Content Hugging and Content Compression resistance priorities.

1. ***According to apple docs:*** *The layout priority is used to indicate to the constraint-based layout system which constraints are more important, allowing the system to make appropriate tradeoffs when satisfying the constraints of the system as a whole.*

?
But what actually does this means

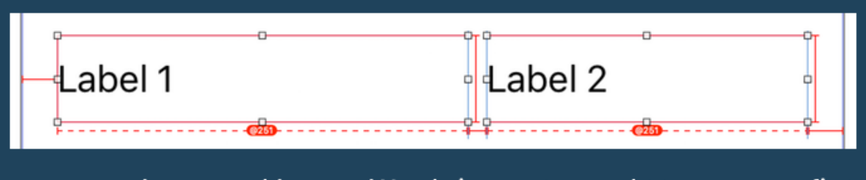
2. *The priority is the final judge when two different constraints conflicts. The system will give importance to the one with higher priority. So, Priority is the kids mom that will say who will sit in the front seat :D but in the autolayout world.*

Content hugging priority.

Sets the priority with which a view resists being made larger than its intrinsic size. Setting a larger value to this priority indicates that we don't want the view to grow larger than its content.

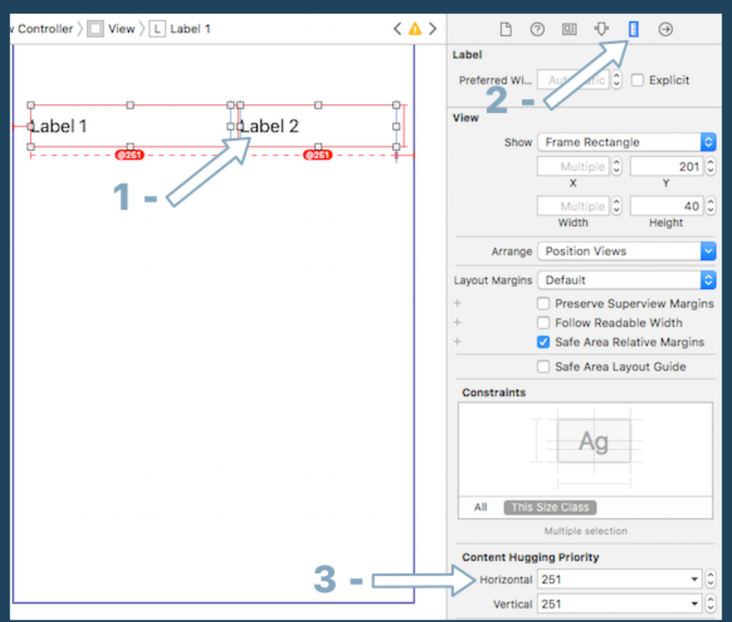
---

1. Imagine this situation, two labels, we set constraints to margin left and right for both, we set a fixed space between both, and of course we set fixed height, but we do not set *width* because we want our text to say which size should choose, how to do that?!



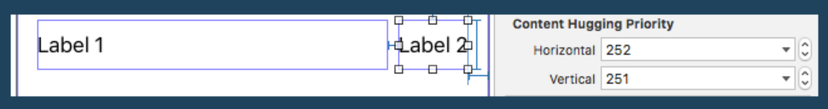
As you can see we have a problem, and Xcode is smart enought to propose a fix to this, as you can see we have in red the ***@251*** *that is the content hugging priority for both labels for the horizontal hugging priority, so, if we set which one is more important auto layout will verify if has any priority that is higher to fix the ambiguity and if have will know which one should fit the label width size!!!*

2. Now what we gonna do is , select the 1 - "*Label 2*", select 2 - "*size inspector*" if not selected, down there you will see the "*Content Hugging Priority*", and now for the 3 - "*Horizontal*" just set the value to "*252*"



** Some considerations, as you saw we have the "Horizontal" and the "Vertical" that is the same like "Width" and "Height", so, when you want to adjust and it's about the "Height" you just need to adjust the "Vertical" content huging priority.*

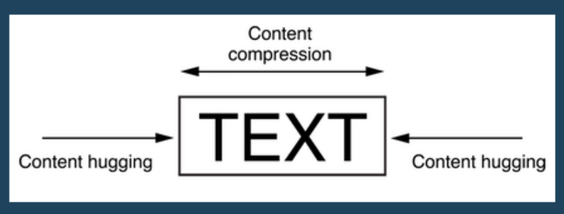
3. And that's it, you see that now the text adjusted itself to fit the size of "Label 2" and there's no more autolayout complaints about which label size should fit, that means that now if our text grows we will always adjust according to our text!



Tip: If you wonder how you can see the content hugging priority on storyboard just go to "Editor -> Canvas -> Show Intrinsic Size Constraints Contributing To Ambiguity". There are even more options to see corners and so on, it's nice to have activate when creating your views.

Anatomy of a Constraint:
https://developer.apple.com/library/archive/documentation/UserExperience/Conceptual/AutolayoutPG/AnatomyofaConstraint.html#//apple_ref/doc/uid/TP40010853-CH9-SW1

Intrinsic content size (Apple docs)



Created by Felipe Florencio Garcia

@dr_nerd

GitHub *https://github.com/felipeflorencio*

Content Compression resistance priorities.

Set the priority that a view resists being made smaller than it's intrinsic size. It's kind of the "opposite" from the "content hugging" that there we are "forcing" to fit, here we are saying that if do not fit the size that we set using autolayout, so look to our compression and if prioritie is higher them adjust the autolayout to match the text size.
Setting a higher value means that we want view fit the intrinsic content size and don't shrink smaller than the intrinsic content size.
Higher the priority is means larger the resistance to get shrunk.

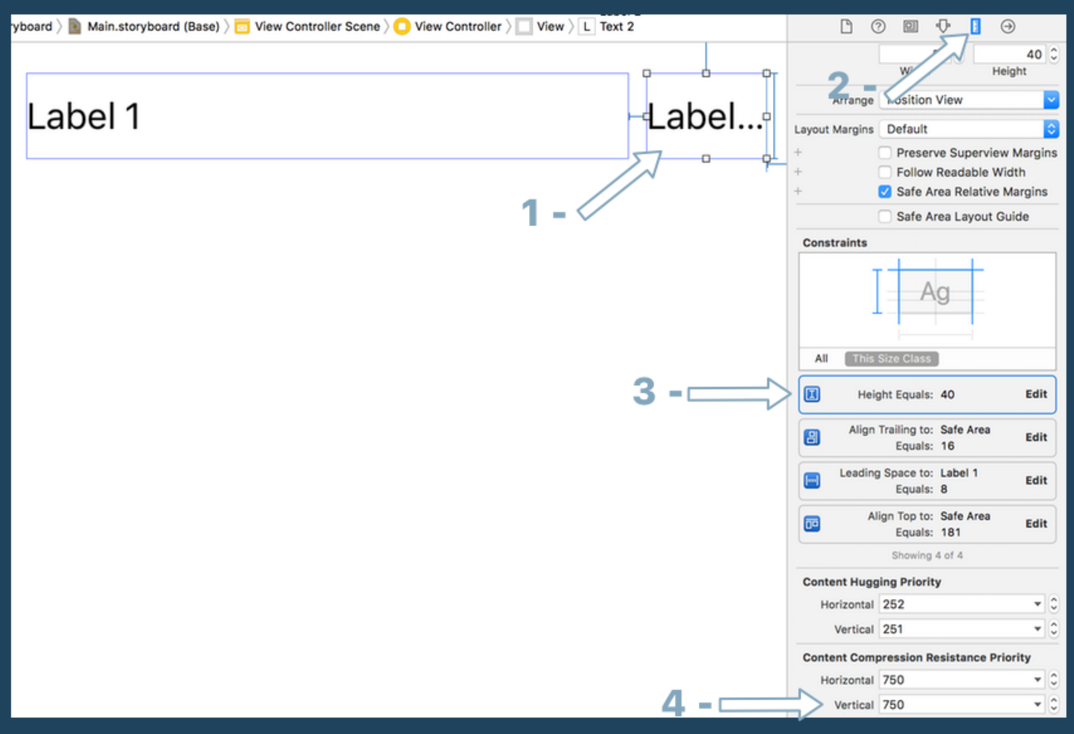
---

1. We will reuse the previous example, so, it's suppose that you have all the constraints working, you adjusted the "content hugging", but now, you want to add some text, but is one below the other, so the intention here is height be higher.

- For Label 2 add this text exactly like this:
Label 2
Text 2
Label 2

The problem it's quite easy to see, as the text do not fit or the text do not adjust or get higher, but it's right, as we set the height constraint to be fixed, we could set ">=", but if we have other's views with anchor to this label we will end up setting more constraint's and so one, and for this the right solution is to use the content compression resistance priority.

2. Now what we gonna do is , select the 1 - "*Label 2*", select 2 - "*size inspector*" if not selected, 3 - look for the "*height*" constraint, click to edit and set "*Priority*" to "*999*", 4 - below you will see the "*Content Compression Resistance Priority*", set the "*Vertical*" value to "*1000*".



3. And that's it, you will see that the "view" from the label will fit the text that we added without need to set any constraint higher than or something like, as we said that our height is a fixed height so any other constraint that anchor on this label will not complaing about ambiguity, but, if you want this label to expand his height according to the text will work too, because when autolayout need to see who has priority will consult the "*compression resistance priority*" and will know how to adjust!