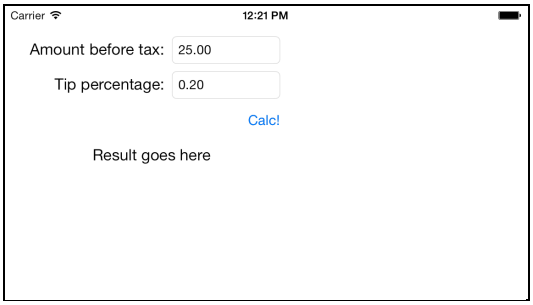
iOS 101

Hands-On Challenges

iOS 101
Hands-On Challenges

Challenge #4: Auto Layout Walkthrough

Although Tip Calculator is a very cool app, it currently doesn't respond to orientation changes well. Here's what it looks like in landscape orientation:

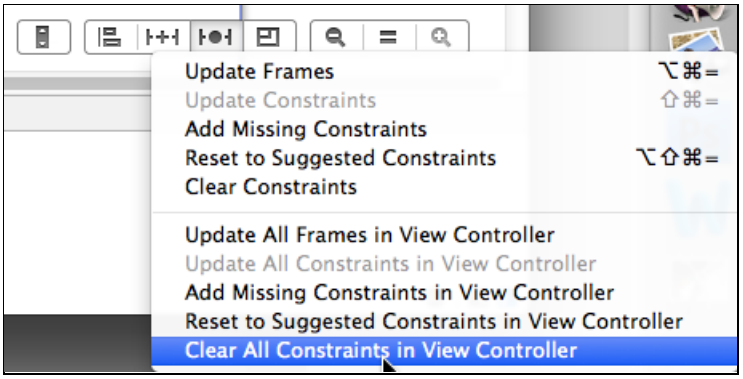


It would be a lot better if the controls repositioned to take advantage of the extra space, like this:

In this challenge, you are going to modify Tip Calculator to do exactly this – through the power of Auto Layout!

Basic Level Walkthrough

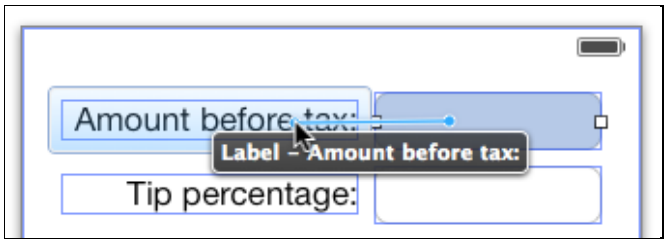
Open your **TipCalc** project from last time, and open **Main.storyboard**. There should be no Auto Layout constraints on any of the controls yet. To verify this, click the **Resolve Auto Layout Issues** button in the bottom right of the Storyboard (it looks like a Tie Fighter), and click **Clear All Constraints in View Controller**:



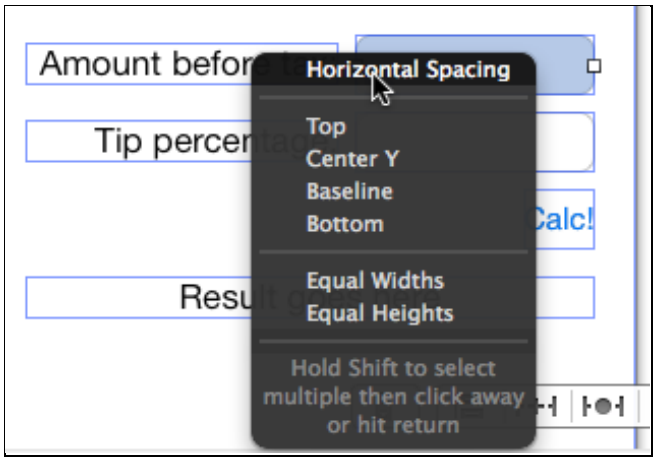
Let's start by getting the top part of the layout working properly – the "Amount before tax:" label and text field.

Start by adding a horizontal space constraint between the text field and label. Remember that there are two ways to do this – by control-dragging, or by using the helper buttons in the bottom right of the Storyboard editor, but this time we'll use control-dragging.

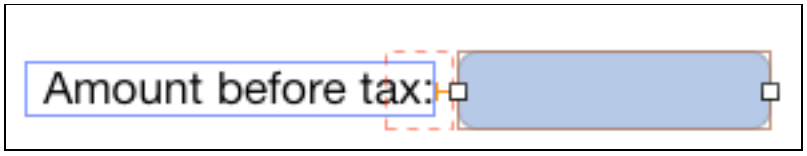
So control-drag from the text field to the label:



And select the Horizontal Spacing constraint from the popup:

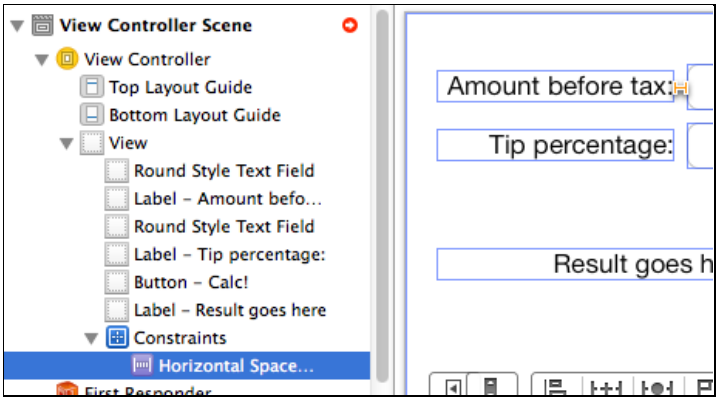


You will now see an orange grid line appear between the label and the text field. This indicates that there is a horizontal space constraint between these two controls.

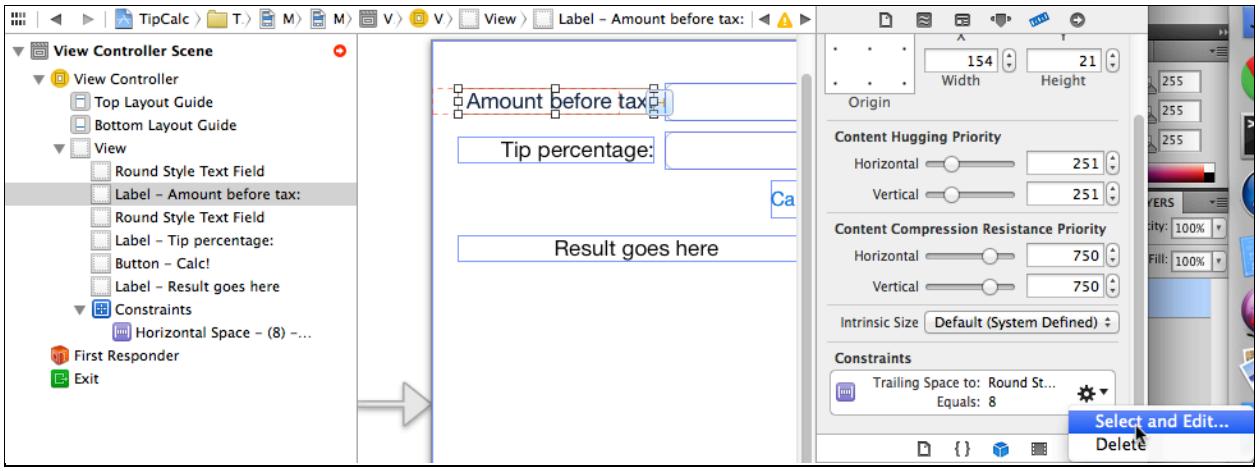


Next, select the constraint you just created. There are three ways to do this and they are all useful in diferent situations, so try each way to make sure you are familiar with each method:

1. In the View Controller hierarchy, expand **View Controller Scene\View Controller\View\Constraints** and select the constraint (Horizontal Space).

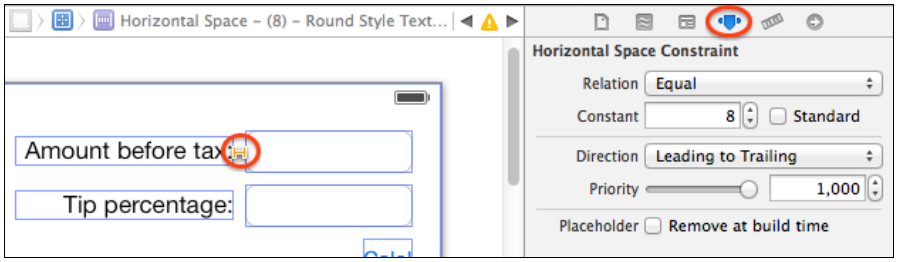


2. Select either the label or the text view and open the Size Inspector (the fifth inspeector tab) to see the constraints that involve that view. Click the constraint, and then click **Select and Edit...**.

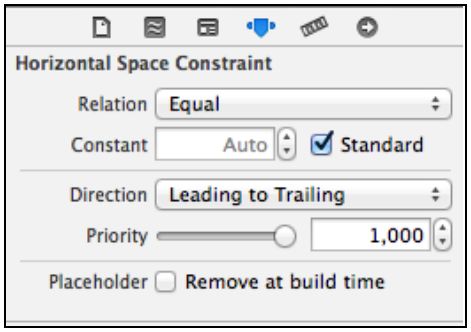


3. Select either the label of the text view so that the constraint appears visually on the Storyboard editor as an orange grid line. Click this orange grid line so that it is selected (it's kinda hard to click, so be accurate).

Once the constraint is selected, open the Attribute Inspector (the fourth inspector tab) to see the properties on the constraint that you can edit:

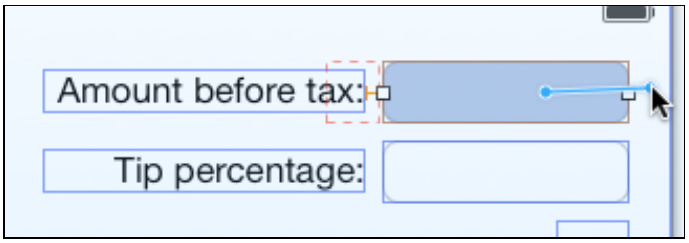


Click the **Standard** button to make the space constraint equal to the recommended spacing in iOS.

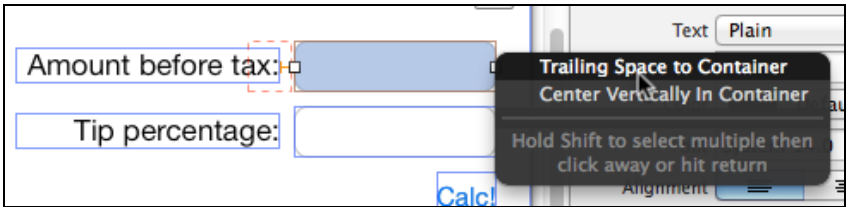


This constraint is orange, which means that there are still not enough constraints so that Auto Layout knows what do do unambiguously. That makes sense – there's still a lot more connecting to do!

Connect the right side of the text field to the right side of the view by control-dragging from the text field to just outside to its right, like this:

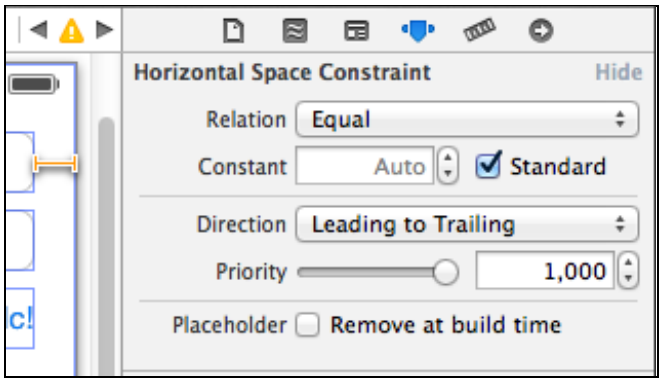


In the popup that appears, select **Trailing Space to Container**:



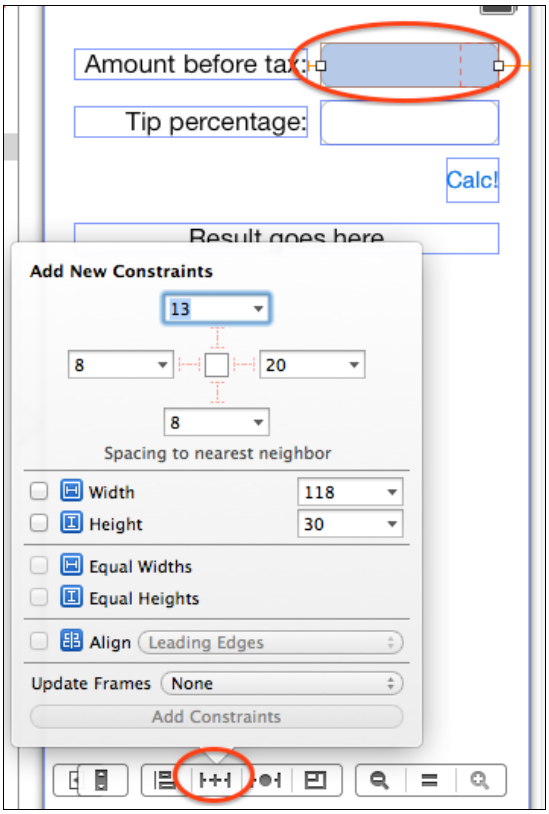
This creates a trailing space constraint between the text field and its containing view – in other words, the view that it is a child view of, which is the main view in this case. So this fixes the right hand side of the text view to the right hand side of the main view.

Select the new constraint (using one of the techniques you learned about earlier), and click the **Standard** button:



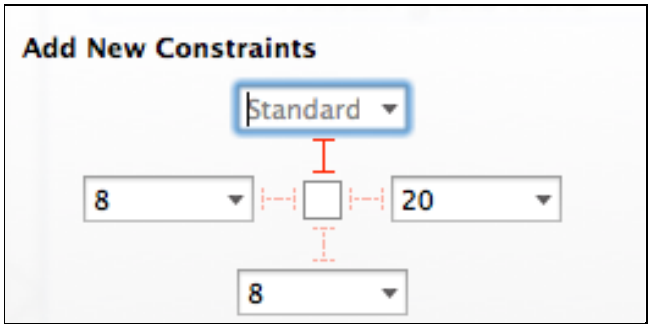
So far, you've seen an example of adding a constraint between two sibling views, and between a view and it's containing view. Both times, you created constraints using the control-drag method, but remember you can also create constraints using the buttons in the bottom right of the Storyboard editor. Let's try that now.

Select the text view, and click the **Pin** button in the bottom right of the Storyboard Editor (it looks like a train track). This brings up a panel that allows you to easily add constraints related to the selected view:

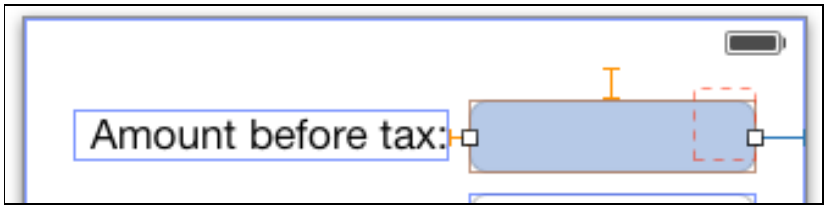


In the **Add New Constraints** section, click on the faded out red grid line on top of the white box. This adds a constraint between the top of the text view, and the top

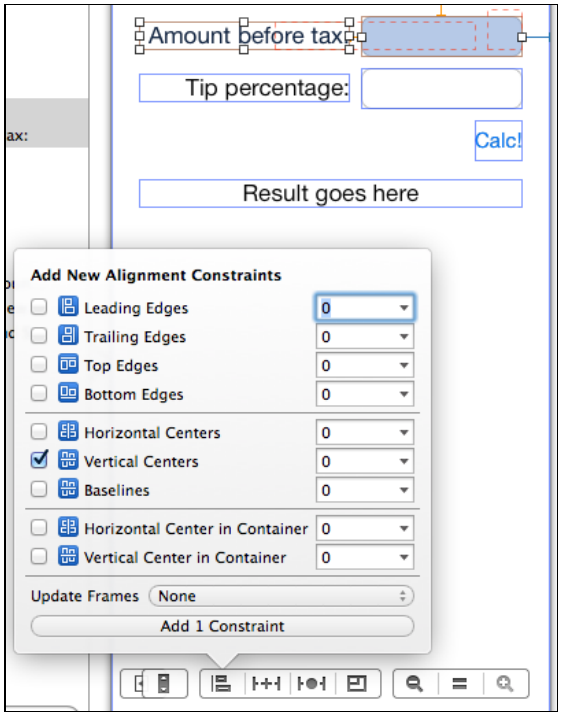
of the nearest neighbor (which is the top of the main view in this case). Then click the down arrow, and set the spacing to **Standard**.



Finally, click **Add 1 Constraint** at the bottom of the Pin panel. At this point, you should see three constraints if you select your text view:



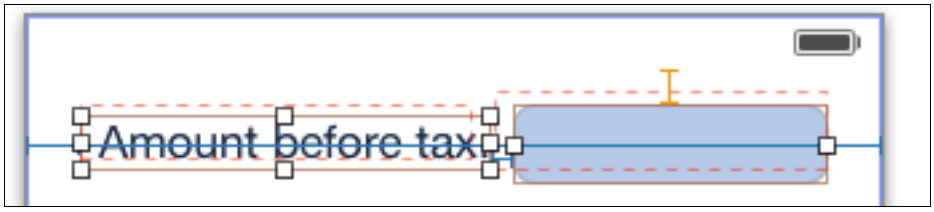
At this point, the text area is fixed to the top and right of the container, and the label is fixed horizontally to the text area. However, there is no constraint that specifies where the label should be placed vertically. To fix this, select both the label and the text area and click the **Align** button in the bottom right of the Storyboard Editor (it looks like a bar graph turned sideways). Click the **Vertical Centers** button, then **Add 1 Constraint**.



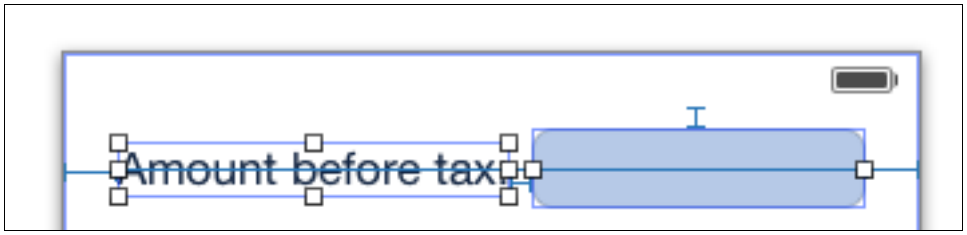
You just need one final constraint to test this out for now. See if you can add it yourself, based on what you've learned so far! If you get stuck, refer back to the instructions earlier in this tutorial. Here is your task:

- Between the left of the **label** and its **containing view**, add a **leading space to container constriant** equal to **Standard**.

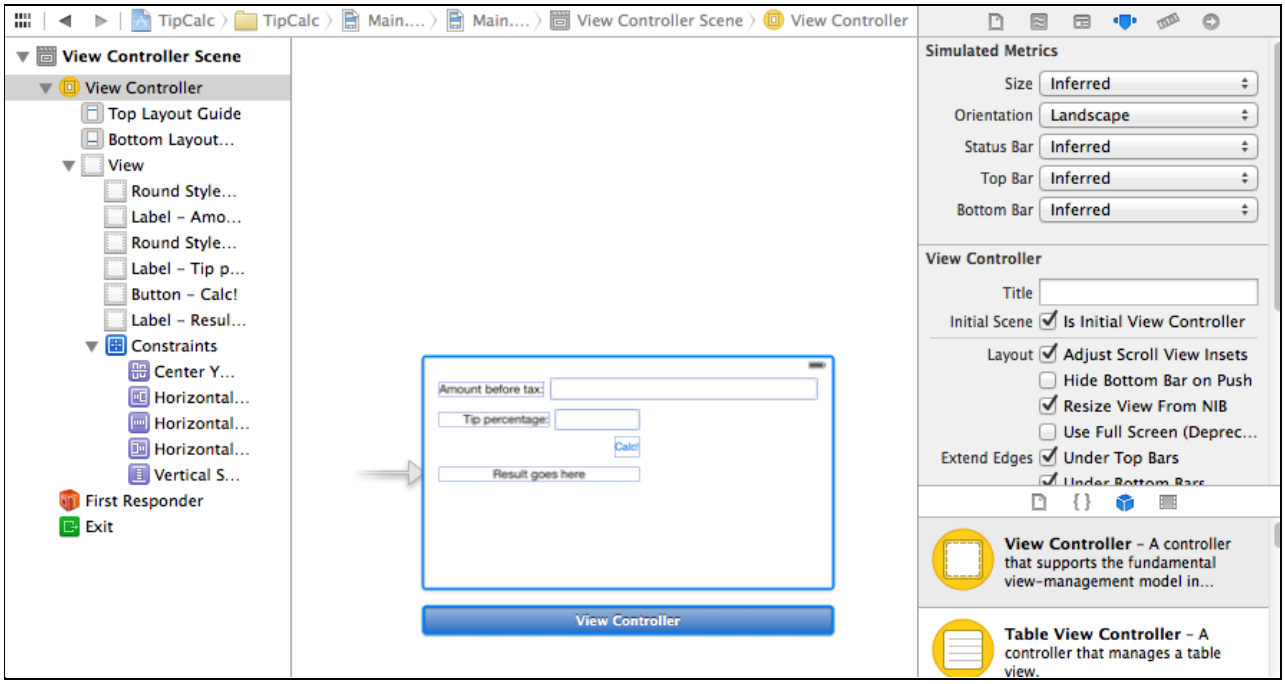
If you did this correctly, if you select both the label and the text view you should see something like this:



Note that the constraints are (mostly) blue now, which means that you have resolved the ambiguity with these controls. However, there is an orange dotted box for each control, that shows a mismatch between where the controls are in your Storyboard versus where they should be according to the constraints. To fix this, make sure both controls are selected, click the **Resolve Auto Layout Issues** button in the bottom right of the Storyboard (it looks like a Tie Fighter), and click **Update Frames**. This updates the frames of both controls in the Storyboard editor to where they should be, according to your Auto Layout constraints.



Everything is blue – looking good! In the left side of the Storyboard editor, select your View Controller and in the Attributes Inspector (the fourth tab), set the Orientation to Landscape. You will see the top controls now seem to stretch properly.

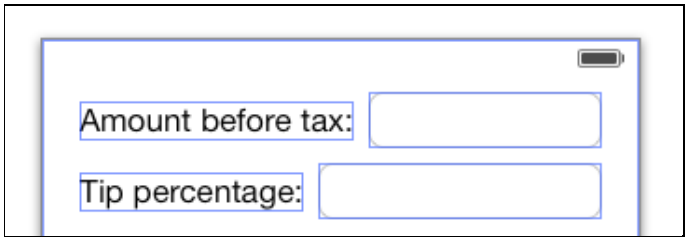


Switch the view controller back to Portrait. Now that you have the basics behind Auto Layout down, add a few more constraints to fix the second row:

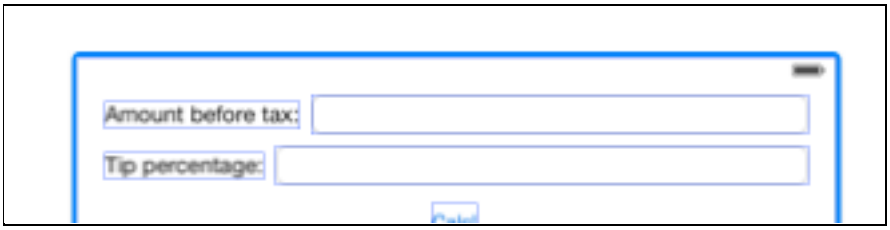
- Between the **two text views**, add a **vertical spacing** constraint set to **Standard**, and an **align trailing edges**.
- Between the **tip percentage label** and its **corresponding text view**, add a **horizontal spacing constraint** set to **Standard** and an **align vertical centers** constraint.
- Between the **tip percentage label** and its **containing view**, add a **leading space to container** constraint set to **Standard**.
- Select the **tip percentage label and text view** and use **Resolve Auto Layout Issues\Update Frames** to update their frames to match the Auto Layout constraints.

As you add these constraints, see if you can explain to yourself why each one is necessary in order to make the layout unambiguous.

If you got this working, your layout should look like the following in Portrait:



And the following in landscape:



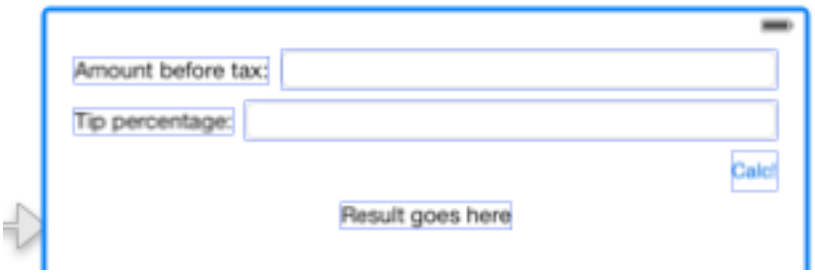
Congratulations – you now know the basics of using Auto Layout! If you're feeling particularly adventurous, try out the next part for an extra challenge.

Uber Haxx0r Level Challenge

There are two parts to this challenge.

The first part is fairly easy: add constraints to the Calc button and results area so that it looks like the following in Portait:

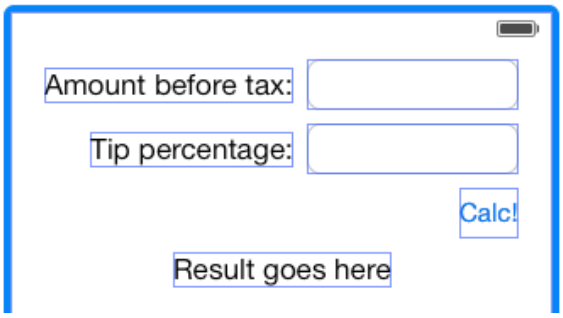
And the following in Landscape:



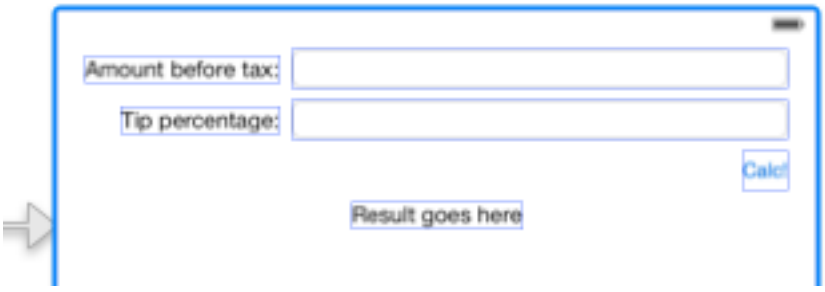
There are multiple ways to accomplish this, but here are the types of constraints I used (in no particular order):

- An align trailing edges constraint
- An align center X constraint
- An align top edge constraint (standard)
- An align bottom edge constraint (standard)

The second part of this challenge is a little bit harder. I think things would look a bit better if the text areas always had the same size, and the labels were always right aligned to the text fields. In other words, it should look like this in Portrait:



And like this in Landscape:



Note how the labels are sized based on their intinsic content size, not stretched to be a particular size.

Try to set up some constraints to accomplish this. It's tricker than you think! Here are a few hints:

- You will need three new constraints (a width constraint, and two additional horizontal space constraints).
- The new horizontal space constraints will need to be set up as Greater Than or Equal.
- You will need to set the priorities of four things correctly (these are listed in no particular order):
    - The greater than or equal horizontal space constraints
    - The normal horizontal space constraints
    - The content hugging priority of the labels
    - The content hugging priority of the text fields

If you get this working, congrats – you've got a solid understanding of Auto Layout and have reached the Uber Haxx0r level!