

# Intro to Auto Layout

Visual Format Language I

# The Problem

---

```
let views = ["leftButton": leftButton, "rightButton": rightButton]
let metrics = ["edgeGap": 20]
NSLayoutConstraint.activateConstraints(NSLayoutConstraint
    .constraintsWithVisualFormat("H: |[leftButton]-[rightButton(==leftButton)]-|",
        options: .AlignAllBottom, metrics: nil, views: views))
NSLayoutConstraint.activateConstraints(NSLayoutConstraint
    .constraintsWithVisualFormat("V: [rightButton]-(edgeGap)-|",
        options: [], metrics: metrics, views: views))
```

# Visual Format Language

“ <orientation>: ? <superview><connection> ? [ <view> ]  
( <connection><view> ) \* <connection><superview> ? ”

```
H: |-[find]-[findNext(100@500)]-[findField(>=20)]-|
```

**H:,V:** - horizontal, vertical

| - superview

- - standard space

<=,>= - less than, greater than

@ - priority

== - equal width

? - Not Required

\* - May appear 0 or more times

# Options

---

```
NSLayoutConstraint.constraintsWithVisualFormat("V:|-20-[label]-[textField]", options: [], metrics: metrics, views: views)
```

- ⚙ AlignAllLeft
- ⚙ AlignAllRight
- ⚙ AlignAllTop
- ⚙ AlignAllBottom
- ⚙ AlignAllLeading
- ⚙ AlignAllTrailing
- ⚙ AlignAllCenterX
- ⚙ AlignAllCenterY
- ⚙ AlignAllBaseline
- ⚙ AlignAllLastBaseline
- ⚙ AlignAllFirstBaseline
- ⚙ DirectionLeadingToTrailing
- ⚙ DirectionLeftToRight
- ⚙ DirectionRightToLeft



# Metrics & Views

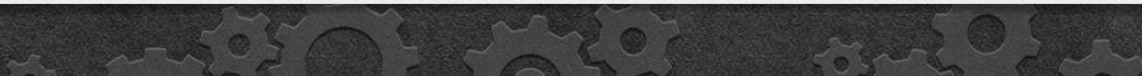
```
let metrics = ["topGap": 20, "edgeGap": 10]
let views = ["leftButton": leftButton, "rightButton": rightButton]

NSLayoutConstraint.constraintsWithVisualFormat("V: |-topGap-[rightButton]",
options: NSLayoutFormatOptions(rawValue: 0), metrics: metrics, views: views)

NSLayoutConstraint.constraintsWithVisualFormat("H: |-edgeGap-[leftButton]",
options: NSLayoutFormatOptions(rawValue: 0), metrics: metrics, views: views)
```

# Demo

---



# Challenge

