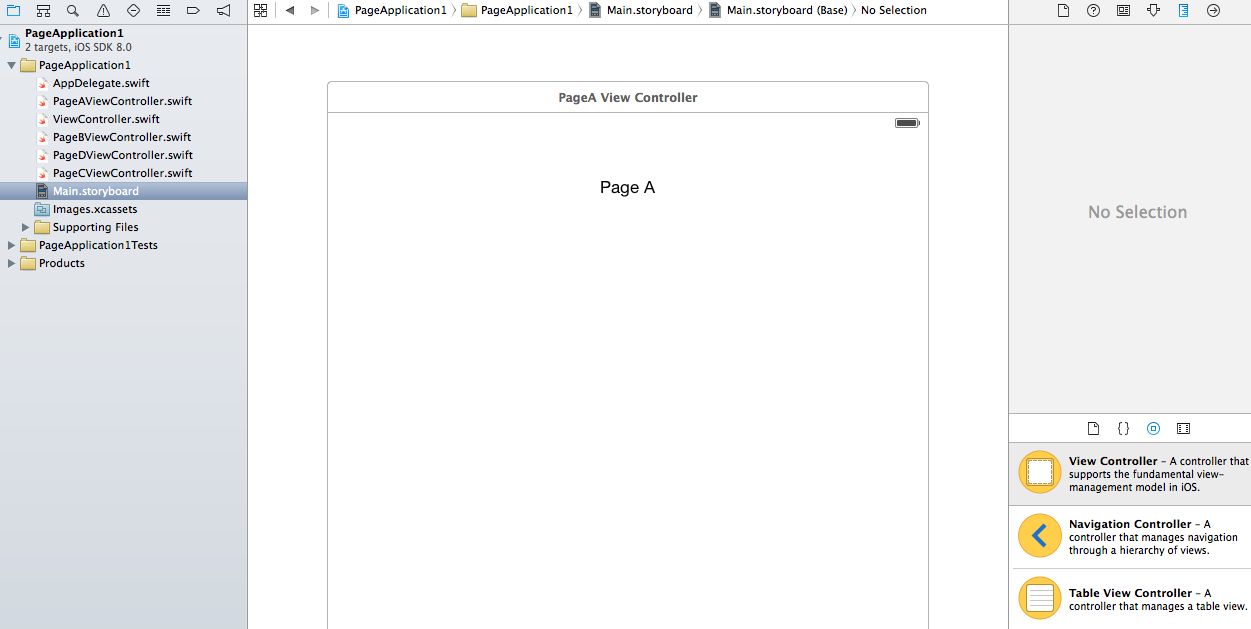
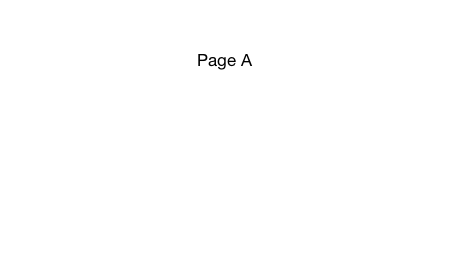
Xcodes interface builder offers an easy way to create views for your applications. Simply drag and dropping particular items such as labels, and buttons. Unlike Java, where you must programmatically do all the work, once you create the items in the views, you can drag the item to your code and it will create a reference to access its functionality. Its very confusing because its such a new language and interface, there aren’t many tutorials out yet and apples documentation is, well, dry.

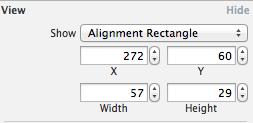
Interface Builder



Understanding how the placement of objects works is confusing. When you initially create something you can place it directly on the view you want.

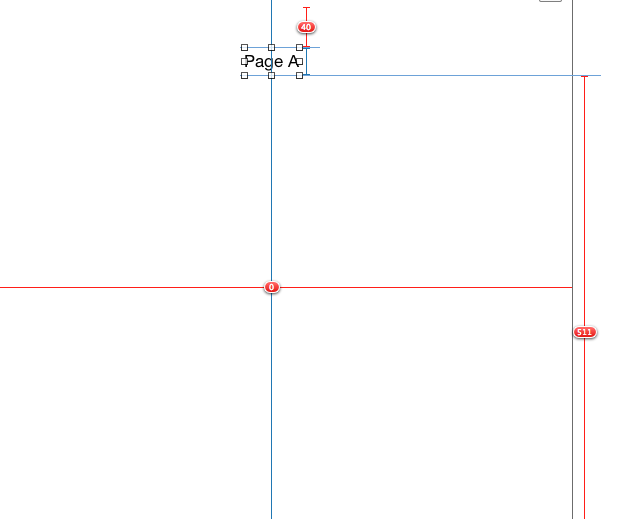


Page A is a Label object, the coordinates for this object are

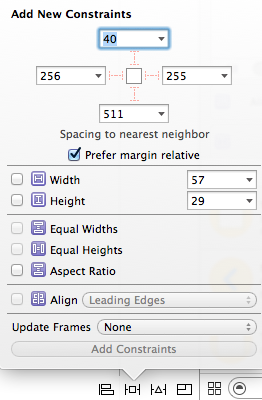
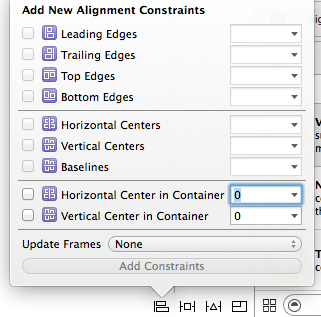


But these coordinates will not hold when you run a simulator, which I still don’t know the reasoning for this. What needs to be done is apply constraints for the objects so they’re held in place when the simulator runs

Horizontal/Vertical Center Alignment constraints are designated with the lines. They are red and orange if the coordinates above do not match the constraints



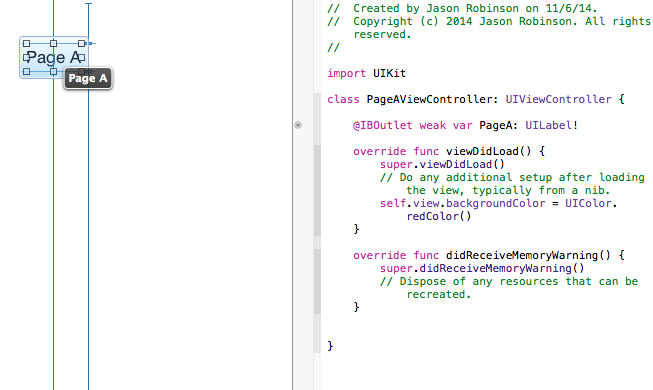
To set the constraints you select your object and apply them with these buttons



Once you’ve positioned your objects to your liking, you want to link your object to your code if its going to be altered or accessed in any way. You can change the objects title info, font style

and size, color, shading, and other properties within the interface builder or using code.

To link the objects to your code you can drag and drop the desired object to the code, or create an object in your code and drag that to which ever view controller holds the object



“@IBOutlet weak var PageA: UILabel!” is the code for a Label in a View, the small dot in the middle of the screen is solid, which means the code is properly linked to some object in the view. If it is hollow then its just empty code.

“weak” refers to the garbage collection preference, in the case of weak, it will be destroyed after usage. Inferring “strong” will hold the data until directly destroyed. “var” defines a regular variable, you can either explicitly declare the type, or let xcode do it for you. “let” would define constants. “UILabel” is the class declaration for the class UIViewController, which is extended into this new class PageAViewController.