.programiz.com/swift-programming/arrays), each element of a tuple is represented by index numbers (**0**, **1**, ...) where the first element is at index **0**.

We use the index number to access tuple elements. For example,

// access the first element

product.0

// access second element

product.1

**Example: Swift Tuple**

// create tuple with two elements

var product = ("MacBook", 1099.99)

// access tuple elements

print("Name:", product.0)

print("Price:", product.1)

**Output**

Name: MacBook

Price: 1099.99

In the above example, we have created a tuple named product with two values.

We have used the index number: product.0 and product.1 to access tuple elements.

**Modify Tuple Element**

We can modify a tuple element by assigning a new value to the particular index. For example,

// create tuple with two elements

var product = ("MacBook", 1099.99)

print("Original Tuple: ")

// access tuple elements

print("Name:", product.0)

print("Price:", product.1)

// modify second value

product.1 = 1299.99

print("\nTuple After Modification: ")

// access tuple elements

print("Name:", product.0)

print("Price:", product.1)

**Named Tuples**

In Swift, we can also provide names for each element of the tuple. For example,

var company = (product: "Programiz App", version: 2.1)

**Example: Nested Tuple**

var alphabets = ("A", "B", "C", ("a", "b", "c"))

// access first element

print(alphabets.0) // prints "A"

// access the third element

print(alphabets.3)

// access nested tuple

print(alphabets.3.0) /

**Add/Remove Elements From Tuple**

We cannot add or remove elements from a tuple in Swift. For example,

var company = ("Programiz","Apple")

company.2 = "Google"

company.remove("Apple")

print(company)

**Dictionary Inside a Tuple**

In Swift, we can use a dictionary to add an element to a tuple. For example,

var laptopLaunch = ("MacBook", 1299, ["Nepal": "10 PM", "England": "10 AM"])

print(laptopLaunch.2)

laptopLaunch.2["USA"] = "11 AM"

print(laptopLaunch.2)