//Different types of serialization
//serial - user interface => one follow another - Main
//concurrently - non-user interface - GLOBAL

First in first out

priority (Quality of service)
- //High
- //default
- // low
- // Global Background

//Different ways of handling tasks
//sync
//async

UIApplication ← every UI view is an instance of UIApplication

main Queue (change interface only in main)

```
DispatchQueue.main.async{
    //labels ←

}
```

DispatchQueue.global(qos:.userInteracctive) ≈ main queue

DispatchQueue.global(qos:.userInitiated) ≈ high priority

DispatchQueue.global(qos:.default) = DispatchQ.global()

DispatchQueue.global(qos:.utility) ≈ low, things with progress bar, ex. download

DispatchQueue.global(qos:.background) ≈ really low, not visiable to user

↳ //io throttled

↳ input output throttled

DispatchQueue.global(qos:.unspecified)

example:

```
DispatchQueue.global(qos: .userInitiated).async{
        longTimeConsumingTask( )
        DispatchQueue.main.async{
              //tableView.reloadData()
        }
}
```

//strong reference cycles

optional

weak var ...   one can not be option

↑ weak   ↑ unowned

two things references each other

make it weak will make it optional

create delegate ⇒ use weak

[weak] ⇒ capture lists