# AutoLayout

- There is a bunch different ways to make a constraints
  "Reset to suggested constraints"
  "Ctrl-drag" to edges or other views
  "Pin Menu"
  by writing a "code"(won't be covered)

- Constraints are inspectable
  Can be inspected by going to "Size Inspector"
  Can be inspected directly from a "storyboard"
  Can be inspected from a "Document outline"

- Mastering an AutoLayout takes an experience
  There is almost no structured way of building an AutoLayout
  Just master a lot of UI and learn by warnings (errors)

# AutoLayout

- But all previous instructions are NOT enough

  Simple auto layout cannot handle when your geometry changes dramatically

  You need relay your views to fit properly on the screen

- Calculator

  For example, you have 20 buttons

  It would be good to have 4 columns and 5 rows in portrait mode

  In landscape 5 columns and 4 rows

  Early constraining instructions won't be able to handle this

- "Size classes"

  The size class can tell you what state your width and height are in

  It's either "compact" or "regular"

  Good job "Apple"

# Size classes

- **iPhone**

  All iPhones are <u>compact</u> in width and <u>regular</u> in height in <u>portrait</u>

  In landscape are <u>compact</u> in both directions(width, height)

- **iPhone Plus**

  iPhone Plus is also compact in width and regular in height in portrait

  But in landscape, it's compact in height and regular in width

- **iPad**

  Always regular width and height in both mode

  Depending on an environment the width can be compact(master of a SplitView)

# Size classes

|  | Compact width | Regular width |
|---|---|---|
| Compact height | iPhones (none Plus) in landscape | iPhone Plus in landscape |
| Regular height | iPhones in portrait or SplitView master | iPads in portrait or landscape |

# AutoLayout

◉ What can we do based on Size classes

We can vary many properties of UIView

For example, fonts, color, background color, isHidden etc..

Basically all properties that have a plus sign on the left

◉ <u>Constraints</u> can be controlled by "Size classes"

by doing this, you can arrange your UI very differently in different situations

All this can be done by a storyboard

# Demo