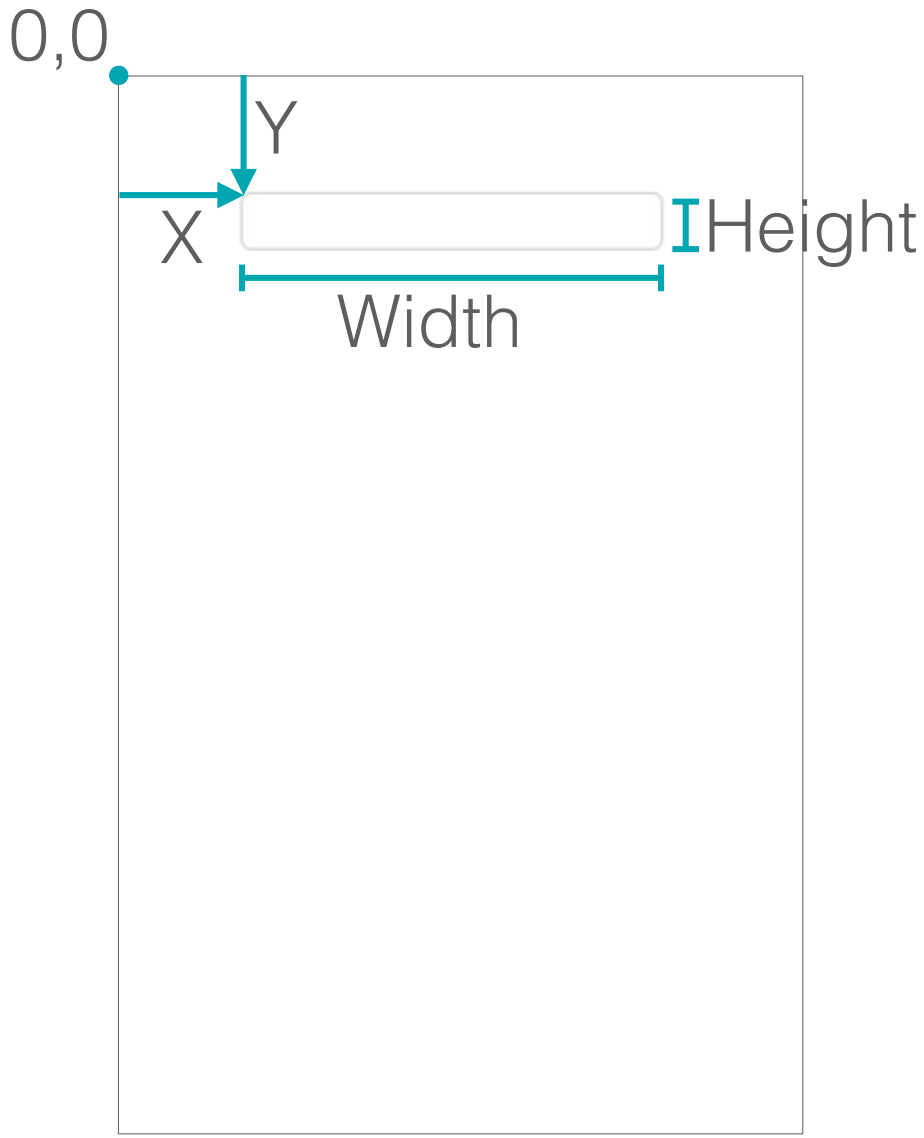
AutoLayout

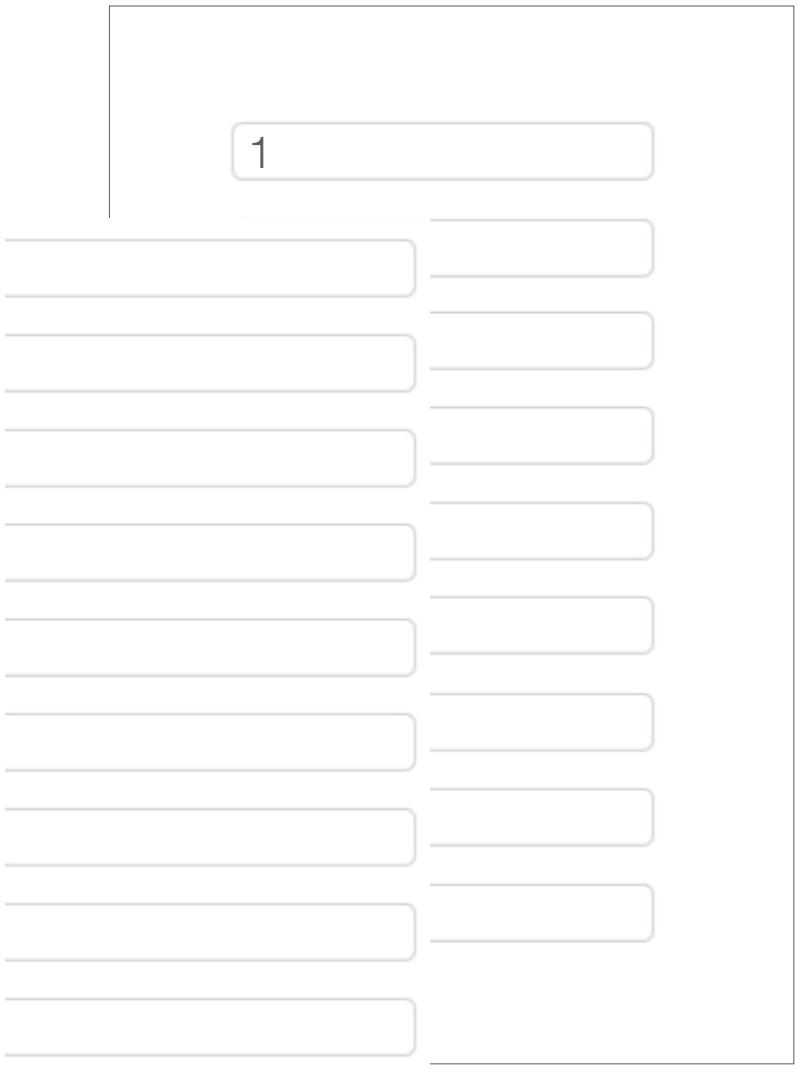
Layout Basics

Position & size UI elements on the screen



Handling Size Changes

But things change…first, it was rotation

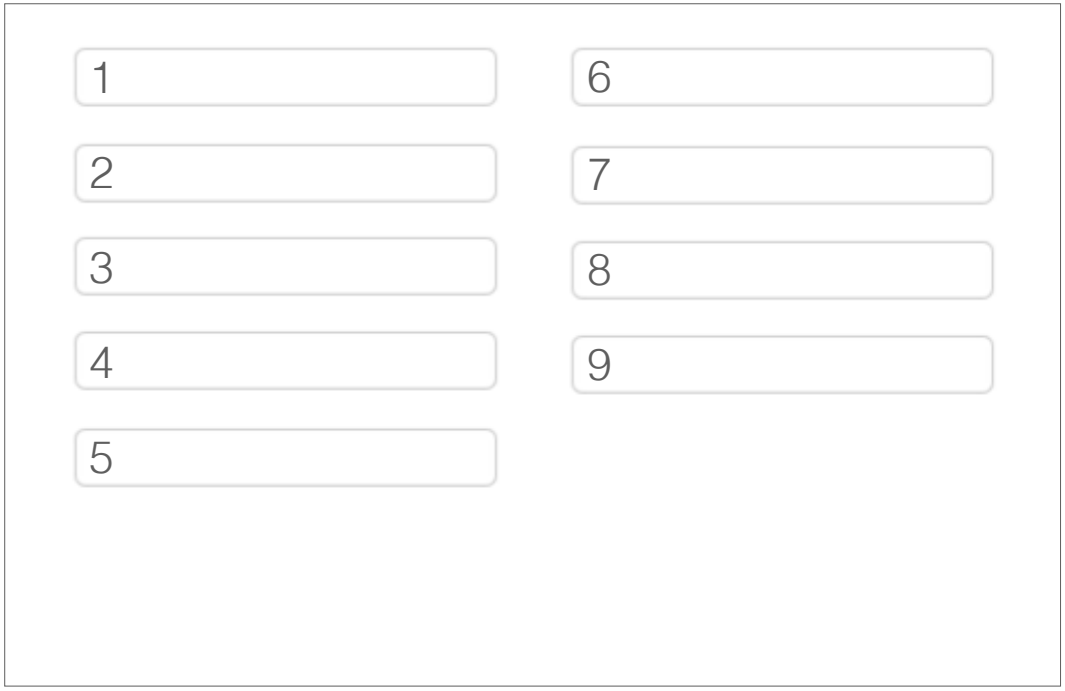


Handling Size Changes

Forces scrolling, which we want to avoid

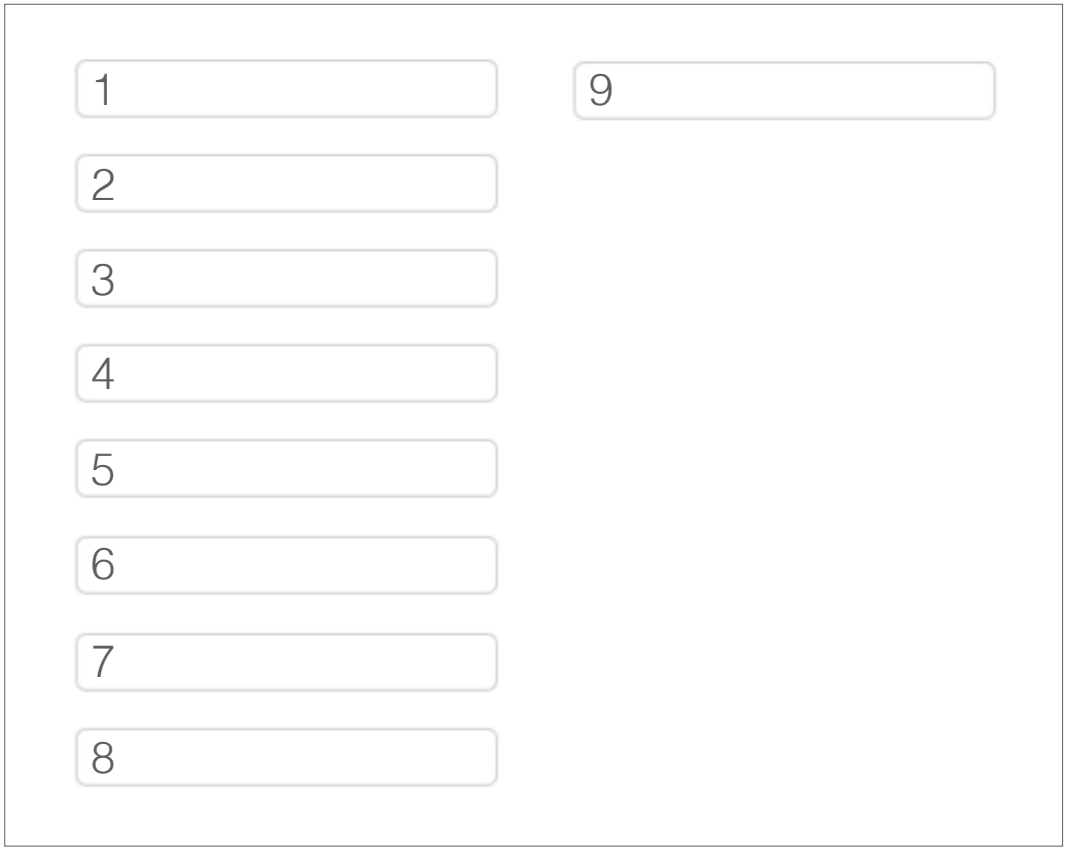
Handling Size Changes

And what if we want a different layout...

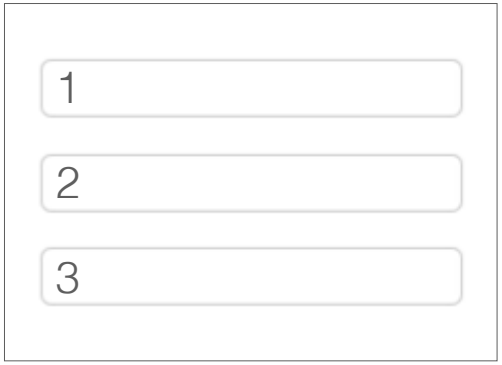


Handling Size Changes

Or what if we have a completely different size or shape...

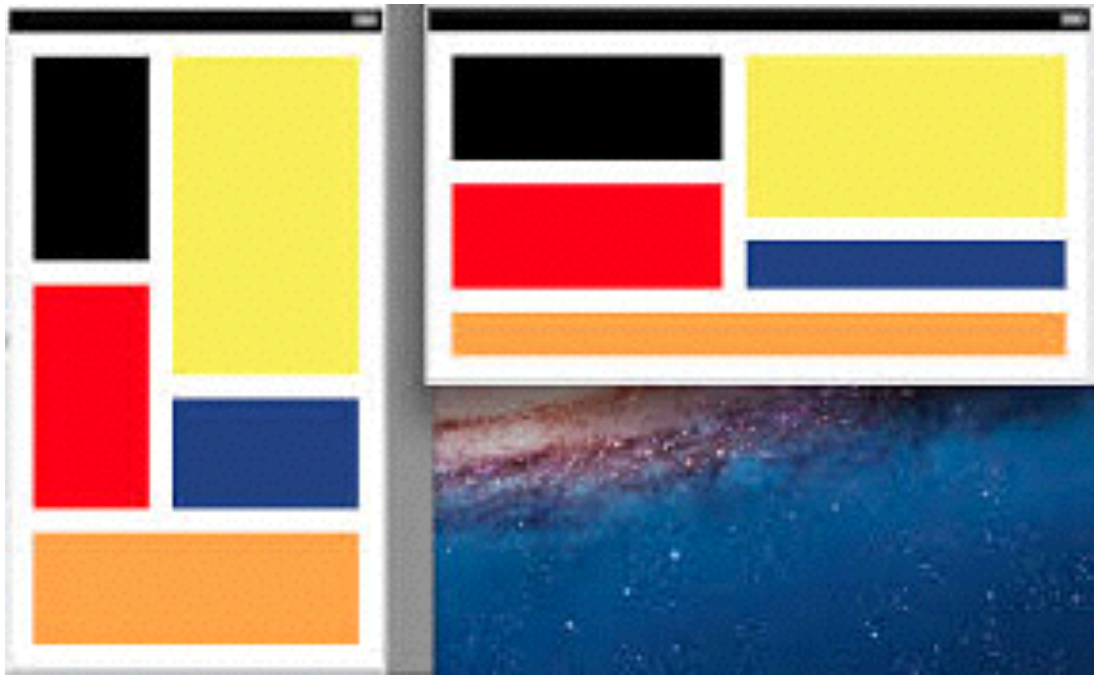


OR



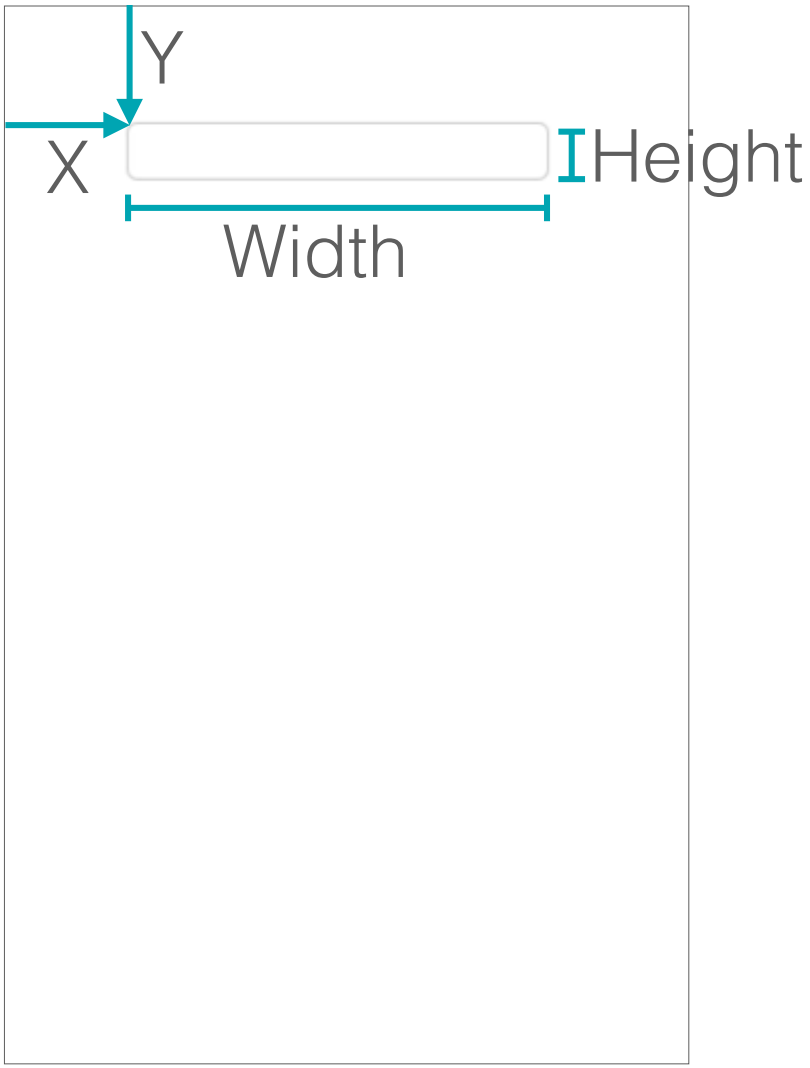
AutoLayout

- How things resize and even reposition in different orientations
- Uses a set of rules (called constraints) for layout
- Generally hated, but getting better and has become essential



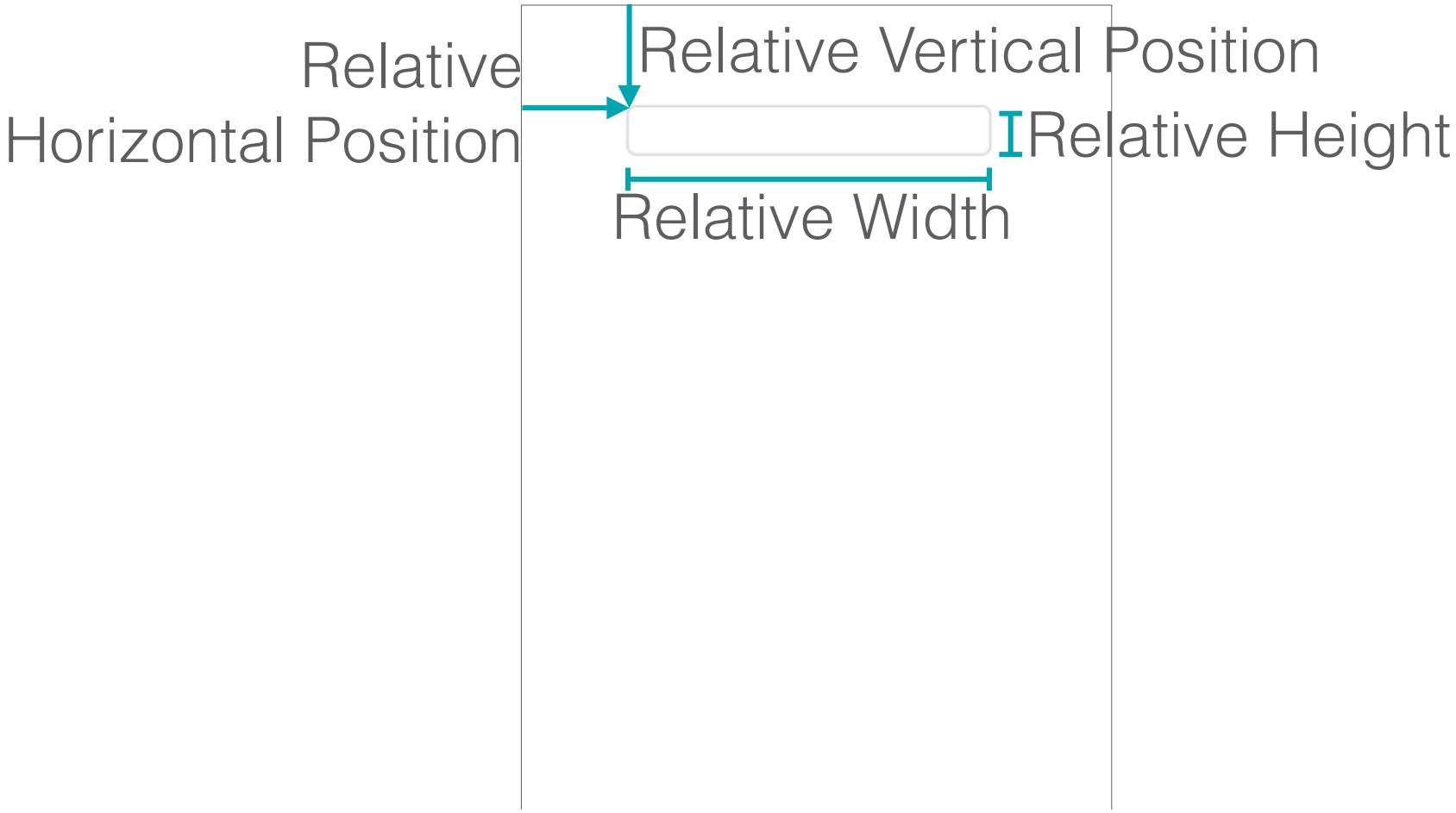
Layout Basics

Still need to know...



AutoLayout Basics

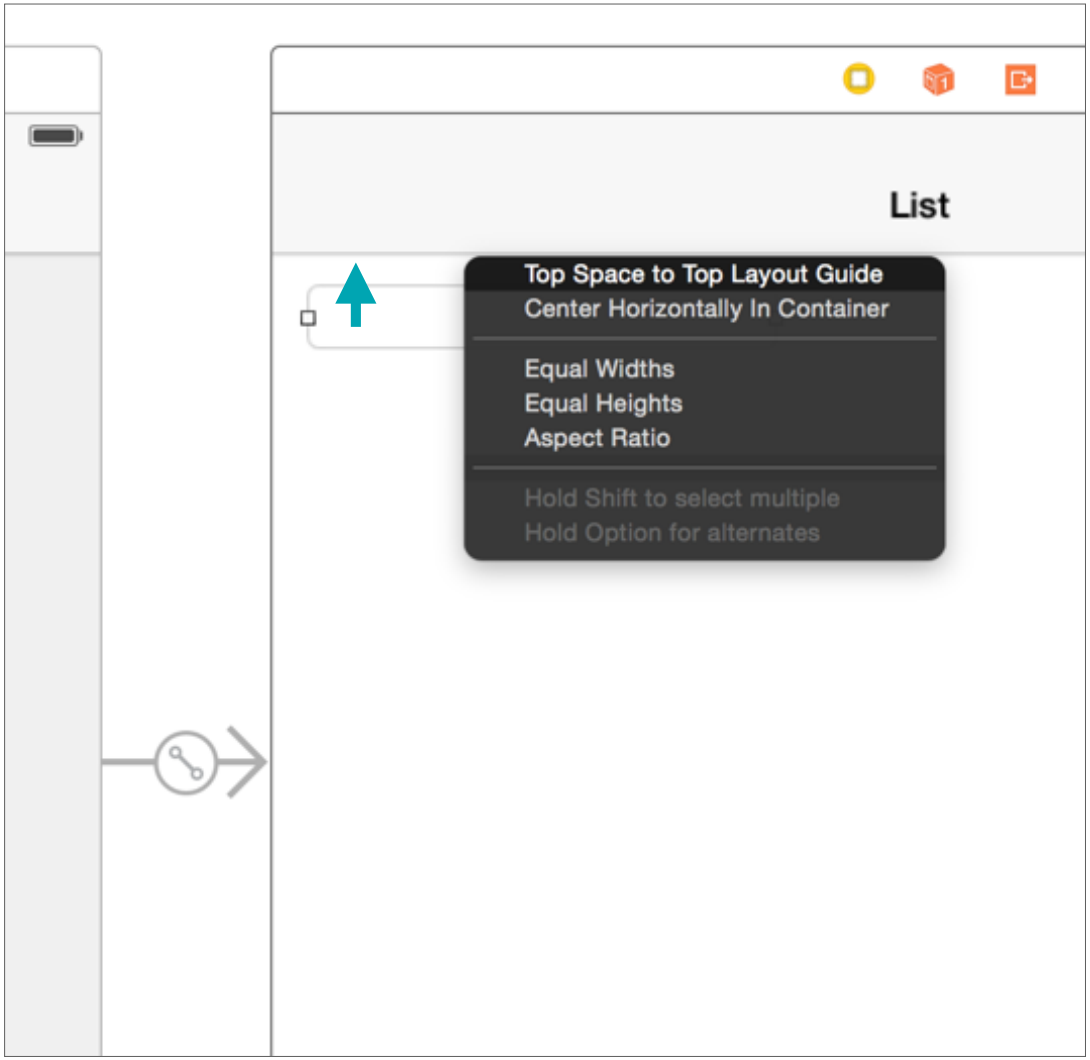
With a slight change...



Everything is positioned relative to something else

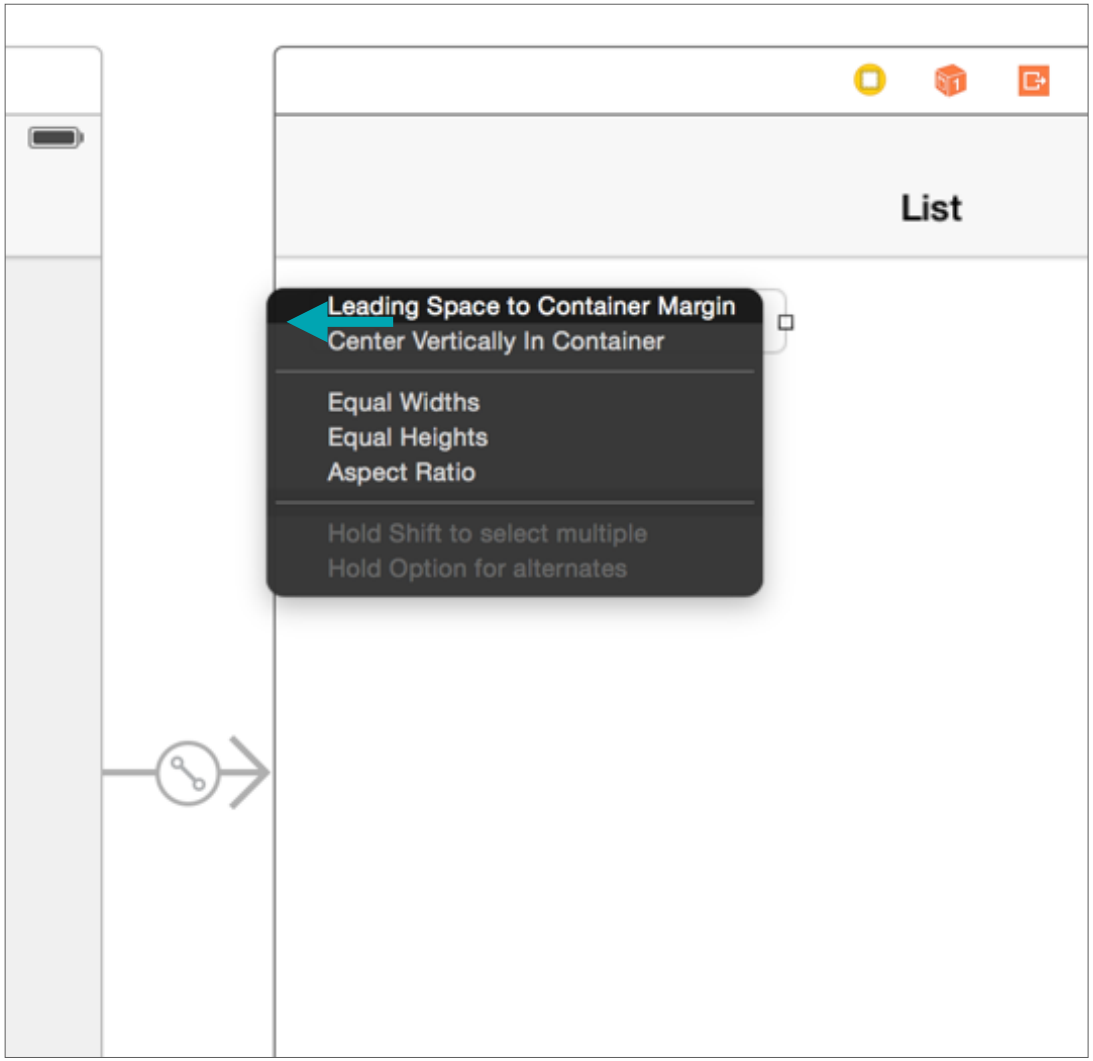
AutoLayout Basics

CTRL-drag from object to top margin or to another object to get Top Space (i.e. Y)



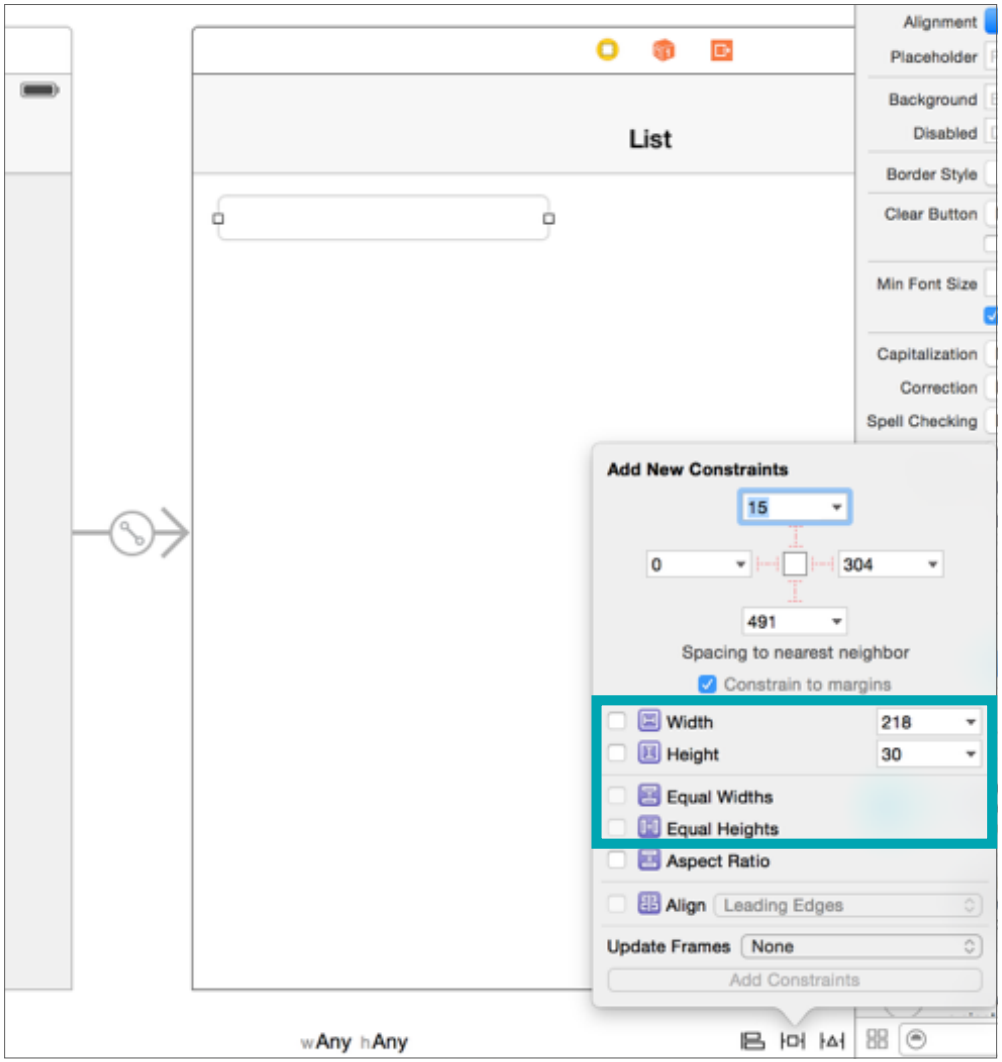
AutoLayout Basics

CTRL-drag from object to left margin or to another object to get Leading Space (i.e. X)



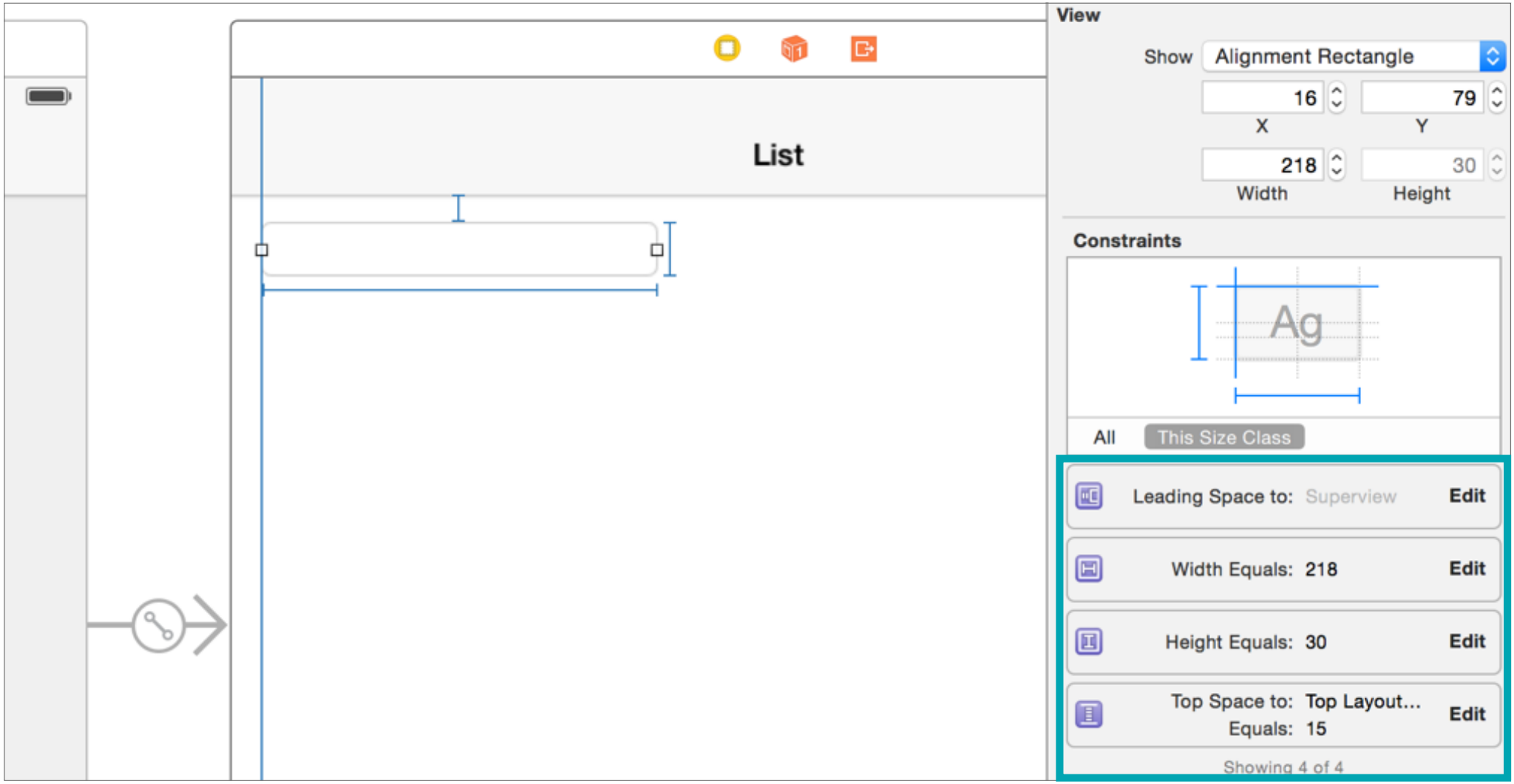
AutoLayout Basics

Set object's specific width & height or make them relative to other objects (equal or percent)



AutoLayout Basics

These create rules for layout (or Constraints, more specifically)



AutoLayout Basics

- Every UI object needs some way to define it's horizontal & vertical placement plus size & width
- These are often derived rather than explicit:
  - Relative to left & right margin gives width
  - Relative to top & bottom margin gives height
  - Equal Width to another control with a width gives width
  - Percentage Width to another control with a width gives width
  - Aspect Ratio with a height gives width

AutoLayout with Size Classes

Size Classes allow different layouts for different sizes of devices or the same device when rotated, using the same (or different) controls