Single Anchor

Core API

```
view.anchors.centerX.equal(container.anchors.centerX + 10)
view.anchors.left.greaterThanOrEqual(container.anchors.left)
view.anchors.right.lessThanOrEqual(container.anchors.right)
```

Semantic API

```
// Edge: Pin
view.anchors.left.pin()
view.anchors.left.pin(inset: 10)
view.anchors.left.pin(to: container.layoutMarginsGuide)

// Edge: Spacing
view.anchors.top.spacing(10, to: view.anchors.bottom)
view.anchors.bottom.spacing(10, to: view.anchors.top)

// Center: Align
view.anchors.centerX.align(offset: 10)

// Dimension: Clamp
view.anchors.width.clamp(to: 10...40)
```

Operators

```
view.anchors.top + 10 / view.anchors.top.offsetting(by: 10)
view.anchors.height * 2 / view.anchors.height.multiplied(by: 2)
```

Constraints

```
// Activates all constraints at the same time
// Allows you to change the constraint priority
Constraints {
    view.anchors.centerX.equal(container.anchors.centerX)
    let c = view.anchors.height.equal(view.anchors.width * 2)
    c.priority = UILayoutPriority(rawValue: 249)
}

// Provides a convenient access to anchors
Constraints(for: title, subtitle) { title, subtitle in
    title.bottom.spacing(10, to: subtitle.top)
}

// Create constraints without activation
let constraints = Constraints(activate: false) {
    view.anchors.height.equal(10)
}
constraints.activate()
```

Multiple Anchors

Core API

```
view.anchors.edges.equal(container)
view.anchors.edges.equal(container, insets: 20)
let insets = EdgeInsets(top: 8, left: 16, bottom: 8, right: 16)
view.anchors.edges.equal(container, insets: insets)
view.anchors.edges.lessThanOrEqual(container, insets: 20)

view.anchors.size.equal(container)
view.anchors.size.equal(CGSize(width: 40, height: 40))
view.anchors.size.lessThanOrEqual(container)

view.anchors.center.equal(container)
```

Semantic API

```
// Center: Align
view.anchors.center.align()
view.anchors.center.align(with: container)

// Edges: Pin
view.anchors.edges.pin()
view.anchors.edges.pin(insets: 20)
let insets = EdgeInsets(top: 8, left: 16, bottom: 8, right: 16)
view.anchors.edges.pin(insets: insets)
view.anchors.edges.pin(axis: .horizontal)
view.anchors.edges.pin(to: container, insets: 20,
  axis: .horizontal, alignment: .center)
```

Anchors

```
// Edges
view.anchors.top
view.anchors.bottom
view.anchors.left
view.anchors.right
view.anchors.leading
view.anchors.trailing

// Center
view.anchors.centerX
view.anchors.centerY

// Baseline
view.anchors.firstBaseline
view.anchors.lastBaseline

// Dimension
view.anchors.width
view.anchors.height

// Collections
view.anchors.edges
view.anchors.size
view.anchors.center
```

Alignment

```
public struct Alignmment {
    public enum Horizontal {
        case fill, center, leading, trailing
    }
    public enum Vertical {
        case fill, center, top, bottom
    }
}

// There are multiple built-in alignments
```

topLeading	top	topTrailing
leading	fill, center	trailing
bottomLeading	bottom	bottomTrailing

```
view.anchors.edges
  .pin(insets: 20, alignment: .center)
```