



# ОСНОВЫ Swift

# **Associated Types**

# Объявление

```
protocol Printable {
    associatedtype PrintableType
    associatedtype OtherPrintableType

    func printValues(_ arg1: PrintableType,
        and arg2: OtherPrintableType)
}
```

---

## Пример реализации протокола

```
class Document: Printable {  
    func printValues(_ arg1: Bool, and arg2: Int) {  
  
    }  
}
```

# Typealias

```
protocol ViewModel {  
    associatedtype ContextType  
  
    var context: ContextType?  
  
    init(with context: ContextType?)  
}
```

```
class ScheduleViewModel:  
    ViewModel {  
    typealias ContextType = Int  
  
    var context: ContextType?  
  
    required init(with context:  
        ContextType?) {  
        self.context = context  
    }  
}
```

# Ограничения для типов

```
protocol Container {  
    associatedtype Item  
    mutating func append(_ item: Item)  
    var count: Int { get }  
    subscript(i: Int) -> Item { get }  
  
    associatedtype Iterator: IteratorProtocol  
    where Iterator.Element == Item  
    func makeIterator() -> Iterator  
}
```