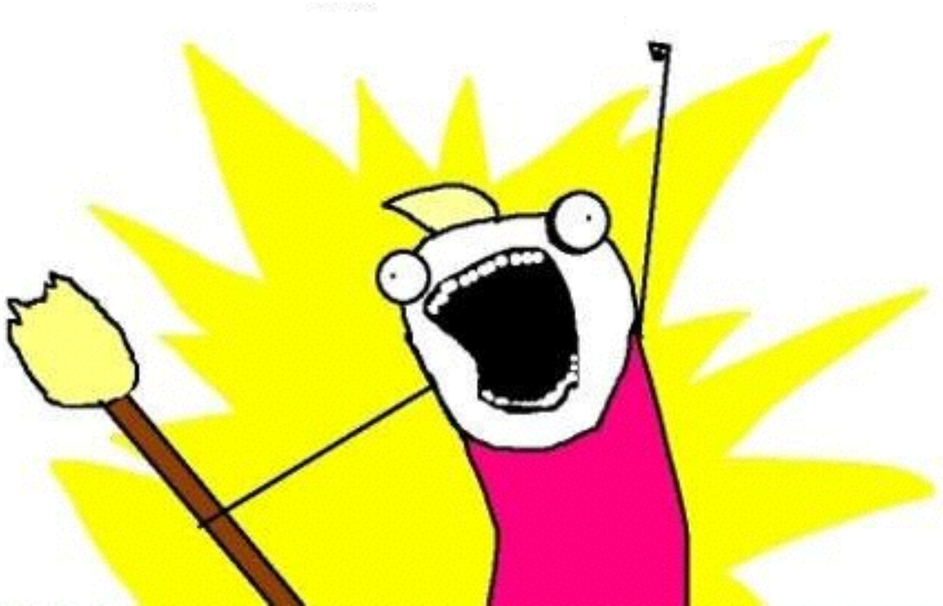
ADVANCED SWIFT 3

Part 2: Protocol Oriented Programming

Protocol All The Things!

- Protocols describe types and their capabilities
- Protocols add interface and functionality



Shape

```
class Shape {
  var origin: CGPoint
  var size: CGSize
  var color: UIColor
  var fillColor: UIColor
  var strokeWidth: CGFloat

  func draw(on context: CGContext) {
    fatalError("override \(#function)")
  }
  func area() -> CGFloat {
    return size.width * size.height
  }
  …
 }
```

Circle

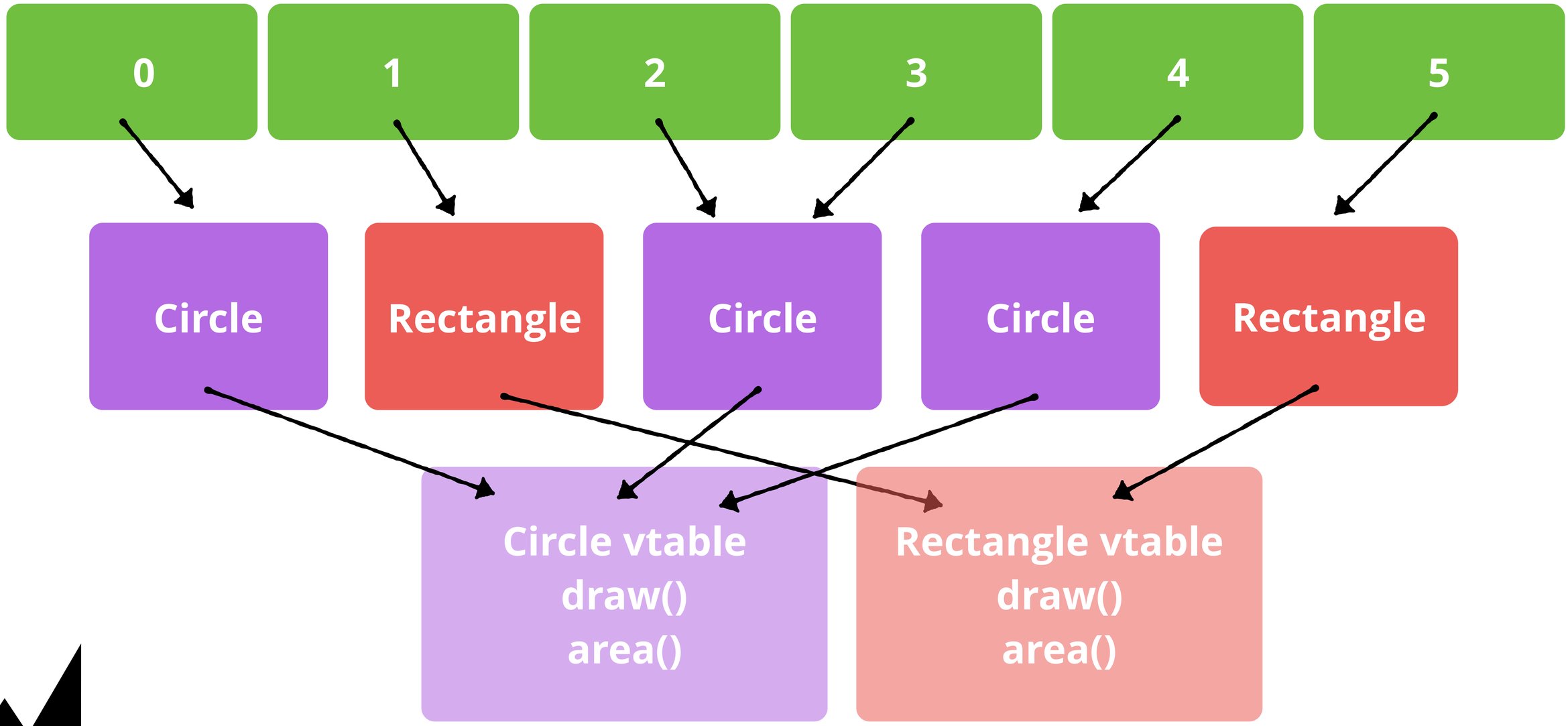
```
class Circle: Shape {
  var diameter: CGFloat {
    return size.width
  }

  var radius: CGFloat {
    return size.width / 2
  }

  override func area() -> CGFloat {
    return .pi * radius * radius
  }

  override func draw(on context: CGContext) {
    …
  }
  …
}
```

Array of the Base Class V-table



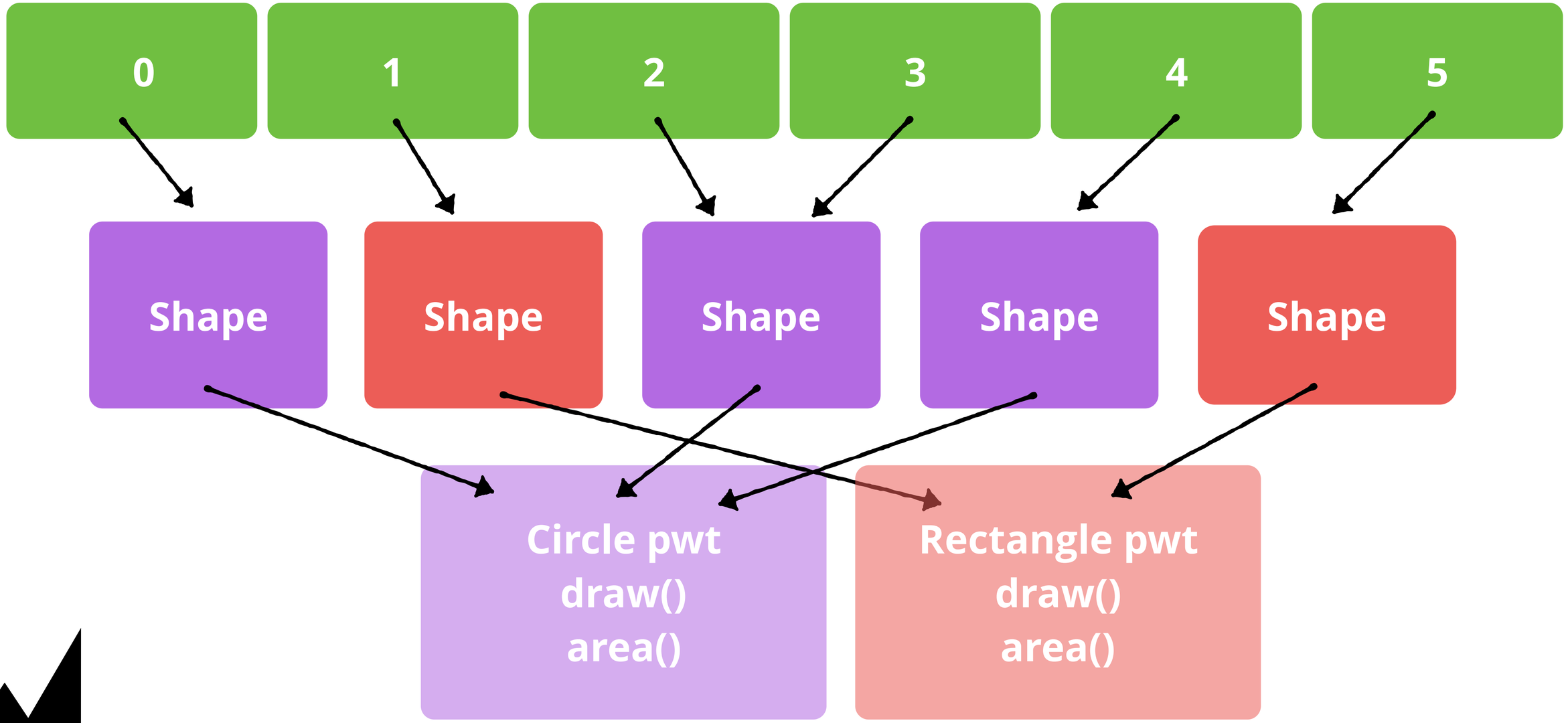
Protocol

```
protocol Shape {
  var origin: CGPoint { get }
  var size: CGSize { get }
  var color: UIColor { get }
  var fillColor: UIColor { get }
  var strokeWidth: CGFloat { get }

  func draw(on context: CGContext)
  func area() -> CGFloat
}
```

```
extension Shape {
 func area() -> CGFloat {
    return size.width * size.height
  }
}
```

Existential Types - Protocol Witness Tables



Refactored Protocols

```
protocol Drawable {
  func draw(on context: CGContext)
}

protocol Geometry {
  var size: CGSize { get }
  func area() -> CGFloat
}

extension Geometry {
  func area() -> CGFloat { return size.width * size.height }
}
```

No Ducks Allowed

- Swift is not duck typed
- You must explicitly conform to protocols
- Soft requirements are not checked by the compiler

No Override Protection

```
protocol Geometry {
  var size: CGSize { get }
  func boundingBoxArea() -> CGFloat
}

extension Geometry {
  func boundingBoxArea() -> CGFloat { return size.width * size.height }
}

extension Circle {
  func area() -> CGFloat { return radius * radius * .pi }
}
```

Challenge Time

- Start with the Challenge page in the starter playground
- Build a Line type that conforms to Drawable
- Make it animate by applying a velocity to each end of the line