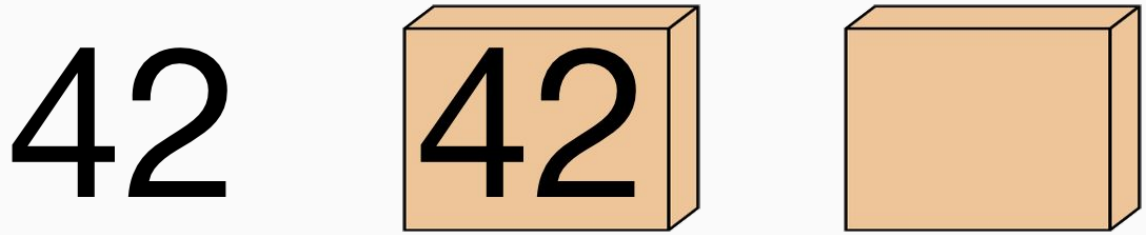
iOS Course

Fifth class - Optionals

Optionals

Optionals

Optional types, which handle the absence of a value. Optionals say either **“there is a value, and it equals x” or “there isn’t a value at all”**.



`Int` `Int?` `Int?`

Working with optionals

(safely)

Ways to handle optionals

- Optional Binding
- **Implicitly Unwrapped**
- Optional Chaining
- Nil coalescing operator

Classes creation and destruction

Initialization

Initialization is the process of preparing an instance of a class, structure, or enumeration for use.

This process involves setting an initial value for each stored property on that instance and performing any other setup or initialization that is required before the new instance is ready for use.

```
init() {
    // perform some initialization here
}
```

Deinitialization

A deinitializer is called immediately before a class instance is deallocated. You write deinitializers with the deinit keyword, similar to how initializers are written with the init keyword.

Deinitializers are only available on class types.

```
deinit {
    // perform the deinitialization
}
```

Swift automatically deallocates your instances when they are no longer needed, to free up resources.

Swift handles the memory management of instances through automatic reference counting (ARC)

Community

Resources

Podcasts

- Swift Over Coffe
- Swift by Sundell

Articles

- Swift by Sundell Basics
- Hacking with Swift

News

- Swift News (YouTube)
- iOS Dev Weekly (Newsletter)

Twitter

- Paul Hudson @twostraws
- Erica Sadun @ericasadun
- Norberto Ortigoza @hiphoox