

# Swift

## - Optional Unwrapping -

yagom

# Optional Unwrapping

- Optional Binding - 옵셔널 바인딩
- Force Unwrapping - 강제 추출

# Optional Binding

옵셔널의 값을 꺼내오는 방법 중 하나

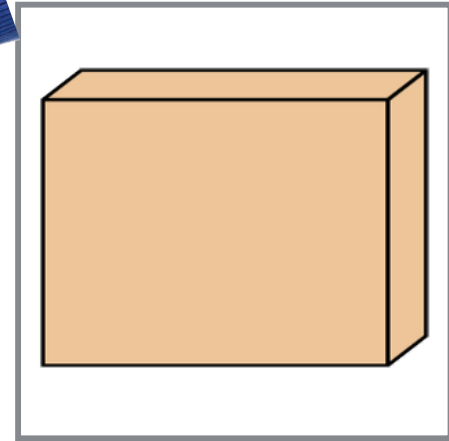
nil 체크 + 안전한 값 추출



Int



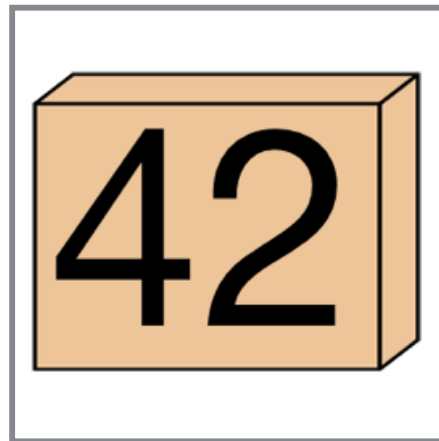
Int?



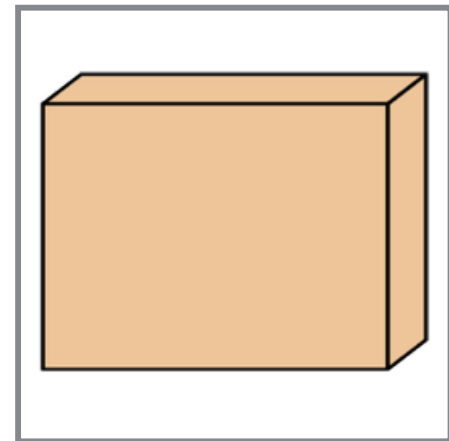
Int?



Int



Int!



Int!

```
func printName(_ name: String) {  
    print(name)  
}
```

```
var myName: String? = nil
```

```
printName(myName)  
// 전달되는 값의 타입이 다르기 때문에 컴파일 오류발생
```

# if-let

```
func printName(_ name: String) {  
    print(name)  
}
```

```
var myName: String? = nil
```

```
if let name: String = myName {  
    printName(name)  
} else {  
    print("myName == nil")  
}
```

```
// name 상수는 if-let 구문 내에서만 사용가능합니다  
// 상수 사용범위를 벗어났기 때문에 컴파일 오류 발생  
printName(name)
```

# if-let

```
func printName(_ name: String) {  
    print(name)  
}
```

```
var myName: String! = nil
```

```
if let name: String = myName {  
    printName(name)  
} else {  
    print("myName == nil")  
}
```

```
// name 상수는 if-let 구문 내에서만 사용가능합니다  
// 상수 사용범위를 벗어났기 때문에 컴파일 오류 발생  
printName(name)
```

# if-let

```
var myName: String? = "yagom"  
var yourName: String? = nil  
  
if let name = myName, let friend = yourName {  
    print("\(name) and \(friend)")  
}  
// yourName이 nil이기 때문에 실행되지 않습니다  
  
yourName = "hana"  
  
if let name = myName, let friend = yourName {  
    print("\(name) and \(friend)")  
}  
// yagom and hana
```

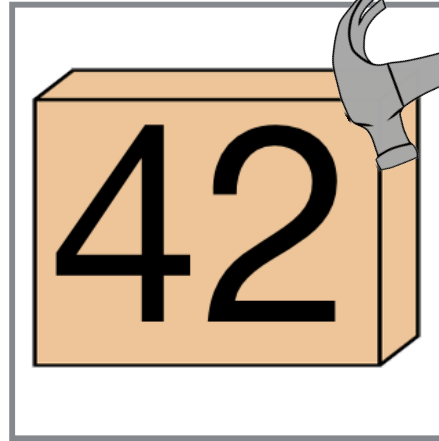


# Force Unwrapping

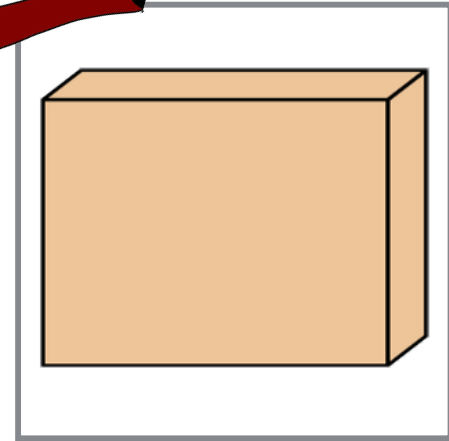
옵셔널의 값을 강제로 추출



Int



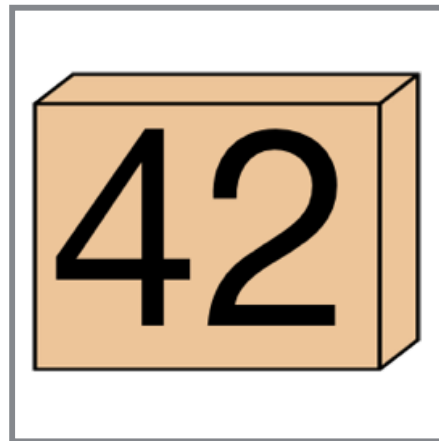
Int?



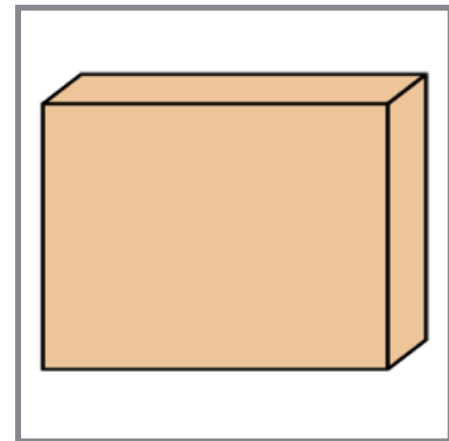
Int?



Int



Int!



Int!

```
func printName(_ name: String) {  
    print(name)  
}
```

```
var myName: String? = "yagom"
```

```
printName(myName!) // yagom
```

```
myName = nil
```

```
print(myName!)  
// 강제추출시 값이 없으므로 런타임 오류 발생
```

```
var yourName: String! = nil
```

```
printName(yourName)  
// nil 값이 전달되기 때문에 런타임 오류발생
```