

# iPhone App Dev

## Lesson 8

# Source

<https://github.com/bryanttang/iOS-Class-2015-6.git>

# Contact

[bryant.tang14mo@gmail.com](mailto:bryant.tang14mo@gmail.com)

# Summary

- MapKit (Review)
- Web view
- Concurrency Programming

# Mapkit

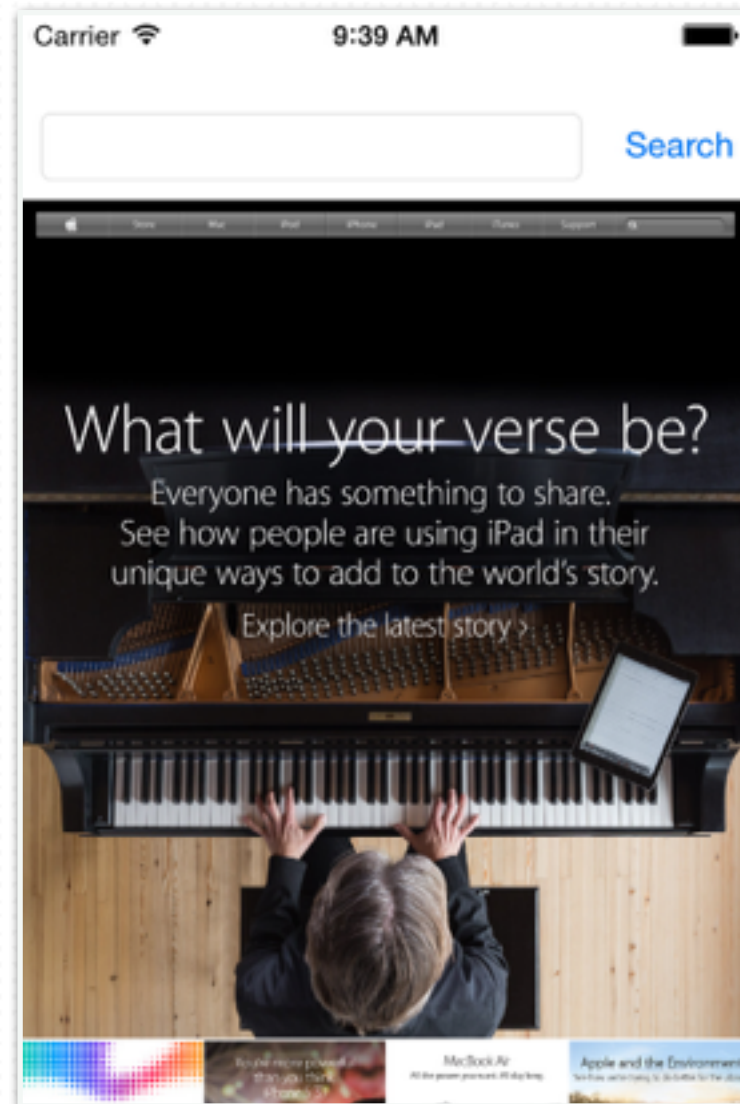
- Homework
- Question?

Web view

Demo

# Web view

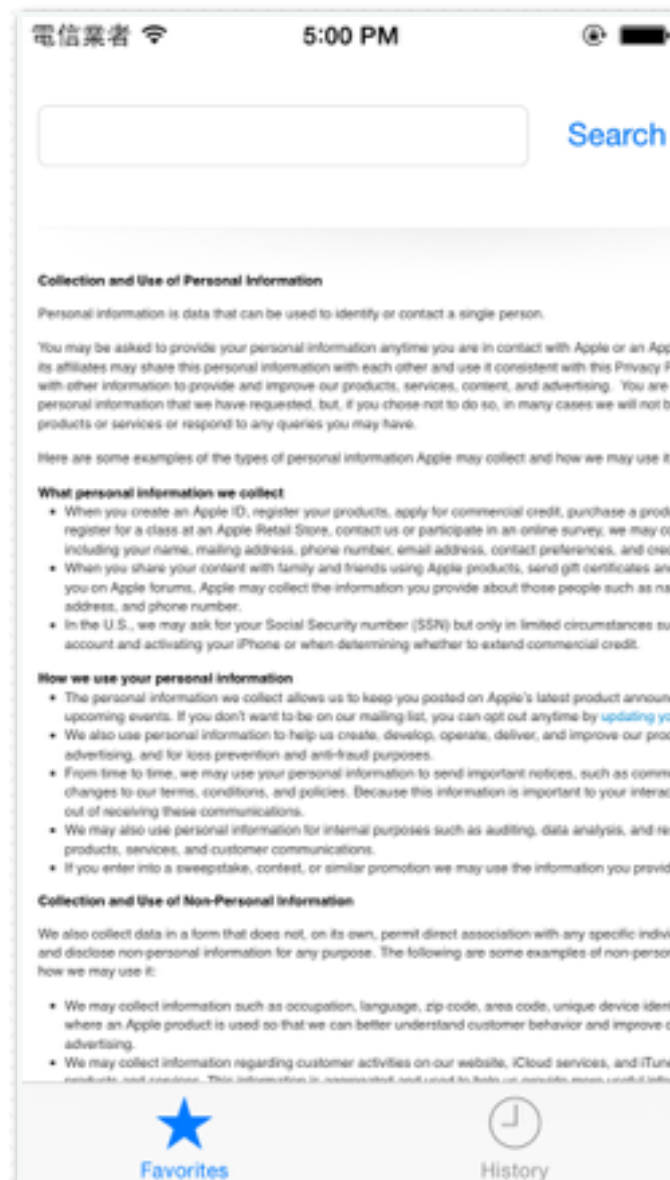
- To show web content just like a browser





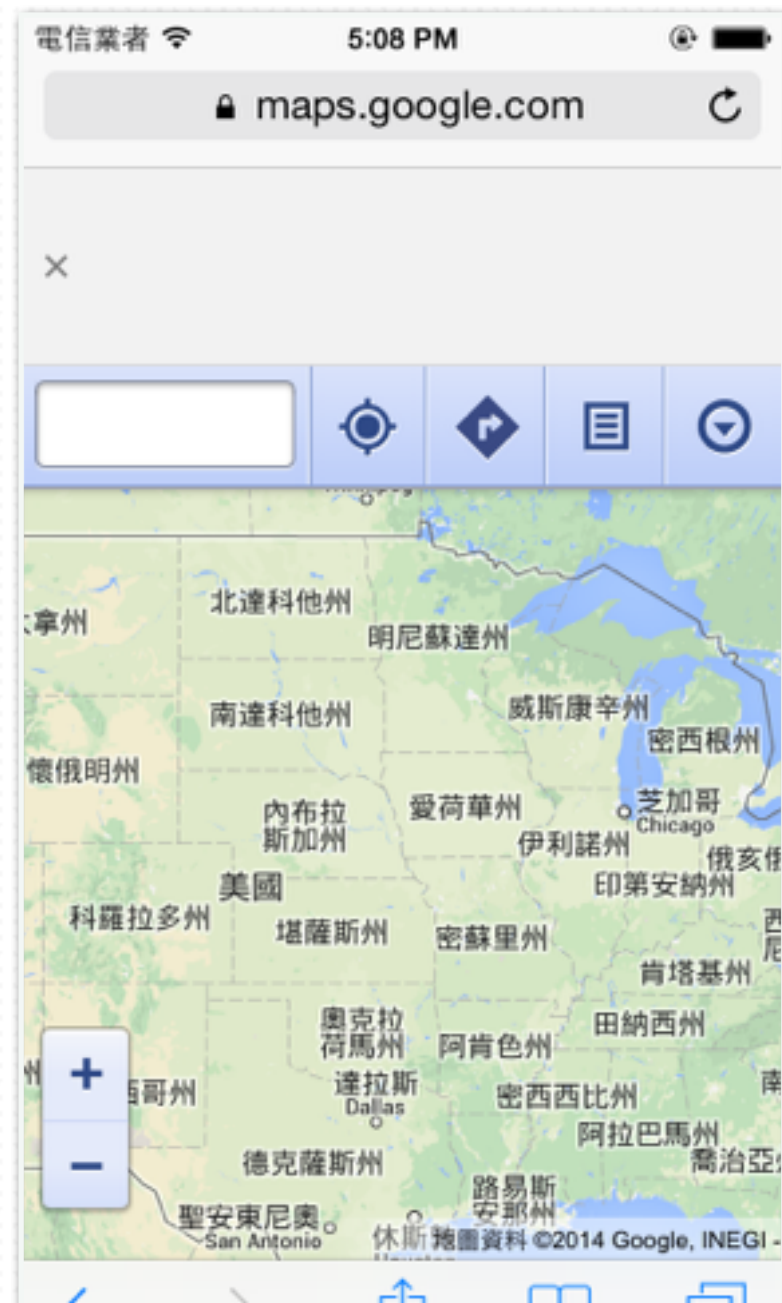
# Web view

- To show a page with static content



# Web view

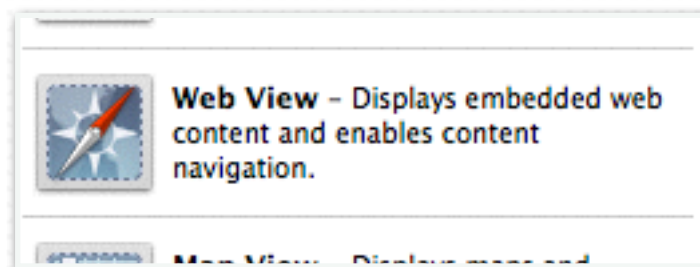
- To develop as a web app



# Web view

- Open a web view in project

Drag a web view from tool bar



# Web view

- Load a website in web view
  - create url string
  - create request with the url
  - load request in web view

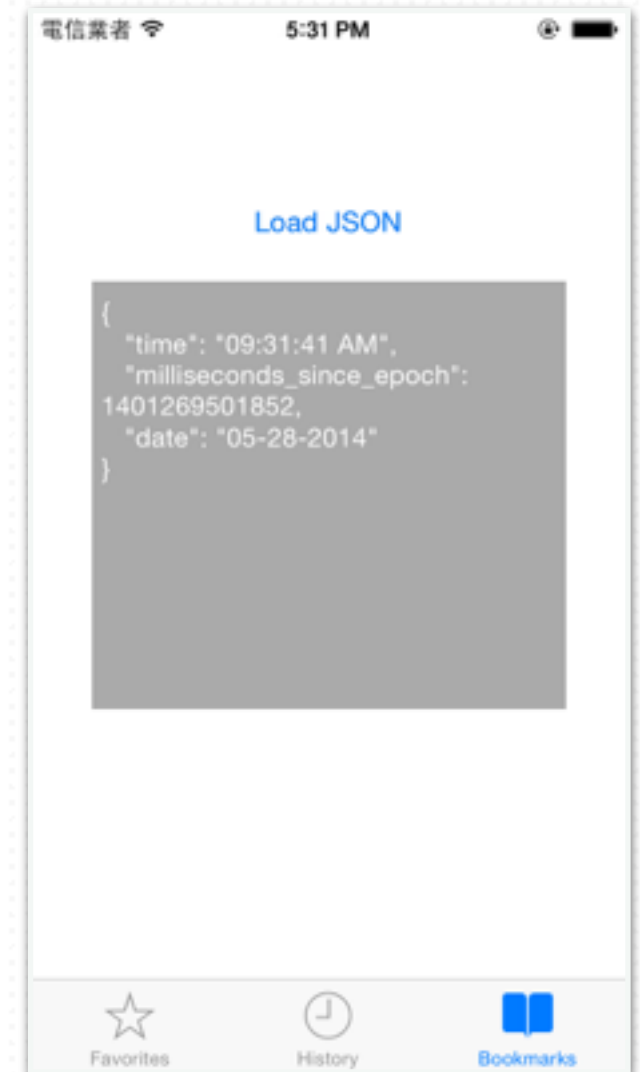
```
NSURL *url = [NSURL URLWithString:@"http://www.apple.com"];
NSURLRequest *request = [NSURLRequest requestWithURL:url];
[_myWebView loadRequest:request];
[_myWebView setScalesPageToFit:YES]; //fit to page
```



# Web view

- Load content by using NSString
  - create url string
  - Load url by using NSString

```
NSURL *url=[NSURL URLWithString:@"http://date.jsontest.com/"];  
  
NSString *json_str = [NSString stringWithContentsOfURL:url  
encoding:NSUTF8StringEncoding error:nil];
```





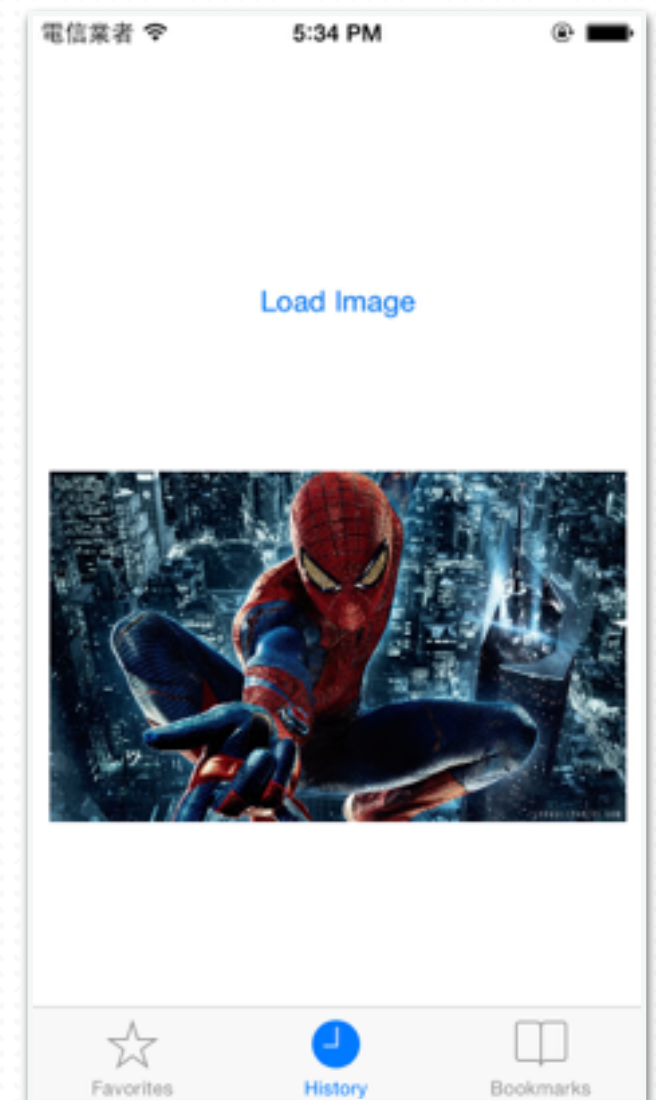
# Web view

- Load content by using NSString
  - create url string
  - Load url by using NSData
  - Convert NSData to UIImage

```
NSURL *url_image = [NSURL URLWithString:@"http://www.newyork-business.com/wp-content/uploads/2014/02/amazing_spider_man_latest-1920x1200.jpg"];
```

```
NSData *image_data = [NSData dataWithContentsOfURL:url_image options:NSDataReadingUncached error:nil];
```

```
_imageView.image = [UIImage imageWithData:image_data];
```



# Concurrency Programming

Demo



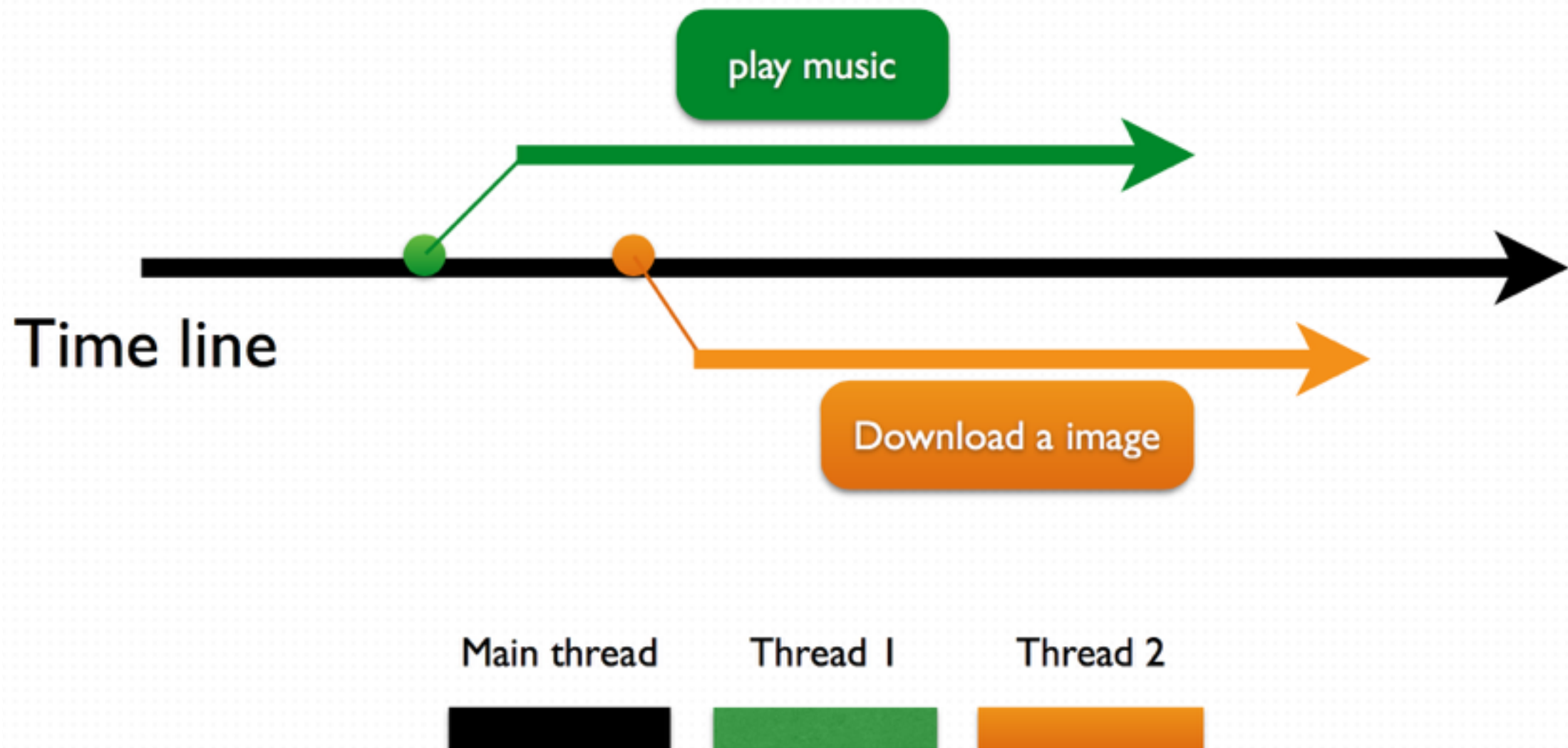
# Concurrency Programming

Two ways to implement

- Use thread
- Use Grand Central Dispatch (GCD)

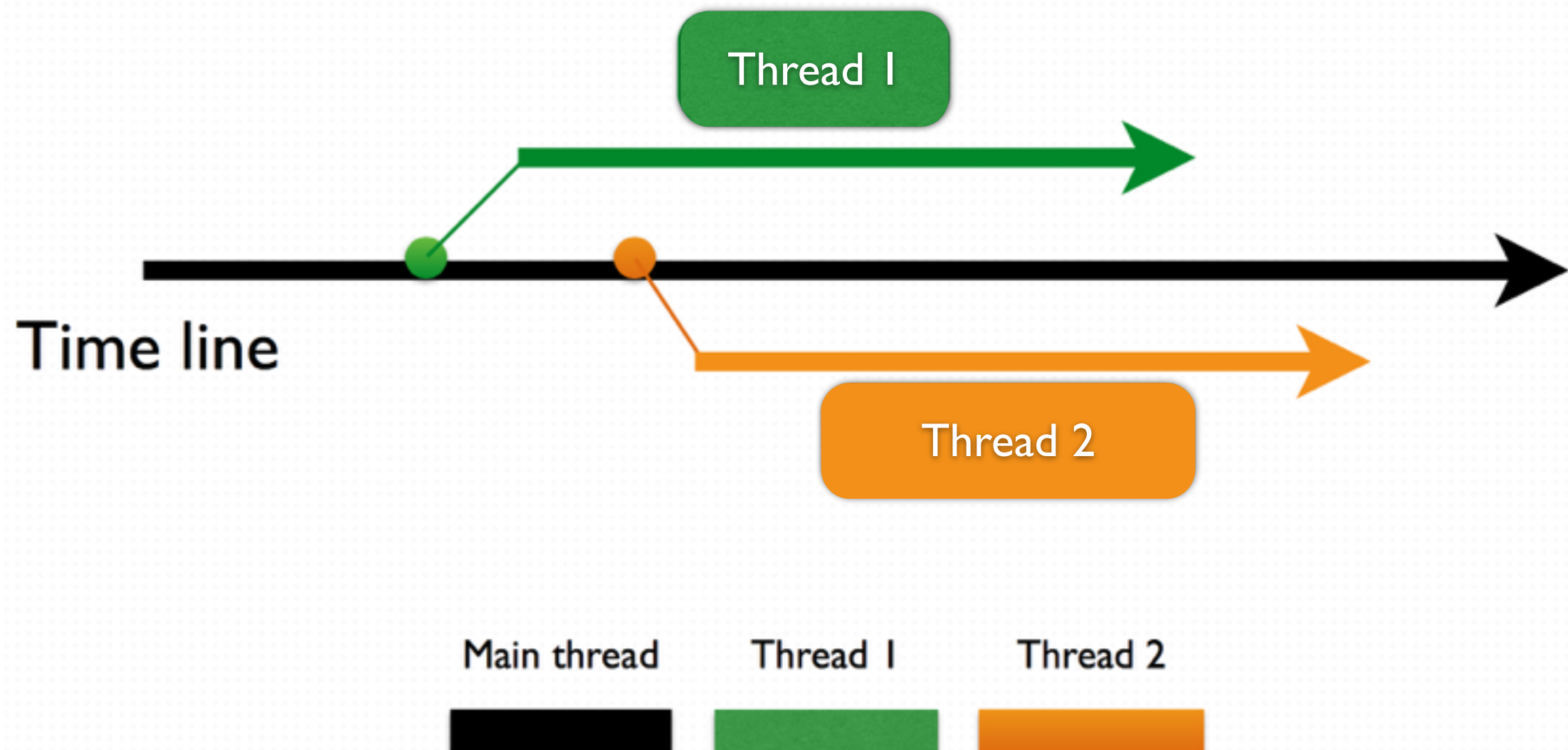
# Concurrency Programming

- Spawn a thread



# Concurrency Programming

- Spawn a thread



# Concurrency Programming

- Use `NSThread` class method

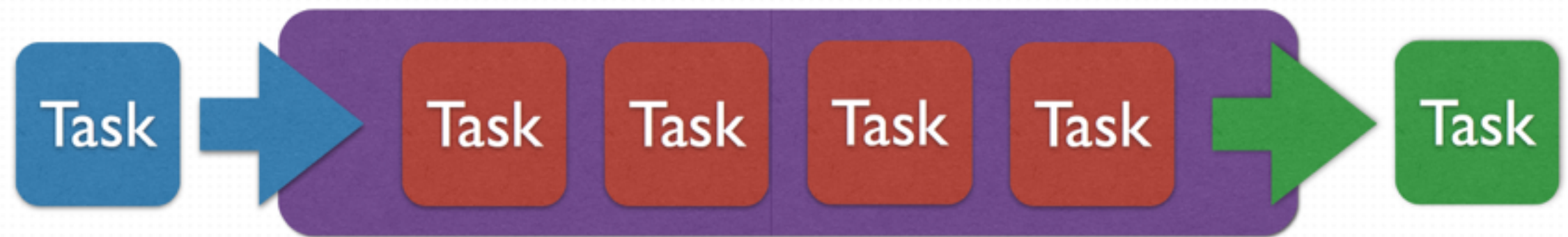
```
[NSThread detachNewThreadSelector:@selector(myThreadMainMethod;) toTarget:self withObject:nil];
```

- Create `NSThread` Object

```
NSThread* myThread = [[NSThread alloc] initWithTarget:self  
                      selector:@selector(myThreadMainMethod;) object:nil];  
[myThread start]; // Actually create the thread
```

# Concurrency Programming

Use Grand Central Dispatch (GCD)



Dispatch queue

# Concurrency Programming

Two ways to implement

- ✓ Serial
- ✓ Concurrent
- ✓ Main dispatch queue

# Concurrency Programming

Two ways to implement

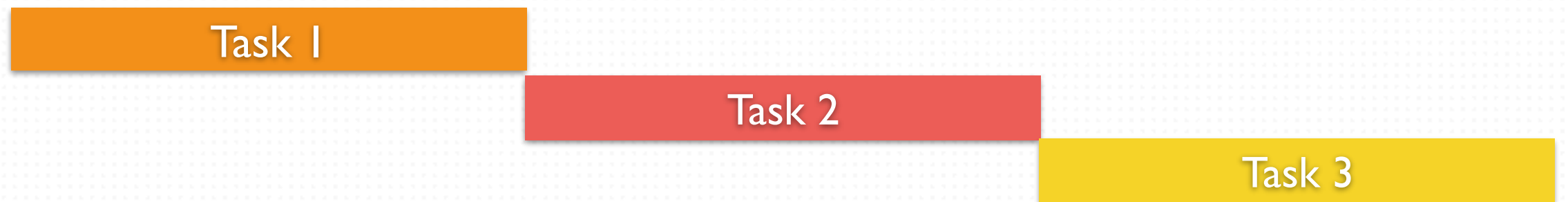
- ✓ Serial
- ✓ Concurrent
- ✓ Main dispatch queue



# Concurrency Programming

## Serial Queue

- ✓ Private dispatch queues
- ✓ Execute one task at a time
- ✓ In the order

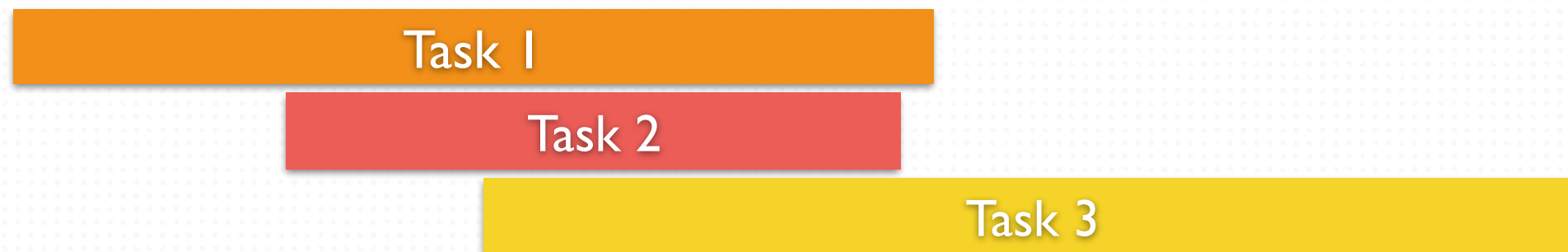




# Concurrency Programming

## Concurrent Queue

- ✓ Global dispatch queue
- ✓ Execute one or more tasks concurrently
- ✓ In the order



# Concurrency Programming

- Use **Serial** queue

```
dispatch_queue_t processQueue = dispatch_queue_create("PROCESS_QUEUE", NULL);
dispatch_async(processQueue, ^{
    //Do somethin
})
```

- Use **concurrent** queue

```
dispatch_queue_t processQueue =
dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0);
dispatch_async(processQueue, ^{
    //Do something
})
```

# Concurrency Programming

- Use **Main dispatch** queue

```
dispatch_queue_t mainQueue = dispatch_get_main_queue("PROCESS_QUEUE", NULL);
dispatch_async(mainQueue, ^{
    //Do somethin
})
```