Swift Bootcamp

try? xyz

- If xyz succeeds, returns an **optional** result.
- If it fails, nil returned
- Thus,

```
guard let x = try? xyz else {
}
```

means, if xyz succeeds, returned result is unwrapped and stored in x. If it fails, execute the block inside else.

- Basically, we don’t care what exception xyz throws, just whether it fails or succeeds

In iOS:

  - DateFormatters can be expensive on CPU.
  - Hence, make them static, so as to init them ONCE, globally (maybe in class)
  - Avoid init them locally in loops.
  - Weak property can automatically turn to nil, and free up the memory, once it’s not needed. Without weak, it’s easy to get a memory leak
  - TableView cells can be
    - Plain
    - Grouped
  - Grouped cells can pull in the empty cells to grey, to show that the cells are grouped and have contents underneath

UI updates should happen on Main thread. ALWAYS.

In CoreData:

- Before NSManagedObject subclass generation, CHANGE CODEGEN to Manual/None, in the entity properties
- File -> Project Settings -> DerivedData. DELETE IT!
- Convenience Init: Is a initializer that calls another initializer in the class
- Remember to save context after inserting data to core data