Getting the Safe Area

```
var safeAreaInsets: UIEdgeInsets
```

The insets that you use to determine the safe area for this view.

```
var safeAreaLayoutGuide: UILayoutGuide
```

The layout guide representing the portion of your view that is unobscured by bars and other content.

```
func safeAreaInsetsDidChange()
```

Called when the safe area of the view changes.

```
var insetsLayoutMarginsFromSafeArea: Bool
```

A Boolean value indicating whether the view's layout margins are updated automatically to reflect the safe area.

Managing the View’s Constraints

```
var constraints: [NSLayoutConstraint]
```

The constraints held by the view.

```
func addConstraint(NSLayoutConstraint)
```

Adds a constraint on the layout of the receiving view or its subviews.

```
func addConstraints([NSLayoutConstraint])
```

Adds multiple constraints on the layout of the receiving view or its subviews.

```
func removeConstraint(NSLayoutConstraint)
```

Removes the specified constraint from the view.

```
func removeConstraints([NSLayoutConstraint])
```

Removes the specified constraints from the view.

Working with Layout Guides

```
func addLayoutGuide(UILayoutGuide)
```

Adds the specified layout guide to the view.

```
var layoutGuides: [UILayoutGuide]
```

The array of layout guide objects owned by this view.

```
var layoutMarginsGuide: UILayoutGuide
```

A layout guide representing the view’s margins.

```
var readableContentGuide: UILayoutGuide
```

A layout guide representing an area with a readable width within the view.

```
func removeLayoutGuide(UILayoutGuide)
```

Removes the specified layout guide from the view.

Measuring in Auto Layout

```
func systemLayoutSizeFitting(CGSize) -> CGSize
```

Returns the optimal size of the view based on its current constraints.

```
func systemLayoutSizeFitting(CGSize, withHorizontalFitting
Priority: UILayoutPriority, verticalFittingPriority: UILayout
Priority) -> CGSize
```

Returns the optimal size of the view based on its constraints and the specified fitting priorities.

```
var intrinsicContentSize: CGSize
```

The natural size for the receiving view, considering only properties of the view itself.

```
func invalidateIntrinsicContentSize()
```

Invalidates the view’s intrinsic content size.

```
func contentCompressionResistancePriority(for: NSLayoutConstraint
.Axis) -> UILayoutPriority
```

Returns the priority with which a view resists being made smaller than its intrinsic size.

```
func setContentCompressionResistancePriority(UILayoutPriority,
for: NSLayoutConstraint.Axis)
```

Sets the priority with which a view resists being made smaller than its intrinsic size.

```
func contentHuggingPriority(for: NSLayoutConstraint.Axis) ->
UILayoutPriority
```

Returns the priority with which a view resists being made larger than its intrinsic size.

```
func setContentHuggingPriority(UILayoutPriority, for: NSLayout
Constraint.Axis)
```

Sets the priority with which a view resists being made larger than its intrinsic size.

Creating Constraints Using Layout Anchors

`func alignmentRect(forFrame: CGRect) -> CGRect`

Returns the view’s alignment rectangle for a given frame.

`func frame(forAlignmentRect: CGRect) -> CGRect`

Returns the view’s frame for a given alignment rectangle.

`var alignmentRectInsets: UIEdgeInsets`

The insets from the view’s frame that define its alignment rectangle.

~~`func forBaselineLayout() -> UIView`~~

Returns a view used to satisfy baseline constraints.

Deprecated

`var forFirstBaselineLayout: UIView`

Returns a view used to satisfy first baseline constraints.

`var forLastBaselineLayout: UIView`

Returns a view used to satisfy last baseline constraints.

Creating Constraints Using Layout Anchors

Attach Auto Layout constraints to one of the view's anchors.

`var bottomAnchor: NSLayoutYAxisAnchor`

A layout anchor representing the bottom edge of the view’s frame.

`var centerXAnchor: NSLayoutXAxisAnchor`

A layout anchor representing the horizontal center of the view’s frame.

`var centerYAnchor: NSLayoutYAxisAnchor`

A layout anchor representing the vertical center of the view’s frame.

`var firstBaselineAnchor: NSLayoutYAxisAnchor`

A layout anchor representing the baseline for the topmost line of text in the view.

`var heightAnchor: NSLayoutDimension`

A layout anchor representing the height of the view’s frame.

`var lastBaselineAnchor: NSLayoutYAxisAnchor`

A layout anchor representing the baseline for the bottommost line of text in the view.

`var leadingAnchor: NSLayoutXAxisAnchor`

A layout anchor representing the leading edge of the view’s frame.

`var leftAnchor: NSLayoutXAxisAnchor`

A layout anchor representing the left edge of the view’s frame.

`var rightAnchor: NSLayoutXAxisAnchor`

A layout anchor representing the right edge of the view’s frame.

`var topAnchor: NSLayoutYAxisAnchor`

A layout anchor representing the top edge of the view’s frame.

`var trailingAnchor: NSLayoutXAxisAnchor`

A layout anchor representing the trailing edge of the view’s frame.

`var widthAnchor: NSLayoutDimension`

A layout anchor representing the width of the view’s frame.

Configuring Content Margins

`var directionalLayoutMargins: NSDirectionalEdgeInsets`

The default spacing to use when laying out content in a view, taking into account the current language direction.

```
var layoutMargins: UIEdgeInsets
```

The default spacing to use when laying out content in the view.

```
var preservesSuperviewLayoutMargins: Bool
```

A Boolean value indicating whether the current view also respects the margins of its superview.

```
func layoutMarginsDidChange()
```

Notifies the view that the layout margins changed.