

Optional

류경혜

content

- a. Optional?
- b. nil?
- c. String? vs. String
- d. Force Unwrapping
- e. Optional Binding
- f. nil coalescing operator
- g. Optional Chaining(JS)

Optional?

변수의 값이 있거나 없거나



```
var optionalString: String = nil  
var optionalInt: Int? = nil
```

nil

"" → String

0 → Int

nil → 비어 있다

String? vs. String

```
let optionalText: String? = "gaori"  
let text: String  
text = optionalText  
print(text)
```



Value of optional type 'String?' must be unwrapped to a value of type 'String'



Coalesce using '??' to provide a default when the optional value contains 'nil'

Fix

Force-unwrap using '!' to abort execution if the optional value contains 'nil'

Fix

Force Unwrapping



```
let optionalText: String? = "gaori"  
let text: String  
text = optionalText  
print(text)
```



```
let optionalText: String? = "gaori"  
let text: String  
text = optionalText!  
print(text)
```

Force Unwrapping



```
var value1: String? = "gaori"  
var value2: String?  
var value3: String! = nil  
var value4: String! = "Ak0"  
  
print(value1) // Optional("gaori")  
print(value2) // nil  
print(value3) // Error  
print(value4) // Optional("Ak0")
```

! 사용하여 강제로 optional 추출

Optional Binding

```
let name = "gaori"
let myAge: Int? = 3
if let age = myAge {
    print("\(name)의 나이는 \(age)입니다.")
} else{
    print("\(myAge)는 변환될 수 없습니다.")
}
// gaori의 나이는 3입니다.
print(age) // Error
```

```
let myAge: Int? = 3
let myName: String? = "gaori"
if let age = myAge, let name = myName{
    print("\(name)는 \(age)살")
} // gaori는 3살
```

if let, guard let 사용하여 optional 추출

Nil coalescing operator



```
let myAge: Int? = 3
var age: Int
if let myAge = myAge {
    age = myAge
    print(age)
}
// 3
```



```
let myAge: Int? = 3
var age: Int

age = myAge ?? 0

print(age) // 3
```

?? 사용하여 아주 간단히 unwrapping

Optional Chaining(JS)



```
animal = {info: {name: "gaori", age: 3}}  
animal.info.name // 'gaori'
```



```
animal?.info?.name    // gaori  
animal?.info?.habitat // undefined
```

- Force Unwrapping: ! 사용하여 강제로 optional 추출
- Optional Binding: if let, guard let 사용하여 optional 추출
- Nil coalescing operator: ?? 사용하여 아주 간단히 unwrapping

감사합니다