옵셔널

optional

값이 있을수도 없을수도 있음

why?

nil의 가능성을 명시적으로 표현

- nil 가능성을 문서화 하지 않아도 코드만으로 충분히 표현가능
    - 문서/주석 작성 시간을 절약
- 전달받은 값이 옵셔널이 아니라면 nil체크를 하지 않더라도 안심하고 사용
    - 효율적인 코딩
    - 예외 상황을 최소화하는 안전한 코딩

```
// enum + general
enum Optional<Wrapped> : ExpressibleByNilLiteral {
  case none
  case some<Wrapped)
}
let optionalValue: Optional<Int> = nil
let optionalValue: Int? = nil
```

!

Implicitly Unwrapped Optional (암시적 추출 옵셔널)

```
var optionalValue: Int! = 100
switch optionalValue {
  case .none:
    print("This Optional variable is nil")
  case .some(let value):
    print("Value is \(value)")
}

// nil 할당 가능
optionalValue = nil

// 기존 변수처럼 사용불가 - 옵셔널과 일반 값은 다른 타입으므로 연산불가
optionalValue = optionValue + 1
```