

OPTIONALS

Zeke Abuhoff

Lead iOS Instructor, General Assembly

OPTIONALS

LEARNING OBJECTIVES

- + Define an optional
- + Write an optional variable
- + Unwrap optionals
- + Use guard
- + Write an optional constant

OPTIONALS

MISSING VALUES

In an app with a user profile,
we may have a username.

```
username: String
```

```
"BlaiseIt"
```

What is the value of that variable
before the user has signed in?

```
username: String
```

```
?
```

OPTIONALS

MISSING VALUES

In an app with a user profile,
we may have a username.

username: String
"BlaiseIt"

What is the value of that variable
before the user has signed in?

username: String
?

And no, an empty
string is not right. That
makes it look like there
is a username but its
value is just "".

OPTIONALS

OPTIONALS

```
var optionalName: Type?
```

Optional variables are allowed to have no value.

```
var username: String?
```

We can use them to represent information we don't have yet or information that's been discarded.

OPTIONALS

OPTIONALS

```
var optionalName: Type?
```

Minimal values are NOT the same thing as not having a value.

`""` \neq `String?`

`0` \neq `Int?`

`False` \neq `Bool?`

Making a variable optional tells us that, in some situations, there is truly no value for that variable.

nil

OPTIONALS

UNWRAPPING OPTIONALS

When we refer to optional variables, first we have to unwrap them.

```
var username: String?  
if let presentUsername = username {  
    // Bring up profile screen using presentUsername  
} else {  
    // Bring up login screen  
}
```

OPTIONALS

UNWRAPPING OPTIONALS

If we're confident that there will be a value, we can force the unwrap.

```
var username: String?  
let presentUsername = username!
```

Forcing an unwrap, though, countermands the purpose of the optional. So avoid doing this.

OPTIONALS

UNWRAPPING OPTIONALS

Practice

- 1) Create an optional string called `liquidName` and an optional double called `boilingPoint`.
- 2) Set the value of `liquidName` to "Mercury" and set the value of `boilingPoint` to 674.1
- 3) Unwrap `liquidName` to a constant called `knownLiquidName`. If it has a value, print it. Otherwise, print "No liquid."
- 4) Force the unwrap of `boilingPoint` to a constant called `knownBoilingPoint`. Print it.

OPTIONALS

GUARD

Another way to unwrap optionals is through guard statements.

```
var favoriteFlavor: String?
func announceFavoriteFlavor() {
    guard let selectedFavoriteFlavor = favoriteFlavor else {
        return
    }
    print("\(selectedFavoriteFlavor) is my favorite!")
}
```

OPTIONALS

OPTIONAL CONSTANTS

A constant may be optional if the expression that defines it doesn't always return a value.

```
let phoneBook = ["Kelly" : "867-5309"]
```

```
let kellysNumber: String? = phoneBook["Kelly"]
```

```
let jakesNumber: String? = phoneBook["Jake"]
```

OPTIONALS

OPTIONALS

Practice

- 1) Write a function named `stringToInteger` that takes an optional string parameter and returns an optional integer. Use a guard statement to check the parameter, returning `nil` if the parameter has no value. After the guard, return an integer that was made from the string.

Hint: You can make an integer using the function `Int()`