

### NSLayoutXAxisAnchor [Horizontal]

**NSLayoutXAxisAnchor**

- \* centerXAnchor
- \* leadingAnchor & trailingAnchor
- \* leftAnchor & rightAnchor

**Swift**

```
myView.centerXAnchor.constraint(equalTo: view.centerXAnchor).isActive = true
```

**Objective-C**

```
[self.myView.centerXAnchor constraintEqualToAnchor:self.view.centerXAnchor].active = YES;
```

### Top and Bottom Layout guides

**UILayoutSupport**

topLayoutGuide bottomLayoutGuide

- \* bottomAnchor
- \* topAnchor
- \* heightAnchor

**// Swift**

```
myView.topAnchor.constraint(equalTo: topLayoutGuide.bottomAnchor, constant: 8.0).isActive = true
```

**// Objective-C**

```
[self.stackView.topAnchor constraintEqualToLayoutGuide.bottomAnchor constant: 8.0].active = YES;
```

### NSLayoutYAxisAnchor [Vertical]

**NSLayoutYAxisAnchor**

- \* centerYAnchor
- \* bottomAnchor & topAnchor
- \* firstBaselineAnchor & lastBaselineAnchor

**Swift**

```
myView.bottomAnchor.constraint(equalTo: view.topAnchor, constant: 8).isActive = true
```

**Objective-C**

```
self.myView.bottomAnchor.constraintEqualToAnchor:self.view.topAnchor constant: 8.0].active = YES
```

### Example 2

Example to make the height of a view twice the height of another view with a multiplier:

**// Swift**

```
myView.heightAnchor.constraint(equalTo: otherView.heightAnchor, multiplier: 2.0).isActive = true
```

**// Objective-C**

```
[self.myView.heightAnchor constraintEqualToAnchor:otherView.heightAnchor multiplier:2.0].active = YES
```

### NSLayoutDimension [Size-Based Constrains]

**NSLayoutDimension**

- \* heightAnchor
- \* widthAnchor

**// Swift**

```
myView.widthAnchor.constraint(equalTo: Constant: 50.0).isActive = true
```

**// Objective-C**

```
[self.myView.widthAnchor constraintEqualToConstant: 50.0].active = YES;
```

### View Margins

**\*layoutMarginGuide & readableContentGuide**

**// Swift**

```
let margins = view.layoutMarginsGuide
myView.leadingAnchor.constraint(equalTo: marginLeadingAnchor).isActive = true
```

**// Objective-C**

```
UILayoutGuide *margins = self.view.layoutMarginsGuide; [self.myView.leadingAnchor constraintEqualToAnchor:margins.leadingAnchor].active = YES;
```