deep dive into **Protocols**

~ Ritesh Gupta

(Swift Delhi Meetup, Chapter 3)

`Protocol-Oriented Programming in Swift`

👉 https://developer.apple.com/videos/play/wwdc2015/408/

WWDC - 2015

`Protocol and Value Oriented Programming in UIKit Apps`

👉 https://developer.apple.com/videos/play/wwdc2016/419/

WWDC - 2016

```
protocol LoadableView {

  var contentView: UIView { get }
}
```

```
protocol LoadableView {

  var contentView: UIView { get }
}
```

```
protocol LoadableView {

  var contentView: Self { get }
}
```

```swift
protocol LoadableView {

  var contentView: UIView { get }
}
```

```swift
protocol LoadableView {

  var contentView: Self { get }
}
```

```swift
protocol LoadableView {

  associatedtype ContentView

  var contentView: ContentView { get }
}
```

playground demos..

protocol -> 😎

protocol -> 😎

protocol extension -> 😎 😎

protocol -> 😎

protocol extension -> 😎 😎

protocol constrained extension -> 😎 😎 😎

self

(refers to the `object`)

vs

Self

(refers to the `type` that conforms to a protocol)

👉 https://krakendev.io/blog/generic-protocols-and-their-shortcomings

👉 http://austinzheng.com/2015/09/29/swift-generics-pt-2/

👉 http://www.russbishop.net/swift-associated-types protocol

is **not** a

silver bullet.. 😅

👉 http://chris.eidhof.nl/post/protocol-oriented-programming/