

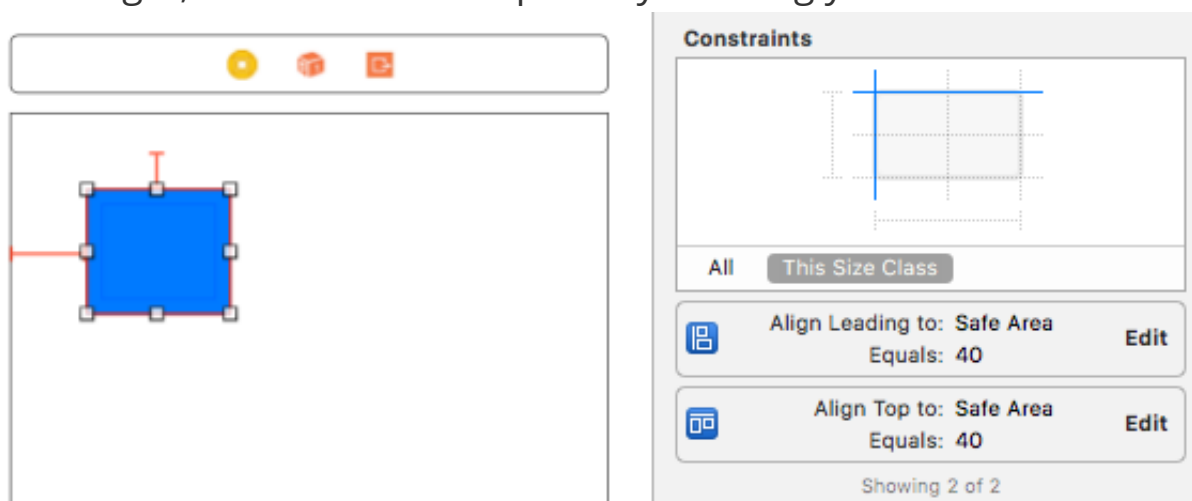
5 - What is intrinsic content size and the usefulness of it

Previously, we have explained why conflicting constraint happen and how to solve them. In this post we will look into what is intrinsic content size and the importance of it.

As a recap, in order for Auto Layout to render the position and size of a view correctly, Auto Layout engine **must know** or **able to calculate** the :

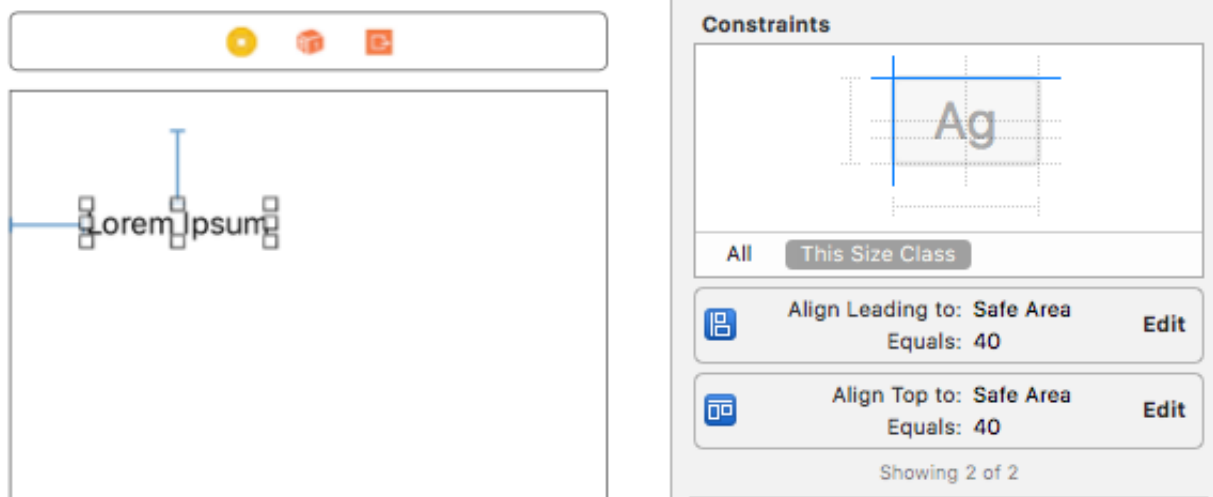
1. **x position** of a view
2. **y position** of a view
3. **width** of a view
4. **height** of a view

If you place a view without enough constraint to calculate its width and height, Xcode would complain by showing you red lines like this :

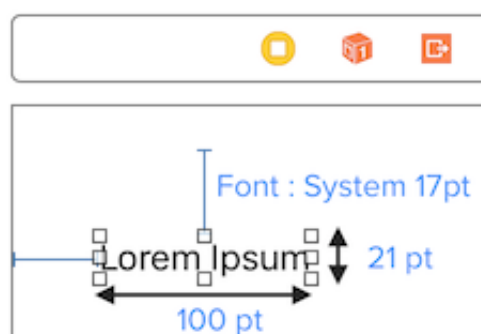


The blue rectangle view has a leading (40 pt) and top (40 pt) constraints, meaning its x position and y position is known, but Auto Layout Engine doesn't know its width and height.

Now lets place a label and define the same leading and top constraints :



Wait... Xcode doesn't complain even though the size is not defined and it show us blue line, what happened? It turns out that Auto Layout Engine is able to calculate the label size (width and height) based on its content (text of the label) and the font type/size of the label. For the label using System font type, font size 17pt and contain text "Lorem Ipsum", its intrinsic size is as follows :



Intrinsic content size means the size of an UI element derived based on its content.

UIView doesn't have intrinsic content size hence you have to manually define constraints that enable Auto Layout Engine to calculate its size.

Label / UILabel has intrinsic content size, its width and height can be derived from the text of the label and the font type/size used.

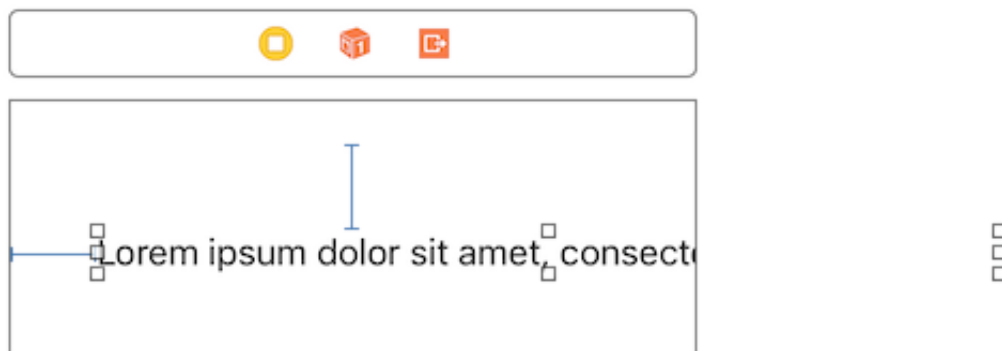
Button and Textfield has intrinsic content size and its calculation is similar to label.

An empty image view doesn't have intrinsic content size, but once you add image to the image view, its intrinsic content size is set to the image size.

[Here is a table of intrinsic content size](#) for common UI element from Apple official documentation.

Intrinsic content size of Label

By default, the intrinsic content width of a label increases as the text gets longer but its intrinsic content height stays fixed as single line. If you have a label containing long text without constraint specifying its width and height, it will grow horizontally and eventually off the screen like this :



To prevent the label from going off the screen, we can specify constraint that enable Auto Layout Engine to calculate its width and this value will replace its default intrinsic content width. Lets add another trailing constrain to the label and it will look like this :



For iPhone SE, Auto Layout Engine will calculate the width for the label like this :

```
screenWidth = 320
leadingConstraint = 40
trailingConstraint = 40

labelWidth = screenWidth - leadingConstraint -
trailingConstraint
labelWidth = 320 - 40 - 40
labelWidth = 240
```

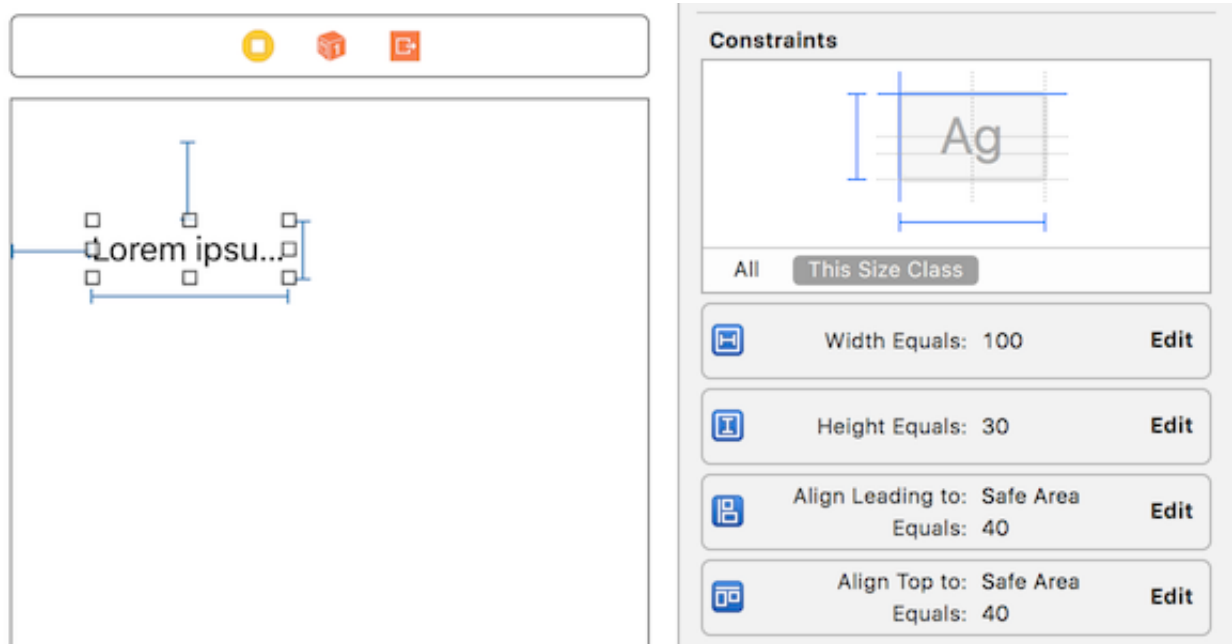
As Auto Layout Engine is able to calculate the label width, it will use this calculated value instead of its original intrinsic width. But now the text is truncated as its intrinsic height is still 1 line tall, how can we change its intrinsic height?

We can change its intrinsic height by changing the number of lines of the label to 0 , 0 means that the label can infinitely expand in height as its content text grows longer (Intrinsic content height increases as the number of line of text increase). We change the number of lines and update the frame of the label like this :

Watch this step on youtube : <https://www.youtube.com/watch?v=5PhJyiglxMA>

(I have to manually drag to increase the label height before updating frame, if I dont do this Xcode doesn't update the label height, its probably due to some Xcode bug)

If you set both explicit width and height constraint, or leading/trailing and top/bottom constraint for the label, the explicit defined size or calculated size from constraints will replace the intrinsic content size making it fixed and non flexible. This causes problem if the text of label exceed the size and unable to expand, causing the text to be cropped :



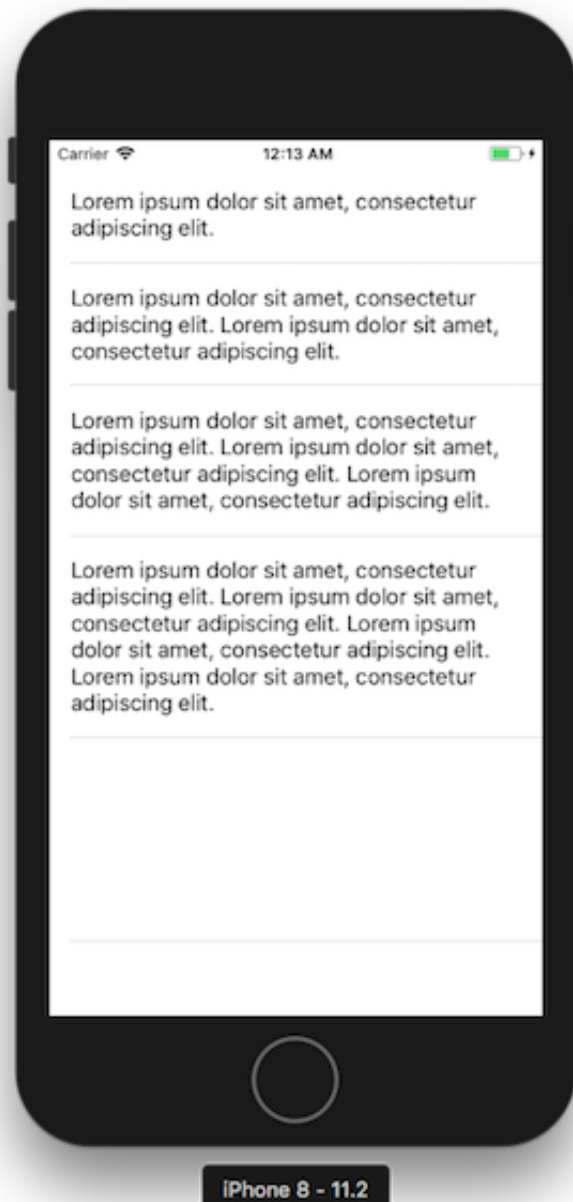
In most cases, you will only need to set constraints to let Auto Layout Engine calculate the width of the label and let its intrinsic height expand based on its text content.

This concept can apply to other UI element that have similar intrinsic content size calculation as label such as button and textfield.

Usefulness of intrinsic content size

Intrinsic content size allows the size of an UI element to be dynamically expanded based on its content, it is especially useful on [designing a scroll view containing dynamic content](#) (content loaded from user input or HTTP request) or [self sizing table view cells](#) (tableview containing rows of different height based on the content of the label inside cell).

Self sizing table view cells look like this :



Notice how the tableview cell expand to cater for longer length of text (content) inside it. Since the width is fixed (label width = screen width - leading - trailing), the text label intrinsic height increases as text got longer. This is useful if you are showing comments with varying length (like Reddit app) from user on table view.

In the next post, we will look into what is constraint priority and when to use it.