

iOS App 開發基礎班

Closure Callbacks & Delegates

蔡智強 Denny Tsai

denny@hpd.io

<https://iosdev.hpd.io>



用途

- 用來處理一些非同步事件使用
- 因為有些事件不是一觸發馬上就會有反應，所以必須以非同步的方式來處理

Delegate

- 將接收事件的物件設定為實際處理物件的 delegate 屬性
- 當實際處理物件遇到事件發生時，便會呼叫 delegate 物件中的特定方法。
- Delegate 通常會搭配一個 protocol 來規範 delegate 物件需要實作的方法。

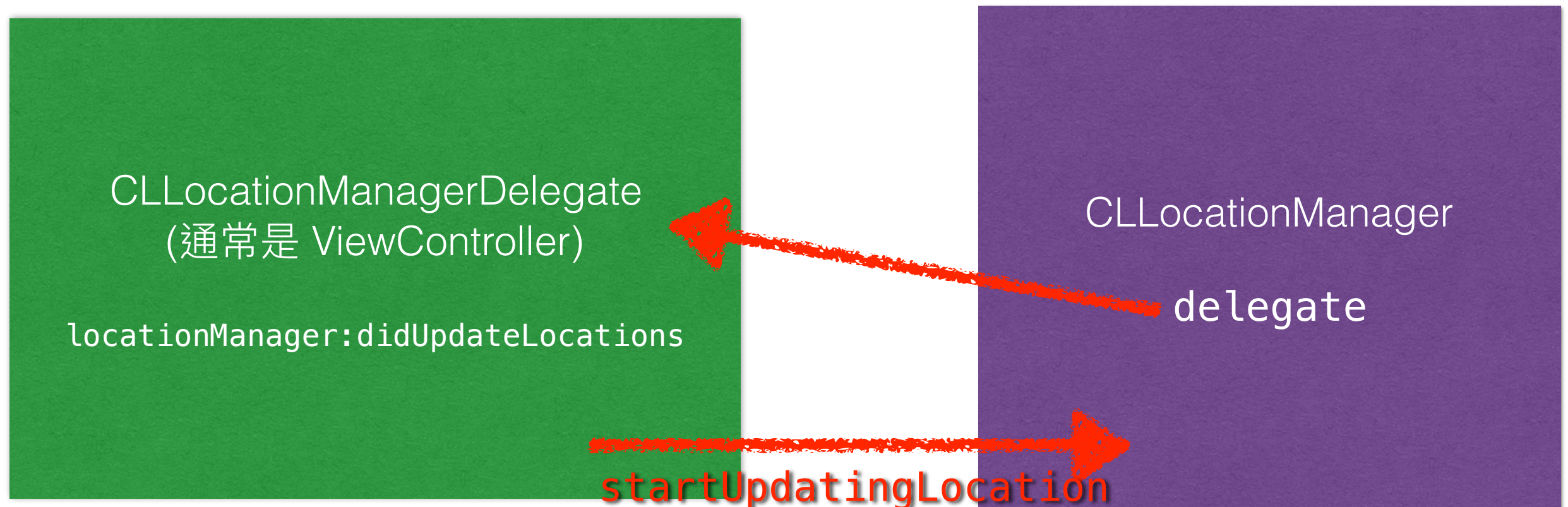
Delegate 範例: CLLocationManager

- CLLocationManager 可以取得目前裝置的所在地理位置、面向方位... 等等
- 將一個物件實作 CLLocationManagerDelegate，便可以接收各種 CLLocationManager 發出來的事件。

Delegate 範例: CLLocationManager

viewController

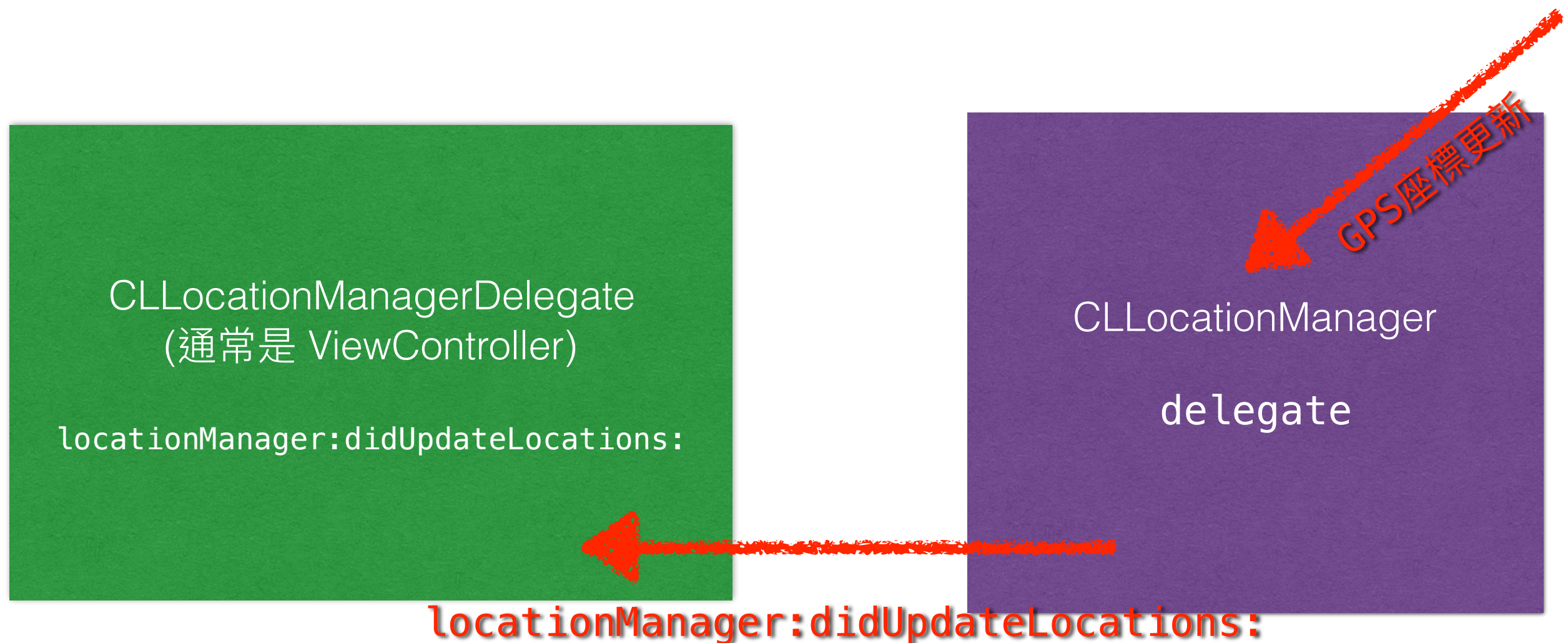
```
let locationManager = CLLocationManager()  
locationManager.delegate = self  
locationManager.startUpdatingLocation()
```



Delegate 範例: CLLocationManager

locationManager

```
self.delegate.locationManager(self, didUpdateLocations: locations)
```

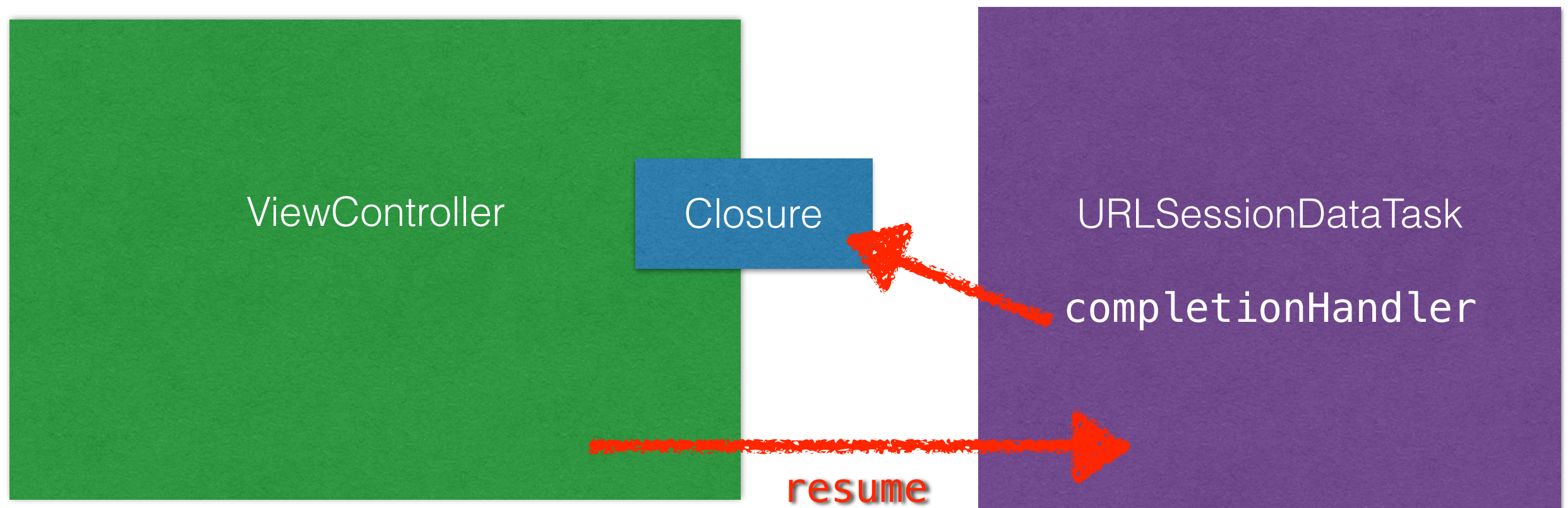


Closure Callback

- 將一段可執行的 closure 傳遞給實際處理的物件
- 比起參照整個 delegate 物件，這裡只參照了一個 closure

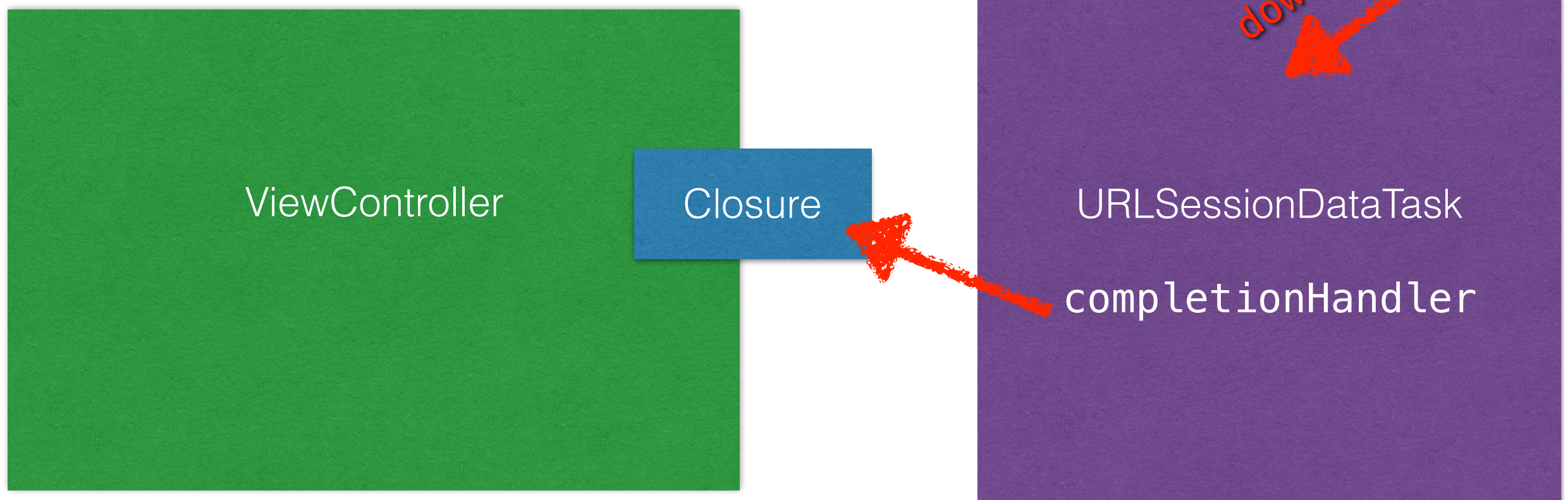
範例: URLSessionDataTask

```
let dataTask = session.dataTaskWithURL(...,  
completionHandler: { data, response, error in  
    // ...  
})  
dataTask.resume()
```



範例: URLSessionDataTask

```
let dataTask = session.dataTaskWithURL(...,  
completionHandler: { data, response, error in  
    // ...  
})  
dataTask.resume()
```



Closure 優勢

- Closure 跟 delegate 做法比起來程式碼更容易理解
- 當事件觸發時會發生的事情直接寫在執行動作之後
- 如URLSessionDataTask把下載完成的事情直接寫在執行下載的方法之後
- 可以直接使用 closure 所在位置的 context (包含區域變數)