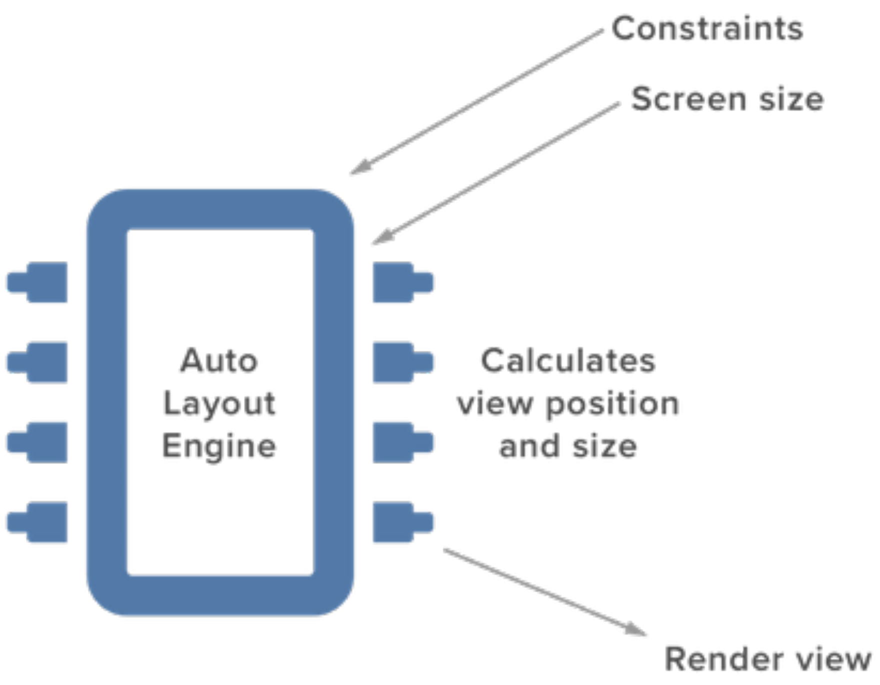
2 - How Auto Layout calculates view position and size

Previously, we have explained the difference between frame based layout and Auto layout with constraint. In this chapter we will take a closer look on how Auto Layout calculates and position view.

For easier understanding, assume there is an Auto Layout Engine that takes input of constraints (the constraint you created in Xcode) and app screen size (depending on iOS device screen resolution, orientation).

Here is the official screen resolution of each iOS devices (https://developer.apple.com/library/archive/documentation/DeviceInformation/Reference/iOSDeviceCompatibility/Displays/Displays.html#//apple_ref/doc/uid/TP40013599-CH108-SW2).

For UI development, we will use the **UIKit Size** (Points) column.



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 :

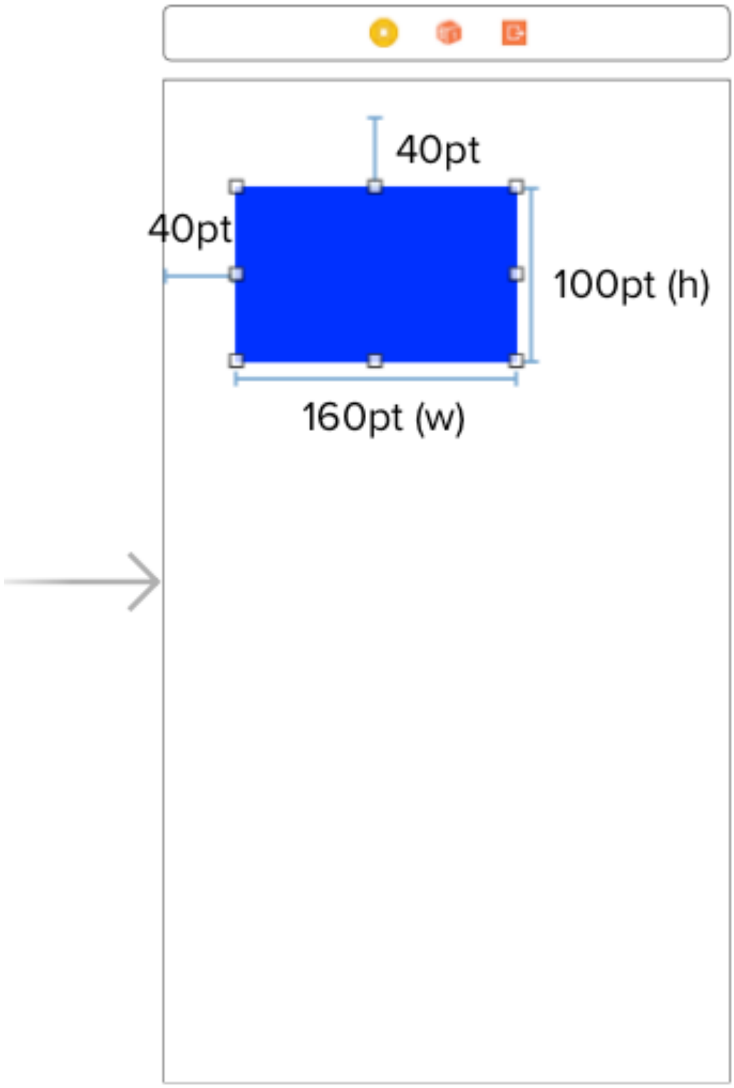
1. x position of a view
2. y position of a view
3. width of a view
4. height of a view

Defining constraint helps Auto Layout Engine to figure all of the above. Below I will show few examples of how Auto Layout calculate the position and size of a view using simple math (around grade school math).

We will discuss 4 examples and 1 exercise question in this chapter.

Example 1

A straight forward example might look like this :



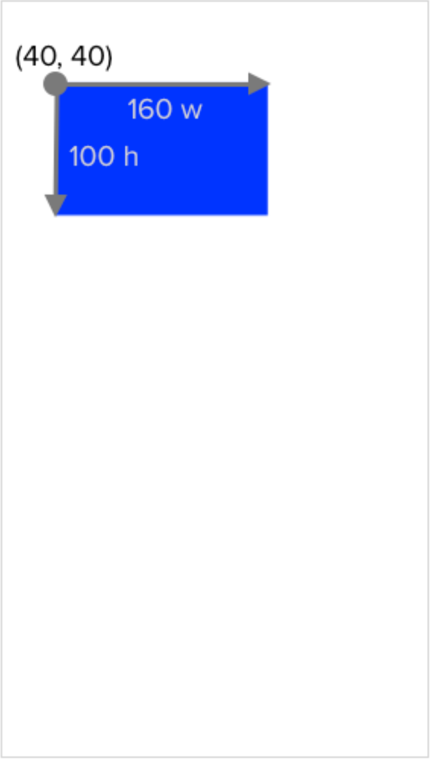
4 constraints are defined :

1. Distance from the screen left to the left of the blue view is 40 pt
2. Distance from the screen top to the top of the blue view is 40 pt
3. Width of the blue view is 160 pt
4. Height of the blue view is 100 pt

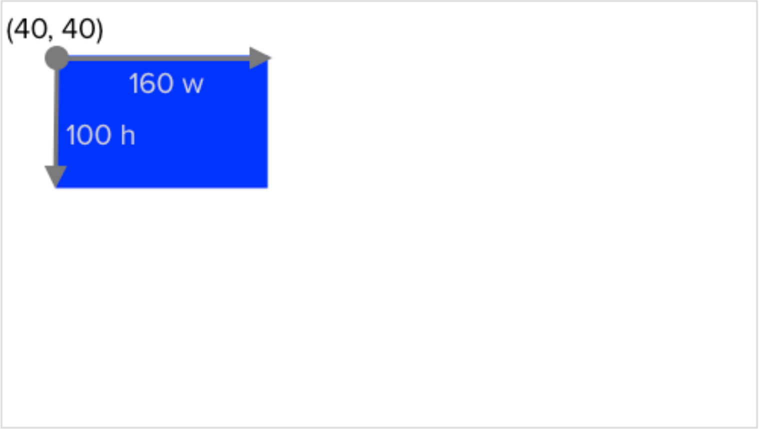
From the 1st constraint, we can deduce that the x position of the view is 40.

From the 2nd constraint, we can deduce that the y position of the view is 40.

Since x, y, width and height of the view is known, Auto Layout Engine will render it nicely across multiple screen size and orientation :



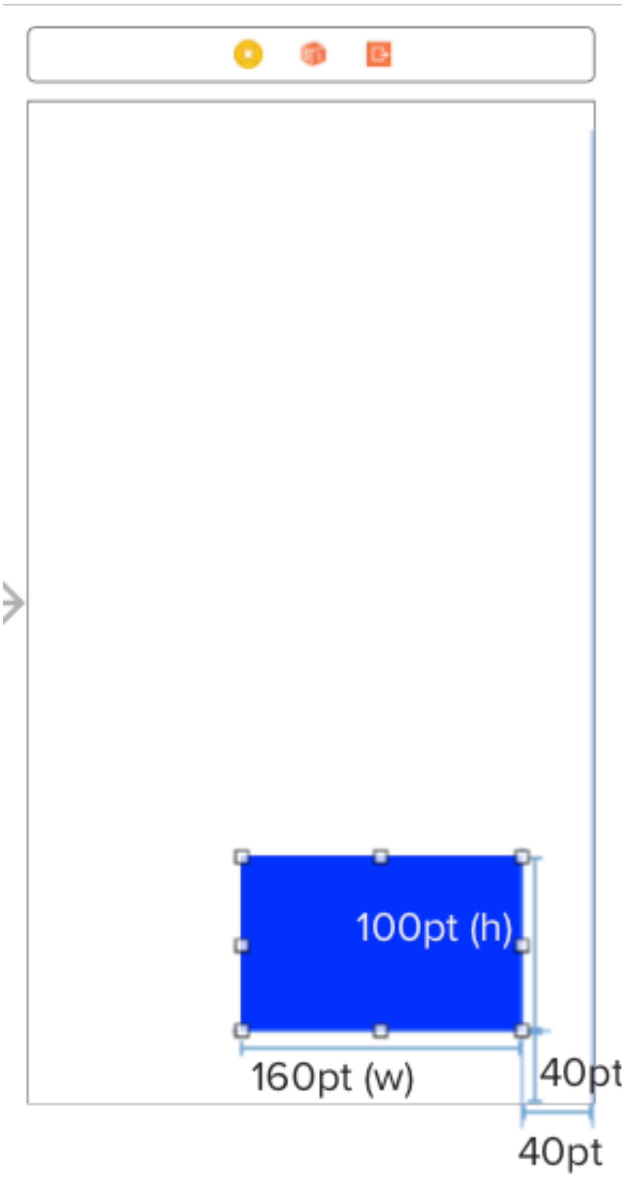
iPhone SE | Portrait



iPhone SE | Landscape

Example 2

Similar to previous example, we use back the same width and height but this time we define a right and bottom constraint like this :



4 constraints are defined :

1. Distance from the screen right to the right of the blue view is 40 pt
2. Distance from the screen bottom to the bottom of the blue view

is 40 pt

3. Width of the blue view is 160 pt
4. Height of the blue view is 100 pt

To calculate **x position** of the blue view, Auto Layout Engine will use the screen width. Calculation on an iPhone SE (screen width 320 pt) will look like this :

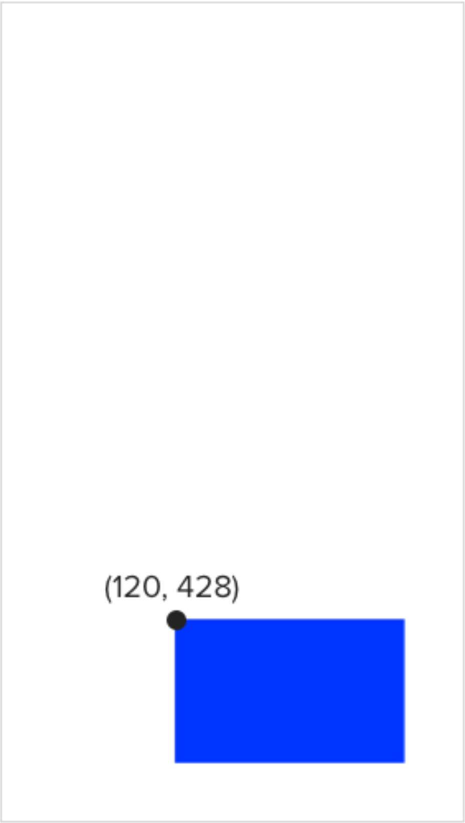
```
screenWidth = 320
distanceFromRight = 40
viewWidth = 160
// x position of the blue view
x = screenWidth - distanceFromRight - viewWidth
x = 320 - 40 - 160
x = 120
```

To calculate **y position** of the blue view, Auto Layout Engine will use the screen height. Calculation on an iPhone SE (screen height 568 pt) will look like this :

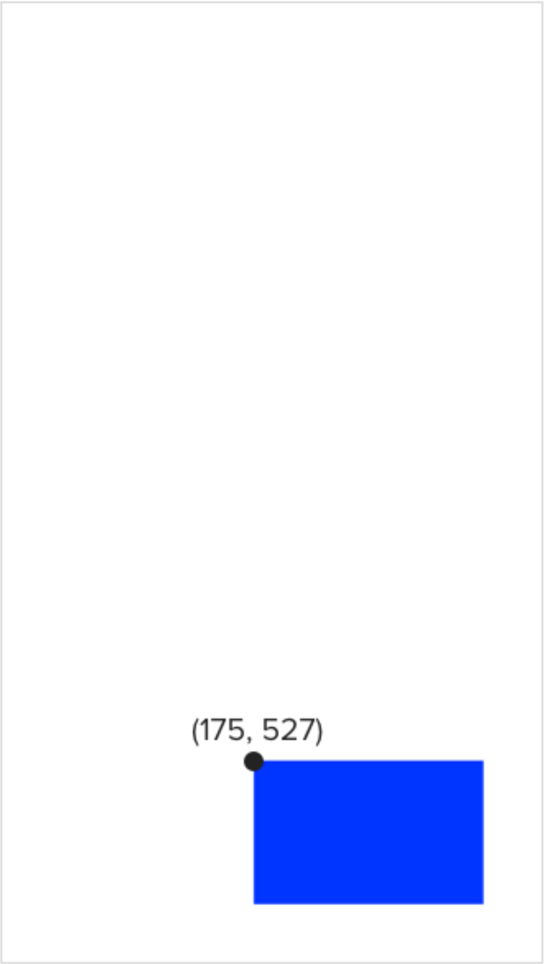
```
screenHeight = 568
distanceFromBottom = 40
viewHeight = 100
// y position of the blue view
y = screenHeight - distanceFromBottom - viewHeight
y = 568 - 40 - 100
y = 428
```

Using the same calculation formula, iPhone 8 (screen width 375, height 667) will calculate the x position as 175, y position as 527.

Now x, y, width and height of the view is known, Auto Layout Engine will render it nicely across multiple screen size and orientation :



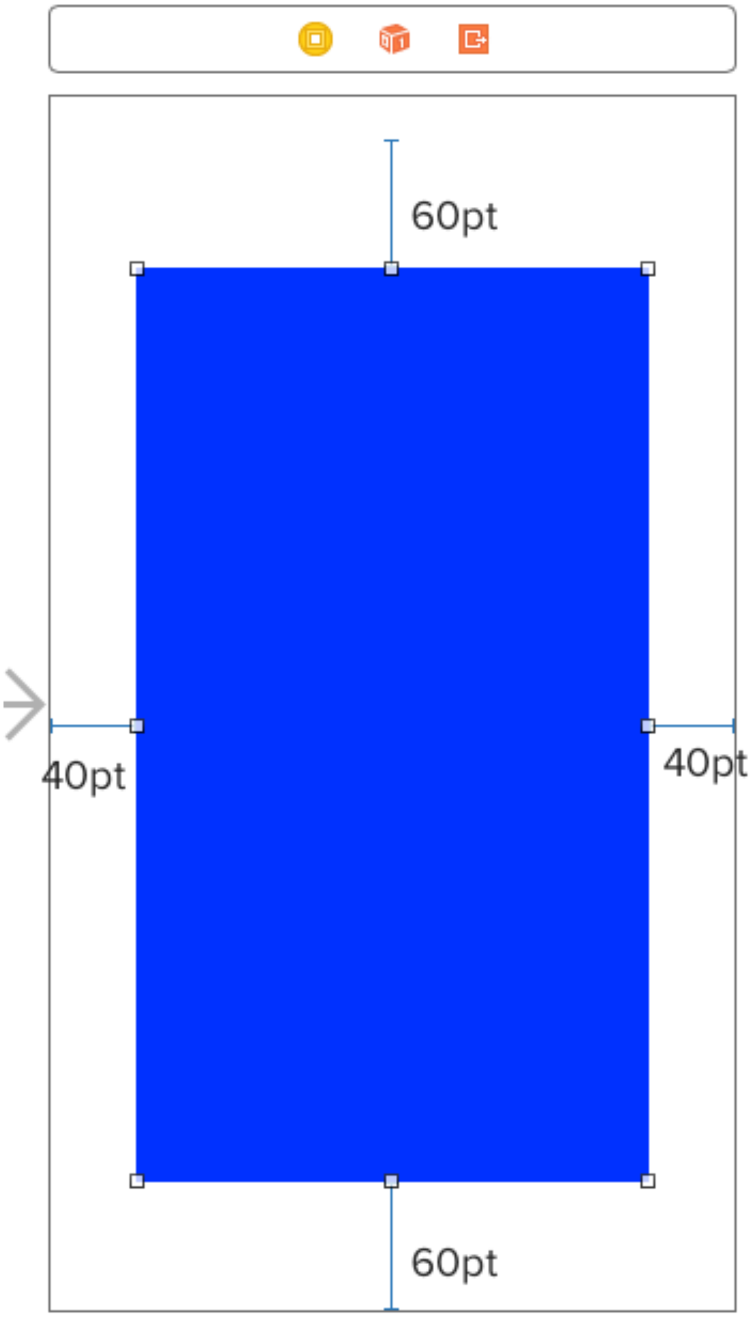
iPhone SE | Portrait



iPhone 8 | Portrait

Example 3

In this example, there is no width or height constraint defined, instead we will set the top, left, right and bottom constraint from the blue view to the screen :



4 constraints are defined :

1. Distance from the screen top to the top of the blue view is 60 pt
2. Distance from the screen left to the left of the blue view is 40 pt
3. Distance from the screen right to the right of the blue view is 40 pt
4. Distance from the screen bottom to the bottom of the blue view is 60 pt

From the 1st constraint, we can deduce that the y position of the view is 60.

From the 2nd constraint, we can deduce that the x position of the view is 40.

To calculate **width** of the blue view, Auto Layout Engine will use the screen width. Calculation on an iPhone SE (screen width 320 pt) will look like this :

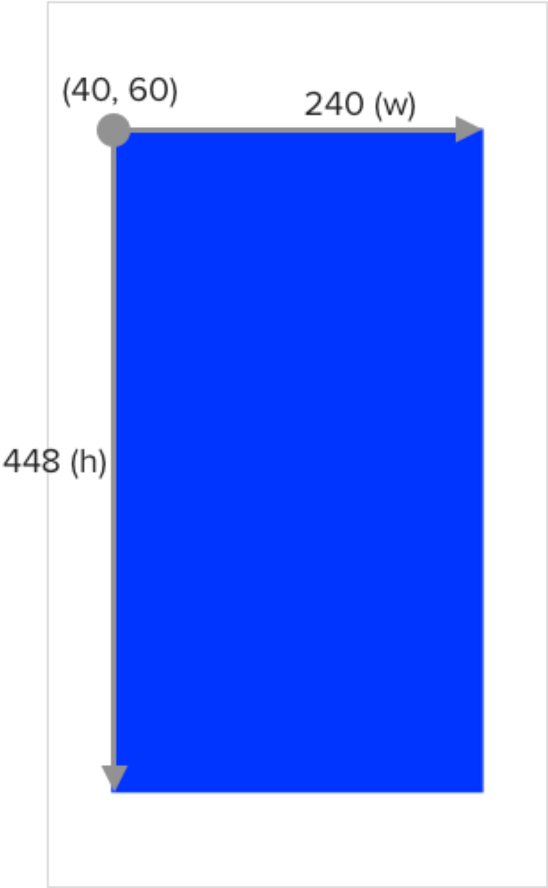
```
screenWidth = 320
distanceFromLeft = 40
distanceFromRight = 40
// width of the blue view
viewWidth = screenWidth - distanceFromLeft -
distanceFromRight
viewWidth = 320 - 40 - 40
viewWidth = 240
```

To calculate **height** of the blue view, Auto Layout Engine will use the screen height. Calculation on an iPhone SE (screen height 568 pt) will look like this :

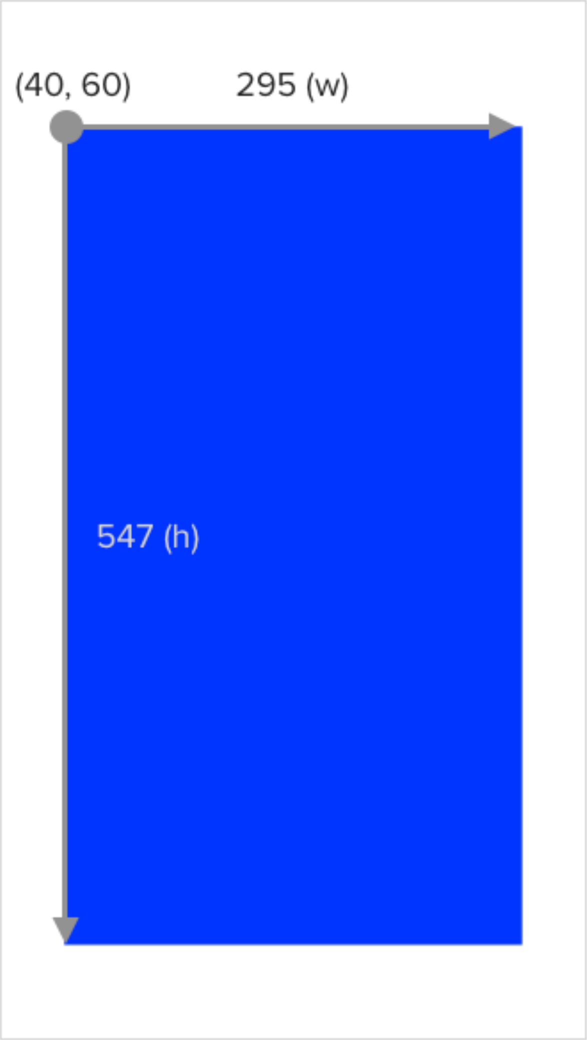
```
screenHeight = 568
distanceFromTop = 60
distanceFromBottom = 60
// height of the blue view
viewHeight = screenHeight - distanceFromTop -
distanceFromBottom
viewHeight = 568 - 60 - 60
viewHeight = 448
```

Using the same calculation formula, iPhone 8 (screen width 375, height 667) will calculate the width as 295, height as 547.

Now x, y, width and height of the view is known, Auto Layout Engine will render it nicely across multiple screen size and orientation :



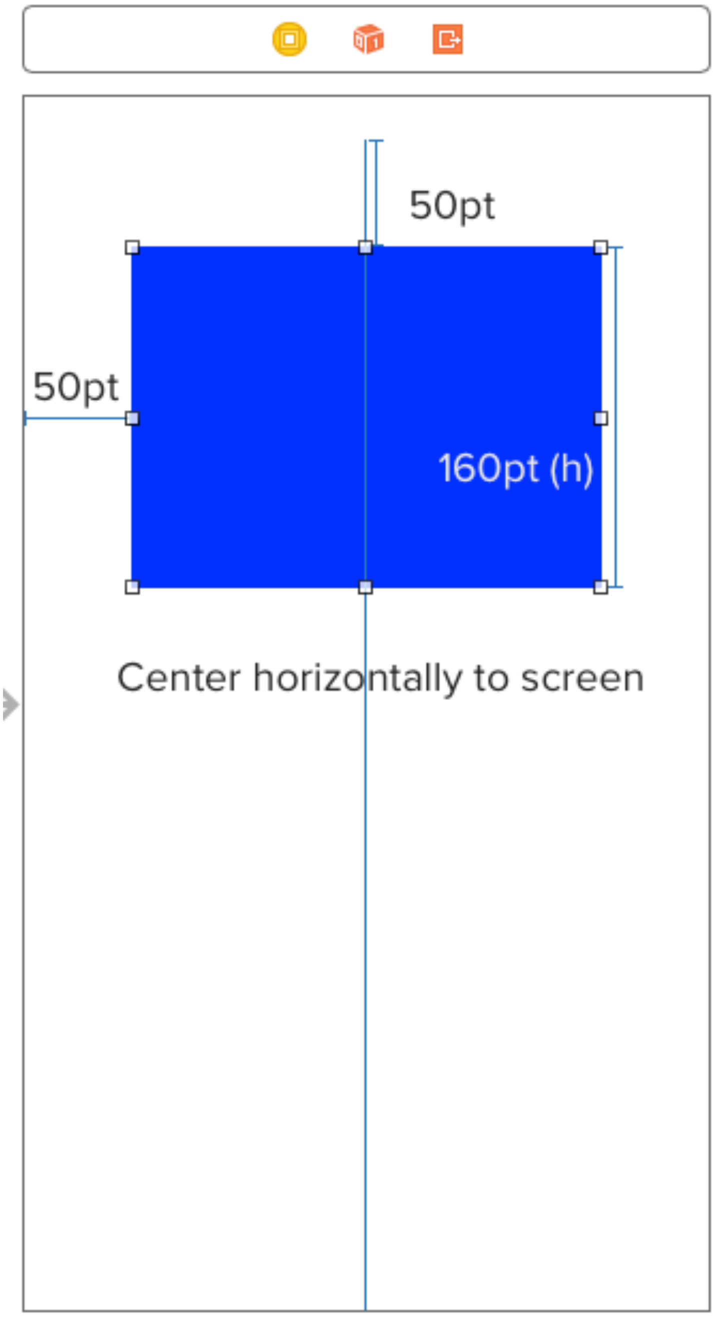
iPhone SE | Portrait



iPhone 8 | Portrait

Example 4

This example will be a bit more difficult to calculate than previous ones, the constraints are defined like this :



4 constraints are defined :

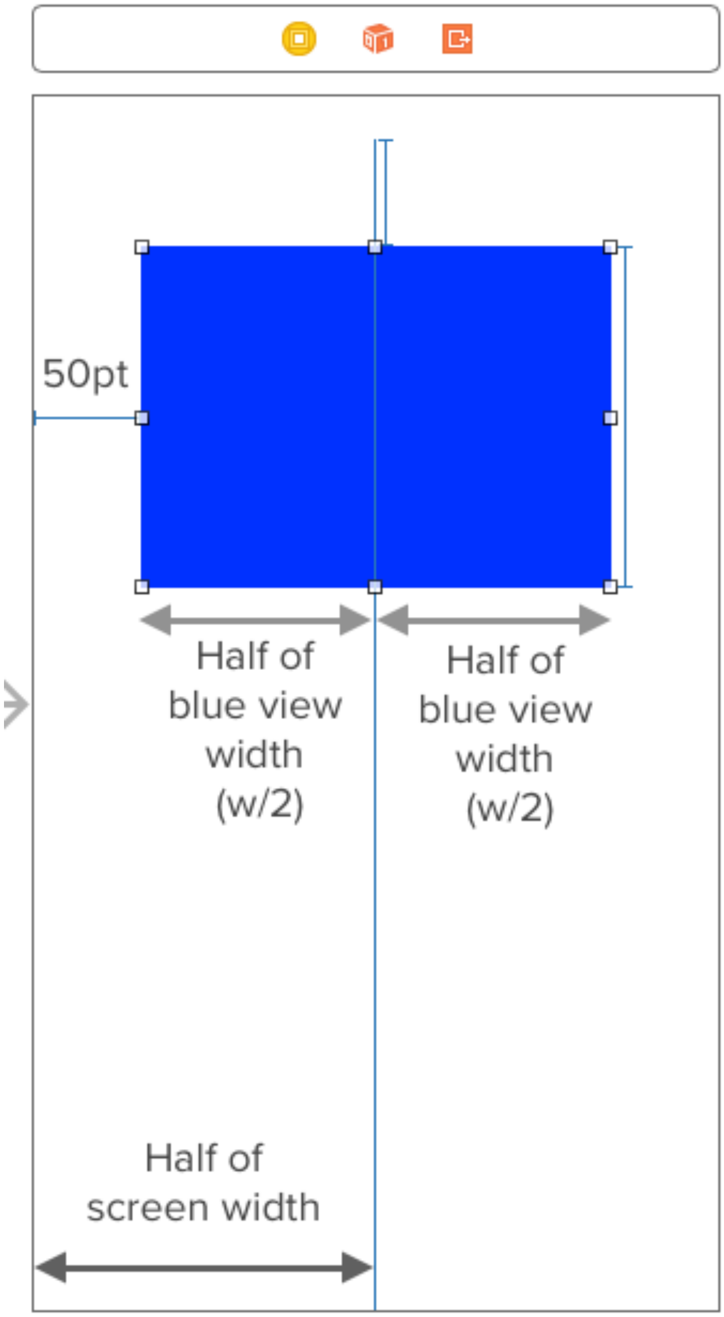
1. Distance from the screen top to the top of the blue view is 50 pt

2. Distance from the screen left to the left of the blue view is 50 pt
3. Height of the blue view is 160 pt
4. Blue view is horizontally centered to the screen

From the 1st constraint, we can deduce that the y position of the view is 50.

From the 2nd constraint, we can deduce that the x position of the view is 50.

The remaining step is to calculate the width of the blue view since height is given. Since blue view is horizontally centered, it means that the left half of the width is same as the right half, like this :



From the above diagram, we can deduce that :

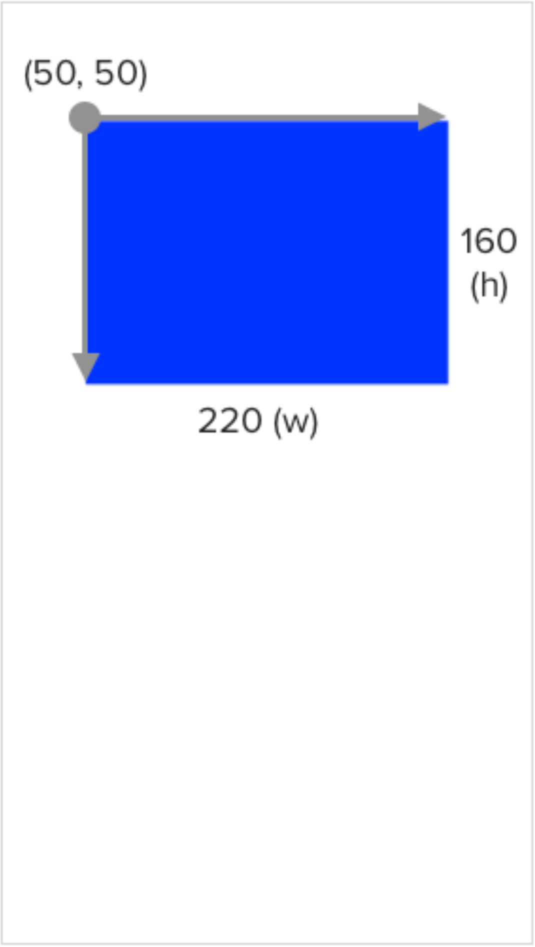
```
50 + half of blue view width = half of screen width
```

To calculate **width** of the blue view, Auto Layout Engine will use the screen width. Calculation on an iPhone SE (screen height 568 pt) will look like this :

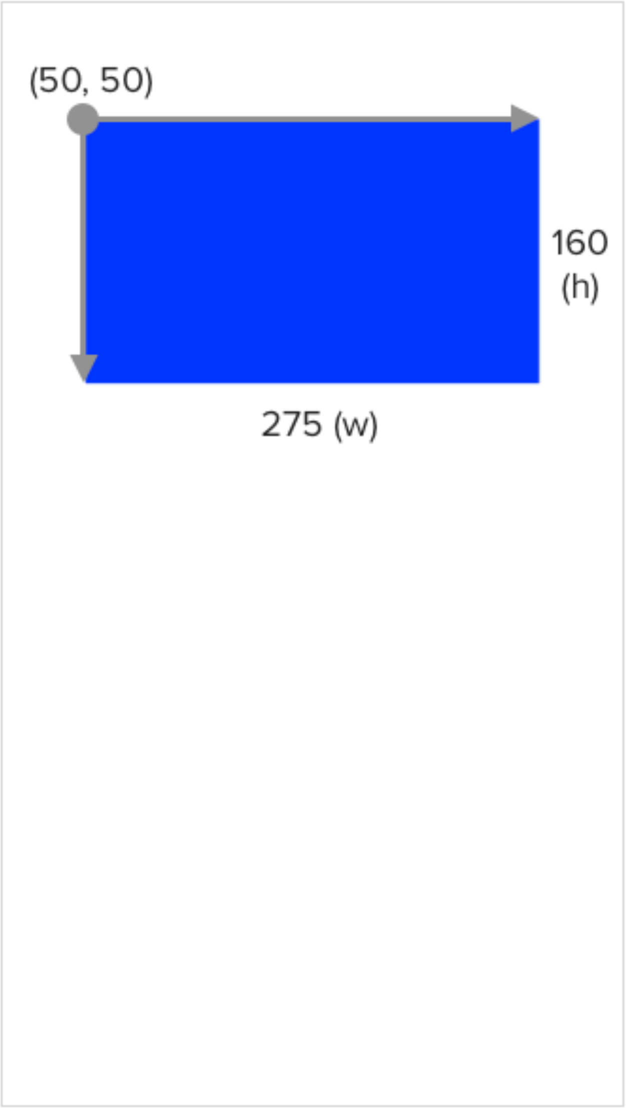
```
screenWidth = 320
distanceFromLeft = 50
distanceFromLeft + (viewWidth / 2 ) = screenWidth / 2
50 + (viewWidth / 2) = 320 / 2
50 + (viewWidth / 2) = 160
viewWidth / 2 = 160 - 50
viewWidth / 2 = 110
viewWidth = 110 * 2
viewWidth = 220
```

Using the same calculation formula, iPhone 8 (screen width 375, height 667) will calculate the width as 275.

Now x, y, width and height of the view is known, Auto Layout Engine will render it nicely across multiple screen size and orientation :



iPhone SE | Portrait



iPhone 8 | Portrait

This example is useful for cases where you want to center a view with equal left spacing and right spacing. Instead of setting left space constraint and right space constraint manually to the same number, you can just set the left space constraint and set a horizontally center constraint on the view.

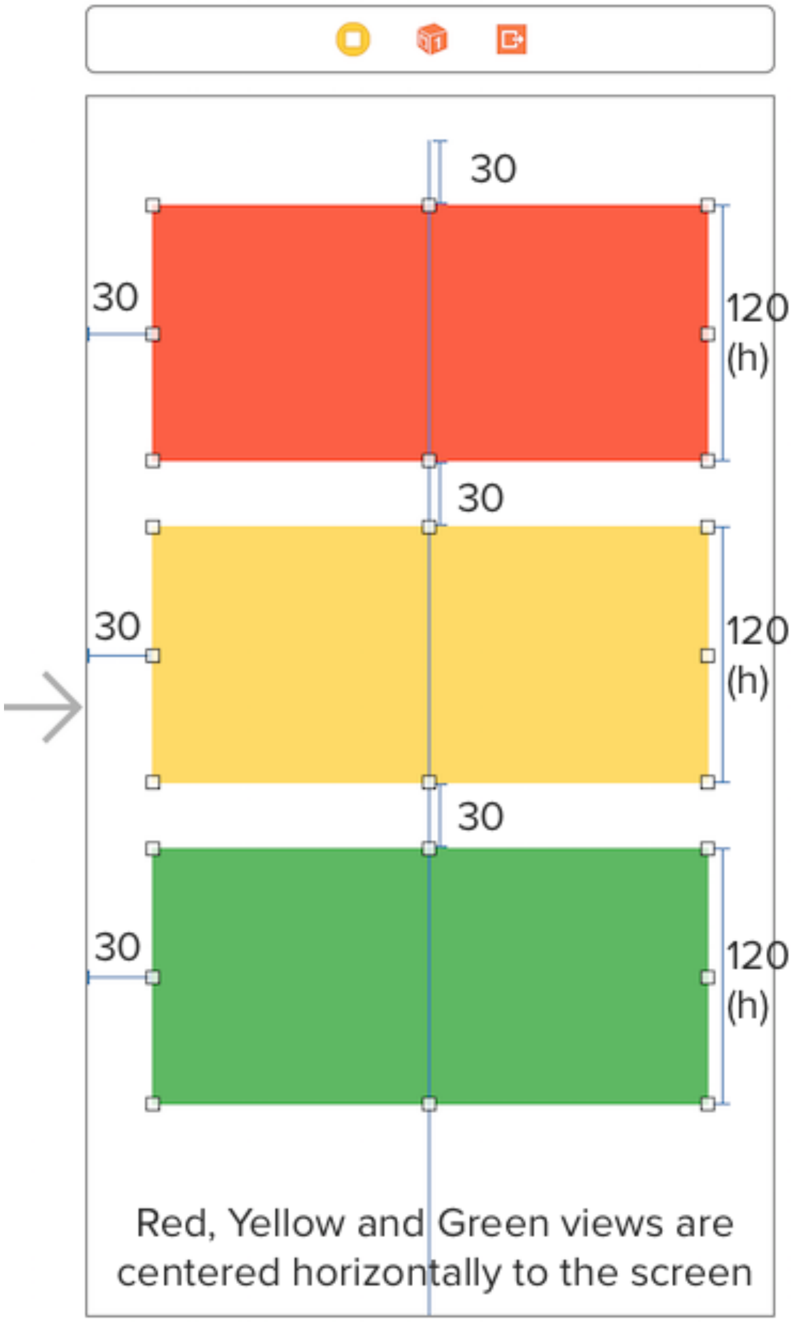
Homework exercise - Multiple View

We have discussed various examples and now its time to do some hands on calculation 📝 ! Although previous examples only use one view, the same concept and calculation still applies for multiple view.

Here is the exercise question :

Calculate the x position, y position, width and height of red, yellow and green view. Assuming iPhone SE screen size is used (320pt x 568 pt)

The constraints are defined as follow :


30
30
120
(h)
30
30
120
(h)
30
30
120
(h)
Red, Yellow and Green views are centered horizontally to the screen

1. Distance from the screen top to the top of red view is 30 pt
2. Distance from the screen left to the left of red view is 30 pt
3. Height of red view is 120 pt
4. Distance from bottom of red view to top of yellow view is 30 pt
5. Distance from the screen left to the left of yellow view is 30 pt
6. Height of yellow view is 120 pt
7. Distance from bottom of yellow view to top of green view is 30 pt
8. Distance from the screen left to the left of green view is 30 pt

9. Height of green view is 120 pt
10. Red view is horizontally centered on the screen
11. Yellow view is horizontally centered on the screen
12. Green view is horizontally centered on the screen

The answer will be at the end of this chapter, be sure to try it before looking at the answer! 🤔

Hint : Start calculating from the view at the top first then proceed to bottom.

Summary

The takeaway of this chapter is that if Auto Layout Engine can calculate all of the views position and size using the given constraints, it will render them nicely. If there is not enough constraint, Auto Layout Engine can't calculate position/size of some view, it will attempt to 'guess' their position/size and thats the reason why you see some layout going haywire when constraints are not defined correctly.

Here is the checklist to ensure Auto Layout Engine can render the view correctly :

Does the constraints defined allow Auto Layout Engine to know / calculate :

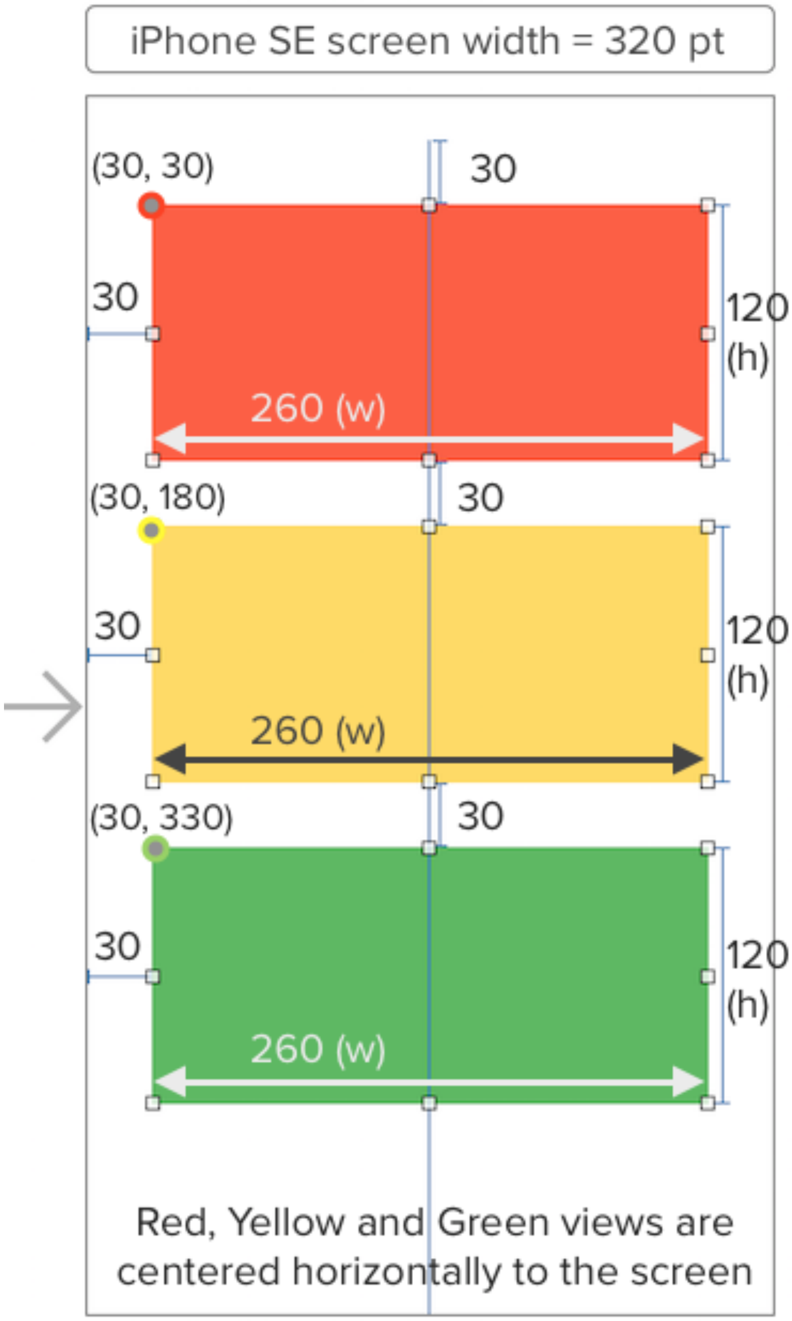
1. x position of all view ✅
2. y position of all view ✅
3. width of all view ✅
4. height of all view ✅

One quick way to determine if Auto Layout Engine is able to calculate them (position and size) is to calculate it yourself, by using pen and paper. If you can't deduce them using math, Auto Layout Engine can't either. Once you get a hang of it, you won't need to calculate these by hand, as long as you know how to define constraints that enable Auto Layout Engine to calculate these, your layout will look great on every device.

In the next chapter, we will look into how to solve missing constraint (when Xcode show you red lines around your view.)

Exercise Answer

On iPhone SE screen :


iPhone SE screen width = 320 pt
(30, 30)
30
30
120
(h)
260 (w)
(30, 180)
30
30
120
(h)
260 (w)
(30, 330)
30
30
120
(h)
260 (w)
Red, Yellow and Green views are
centered horizontally to the screen

Red View :

1. X position is 30
2. Y position is 30
3. Width is 260

4. Height is 120

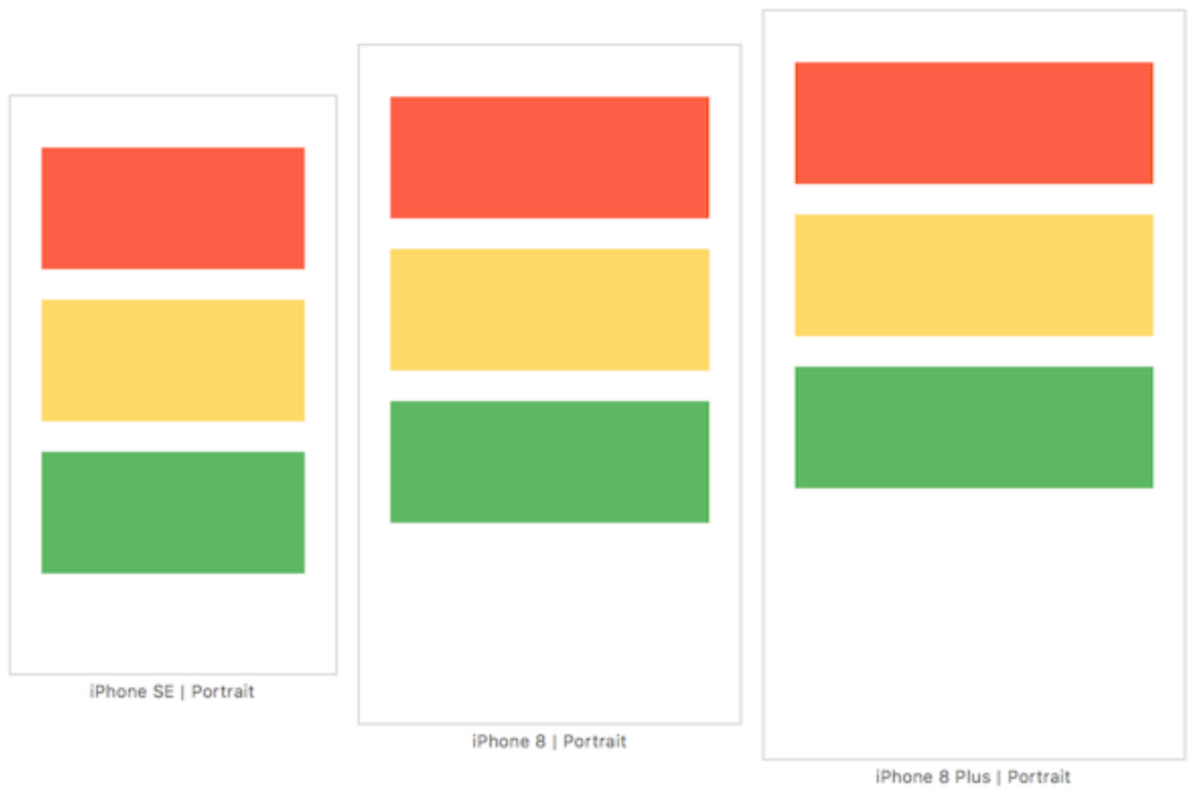
Yellow View :

1. X position is 30
2. Y position is 180
3. Width is 260
4. Height is 120

Green View :

1. X position is 30
2. Y position is 330
3. Width is 260
4. Height is 120

This is how it looks like in multiple screen sizes :


iPhone SE | Portrait
iPhone 8 | Portrait
iPhone 8 Plus | Portrait