



Optional

열거형

Optional

래핑된 값 또는 값이 없음을 나타내는 형식입니다.

선언

```
@frozen enum Optional<Wrapped>
```

개요

Optional이라는 단어를 입력하지 않더라도 선택적 값을 사용할 때마다 유형을 사용합니다.

Swift의 유형 시스템은 일반적으로 전체 유형 이름을 표시하는 대신 후행 물음표(?)와 함께 래핑된 유형의 이름을 표시합니다.

예를 들어, 변수에 Optional 유형이 있는 경우 Int?는 Optional<Int>를 작성하는 또 다른 방법일 뿐입니다.

코드를 쉽게 읽고 쓸 수 있도록 축약형이 선호 됩니다.

다음 코드 샘플 및 유형은 동일합니다.

```
let shortForm: Int? = Int("42")  
let longForm: Optional<Int> = Int("42")
```

Optional 유형은 두 가지 경우가 있는 열거형입니다.

리터럴과 동일합니다.

래핑된 값을 저장합니다.

예를 들어: `Optional.nonNilOptional.some(Wrapped)`

```
let number: Int? = Optional.some(42)
let noNumber: Int? = Optional.none
print(noNumber == nil)
// Prints "true"
```

많은 컨텍스트에서 인스턴스를 사용하려면 인스턴스 값을 언래핑해야 합니다.

Swift는 선택적 값을 안전하게 언래핑하는 여러 가지 방법을 제공하므로 명확하고 간결한 코드를 작성하는데 도움이 되는 방법을 선택할 수 있습니다.

다음 예에서는 이 이미지 이름 및 파일 경로 딕셔너리를 사용합니다.

```
let imagePaths = ["star": "/glyphs/star.png",
                  "portrait": "/images/content/portrait.jpg",
                  "spacer": "/images/shared/spacer.gif"]
```

키를 사용하여 딕셔너리의 값을 가져오면 `Optional` 값이 반환되므로 선호하는 방식으로 작성된 유형이 있습니다.

`imagePaths["star"]`는 `Optional<String>` 또는 `String?`