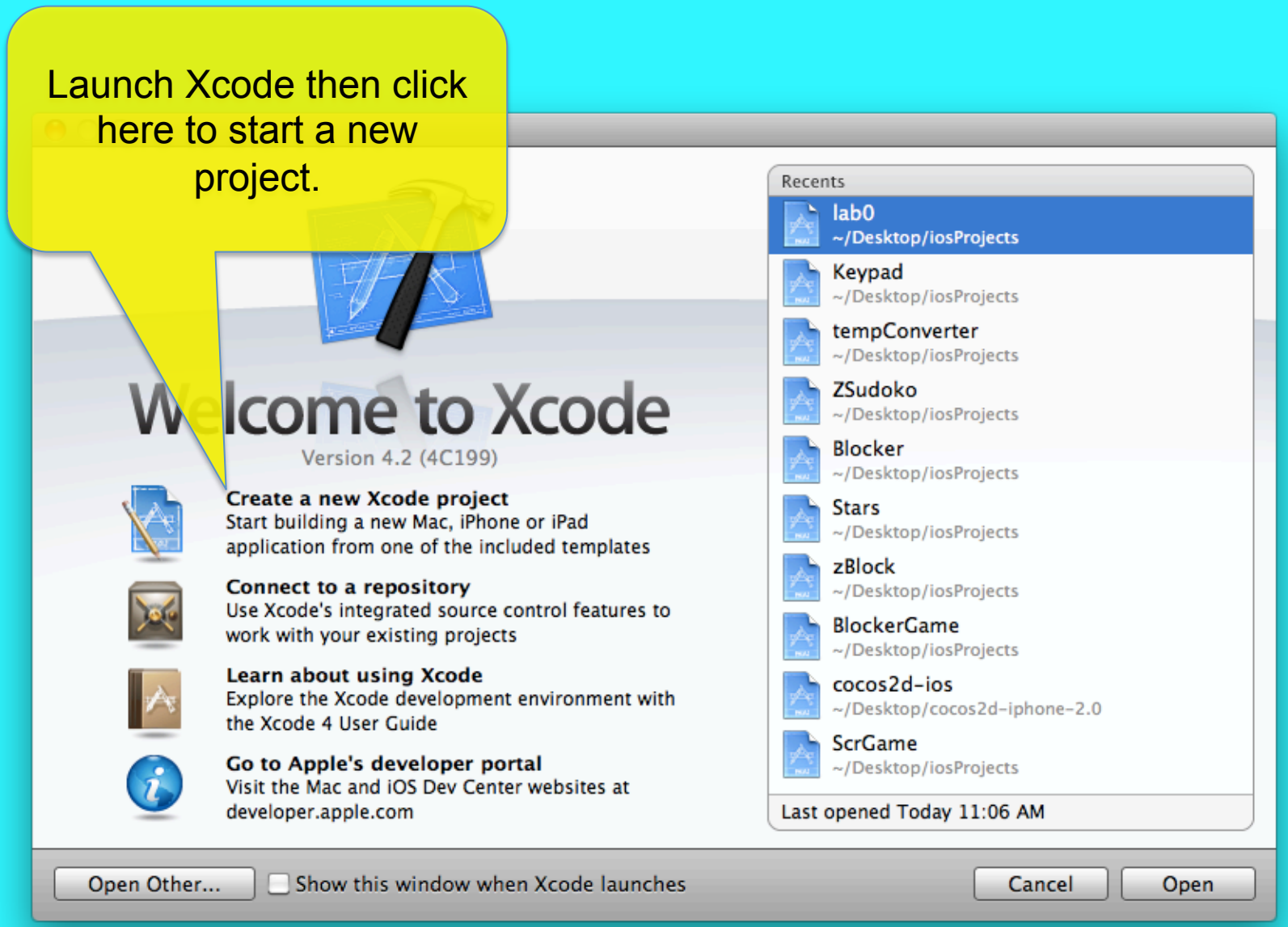
CS121 iOS Tutorial 1

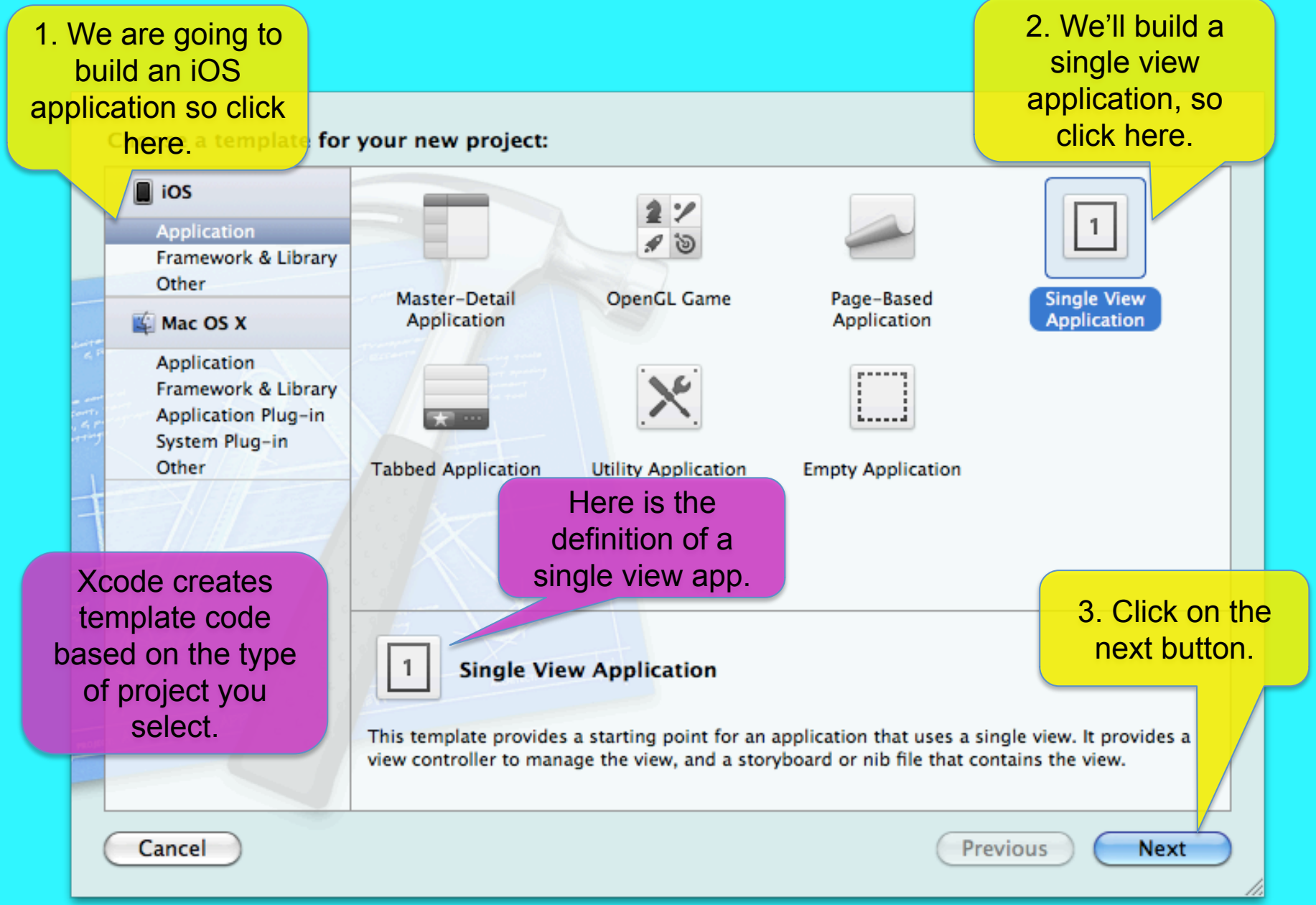
This tutorial provides a basic introduction to Xcode, iOS development and Objective-C. In the process you'll develop a simple app.

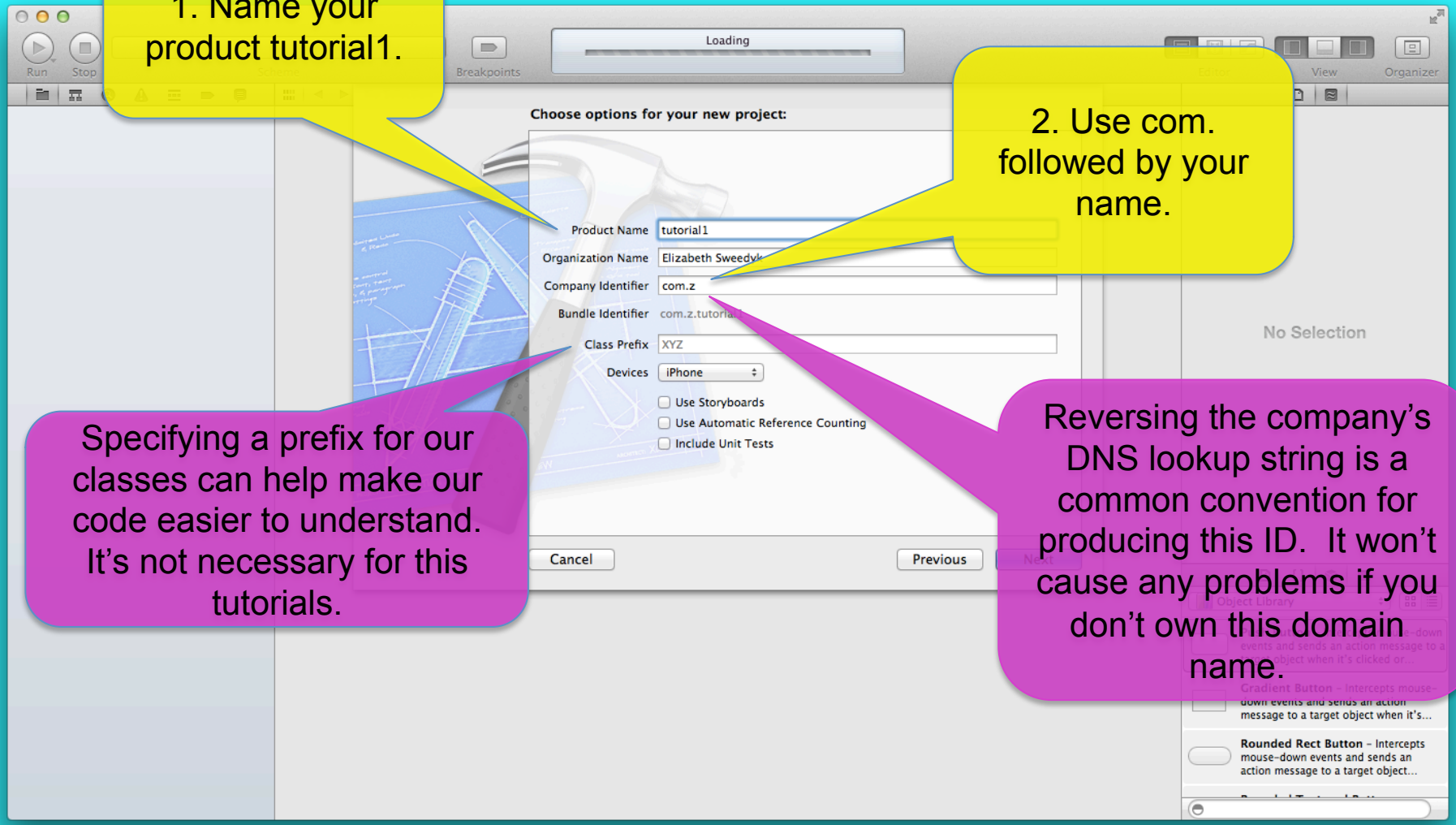
Purple bubbles give you information you'll need to know.

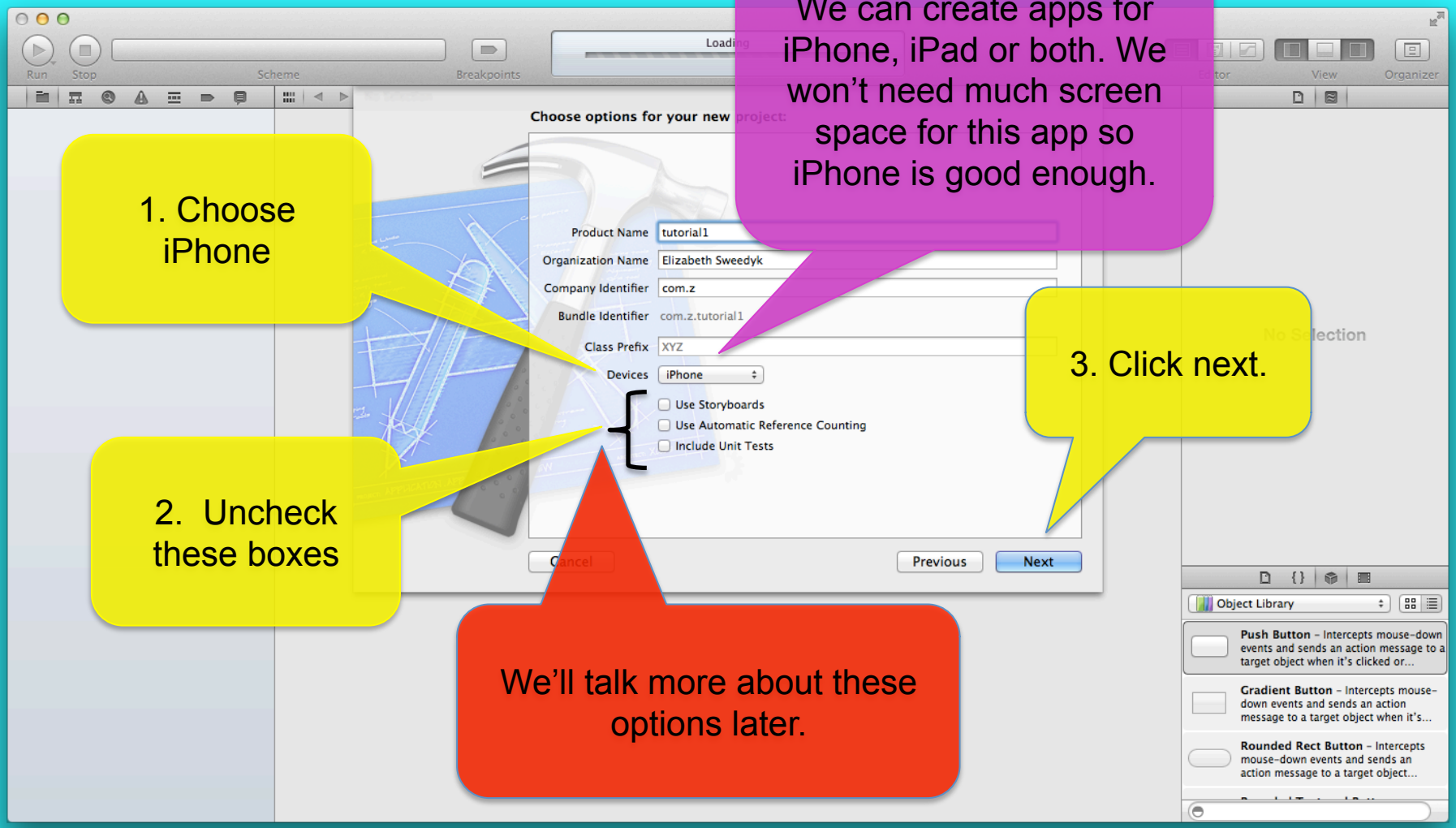
Yellow Bubbles tell you what to do.

Orange bubbles tell you what you're not expected to understand yet. ☺


Launch Xcode then click here to start a new project.
Welcome to Xcode
Version 4.2 (4C199)
Create a new Xcode project
Start building a new Mac, iPhone or iPad application from one of the included templates
Connect to a repository
Use Xcode's integrated source control features to work with your existing projects
Learn about using Xcode
Explore the Xcode development environment with the Xcode 4 User Guide
Go to Apple's developer portal
Visit the Mac and iOS Dev Center websites at developer.apple.com
Recents
lab0
~/Desktop/iosProjects
Keypad
~/Desktop/iosProjects
tempConverter
~/Desktop/iosProjects
ZSudoko
~/Desktop/iosProjects
Blocker
~/Desktop/iosProjects
Stars
~/Desktop/iosProjects
zBlock
~/Desktop/iosProjects
BlockerGame
~/Desktop/iosProjects
cocos2d-ios
~/Desktop/cocos2d-iphone-2.0
ScrGame
~/Desktop/iosProjects
Last opened Today 11:06 AM
Open Other...
Show this window when Xcode launches
Cancel
Open

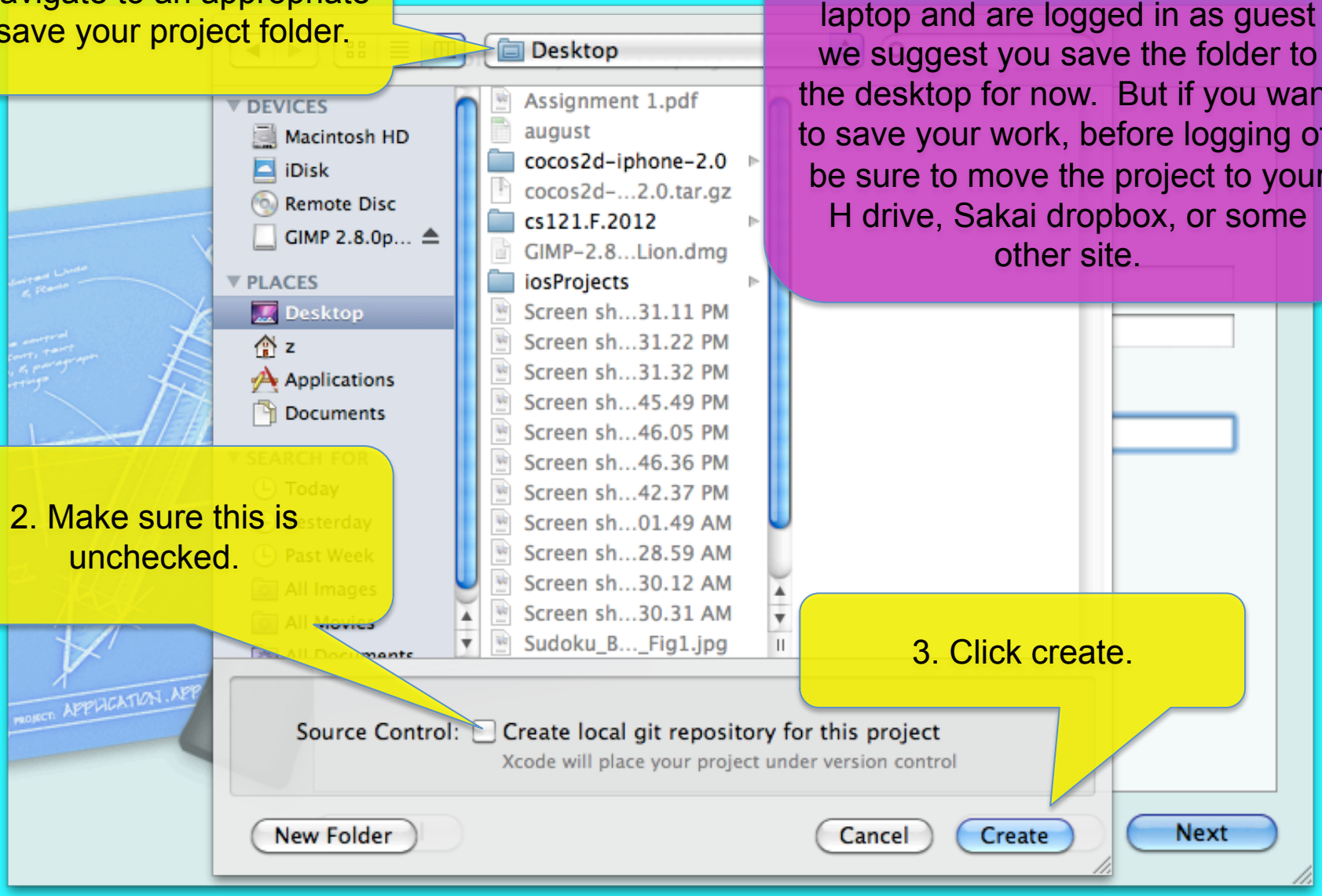

1. We are going to build an iOS application so click here.
2. We'll build a single view application, so click here.
for your new project:
iOS
Application
Framework & Library
Other
Mac OS X
Application
Framework & Library
Application Plug-in
System Plug-in
Other
Master-Detail Application
OpenGL Game
Page-Based Application
Single View Application
Tabbed Application
Utility Application
Empty Application
Here is the definition of a single view app.
Xcode creates template code based on the type of project you select.
3. Click on the next button.
Single View Application
This template provides a starting point for an application that uses a single view. It provides a view controller to manage the view, and a storyboard or nib file that contains the view.
Cancel
Previous
Next


1. Name your product tutorial1.
2. Use com. followed by your name.
Specifying a prefix for our classes can help make our code easier to understand. It's not necessary for this tutorials.
Reversing the company's DNS lookup string is a common convention for producing this ID. It won't cause any problems if you don't own this domain name.
Choose options for your new project:
Product Name
tutorial1
Organization Name
Company Identifier
com.z
Bundle Identifier
Class Prefix
XYZ
Devices
iPhone
Use Storyboards
Use Automatic Reference Counting
Include Unit Tests
Cancel
Previous
No Selection
Loading
Run
Stop
Breakpoints
View
Organizer


We can create apps for iPhone, iPad or both. We won't need much screen space for this app so iPhone is good enough.
1. Choose iPhone
2. Uncheck these boxes
3. Click next.
We'll talk more about these options later.
Choose options for your new project:
Product Name
tutorial1
Organization Name
Elizabeth Sweedyk
Company Identifier
com.z
Bundle Identifier
com.z.tutorial1
Class Prefix
XYZ
Devices
iPhone
Use Storyboards
Use Automatic Reference Counting
Include Unit Tests
Cancel
Previous
Next
Object Library
Push Button – Intercepts mouse-down events and sends an action message to a target object when it's clicked or...
Gradient Button – Intercepts mouse-down events and sends an action message to a target object when it's...
Rounded Rect Button – Intercepts mouse-down events and sends an action message to a target object...

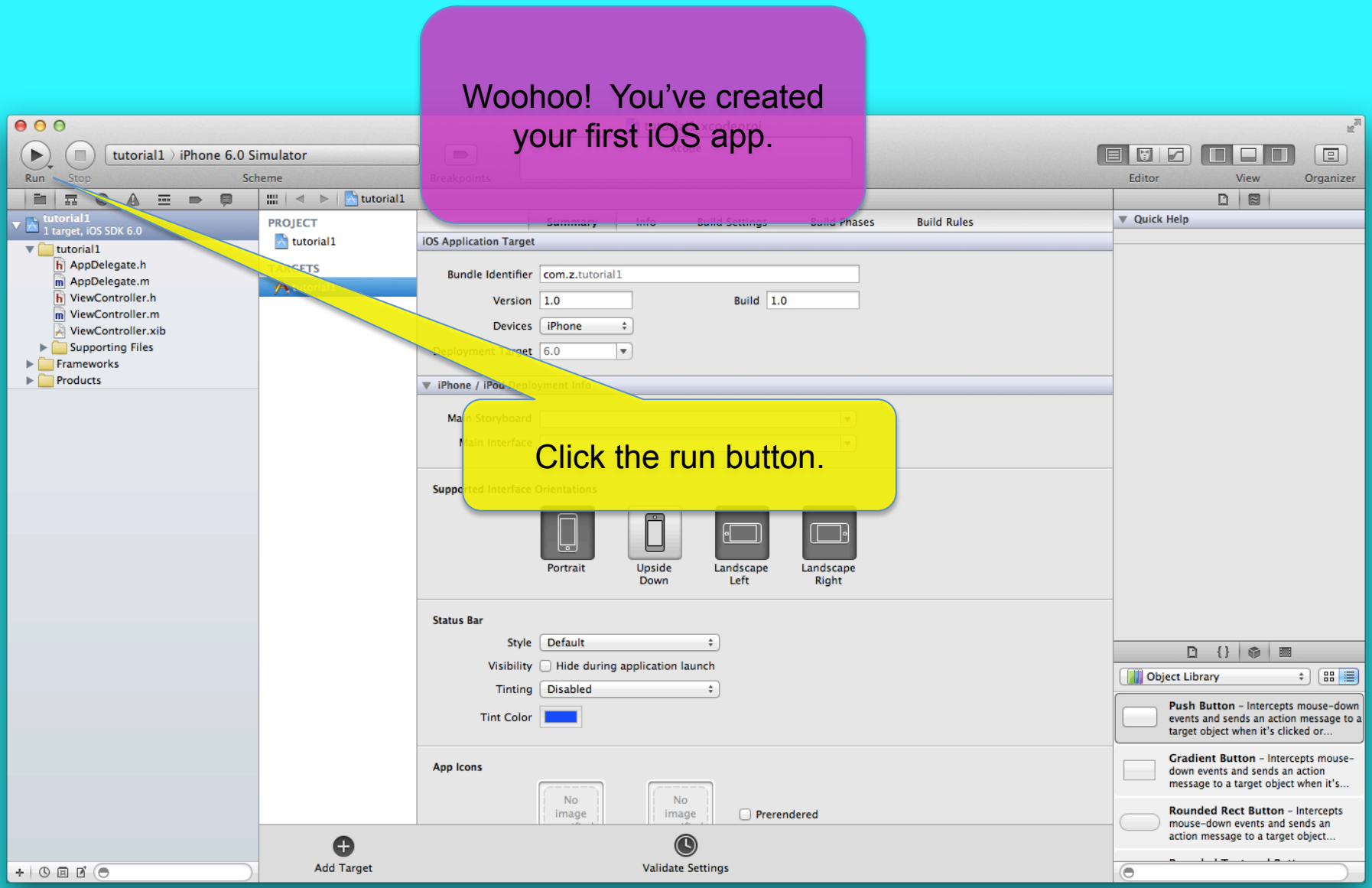
1. Navigate to an appropriate to save your project folder.

WARNING: If you are using a CIS laptop and are logged in as guest we suggest you save the folder to the desktop for now. But if you want to save your work, before logging off be sure to move the project to your H drive, Sakai dropbox, or some other site.

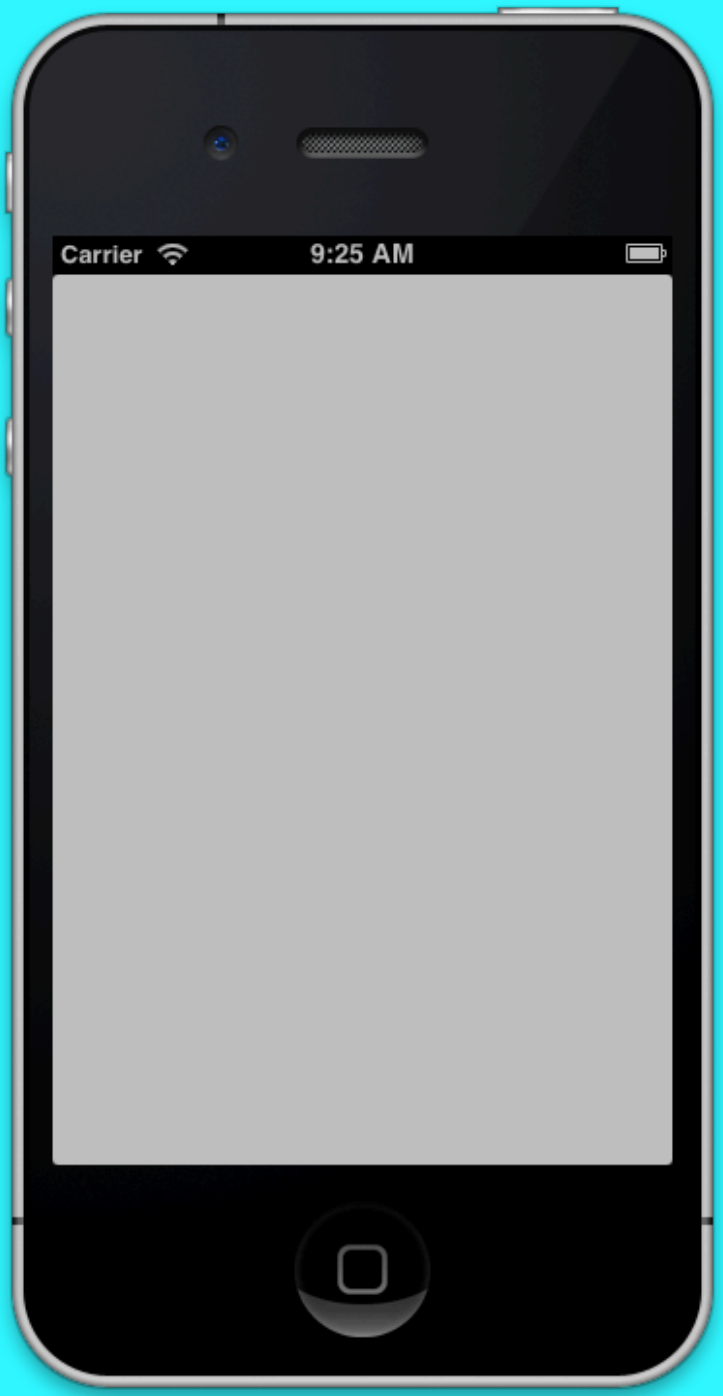


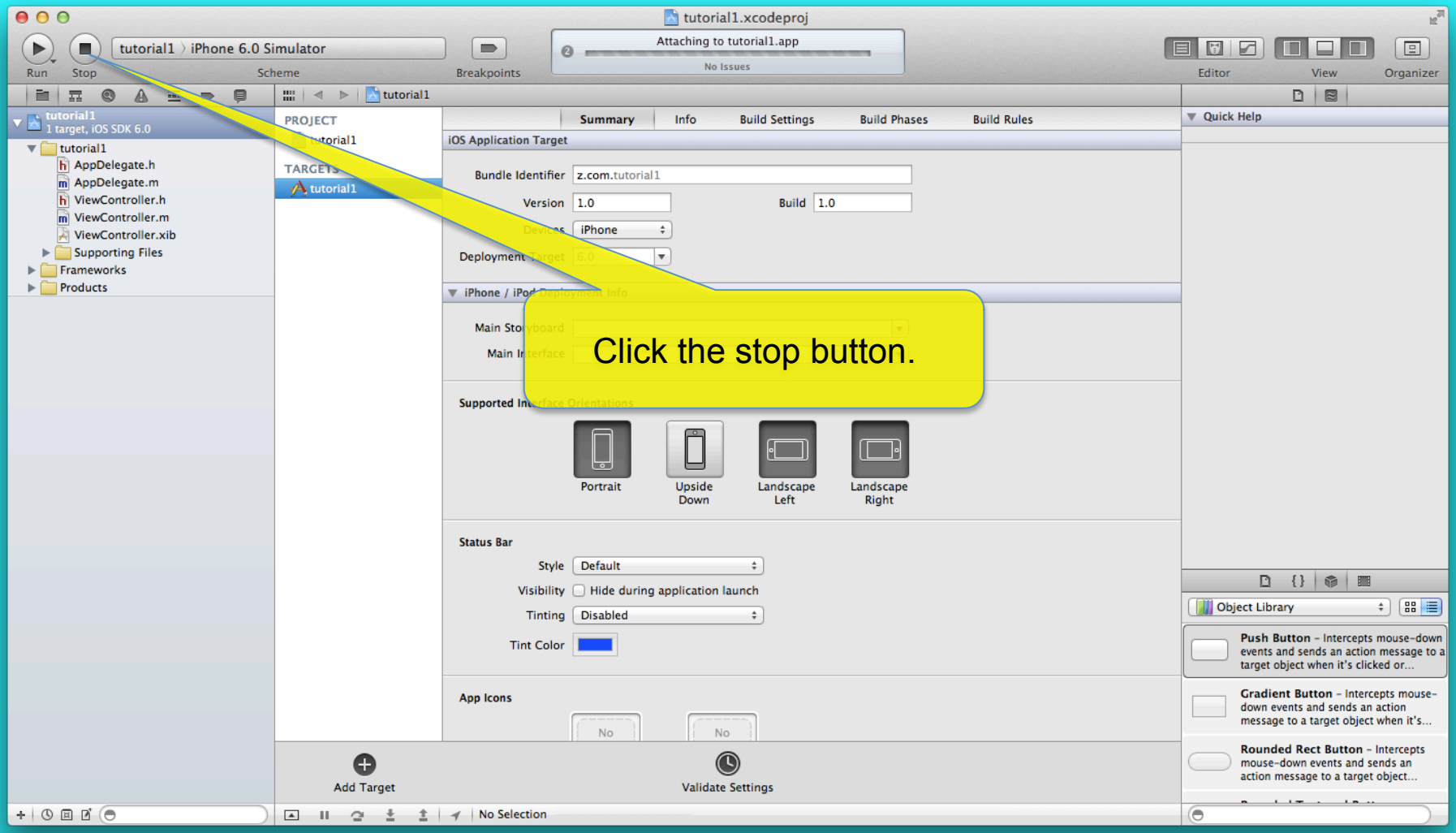
2. Make sure this is unchecked.

3. Click create.

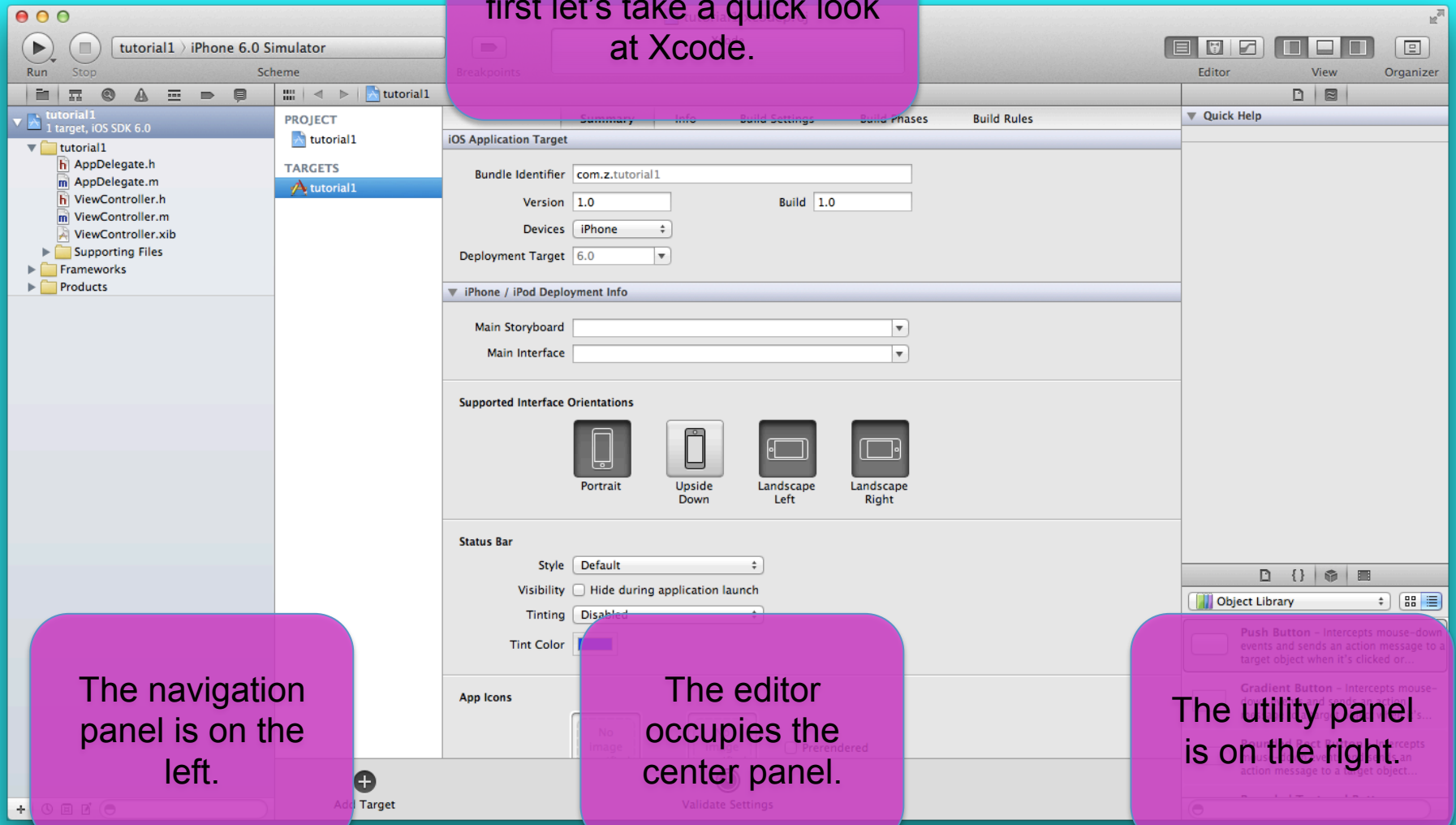

Woohoo! You've created your first iOS app.
Click the run button.
Run
Stop
tutorial1 〉 iPhone 6.0 Simulator
Scheme
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
PROJECT
TARGETS
Summary
Info
Build Settings
Build Phases
Build Rules
iOS Application Target
Bundle Identifier
com.z.tutorial1
Version
1.0
Build
1.0
Devices
iPhone
Deployment Target
6.0
Main Storyboard
Supported Interface Orientations
Portrait
Upside Down
Landscape Left
Landscape Right
Status Bar
Style
Default
Visibility
Hide during application launch
Tinting
Disabled
Tint Color
App Icons
No image
Prerendered
Add Target
Validate Settings
Quick Help
Object Library
Push Button – Intercepts mouse-down events and sends an action message to a target object when it's clicked or...
Gradient Button – Intercepts mouse-down events and sends an action message to a target object when it's...
Rounded Rect Button – Intercepts mouse-down events and sends an action message to a target object...

The app is now running in the iPhone simulator, which appears on your desktop.




tutorial1.xcodeproj
Run
Stop
tutorial1 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Attaching to tutorial1.app
No Issues
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
PROJECT
TARGETS
Summary
Info
Build Settings
Build Phases
Build Rules
iOS Application Target
Bundle Identifier
z.com.tutorial1
Version
1.0
Build
1.0
iPhone
Click the stop button.
Portrait
Upside Down
Landscape Left
Landscape Right
Status Bar
Style
Default
Visibility
Hide during application launch
Tinting
Disabled
Tint Color
App Icons
Add Target
Validate Settings
No Selection
Quick Help
Object Library
Push Button – Intercepts mouse-down events and sends an action message to a target object when it's clicked or...
Gradient Button – Intercepts mouse-down events and sends an action message to a target object when it's...
Rounded Rect Button – Intercepts mouse-down events and sends an action message to a target object...

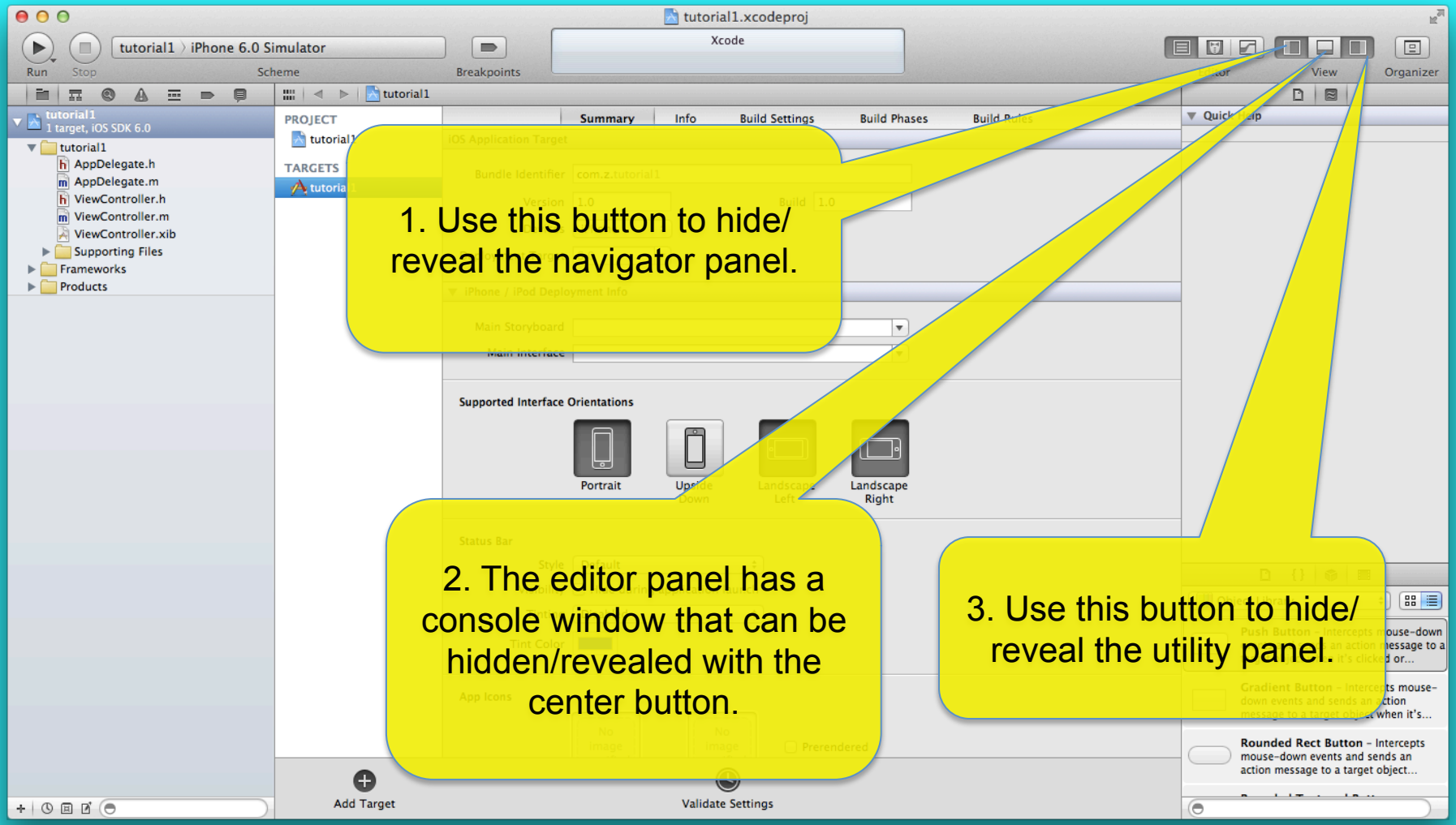
We'll add to the app but first let's take a quick look at Xcode.

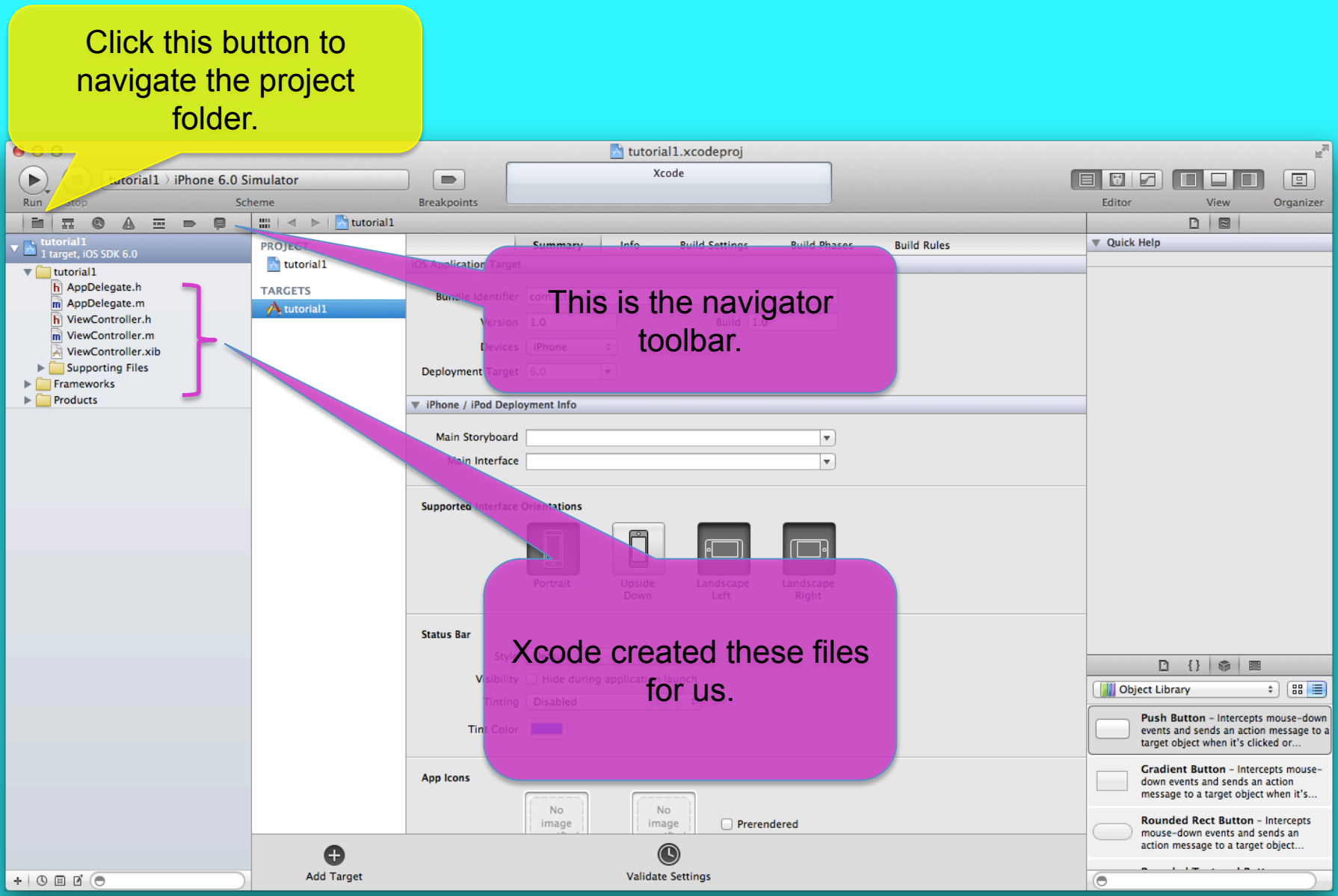


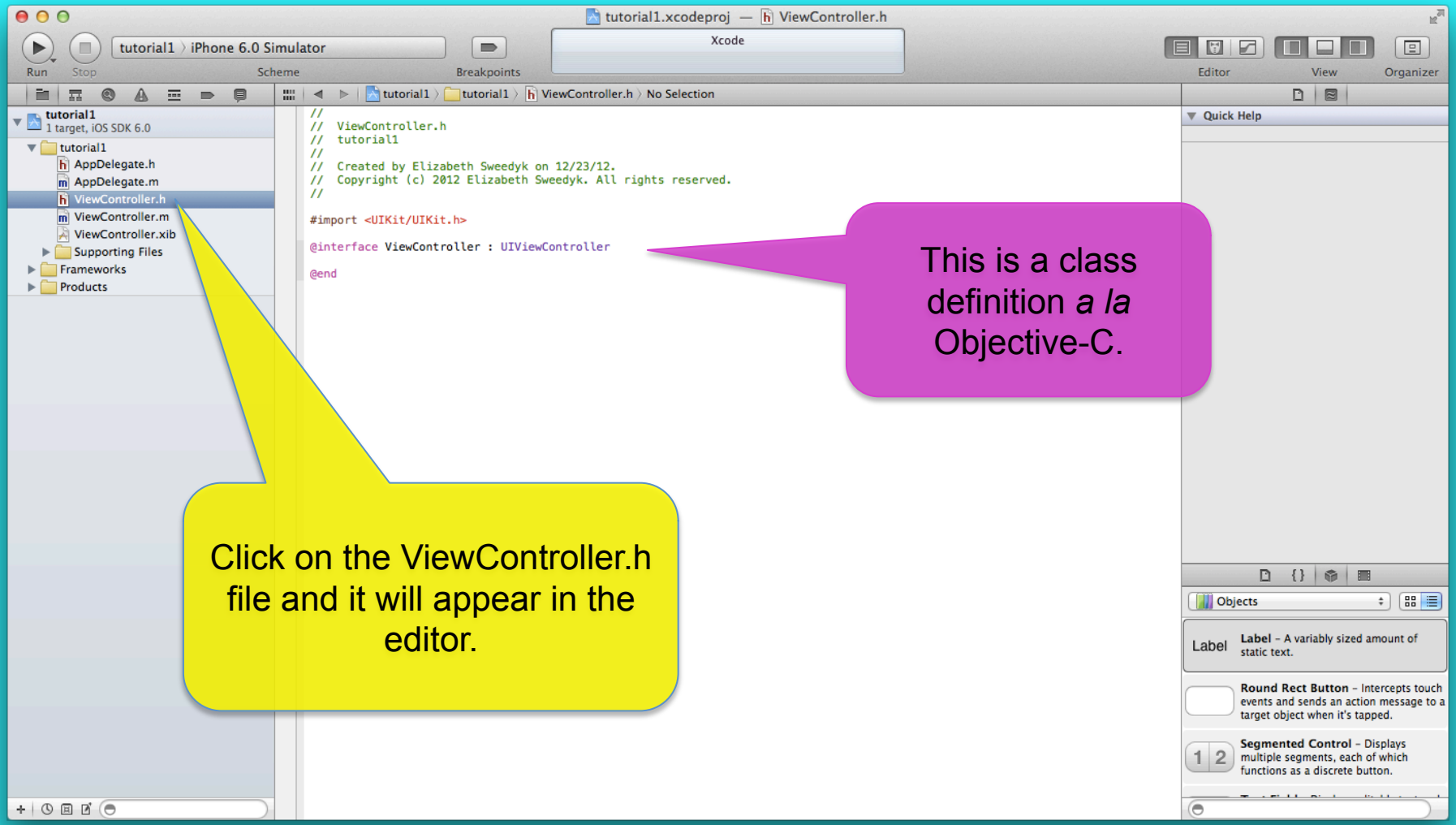
The navigation panel is on the left.

The editor occupies the center panel.

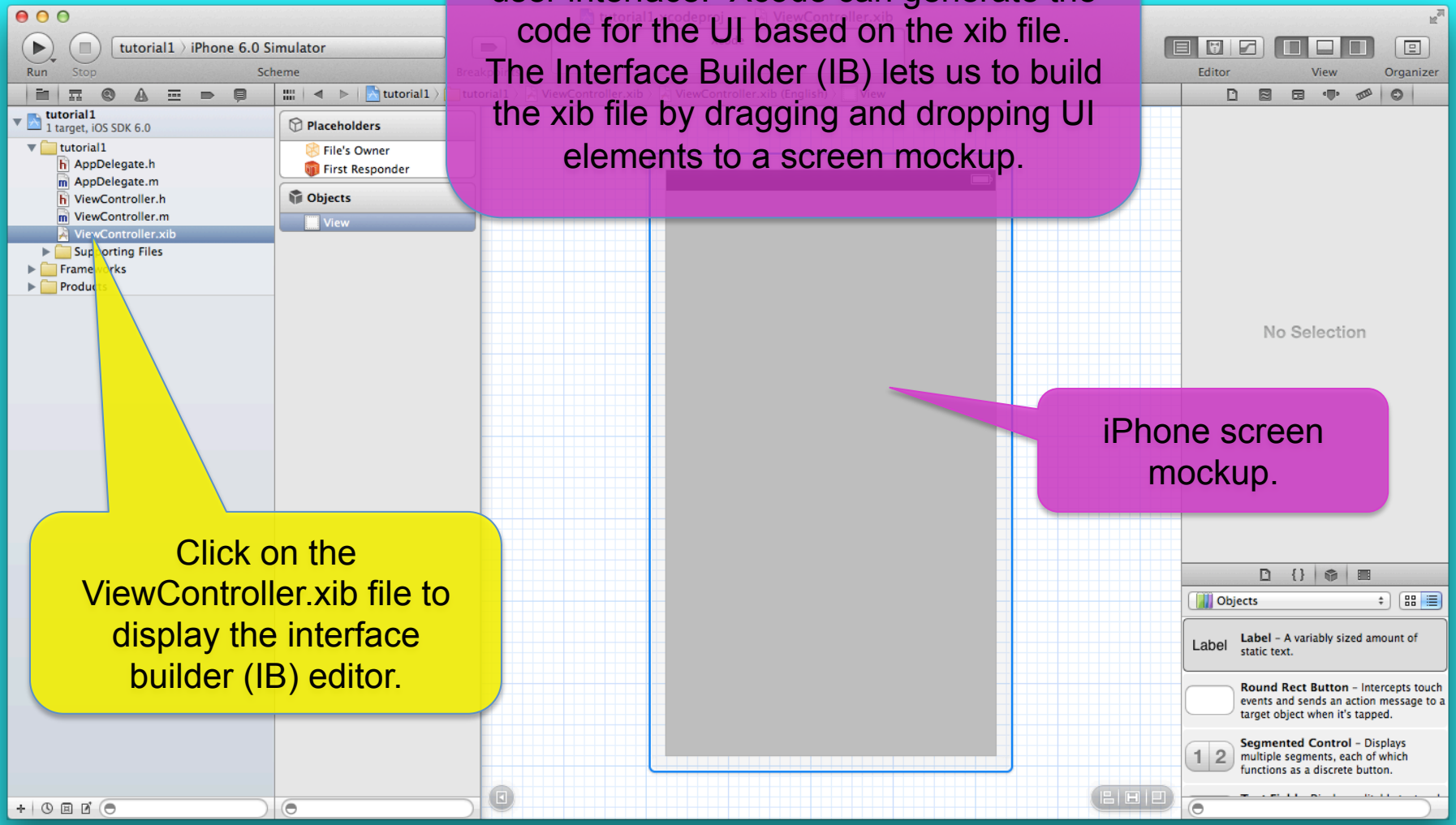
The utility panel is on the right.


tutorial1.xcodeproj
Run
Stop
tutorial1 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Xcode
View
Organizer
tutorial1
1 target, iOS SDK 6.0
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
PROJECT
TARGETS
Summary
Info
Build Settings
Build Phases
Supported Interface Orientations
Portrait
Landscape Right
1. Use this button to hide/ reveal the navigator panel.
2. The editor panel has a console window that can be hidden/revealed with the center button.
3. Use this button to hide/ reveal the utility panel.
Rounded Rect Button – Intercepts mouse-down events and sends an action message to a target object...
Add Target
Validate Settings


Click this button to navigate the project folder.
tutorial1.xcodeproj
Run
Scheme
Breakpoints
Xcode
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
PROJECT
TARGETS
Summary
Info
Build Settings
Build Phases
Build Rules
Quick Help
This is the navigator toolbar.
Deployment Target
iPhone / iPod Deployment Info
Main Storyboard
Main Interface
Supported Interface Orientations
Portrait
Upside Down
Landscape Left
Landscape Right
Status Bar
Tint Color
Xcode created these files for us.
App Icons
No image
Prerendered
Object Library
Push Button – Intercepts mouse-down events and sends an action message to a target object when it's clicked or...
Gradient Button – Intercepts mouse-down events and sends an action message to a target object when it's...
Rounded Rect Button – Intercepts mouse-down events and sends an action message to a target object...
Add Target
Validate Settings

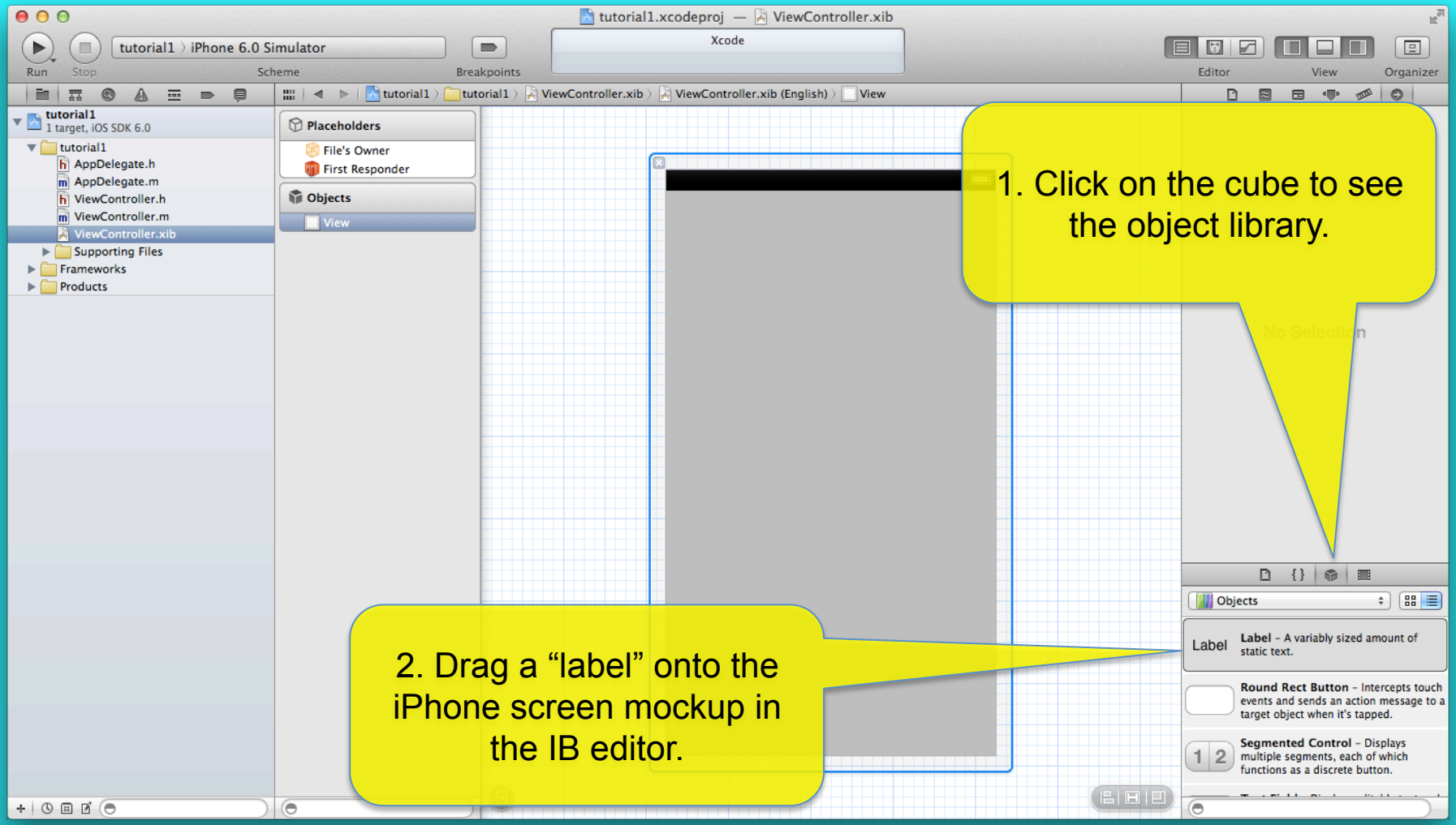

tutorial1.xcodeproj — ViewController.h
Run
Stop
tutorial1 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Xcode
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
tutorial1
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
tutorial1 〉 tutorial1 〉 ViewController.h 〉 No Selection
//
// ViewController.h
// tutorial1
//
// Created by Elizabeth Sweedyk on 12/23/12.
// Copyright (c) 2012 Elizabeth Sweedyk. All rights reserved.
//
#import <UIKit/UIKit.h>
@interface ViewController : UIViewController
@end
Quick Help
Objects
Label
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
1 2
Segmented Control – Displays multiple segments, each of which functions as a discrete button.
This is a class definition a la Objective-C.
Click on the ViewController.h file and it will appear in the editor.

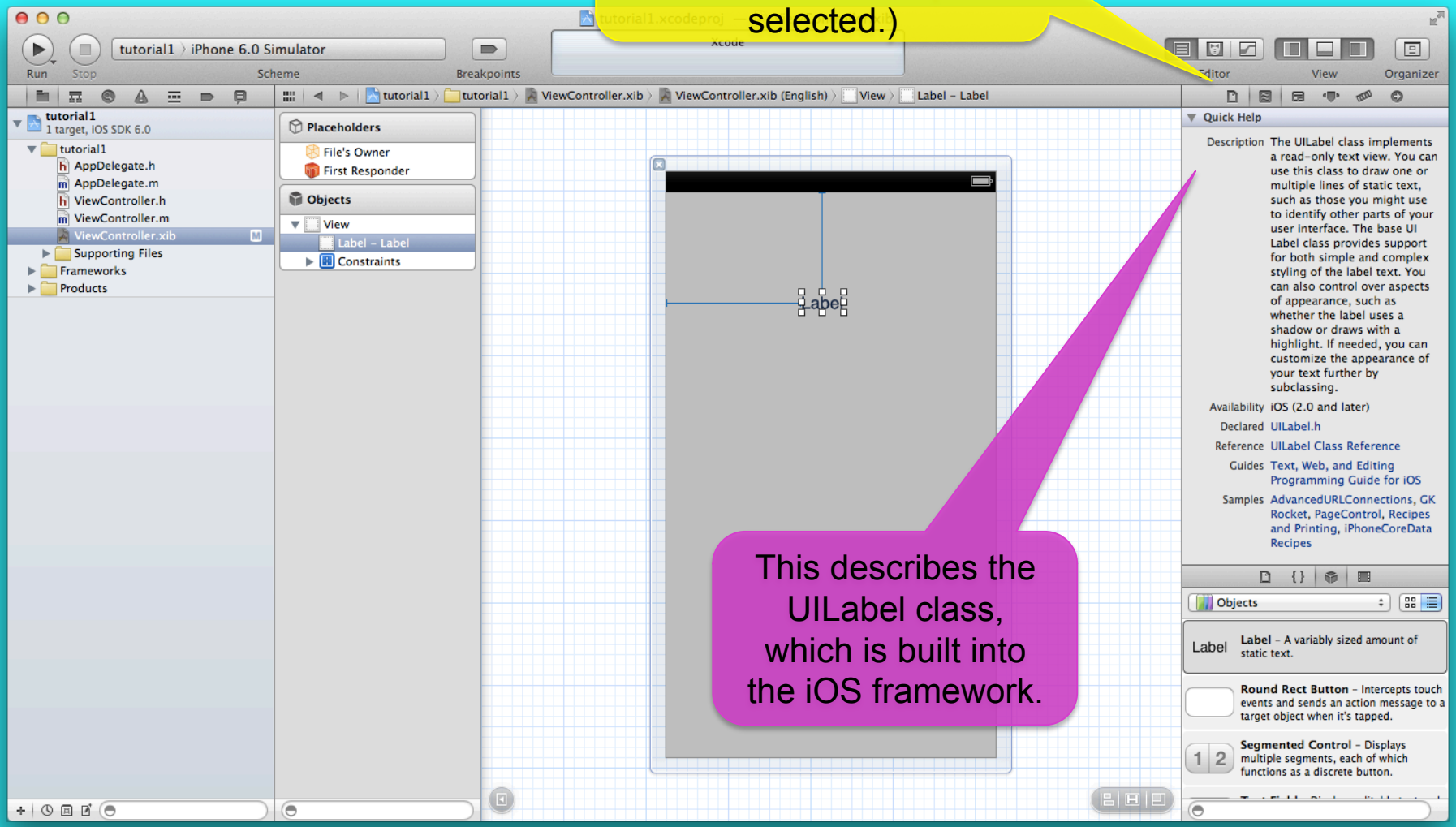
A xib file is an XML file that describes the user interface. Xcode can generate the code for the UI based on the xib file. The Interface Builder (IB) lets us to build the xib file by dragging and dropping UI elements to a screen mockup.

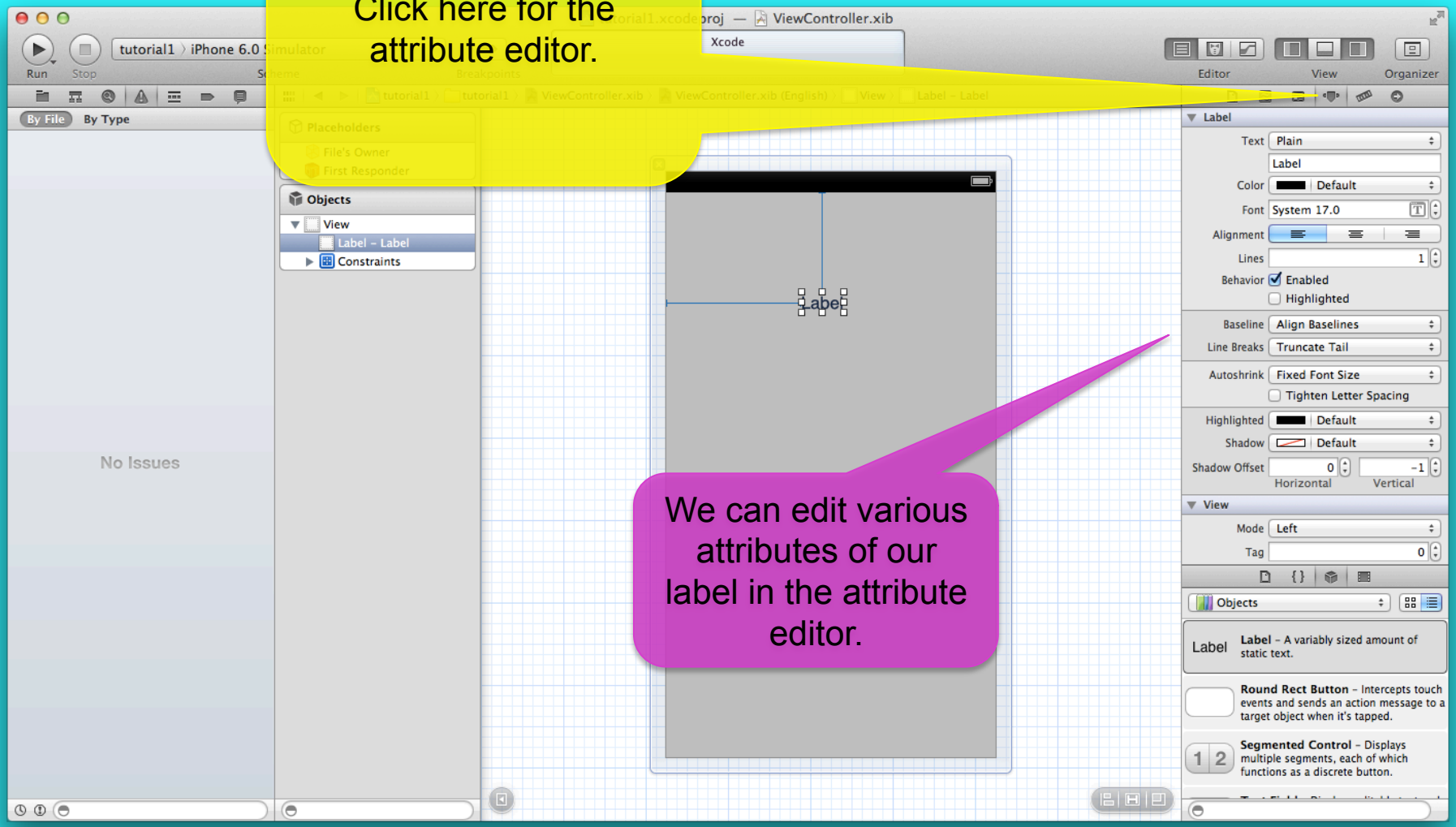


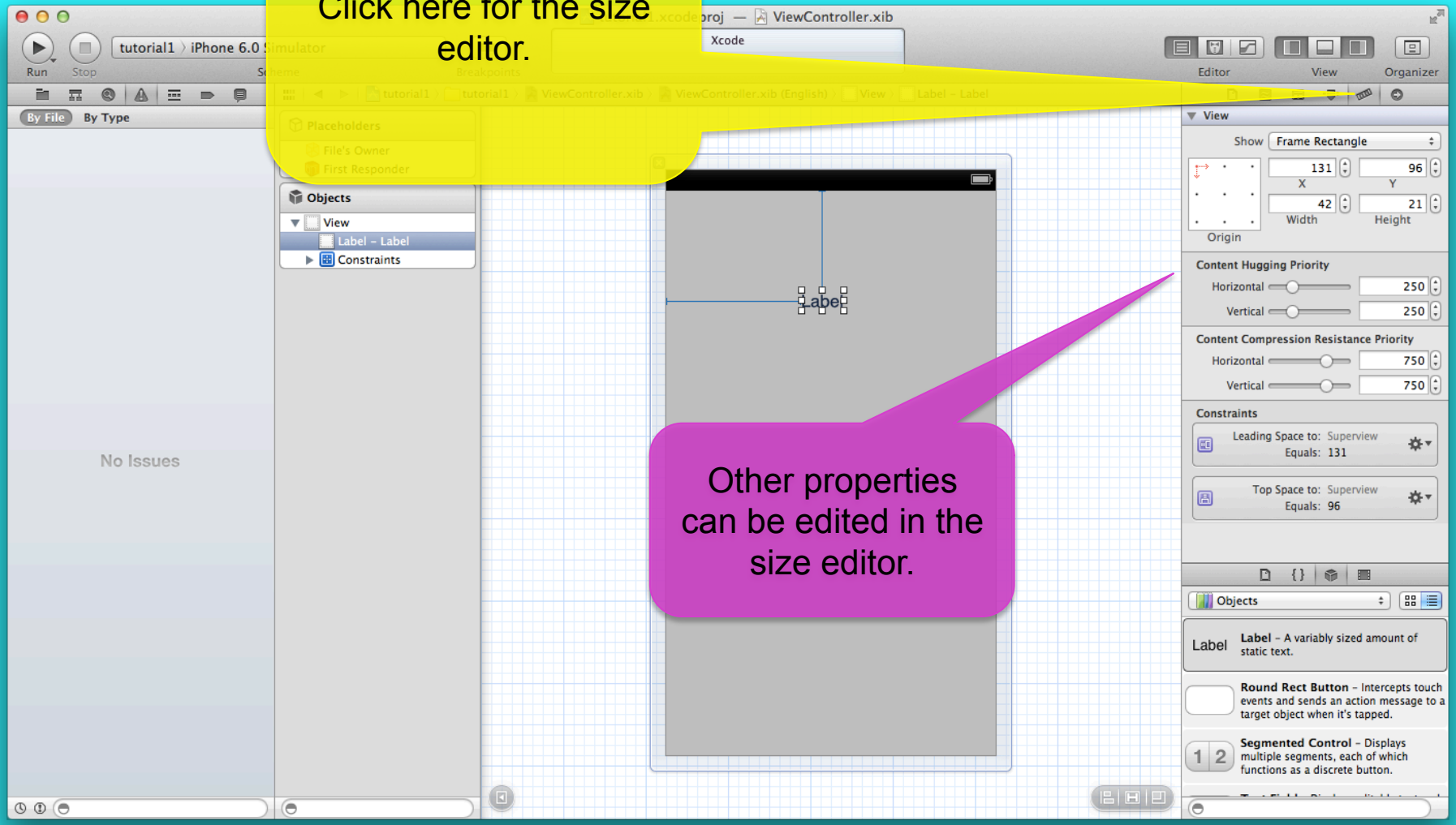
Click on the ViewController.xib file to display the interface builder (IB) editor.

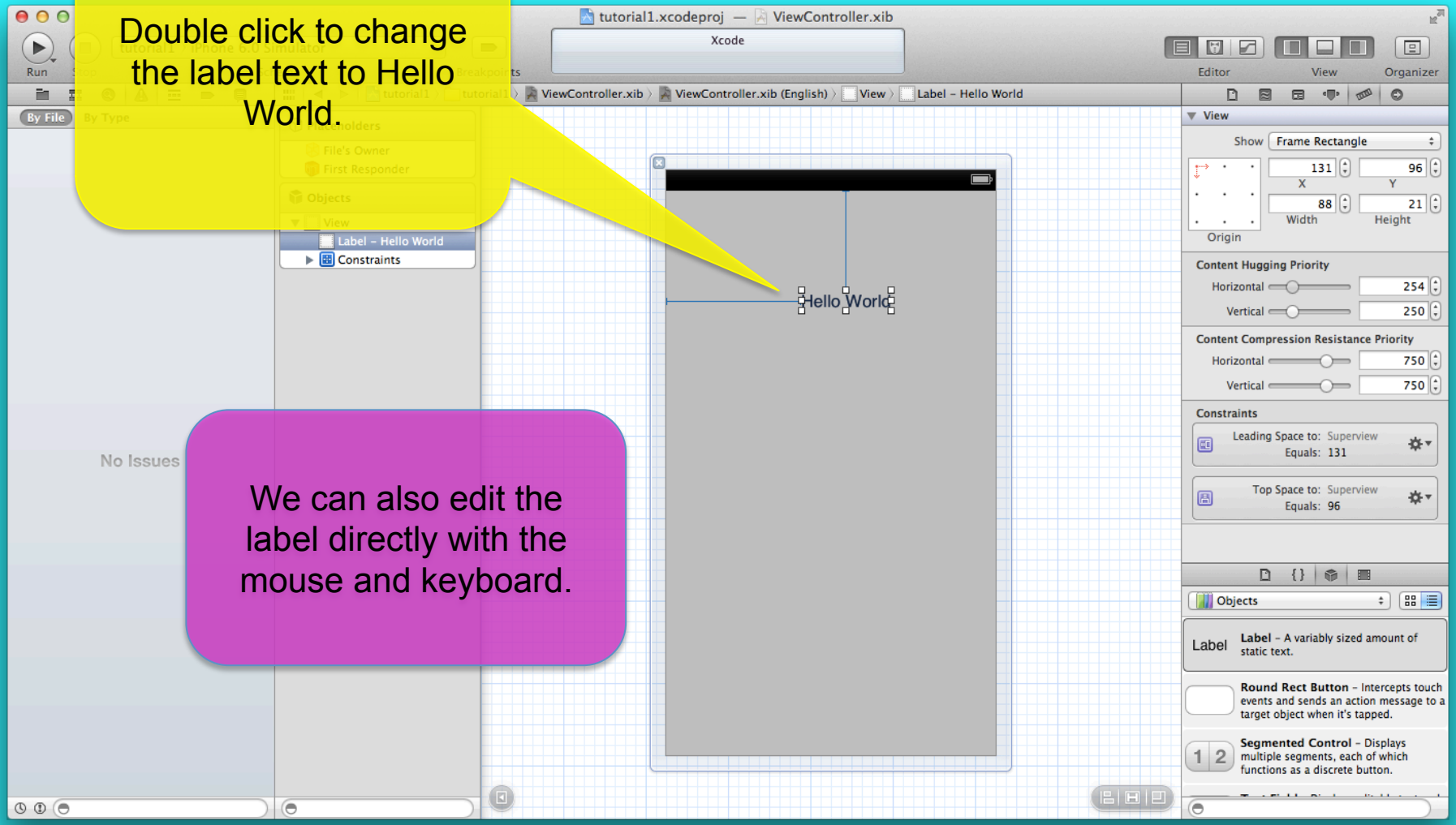
iPhone screen mockup.

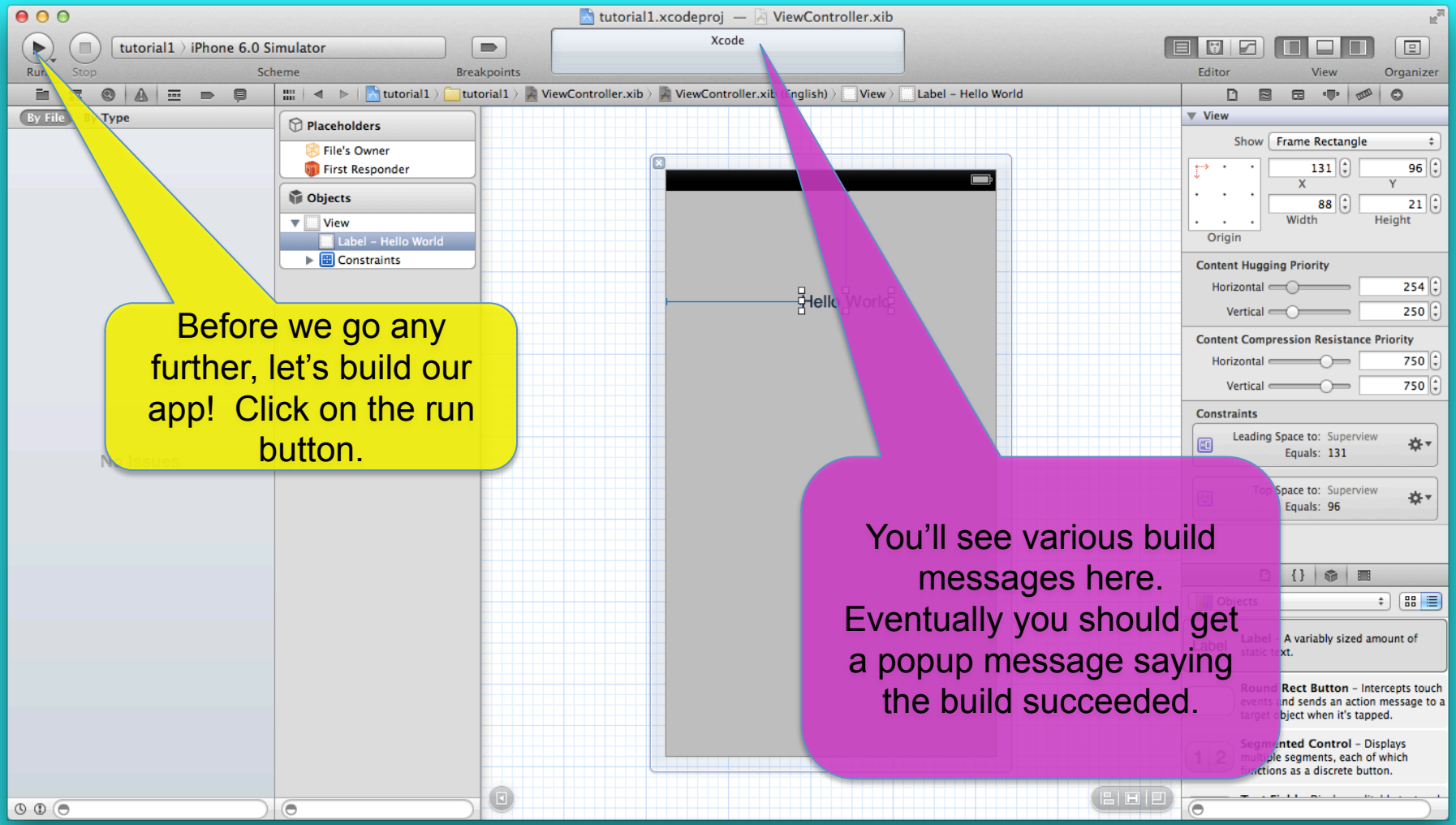

tutorial1.xcodeproj — ViewController.xib
Run
Stop
tutorial1 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Xcode
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
tutorial1
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
Placeholders
File's Owner
First Responder
Objects
View
1. Click on the cube to see the object library.
2. Drag a “label” onto the iPhone screen mockup in the IB editor.
No Selection
Objects
Label
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.


Click here for quick help. (Make sure the label on the screen mockup is selected.)
This describes the UILabel class, which is built into the iOS framework.
tutorial1
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
Placeholders
File's Owner
First Responder
Objects
View
Label – Label
Constraints
Label
Quick Help
Description
The UILabel class implements a read-only text view. You can use this class to draw one or multiple lines of static text, such as those you might use to identify other parts of your user interface. The base UI Label class provides support for both simple and complex styling of the label text. You can also control over aspects of appearance, such as whether the label uses a shadow or draws with a highlight. If needed, you can customize the appearance of your text further by subclassing.
Availability
iOS (2.0 and later)
Declared
UILabel.h
Reference
UILabel Class Reference
Guides
Text, Web, and Editing Programming Guide for iOS
Samples
AdvancedURLConnections, GK Rocket, PageControl, Recipes and Printing, iPhoneCoreData Recipes
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.

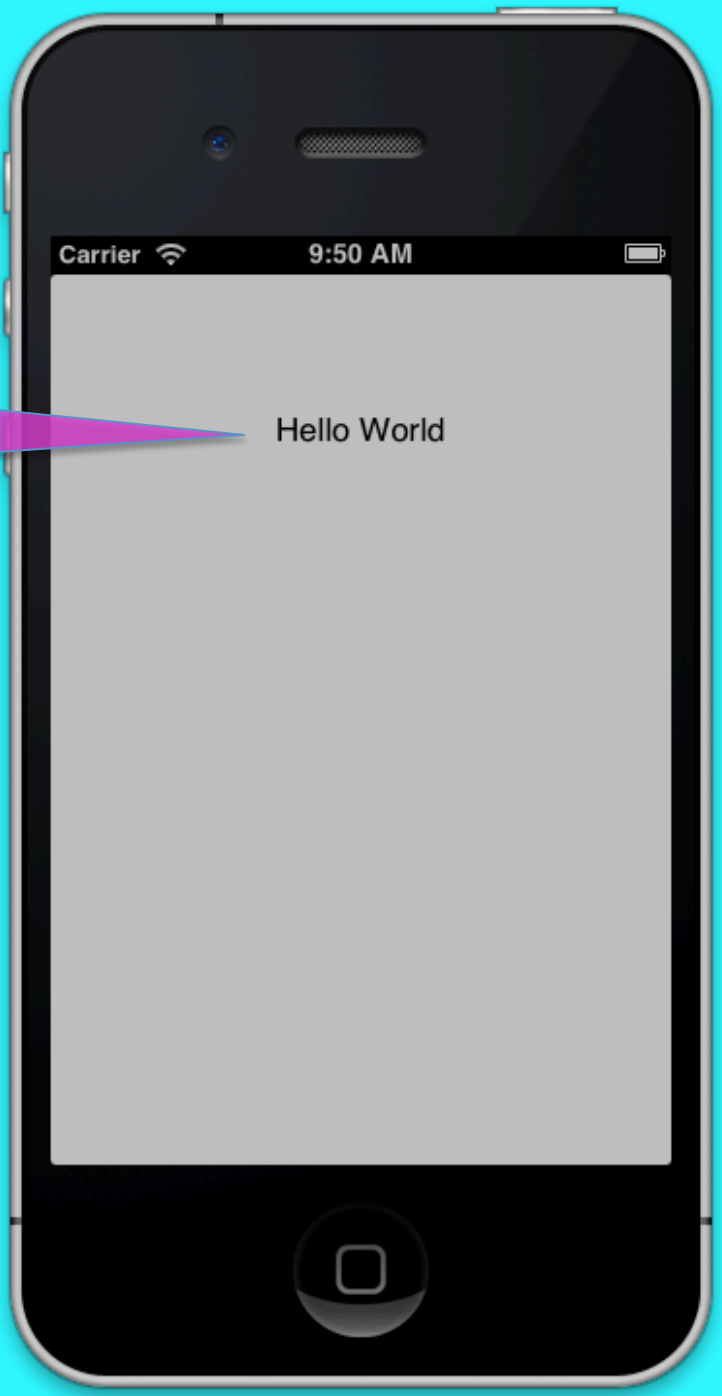

Click here for the attribute editor.
We can edit various attributes of our label in the attribute editor.
ViewController.xib
Xcode
Run
Stop
Editor
View
Organizer
By File
By Type
No Issues
Placeholders
File's Owner
First Responder
Objects
View
Label – Label
Constraints
Label
Text
Plain
Color
Default
Font
System 17.0
Alignment
Lines
Behavior
Enabled
Highlighted
Baseline
Align Baselines
Line Breaks
Truncate Tail
Autoshrink
Fixed Font Size
Tighten Letter Spacing
Highlighted
Shadow
Shadow Offset
Horizontal
Vertical
Mode
Left
Tag
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.

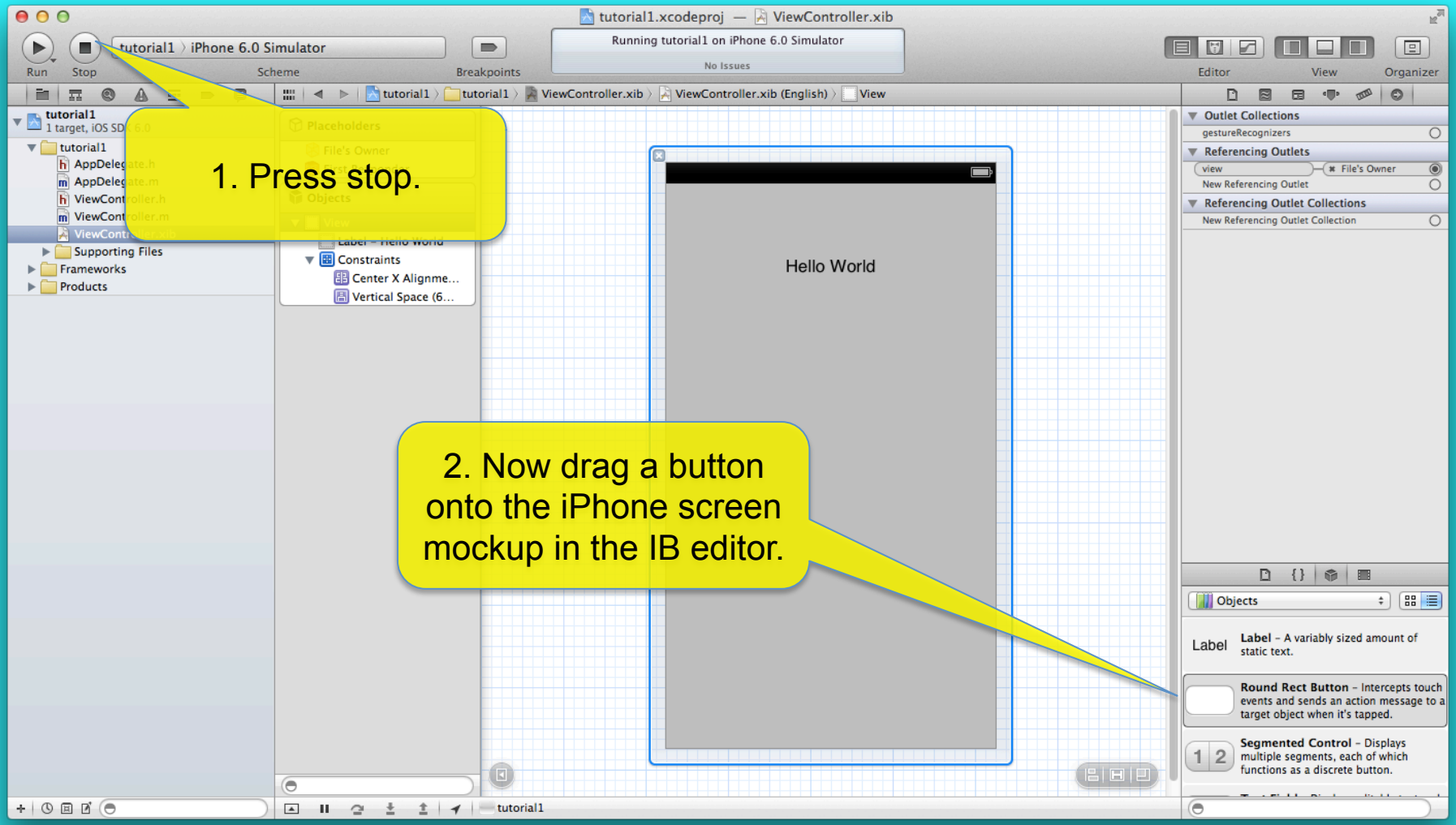

Click here for the size editor.
Other properties can be edited in the size editor.
Run
Stop
Editor
View
Organizer
ViewController.xib
Xcode
By File
By Type
No Issues
Placeholders
File's Owner
First Responder
Objects
View
Label – Label
Constraints
Show
Frame Rectangle
131
96
X
Y
42
21
Width
Height
Origin
Content Hugging Priority
Horizontal
250
Vertical
250
Content Compression Resistance Priority
Horizontal
750
Vertical
750
Constraints
Leading Space to: Superview
Equals: 131
Top Space to: Superview
Equals: 96
Label
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.

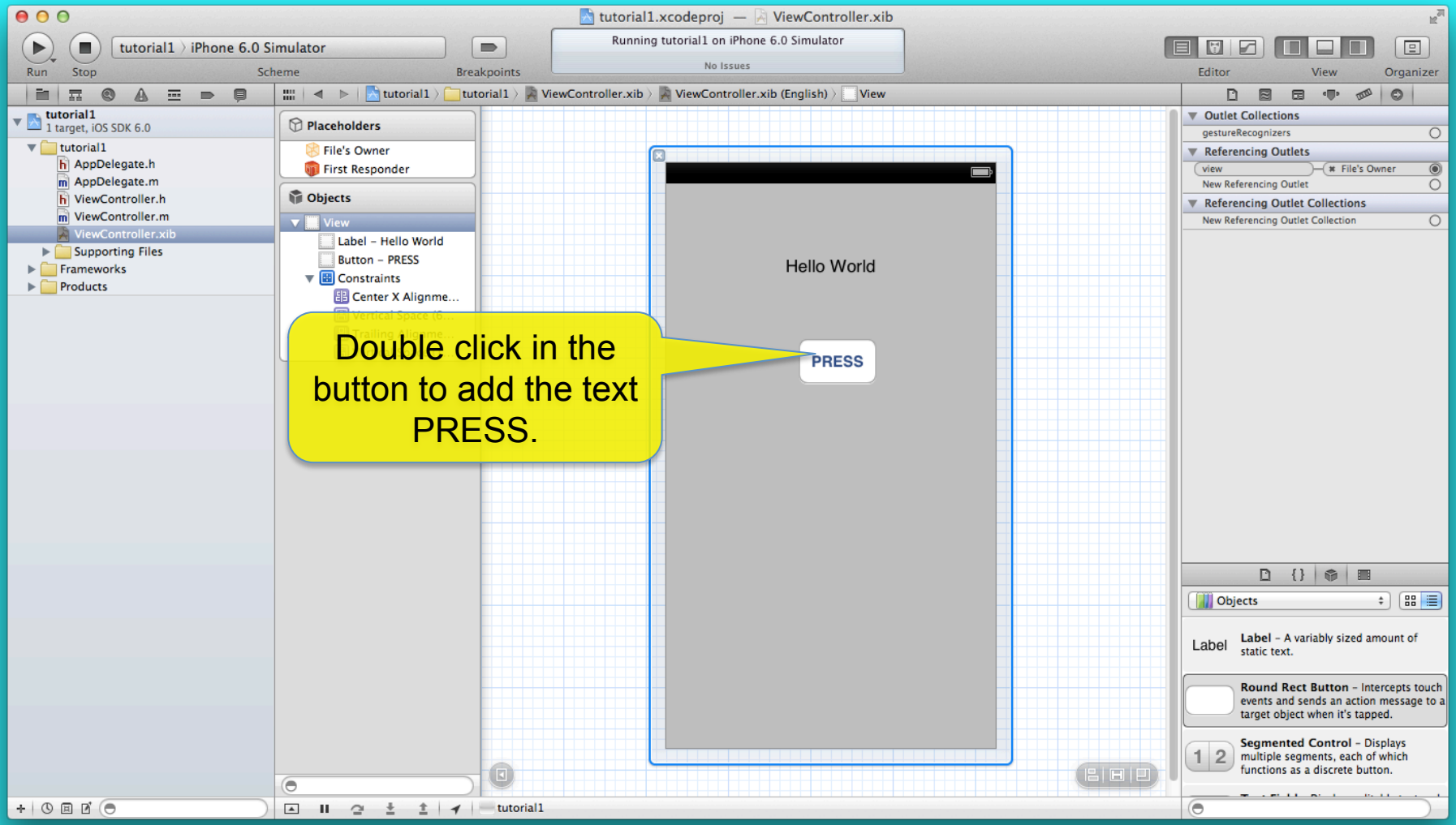

Double click to change the label text to Hello World.
We can also edit the label directly with the mouse and keyboard.
tutorial1.xcodeproj — ViewController.xib
Xcode
Run
Editor
View
Organizer
ViewController.xib
ViewController.xib (English)
View
Label – Hello World
By File
By Type
File's Owner
First Responder
Objects
View
Label – Hello World
Constraints
No Issues
Hello World
Show
Frame Rectangle
131
X
96
Y
88
Width
21
Height
Origin
Content Hugging Priority
Horizontal
254
Vertical
250
Content Compression Resistance Priority
Horizontal
750
Vertical
750
Constraints
Leading Space to: Superview
Equals: 131
Top Space to: Superview
Equals: 96
Objects
Label
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.

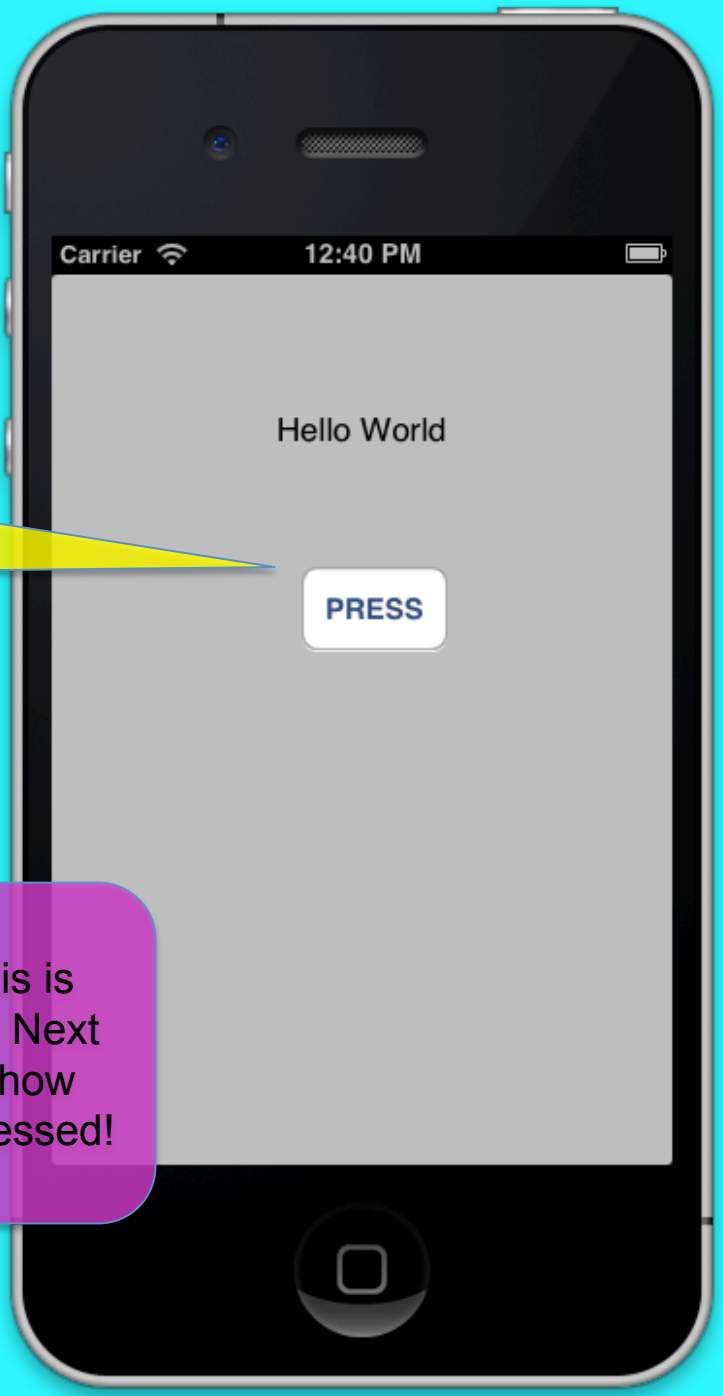

tutorial1.xcodeproj — ViewController.xib
Xcode
tutorial1 › iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Editor
View
Organizer
Placeholders
File's Owner
First Responder
Objects
View
Label – Hello World
Constraints
Hello World
Before we go any further, let's build our app! Click on the run button.
You'll see various build messages here. Eventually you should get a popup message saying the build succeeded.

Here is our label.



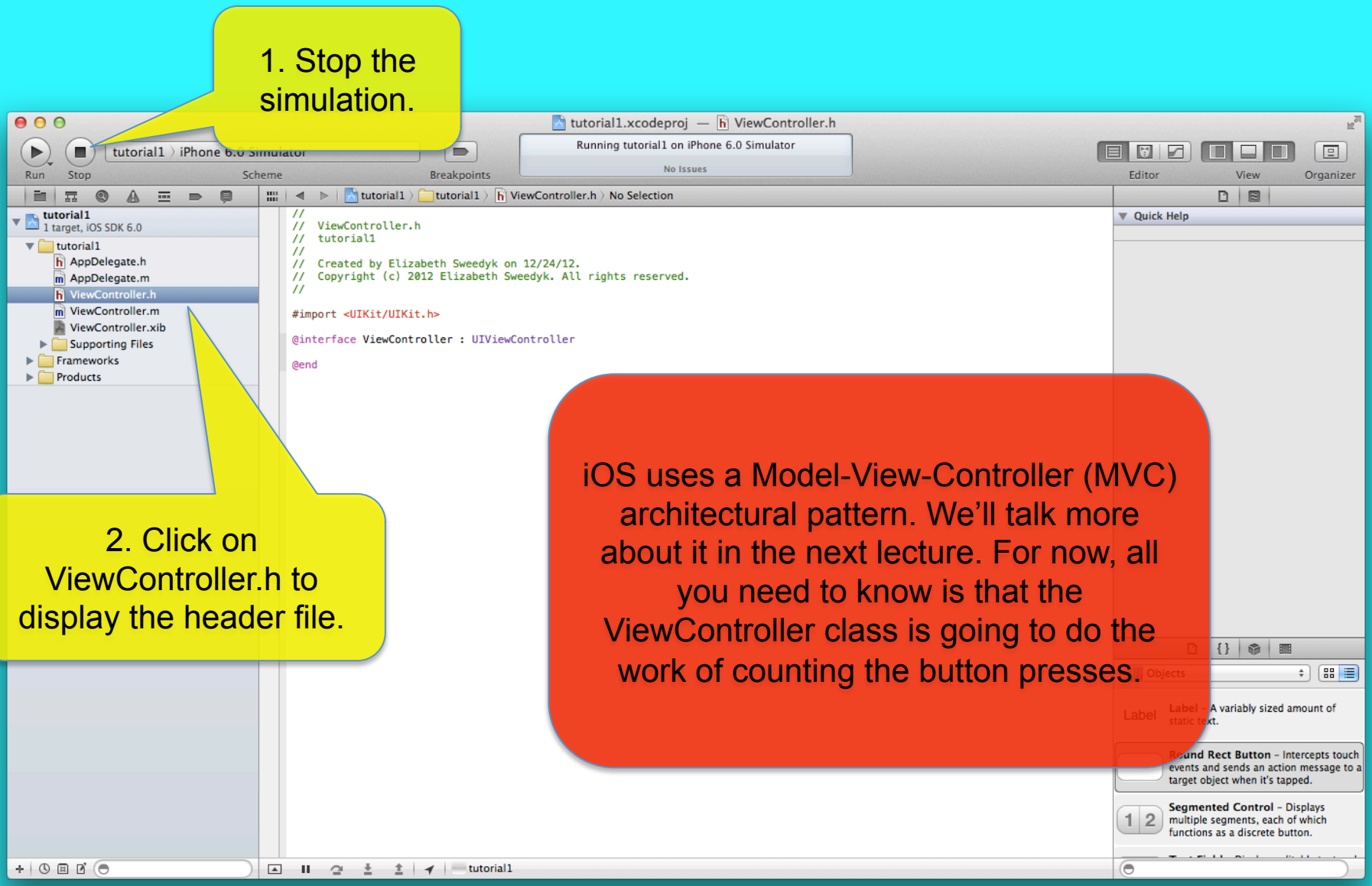

tutorial1.xcodeproj — ViewController.xib
Run
Stop
tutorial1 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running tutorial1 on iPhone 6.0 Simulator
No Issues
Editor
View
Organizer
1. Press stop.
Hello World
2. Now drag a button onto the iPhone screen mockup in the IB editor.
Outlet Collections
gestureRecognizers
Referencing Outlets
view
File's Owner
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Objects
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.

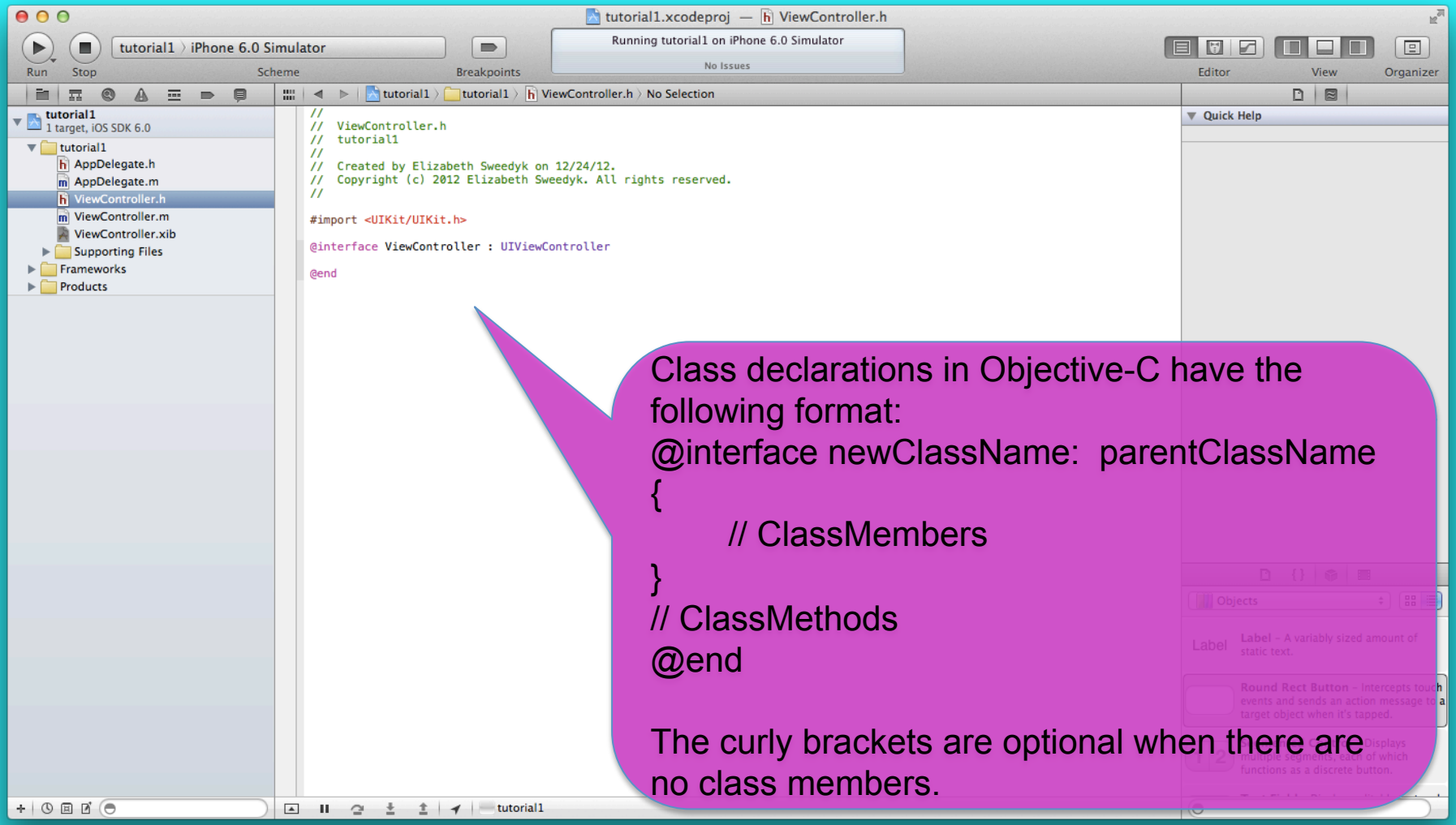

tutorial1.xcodeproj — ViewController.xib
Run
Stop
tutorial1 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running tutorial1 on iPhone 6.0 Simulator
No Issues
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
tutorial1
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
Placeholders
File's Owner
First Responder
Objects
View
Label – Hello World
Button – PRESS
Constraints
Center X Alignme...
Double click in the button to add the text PRESS.
Hello World
PRESS
Outlet Collections
gestureRecognizers
Referencing Outlets
view
File's Owner
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.

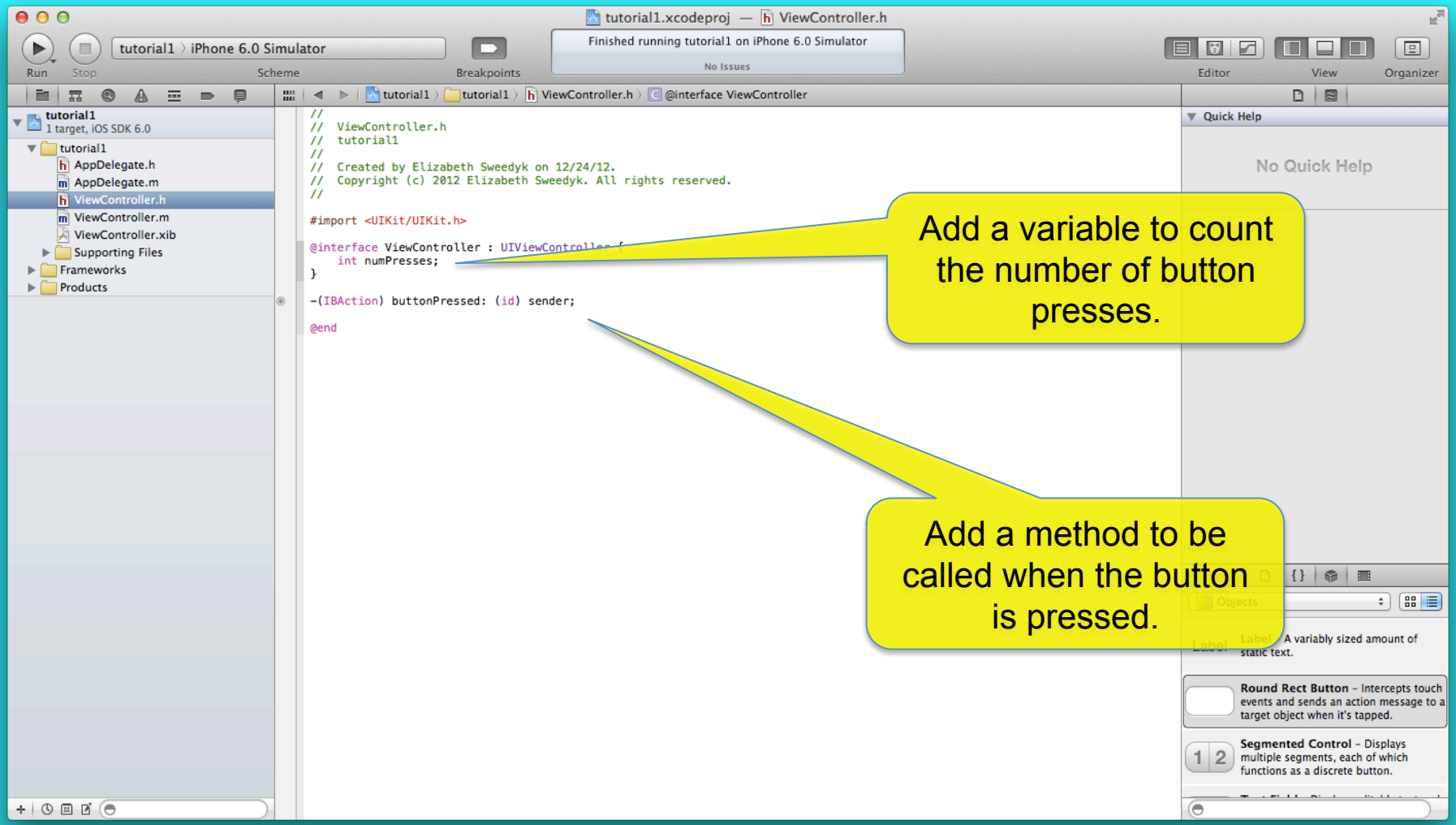


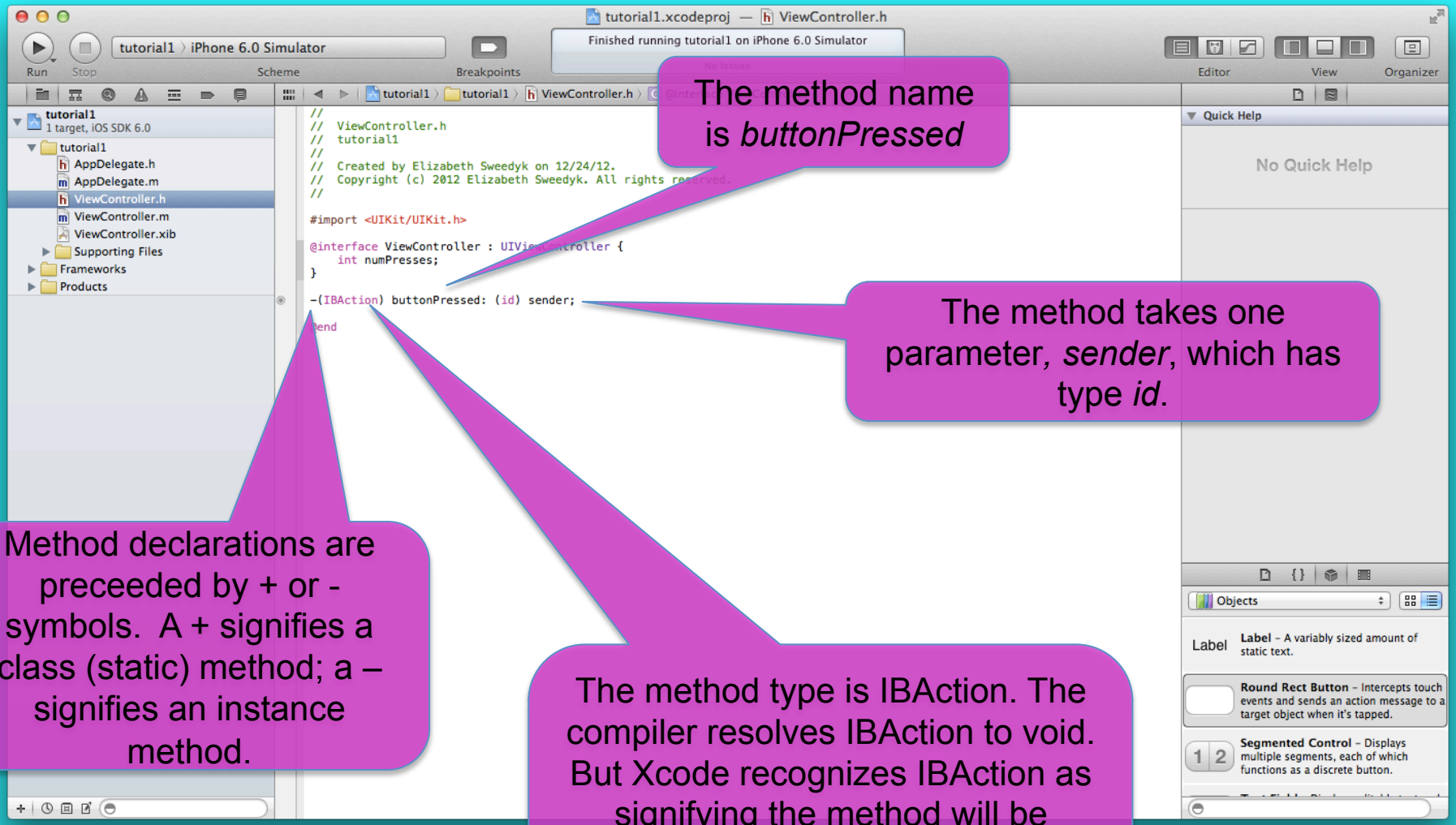
Run the app. Press the button!

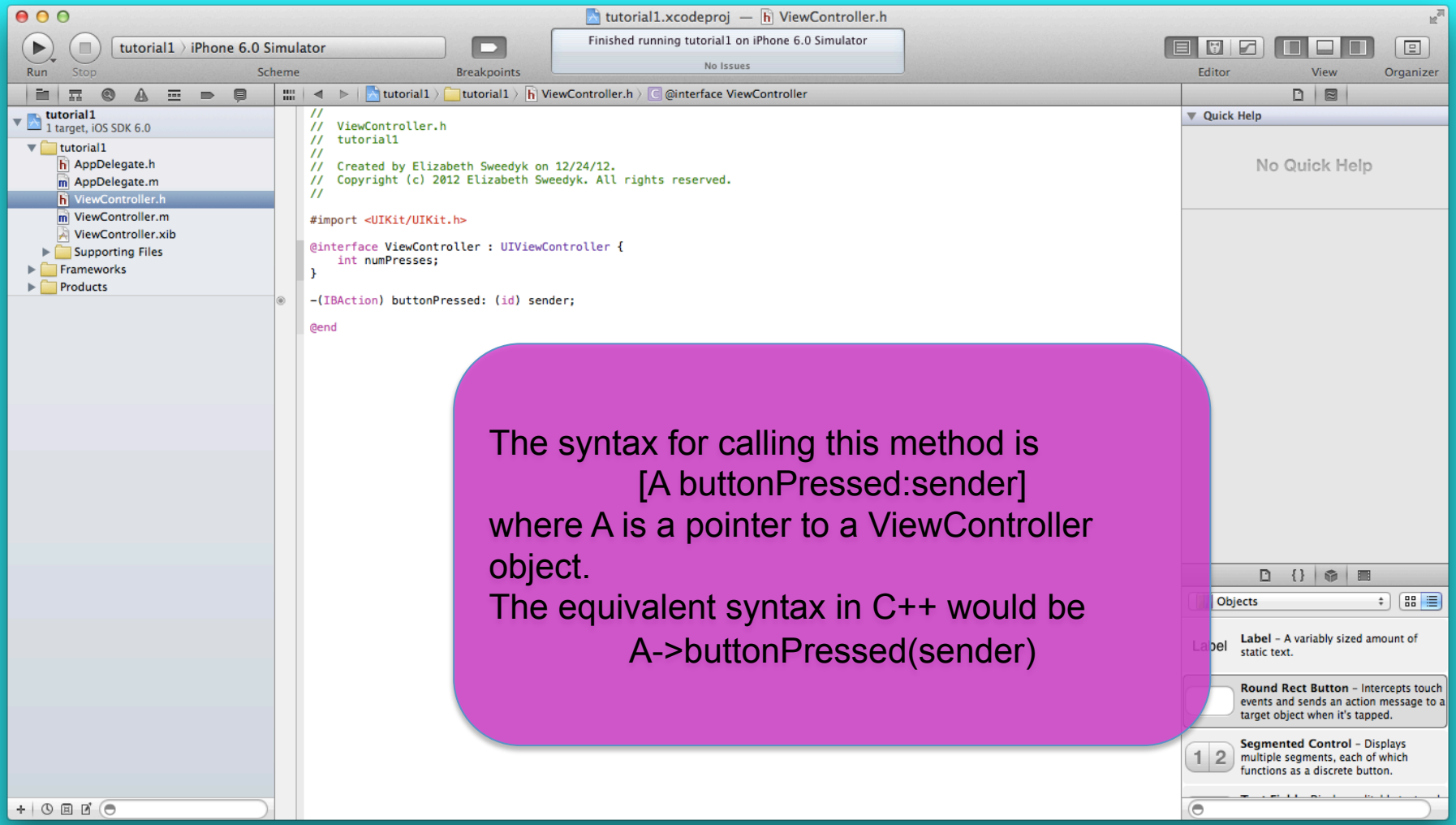
I know you are thinking this is awesome. ☺ But just wait. Next we'll make the app count how many times the button is pressed!

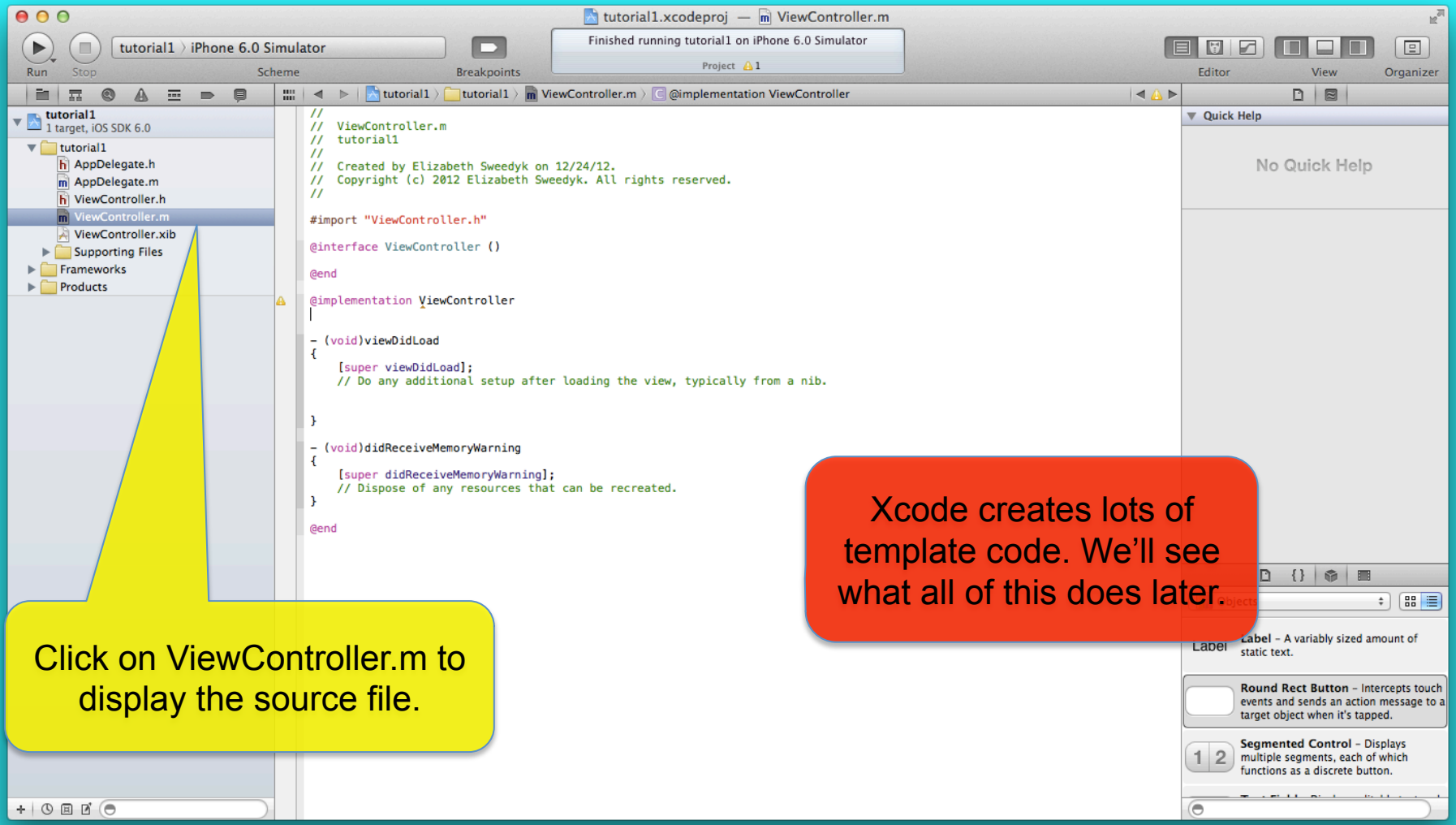

1. Stop the simulation.
2. Click on ViewController.h to display the header file.
iOS uses a Model-View-Controller (MVC) architectural pattern. We'll talk more about it in the next lecture. For now, all you need to know is that the ViewController class is going to do the work of counting the button presses.
Run
Stop
Scheme
Breakpoints
Editor
View
Organizer
tutorial1.xcodeproj — ViewController.h
Running tutorial1 on iPhone 6.0 Simulator
No Issues
tutorial1
1 target, iOS SDK 6.0
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
// ViewController.h // tutorial1 // // Created by Elizabeth Sweedyk on 12/24/12. // Copyright (c) 2012 Elizabeth Sweedyk. All rights reserved. //
#import <UIKit/UIKit.h>
@interface ViewController : UIViewController
@end
Quick Help
Segmented Control – Displays multiple segments, each of which functions as a discrete button.

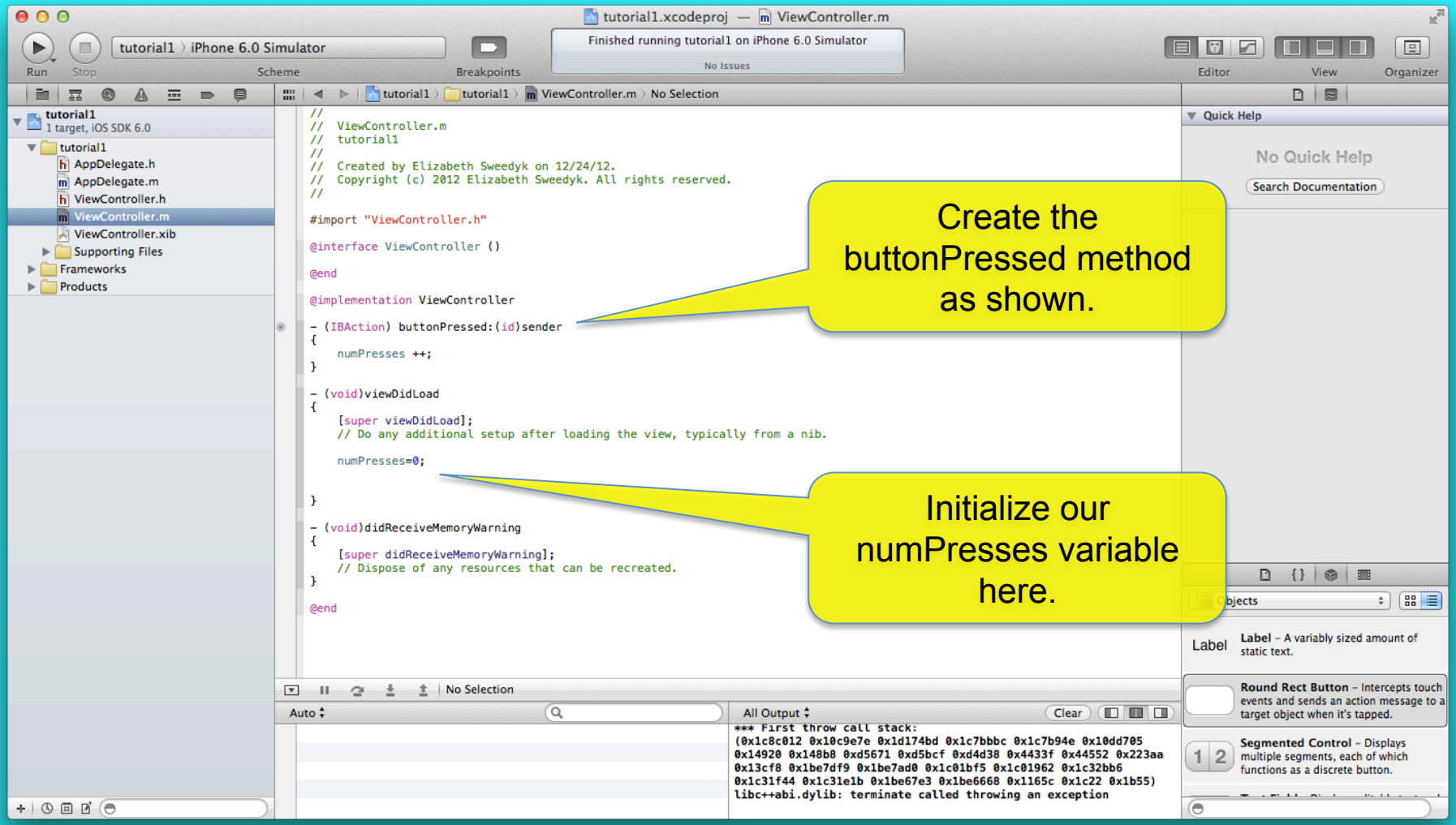

tutorial1.xcodeproj — ViewController.h
Run
Stop
tutorial1 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running tutorial1 on iPhone 6.0 Simulator
No Issues
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
tutorial1
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
tutorial1 〉 tutorial1 〉 ViewController.h 〉 No Selection
//
// ViewController.h
// tutorial1
//
// Created by Elizabeth Sweedyk on 12/24/12.
// Copyright (c) 2012 Elizabeth Sweedyk. All rights reserved.
//
#import <UIKit/UIKit.h>
@interface ViewController : UIViewController
@end
Quick Help
Class declarations in Objective-C have the following format:
@interface newClassName: parentClassName
{
// ClassMembers
}
// ClassMethods
@end
The curly brackets are optional when there are no class members.
tutorial1

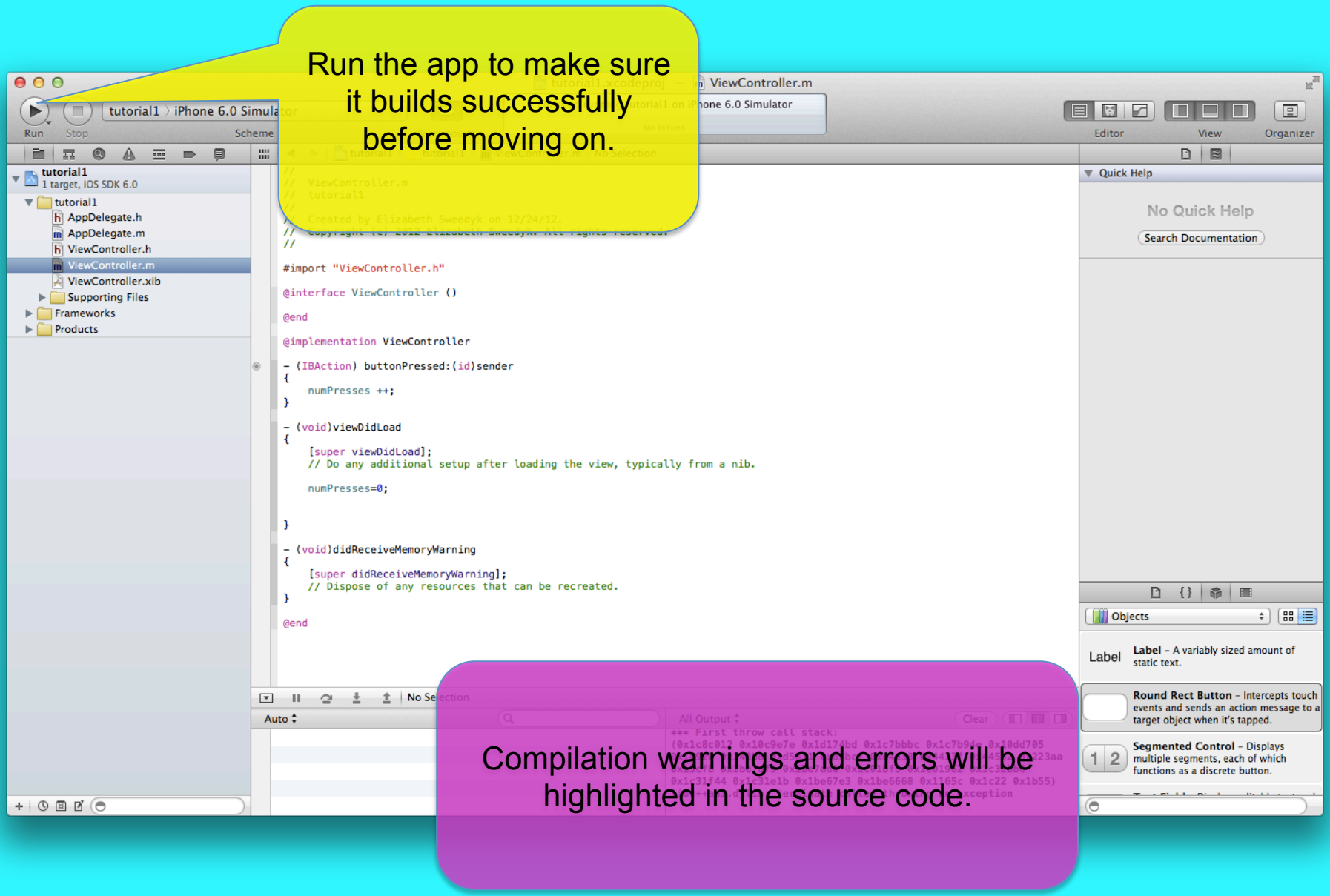

tutorial1.xcodeproj — ViewController.h
tutorial1 › iPhone 6.0 Simulator
Finished running tutorial1 on iPhone 6.0 Simulator
No Issues
Run
Stop
Scheme
Breakpoints
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
tutorial1
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
tutorial1 › tutorial1 › ViewController.h › @interface ViewController
//
// ViewController.h
// tutorial1
//
// Created by Elizabeth Sweedyk on 12/24/12.
// Copyright (c) 2012 Elizabeth Sweedyk. All rights reserved.
//
#import <UIKit/UIKit.h>
@interface ViewController : UIViewController {
int numPresses;
}
-(IBAction) buttonPressed: (id) sender;
@end
Quick Help
No Quick Help
Add a variable to count the number of button presses.
Add a method to be called when the button is pressed.
A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.


tutorial1.xcodeproj — ViewController.h
Finished running tutorial1 on iPhone 6.0 Simulator
tutorial1 › iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
tutorial1
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
//
// ViewController.h
// tutorial1
//
// Created by Elizabeth Sweedyk on 12/24/12.
// Copyright (c) 2012 Elizabeth Sweedyk. All rights reserved.
//
#import <UIKit/UIKit.h>
@interface ViewController : UIViewController {
int numPresses;
}
-(IBAction) buttonPressed: (id) sender;
@end
Quick Help
No Quick Help
Objects
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.
The method name is buttonPressed
The method takes one parameter, sender, which has type id.
Method declarations are preceeded by + or - symbols. A + signifies a class (static) method; a – signifies an instance method.
The method type is IBAction. The compiler resolves IBAction to void. But Xcode recognizes IBAction as signifying the method will be connected to a UI element created in the IB. (We'll make the connection in slide 37.)


tutorial1.xcodeproj — ViewController.h
Run
Stop
tutorial1 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running tutorial1 on iPhone 6.0 Simulator
No Issues
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
tutorial1
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
tutorial1 〉 tutorial1 〉 ViewController.h 〉 @interface ViewController
//
// ViewController.h
// tutorial1
//
// Created by Elizabeth Sweedyk on 12/24/12.
// Copyright (c) 2012 Elizabeth Sweedyk. All rights reserved.
//
#import <UIKit/UIKit.h>
@interface ViewController : UIViewController {
int numPresses;
}
-(IBAction) buttonPressed: (id) sender;
@end
Quick Help
No Quick Help
Objects
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.
The syntax for calling this method is
[A buttonPressed:sender]
where A is a pointer to a ViewController object.
The equivalent syntax in C++ would be
A->buttonPressed(sender)

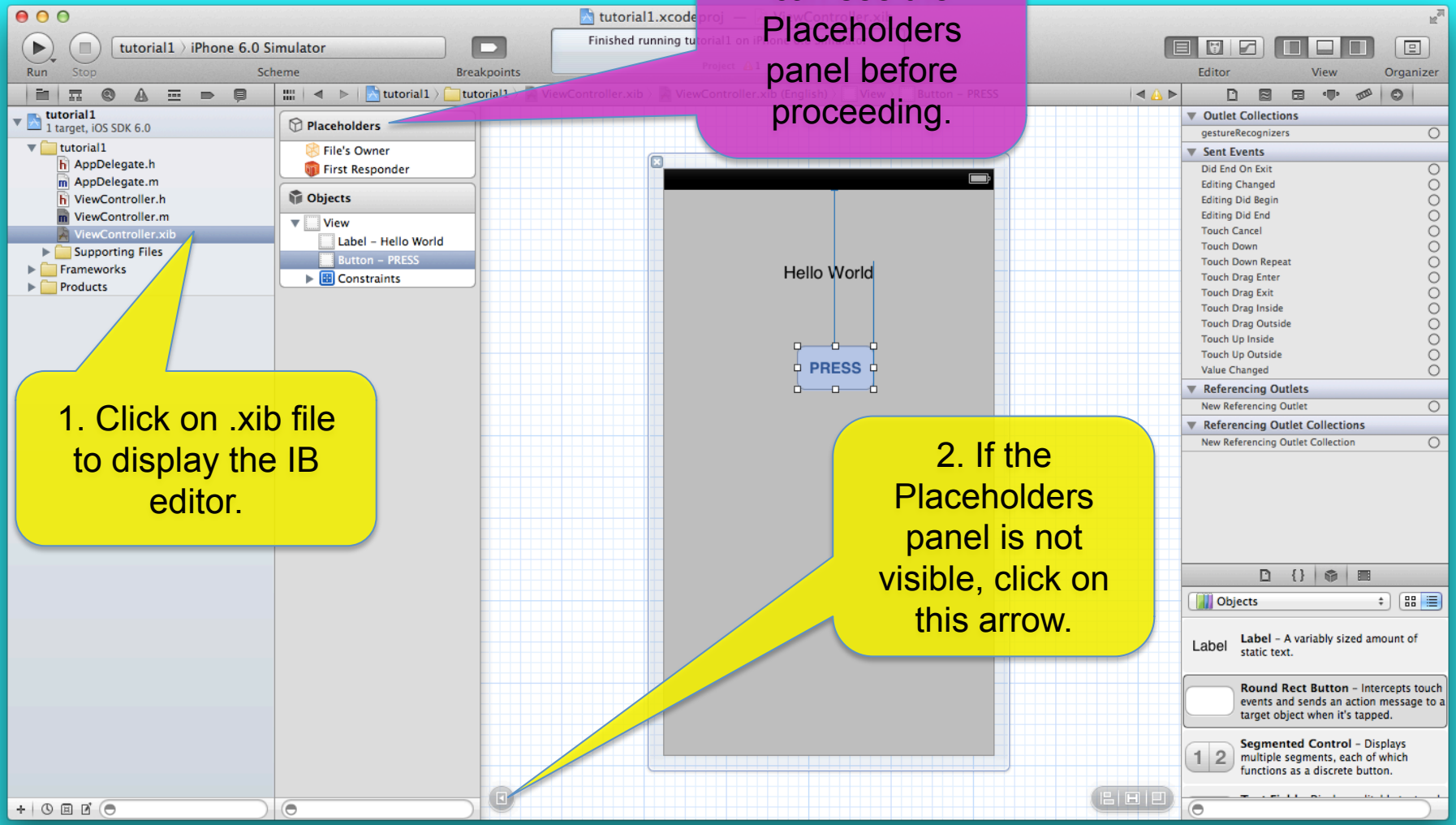

tutorial1.xcodeproj — ViewController.m
tutorial1 › iPhone 6.0 Simulator
Finished running tutorial1 on iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
tutorial1
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
tutorial1 › tutorial1 › ViewController.m › @implementation ViewController
//
// ViewController.m
// tutorial1
//
// Created by Elizabeth Sweedyk on 12/24/12.
// Copyright (c) 2012 Elizabeth Sweedyk. All rights reserved.
//
#import "ViewController.h"
@interface ViewController ()
@end
@implementation ViewController
- (void)viewDidLoad
{
[super viewDidLoad];
// Do any additional setup after loading the view, typically from a nib.
}
- (void)didReceiveMemoryWarning
{
[super didReceiveMemoryWarning];
// Dispose of any resources that can be recreated.
}
@end
Quick Help
No Quick Help
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.
Click on ViewController.m to display the source file.
Xcode creates lots of template code. We'll see what all of this does later.

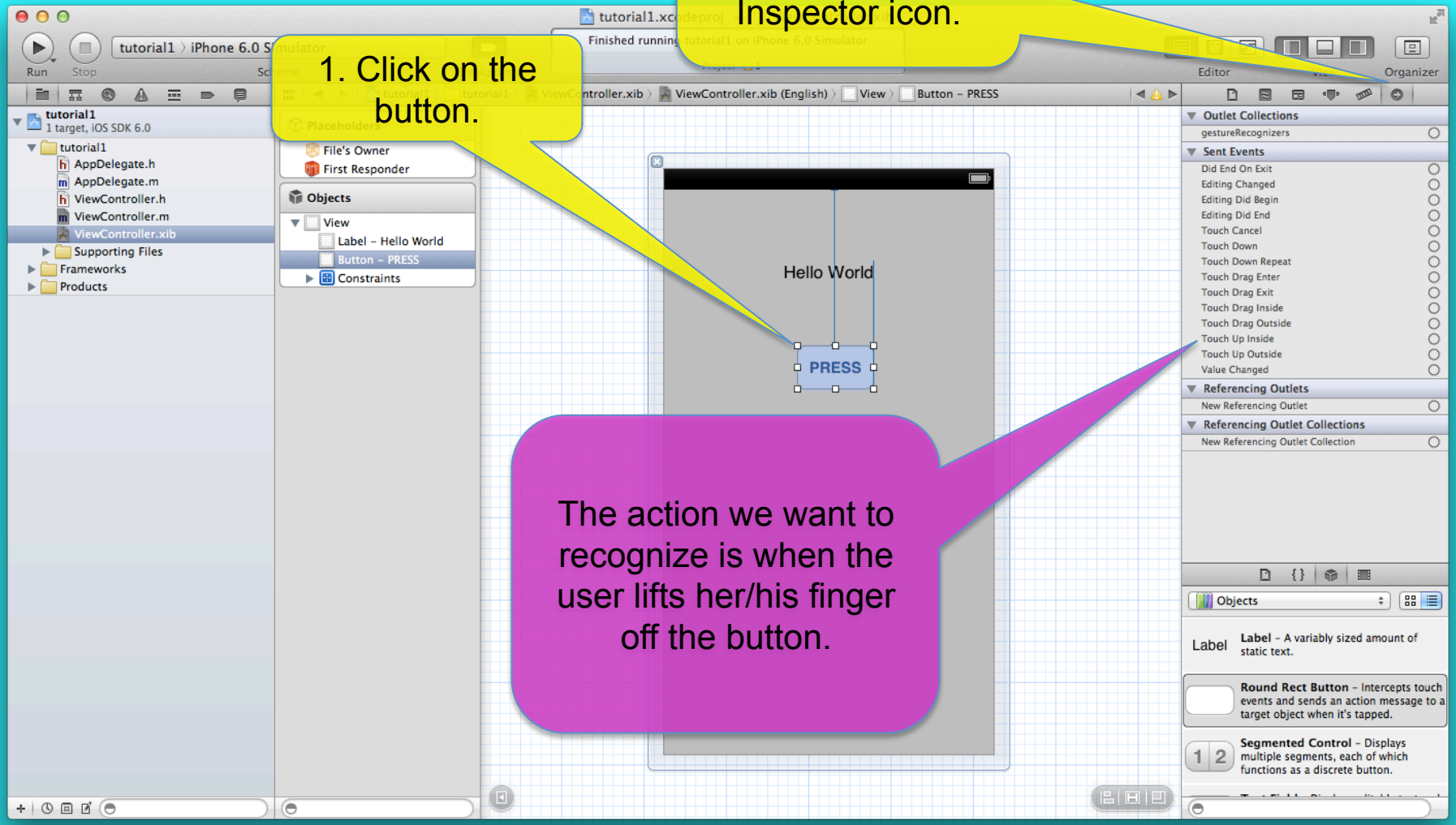

tutorial1.xcodeproj — ViewController.m
tutorial1 › iPhone 6.0 Simulator
Finished running tutorial1 on iPhone 6.0 Simulator
No Issues
Run
Stop
Scheme
Breakpoints
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
tutorial1
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
tutorial1 › tutorial1 › ViewController.m › No Selection
//
// ViewController.m
// tutorial1
//
// Created by Elizabeth Sweedyk on 12/24/12.
// Copyright (c) 2012 Elizabeth Sweedyk. All rights reserved.
//
#import "ViewController.h"
@interface ViewController ()
@end
@implementation ViewController
- (IBAction) buttonPressed:(id)sender
{
numPresses ++;
}
- (void)viewDidLoad
{
[super viewDidLoad];
// Do any additional setup after loading the view, typically from a nib.
numPresses=0;
}
- (void)didReceiveMemoryWarning
{
[super didReceiveMemoryWarning];
// Dispose of any resources that can be recreated.
}
@end
Create the buttonPressed method as shown.
Initialize our numPresses variable here.
Quick Help
No Quick Help
Search Documentation
No Selection
Auto
All Output
Clear
*** First throw call stack:
(0x1c8c012 0x10c9e7e 0x1d174bd 0x1c7bbbc 0x1c7b94e 0x10dd705
0x14920 0x148b8 0xd5671 0xd5bcf 0xd4d38 0x4433f 0x44552 0x223aa
0x13cf8 0x1be7df9 0x1be7ad0 0x1c01bf5 0x1c01962 0x1c32bb6
0x1c31f44 0x1c31e1b 0x1be67e3 0x1be6668 0x1165c 0x1c22 0x1b55)
libc++abi.dylib: terminate called throwing an exception
Label
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.

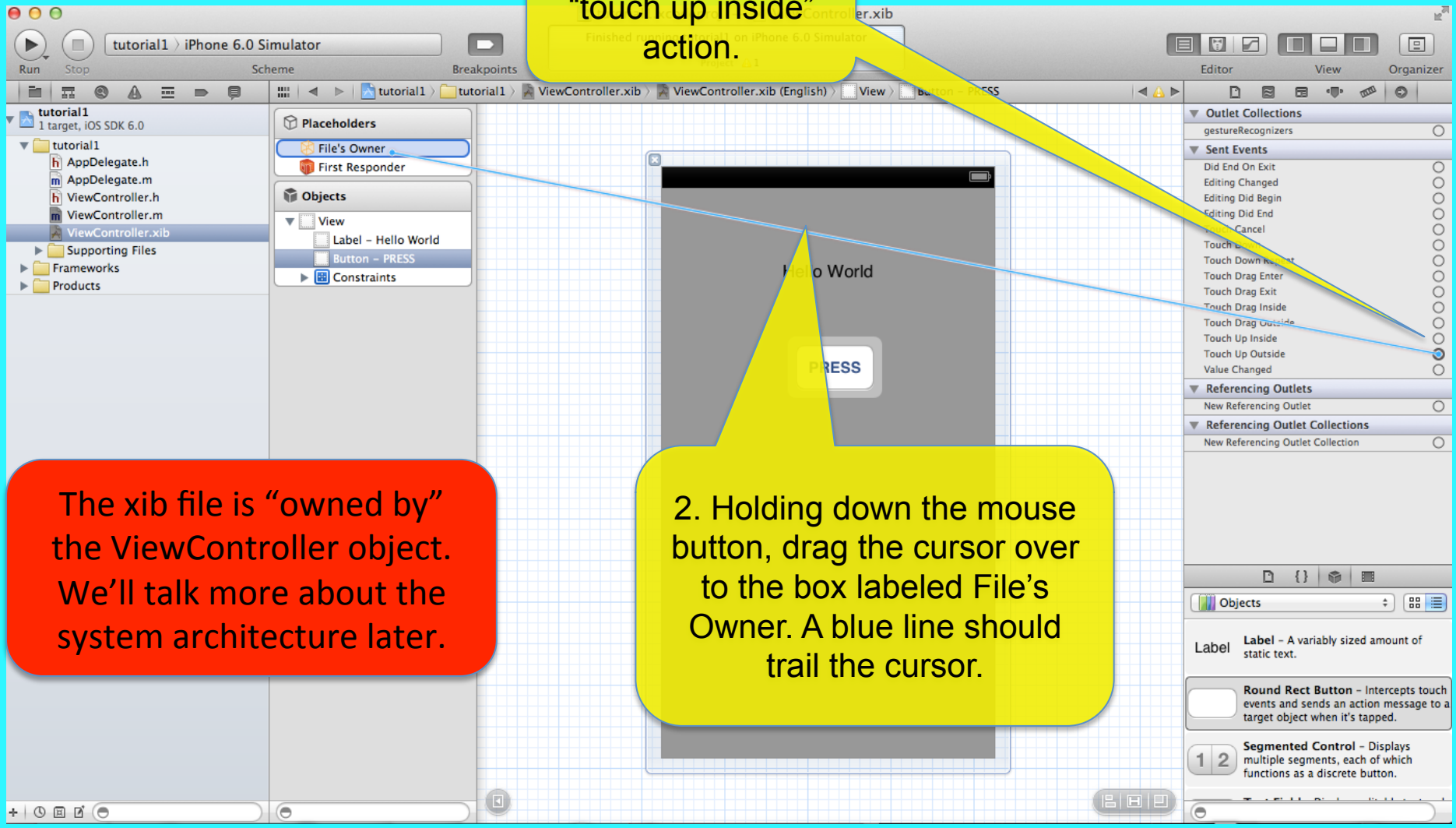


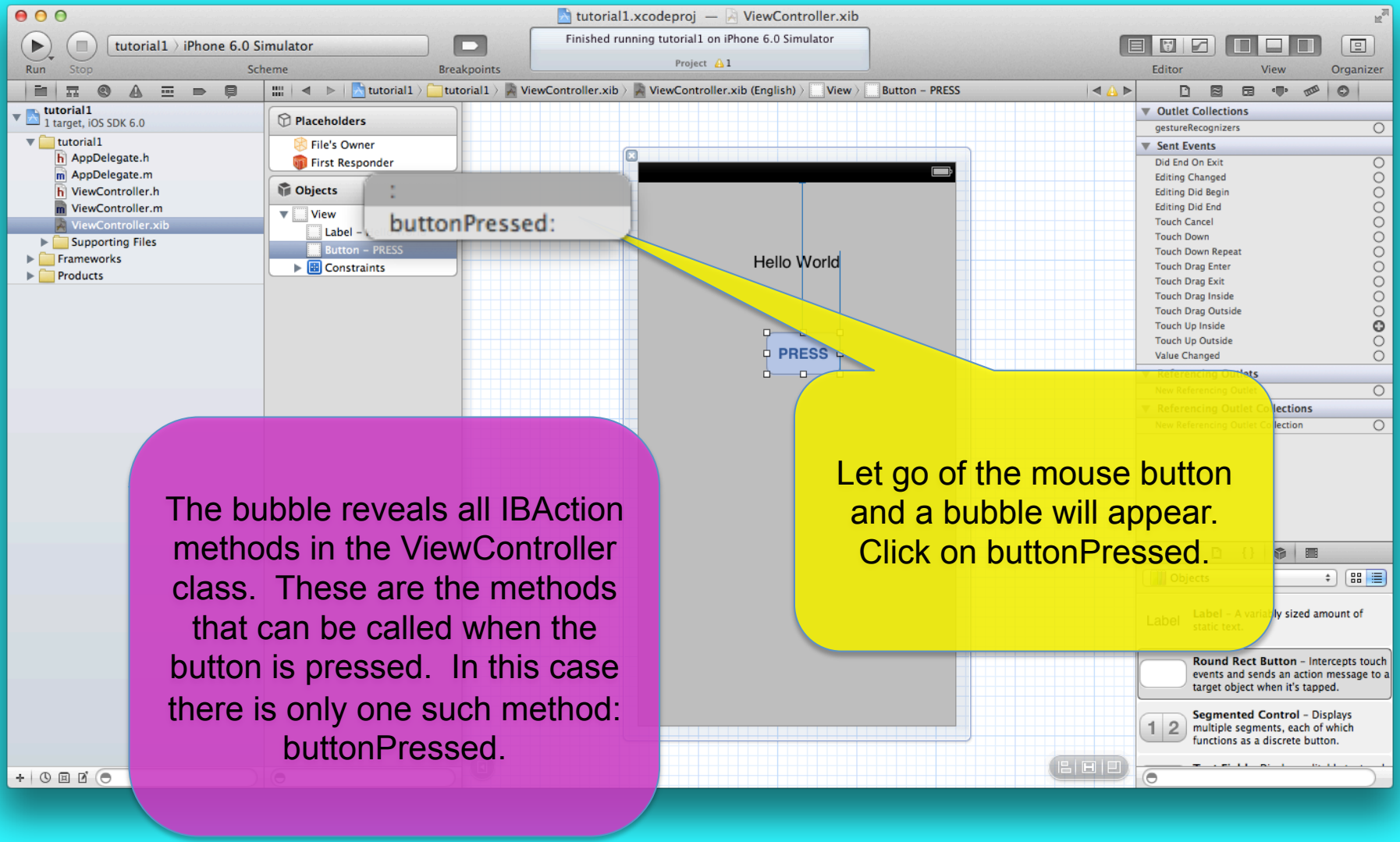

```
//
//  ViewController.m
//  tutorial1
//
//  Created by Elizabeth Sweedyk on 12/24/12.
//  Copyright (c) 2012 Elizabeth Sweedyk. All rights reserved.
//

#import "ViewController.h"

@interface ViewController ()

@end

@implementation ViewController

- (IBAction) buttonPressed:(id)sender
{
    numPresses ++;
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.

    numPresses=0;


}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

@end
```

Next we need to “connect” our PRESS button in the UI to our buttonPressed method.


Make sure you can see the Placeholders panel before proceeding.
1. Click on .xib file to display the IB editor.
2. If the Placeholders panel is not visible, click on this arrow.
tutorial1
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
Placeholders
File's Owner
First Responder
Objects
View
Label – Hello World
Button – PRESS
Constraints
Hello World
PRESS
Outlet Collections
gestureRecognizers
Sent Events
Did End On Exit
Editing Changed
Editing Did Begin
Editing Did End
Touch Cancel
Touch Down
Touch Down Repeat
Touch Drag Enter
Touch Drag Exit
Touch Drag Inside
Touch Drag Outside
Touch Up Inside
Touch Up Outside
Value Changed
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Objects
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.

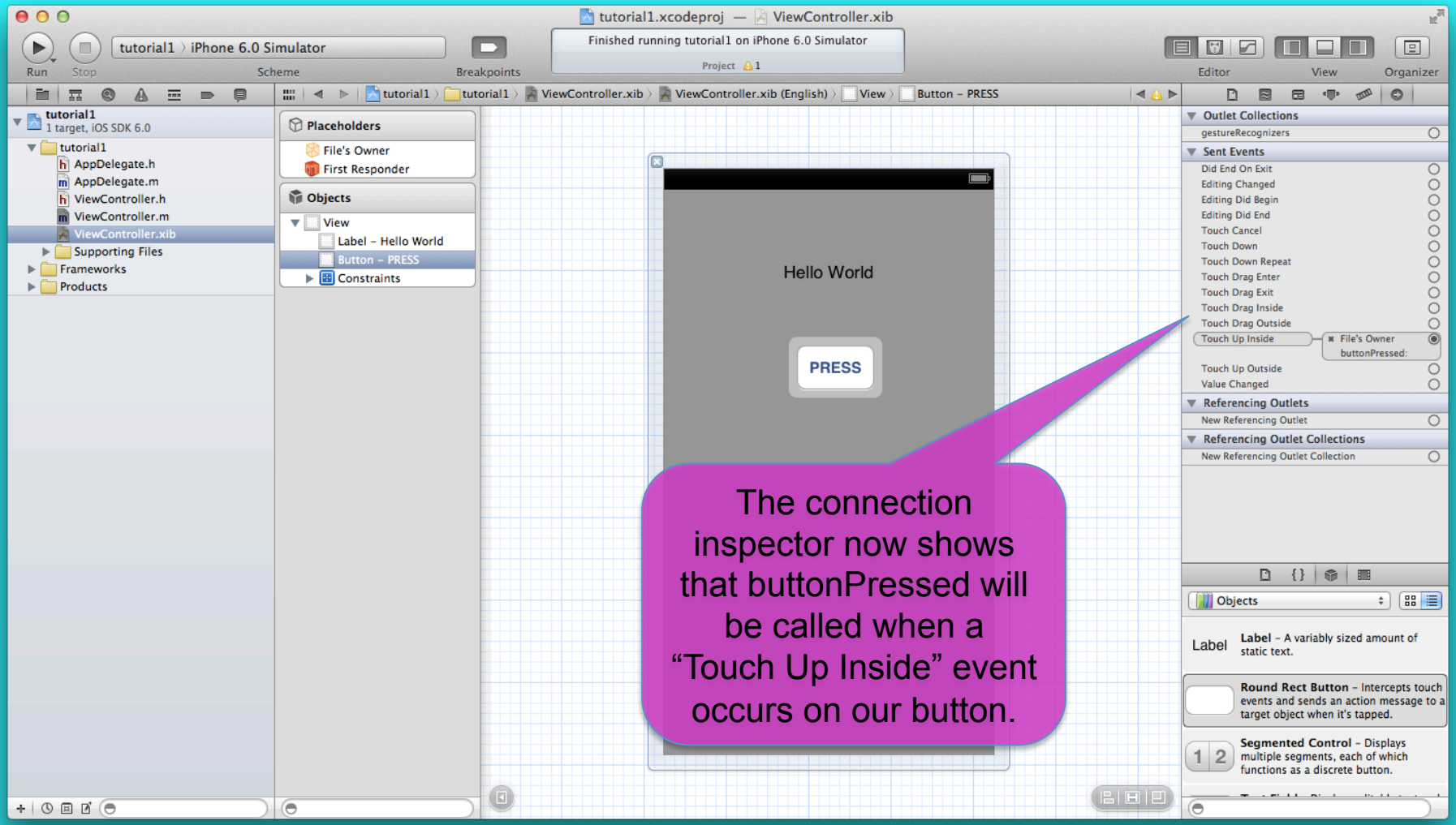

2. Click the Connections Inspector icon.
1. Click on the button.
Hello World
PRESS
The action we want to recognize is when the user lifts her/his finger off the button.


1. Click on the "touch up inside" action.
The xib file is "owned by" the ViewController object. We'll talk more about the system architecture later.
2. Holding down the mouse button, drag the cursor over to the box labeled File's Owner. A blue line should trail the cursor.
tutorial1 〉 iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
Placeholders
File's Owner
First Responder
Objects
View
Label – Hello World
Button – PRESS
Constraints
Outlet Collections
gestureRecognizers
Sent Events
Did End On Exit
Editing Changed
Editing Did Begin
Editing Did End
Touch Drag Enter
Touch Drag Exit
Touch Drag Inside
Touch Up Inside
Touch Up Outside
Value Changed
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.



The bubble reveals all IBAction methods in the ViewController class. These are the methods that can be called when the button is pressed. In this case there is only one such method: buttonPressed.

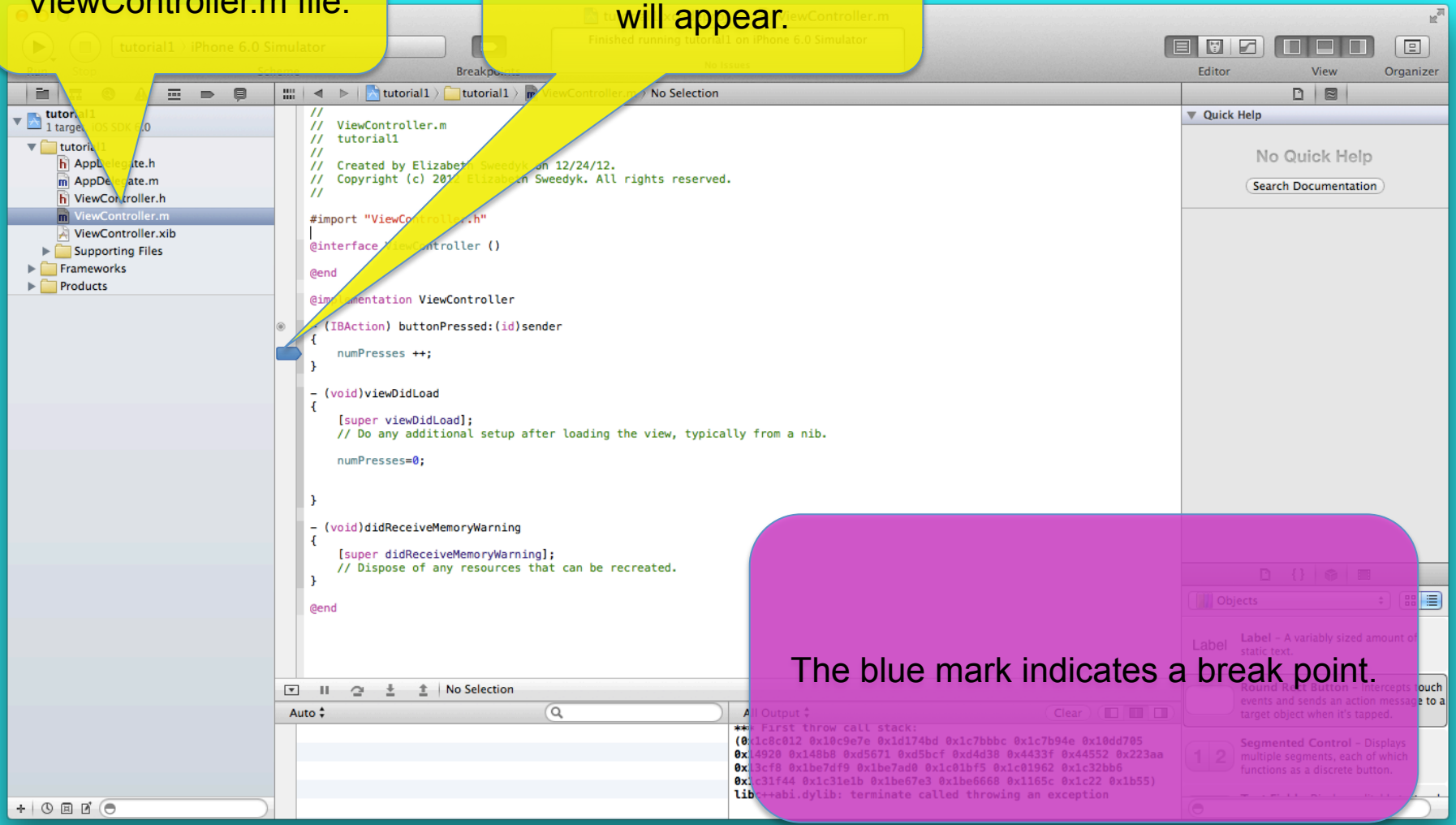
Let go of the mouse button and a bubble will appear. Click on buttonPressed.


tutorial1.xcodeproj — ViewController.xib
Run
Stop
tutorial1 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running tutorial1 on iPhone 6.0 Simulator
Project ⚠1
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
tutorial1
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
Placeholders
File's Owner
First Responder
Objects
View
Label – Hello World
Button – PRESS
Constraints
Hello World
PRESS
Outlet Collections
gestureRecognizers
Sent Events
Did End On Exit
Editing Changed
Editing Did Begin
Editing Did End
Touch Cancel
Touch Down
Touch Down Repeat
Touch Drag Enter
Touch Drag Exit
Touch Drag Inside
Touch Drag Outside
Touch Up Inside
File's Owner
buttonPressed:
Touch Up Outside
Value Changed
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Objects
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.
The connection inspector now shows that buttonPressed will be called when a "Touch Up Inside" event occurs on our button.

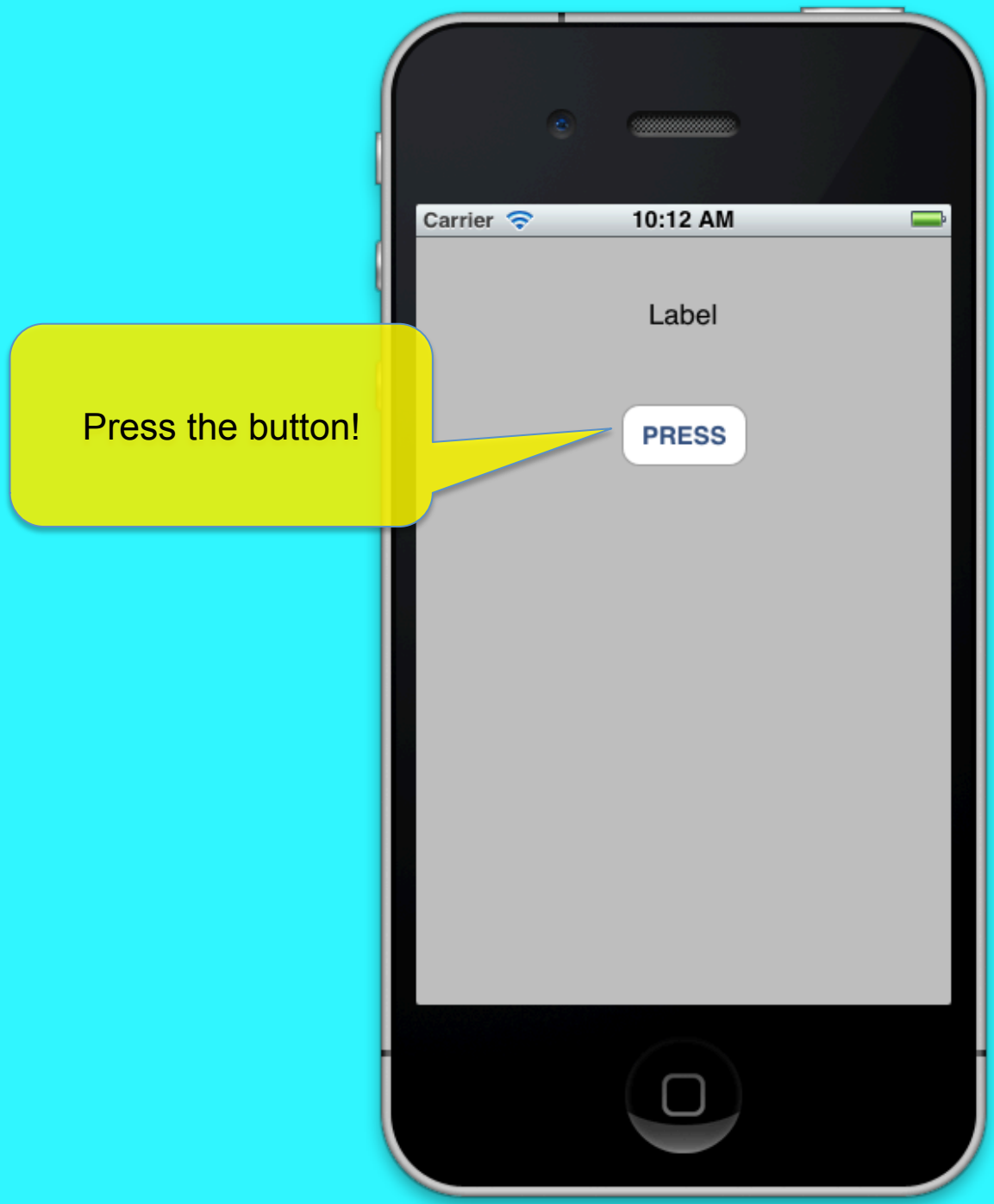
At this point the easiest way to check that we are counting button presses is in the debugger. (It is also a great time to introduce you to the debugger!)

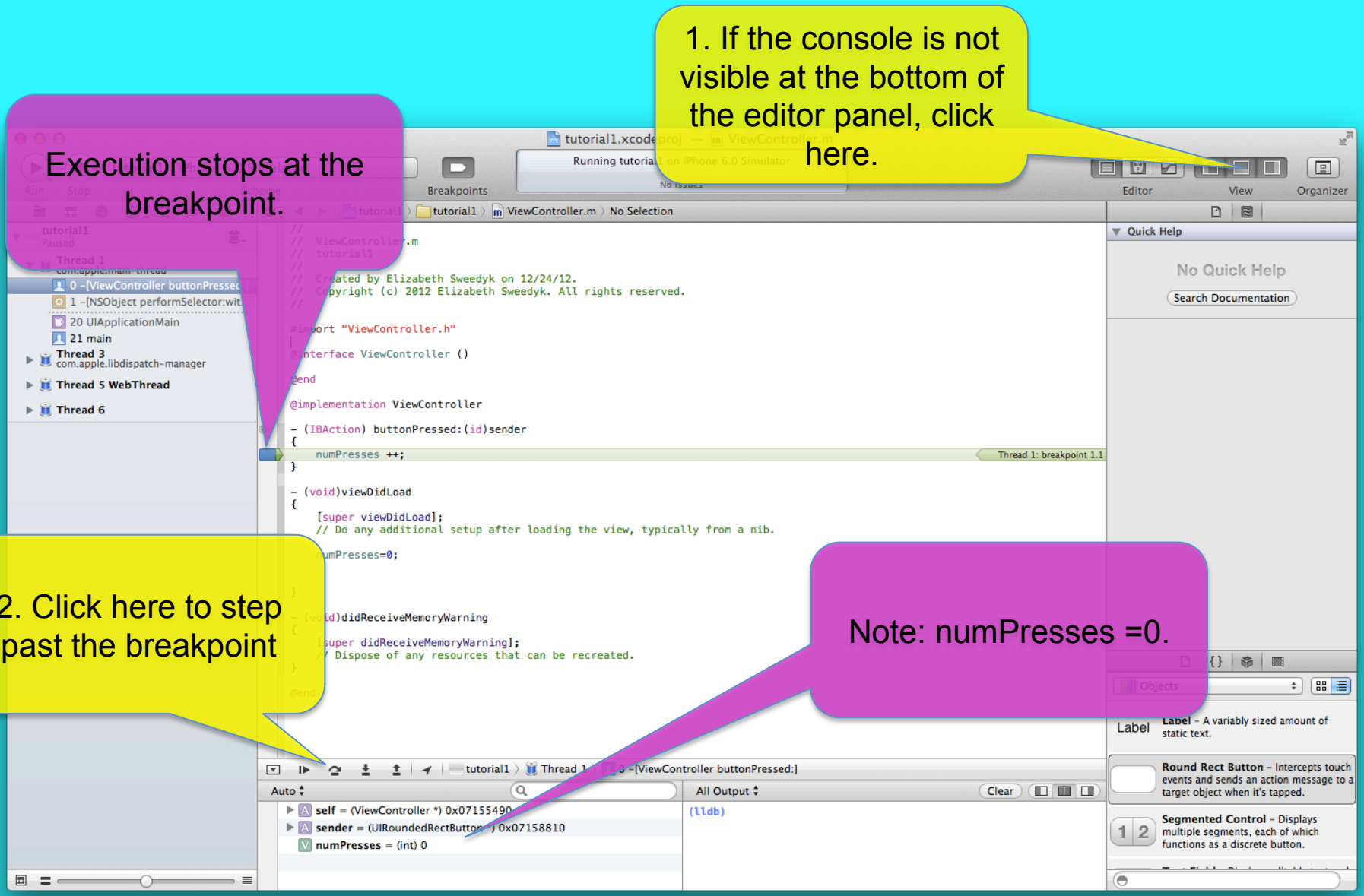
1. Open the ViewController.m file.

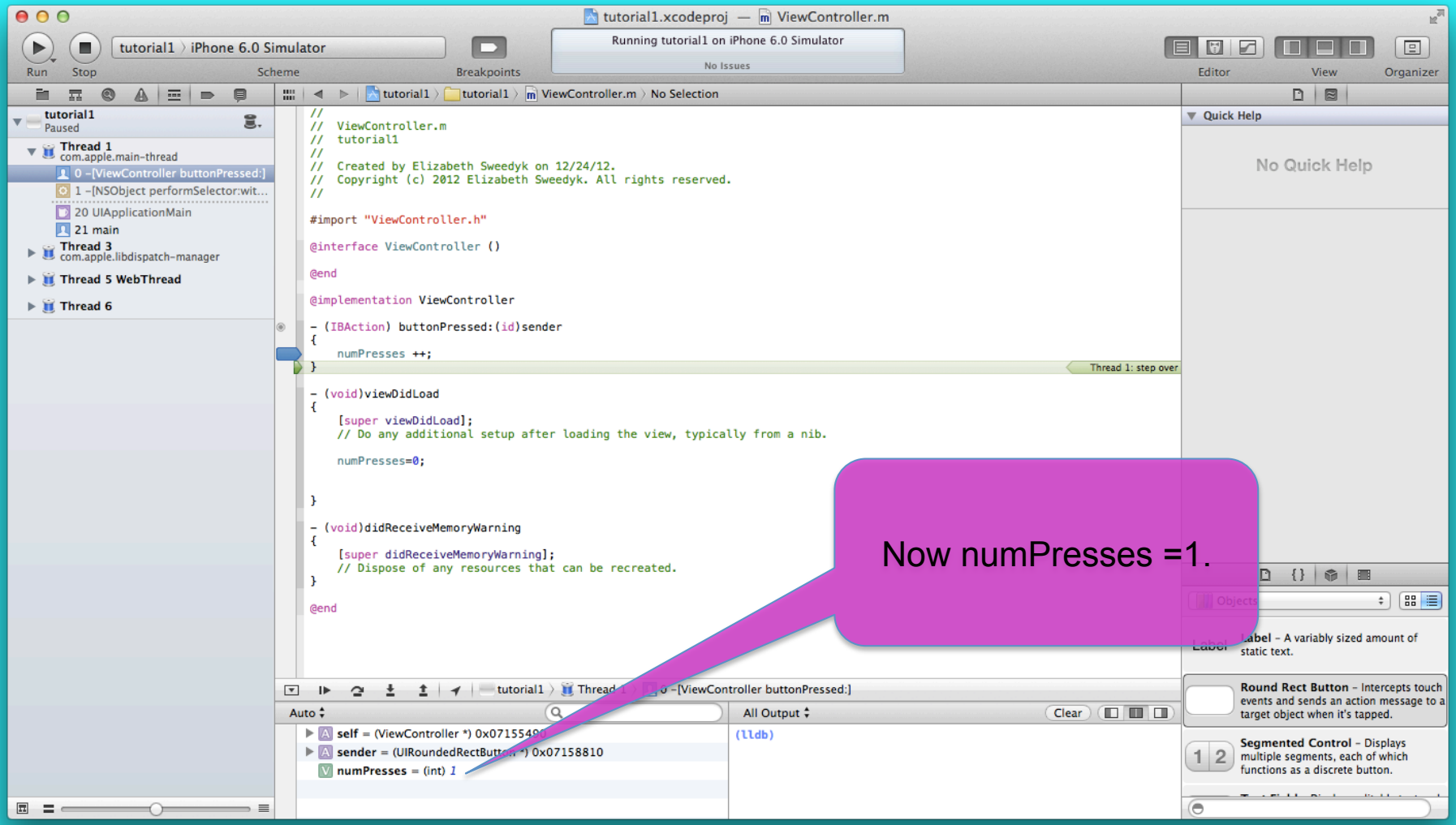
2. Click here. A blue mark will appear.



The blue mark indicates a break point.


Carrier
10:12 AM
Label
PRESS
Press the button!

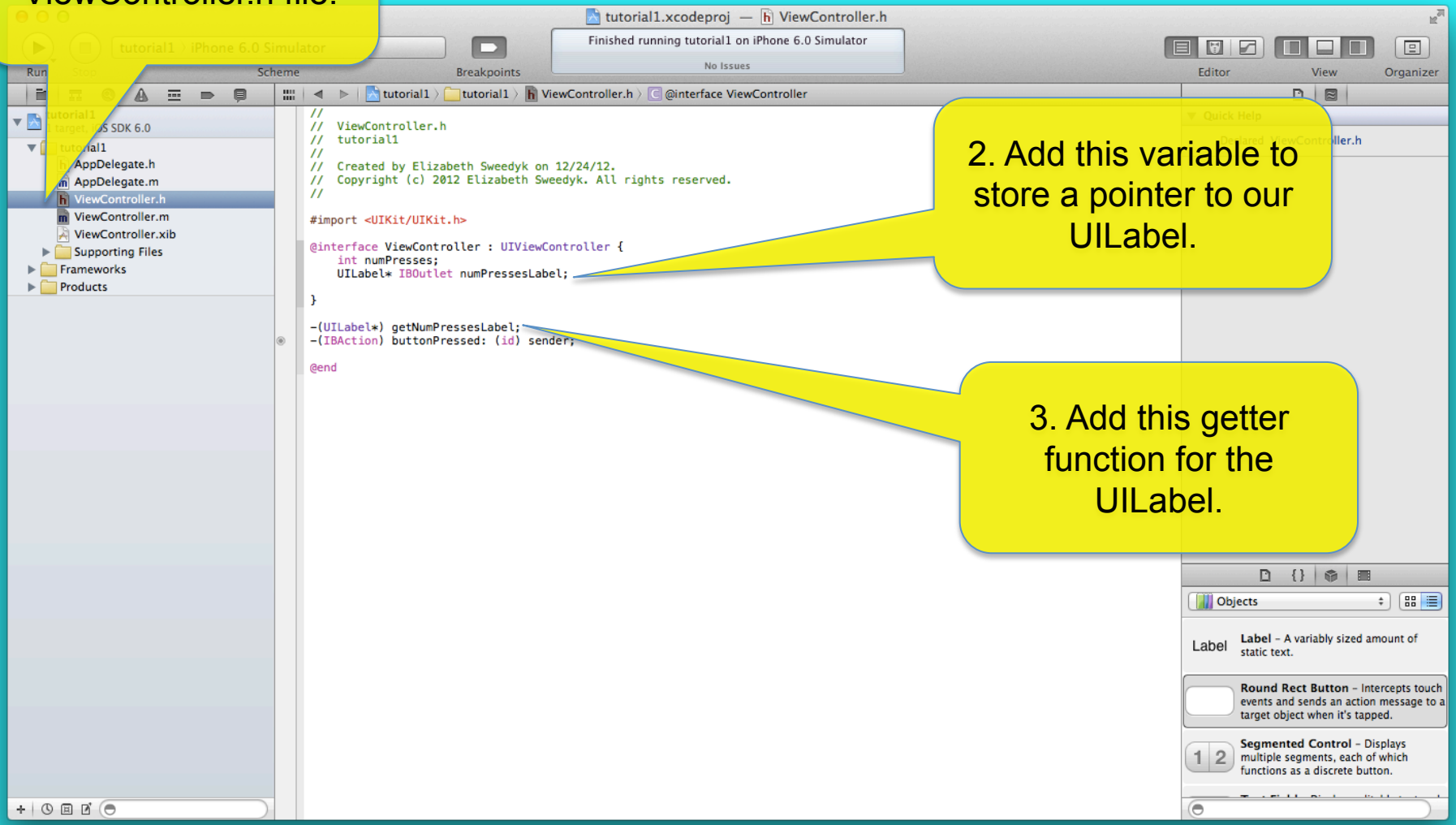

1. If the console is not visible at the bottom of the editor panel, click here.
Execution stops at the breakpoint.
2. Click here to step past the breakpoint
Note: numPresses =0.
tutorial1.xcodeproj — ViewController.m
Breakpoints
Editor
View
Organizer
tutorial1 › ViewController.m › No Selection
Quick Help
No Quick Help
Search Documentation
numPresses ++;
Thread 1: breakpoint 1.1
- (void)viewDidLoad
[super viewDidLoad];
// Do any additional setup after loading the view, typically from a nib.
- (void)didReceiveMemoryWarning
[super didReceiveMemoryWarning];
// Dispose of any resources that can be recreated.
Thread 1
0 -[ViewController buttonPressed:]
Auto
All Output
Clear
(lldb)
self = (ViewController *) 0x07155490
sender = (UIRoundedRectButton *) 0x07158810
numPresses = (int) 0
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.


tutorial1.xcodeproj — ViewController.m
Run
Stop
tutorial1 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running tutorial1 on iPhone 6.0 Simulator
No Issues
Editor
View
Organizer
tutorial1
Paused
Thread 1
com.apple.main-thread
0 -[ViewController buttonPressed:]
1 -[NSObject performSelector:wit...
20 UIApplicationMain
21 main
Thread 3
com.apple.libdispatch-manager
Thread 5 WebThread
Thread 6
tutorial1 〉 tutorial1 〉 ViewController.m 〉 No Selection
//
// ViewController.m
// tutorial1
//
// Created by Elizabeth Sweedyk on 12/24/12.
// Copyright (c) 2012 Elizabeth Sweedyk. All rights reserved.
//
#import "ViewController.h"
@interface ViewController ()
@end
@implementation ViewController
- (IBAction) buttonPressed:(id)sender
{
numPresses ++;
}
Thread 1: step over
- (void)viewDidLoad
{
[super viewDidLoad];
// Do any additional setup after loading the view, typically from a nib.
numPresses=0;
}
- (void)didReceiveMemoryWarning
{
[super didReceiveMemoryWarning];
// Dispose of any resources that can be recreated.
}
@end
Now numPresses =1.
Quick Help
No Quick Help
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.
tutorial1 〉 Thread 〉 0 -[ViewController buttonPressed:]
Auto
All Output
Clear
self = (ViewController *) 0x0715549
sender = (UIRoundedRectButton *) 0x07158810
numPresses = (int) 1
(lldb)

Next we will display the number of button presses in the label field.

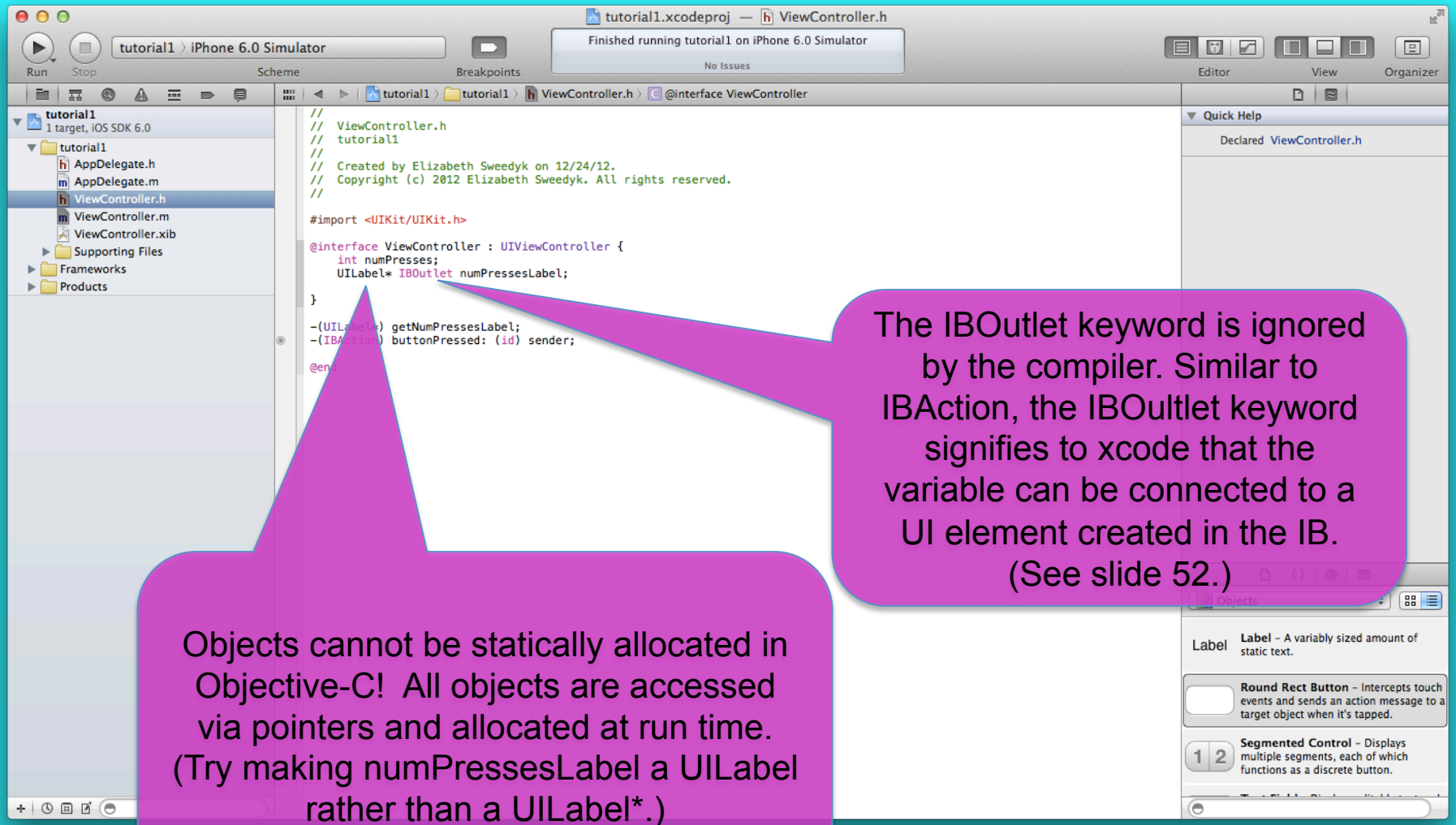
Our viewController needs to provide a UILabel reference to the “screen.”

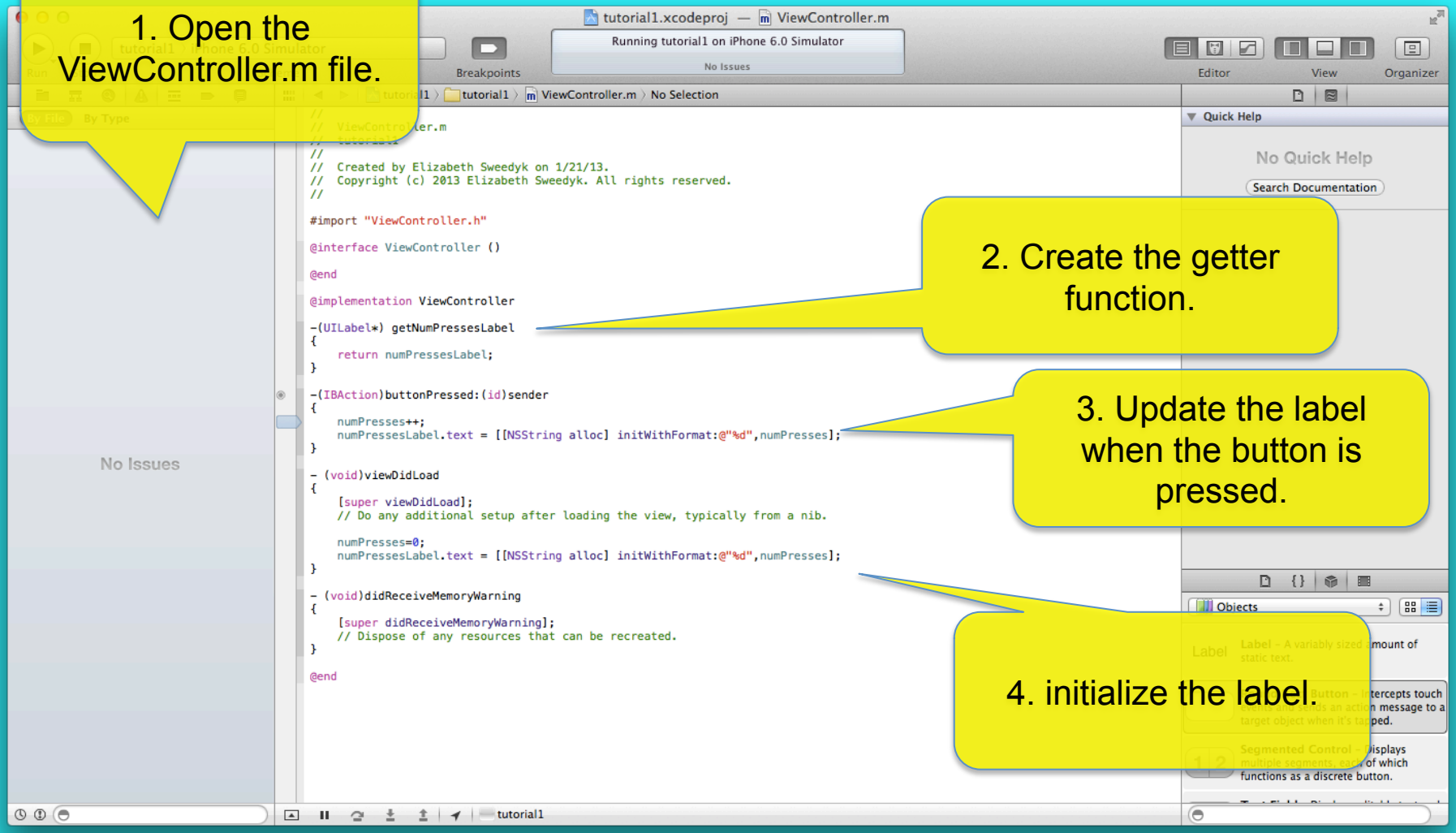
1. Open the ViewController.h file.



2. Add this variable to store a pointer to our UILabel.

3. Add this getter function for the UILabel.


tutorial1.xcodeproj — ViewController.h
Run
Stop
tutorial1 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running tutorial1 on iPhone 6.0 Simulator
No Issues
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
tutorial1
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
tutorial1 〉 tutorial1 〉 ViewController.h 〉 @interface ViewController
//
// ViewController.h
// tutorial1
//
// Created by Elizabeth Sweedyk on 12/24/12.
// Copyright (c) 2012 Elizabeth Sweedyk. All rights reserved.
//
#import <UIKit/UIKit.h>
@interface ViewController : UIViewController {
int numPresses;
UILabel* IBOutlet numPressesLabel;
}
-(UILabel*) getNumPressesLabel;
-(IBAction) buttonPressed: (id) sender;
@end
Quick Help
Declared ViewController.h
Label
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.
The IBOutlet keyword is ignored by the compiler. Similar to IBAction, the IBOultlet keyword signifies to xcode that the variable can be connected to a UI element created in the IB. (See slide 52.)
Objects cannot be statically allocated in Objective-C! All objects are accessed via pointers and allocated at run time. (Try making numPressesLabel a UILabel rather than a UILabel*.)


1. Open the ViewController.m file.
tutorial1.xcodeproj — ViewController.m
Running tutorial1 on iPhone 6.0 Simulator
No Issues
Breakpoints
Editor
View
Organizer
tutorial1 › tutorial1 › ViewController.m › No Selection
//
// ViewController.m
//
// Created by Elizabeth Sweedyk on 1/21/13.
// Copyright (c) 2013 Elizabeth Sweedyk. All rights reserved.
//
#import "ViewController.h"
@interface ViewController ()
@end
@implementation ViewController
-(UILabel*) getNumPressesLabel
{
return numPressesLabel;
}
-(IBAction)buttonPressed:(id)sender
{
numPresses++;
numPressesLabel.text = [[NSString alloc] initWithFormat:@"%d",numPresses];
}
- (void)viewDidLoad
{
[super viewDidLoad];
// Do any additional setup after loading the view, typically from a nib.
numPresses=0;
numPressesLabel.text = [[NSString alloc] initWithFormat:@"%d",numPresses];
}
- (void)didReceiveMemoryWarning
{
[super didReceiveMemoryWarning];
// Dispose of any resources that can be recreated.
}
@end
Quick Help
No Quick Help
Search Documentation
2. Create the getter function.
3. Update the label when the button is pressed.
4. initialize the label.
No Issues
Objects
tutorial1

```
numPressesLabel.text = [[NSString alloc] initWithFormat: @"%d",numPresses;]
```

We are going to set the *text* field of our UILabel object.

In C++ we'd say *numPressesLabel->text* = …because *numPressesLabel* is a pointer.

But the only way to access an object in Objective-C is through a pointer and the dot notation is used.

The good news is you don't have to remember whether you have an object or a pointer to an object; in Objective-C it will always be the latter.

```
numPressesLabel.text = [[NSString alloc] initWithFormat: @"%d",numPresses;]
```

text is a data member of the UILabel class. It is a pointer to a string; more specifically a pointer to an NSString, which is iOS's string class.

NSString is immutable. (There is another string class that is mutable.) If we want to change the string pointed to by *text* we have to allocate new space and initialize it.

```
numPressesLabel.text = [[NSString alloc] initWithFormat: @"%d",numPresses;]
```

Object creation is a two step process. We first allocate space for the string with [NSString alloc].

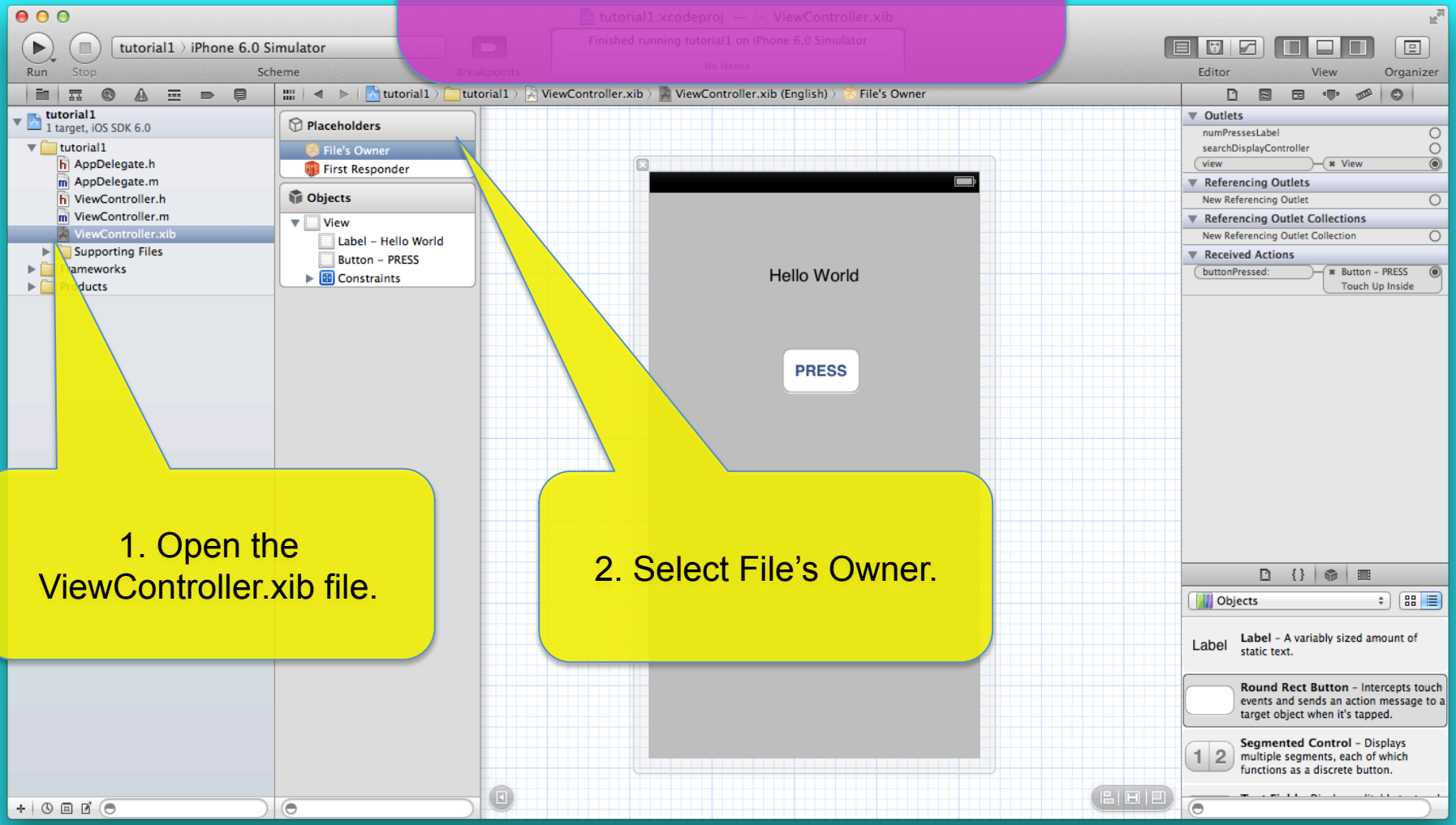
```
numPressesLabel.text = [[NSString alloc] initWithFormat: @"%d",numPresses;]
```

The second step is to initialize the newly allocated space.

NSString has a variety of initialization methods; here we use *initWithFormat*.

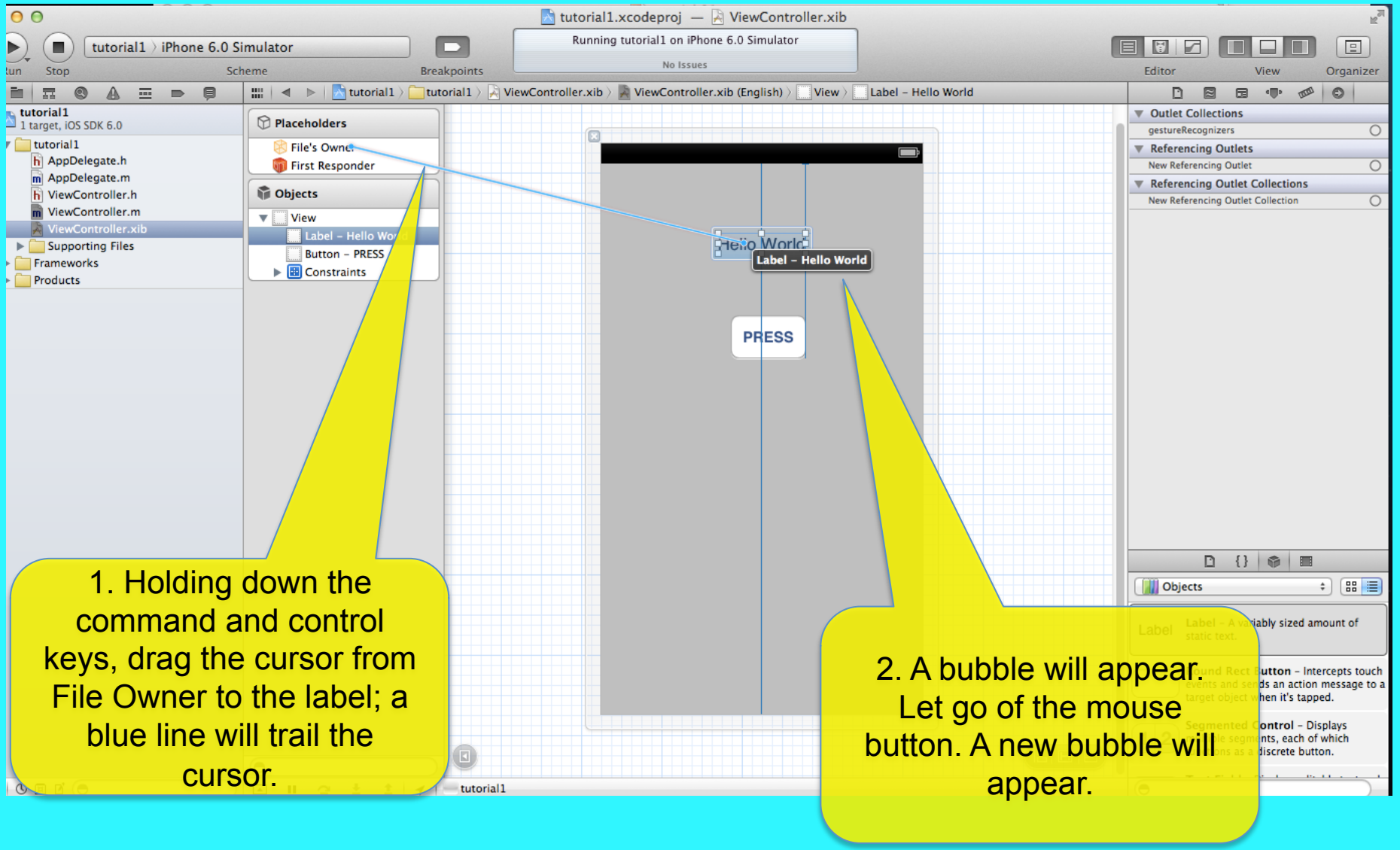
Note an @ before a string means it is an NSString as opposed to a C-style string.

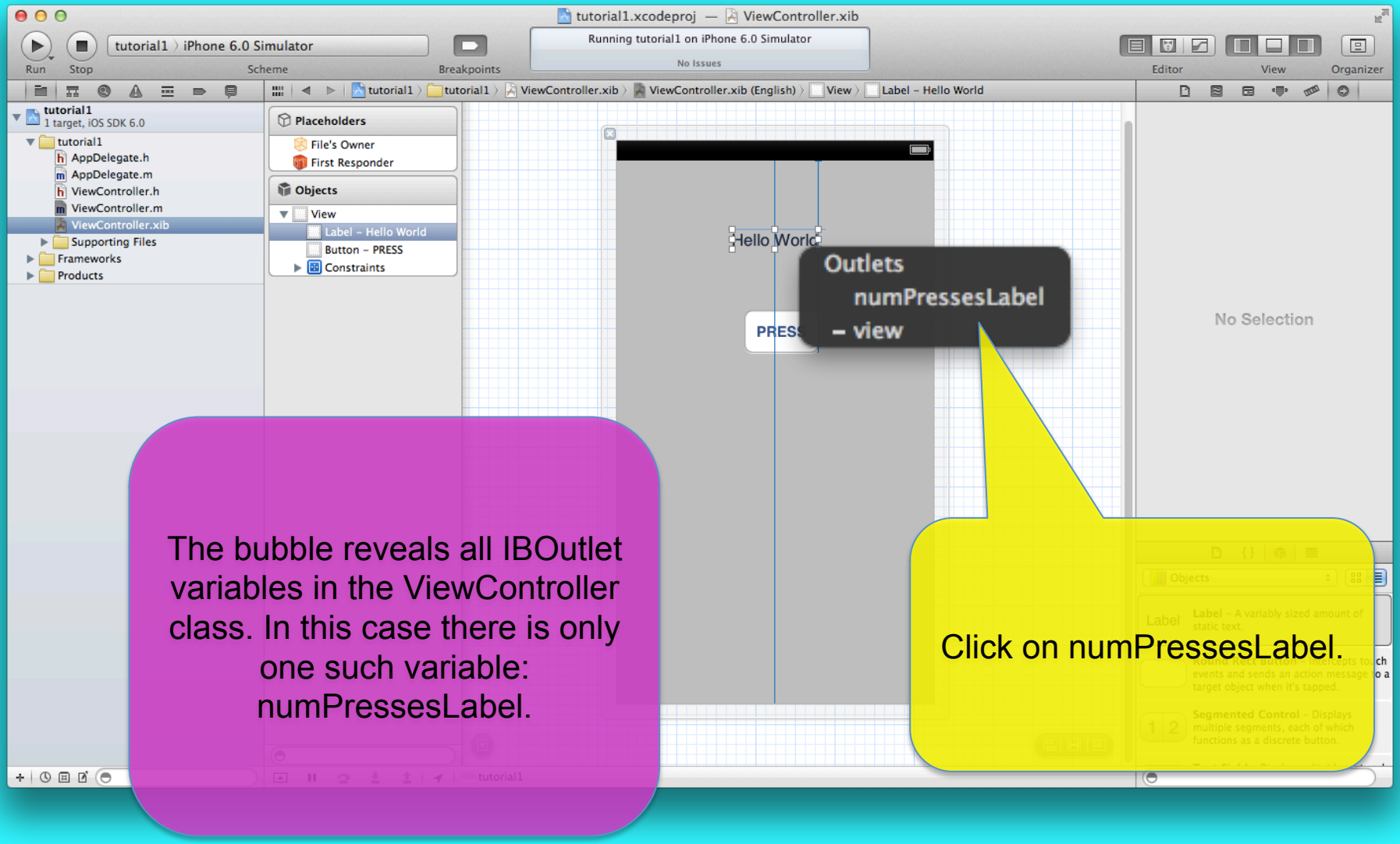
Now we'll connect *numPressesLabel* to the label on the screen.

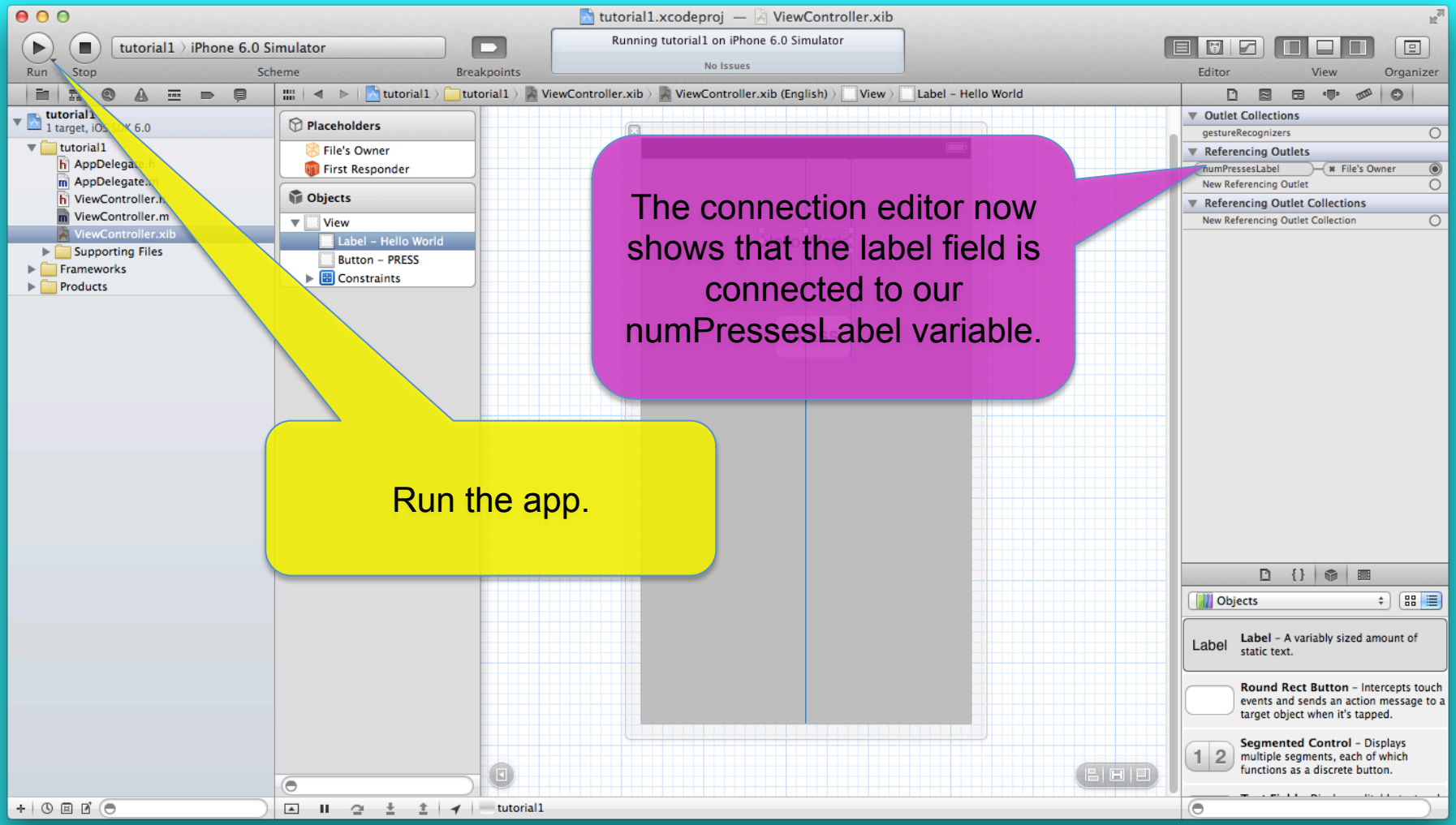


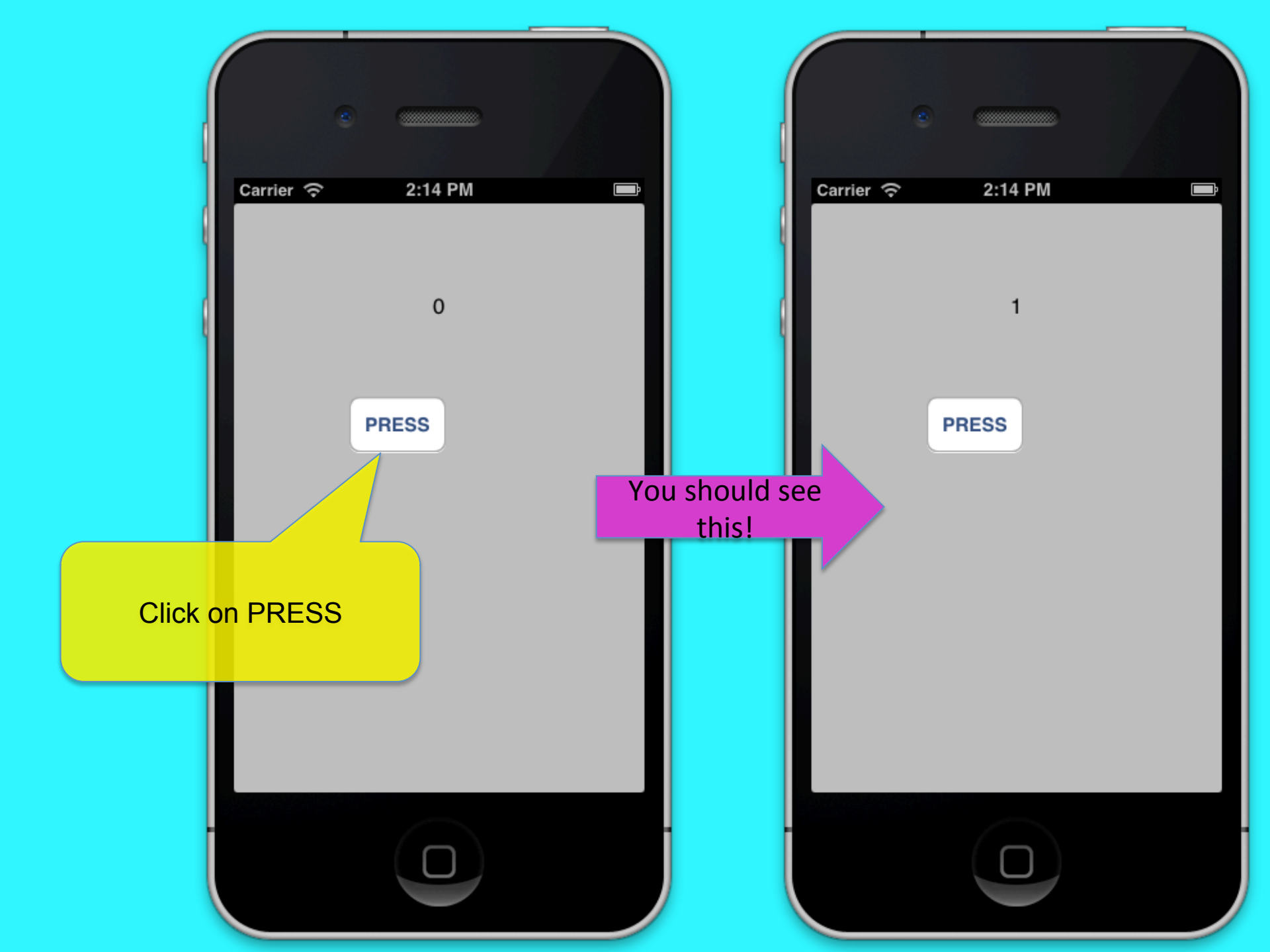
1. Open the ViewController.xib file.

2. Select File's Owner.


tutorial1.xcodeproj — ViewController.xib
tutorial1 〉 iPhone 6.0 Simulator
Running tutorial1 on iPhone 6.0 Simulator
No Issues
Stop
Scheme
Breakpoints
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
Placeholders
File's Owner
First Responder
Objects
View
Label – Hello World
Button – PRESS
Constraints
Label – Hello World
PRESS
Outlet Collections
gestureRecognizers
Referencing Outlets
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Objects
1. Holding down the command and control keys, drag the cursor from File Owner to the label; a blue line will trail the cursor.
2. A bubble will appear. Let go of the mouse button. A new bubble will appear.

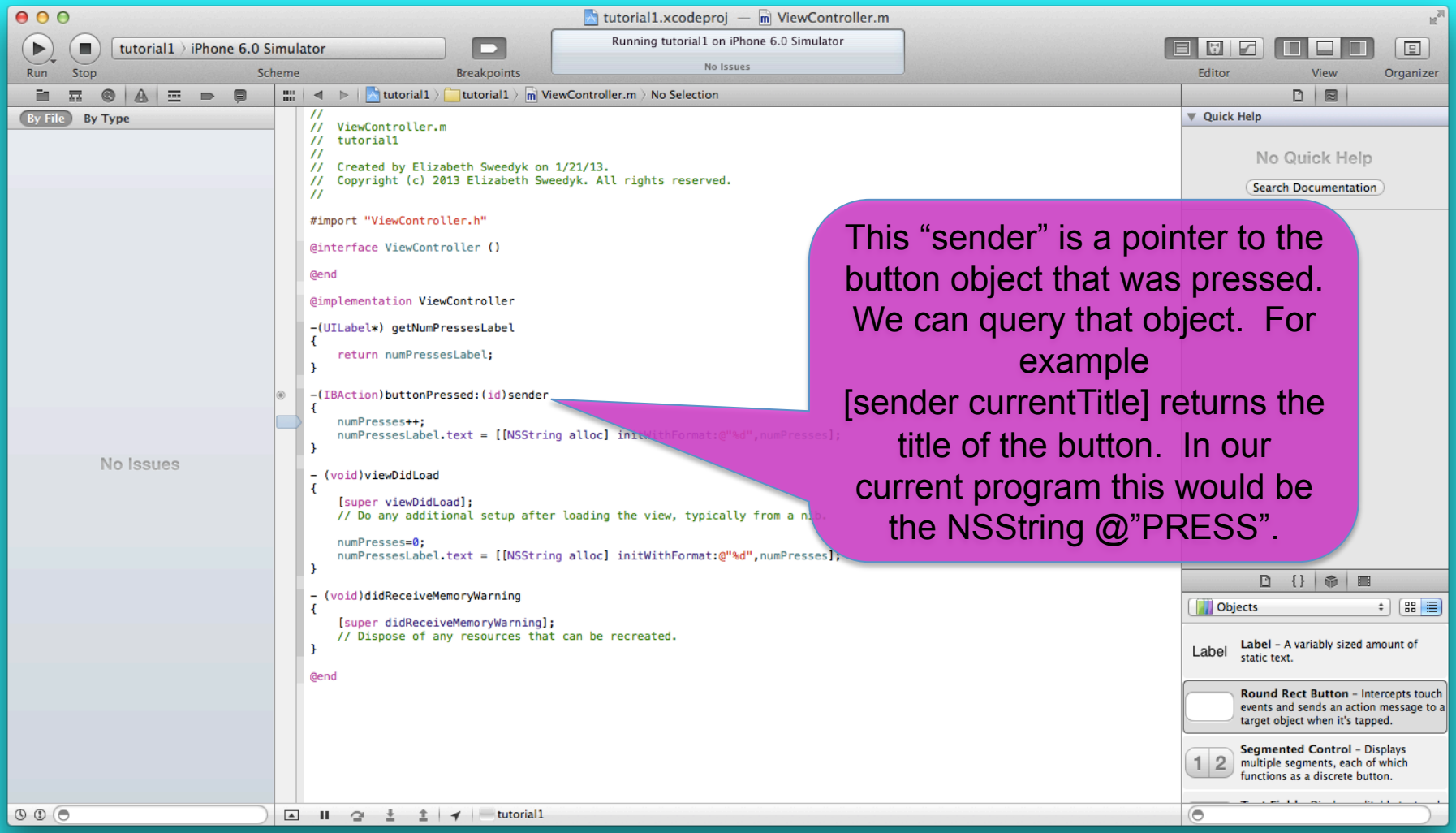

tutorial1.xcodeproj — ViewController.xib
Run
Stop
tutorial1 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running tutorial1 on iPhone 6.0 Simulator
No Issues
Editor
View
Organizer
tutorial1
1 target, iOS SDK 6.0
tutorial1
AppDelegate.h
AppDelegate.m
ViewController.h
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
Placeholders
File's Owner
First Responder
Objects
View
Label – Hello World
Button – PRESS
Constraints
Hello World
PRESS
Outlets
numPressesLabel
– view
No Selection
The bubble reveals all IBOutlet variables in the ViewController class. In this case there is only one such variable: numPressesLabel.
Click on numPressesLabel.


tutorial1.xcodeproj — ViewController.xib
Run
Stop
tutorial1 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running tutorial1 on iPhone 6.0 Simulator
No Issues
Editor
View
Organizer
tutorial1
tutorial1
ViewController.m
ViewController.xib
Supporting Files
Frameworks
Products
Placeholders
File's Owner
First Responder
Objects
View
Label – Hello World
Button – PRESS
Constraints
Outlet Collections
gestureRecognizers
Referencing Outlets
numPressesLabel
File's Owner
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.
The connection editor now shows that the label field is connected to our numPressesLabel variable.
Run the app.


Carrier
2:14 PM
0
PRESS
Click on PRESS
You should see this!
Carrier
2:14 PM
1
PRESS

Next you are going to modify the app so that there are two buttons labeled 0 and 1. The text field of the UILabel should say “The last button pressed was n”, where n is either 0 or 1.

There are two things you need to know.


tutorial1.xcodeproj — ViewController.m
Run
Stop
tutorial1 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running tutorial1 on iPhone 6.0 Simulator
No Issues
Editor
View
Organizer
By File
By Type
No Issues
tutorial1 〉 tutorial1 〉 ViewController.m 〉 No Selection
//
// ViewController.m
// tutorial1
//
// Created by Elizabeth Sweedyk on 1/21/13.
// Copyright (c) 2013 Elizabeth Sweedyk. All rights reserved.
//
#import "ViewController.h"
@interface ViewController ()
@end
@implementation ViewController
-(UILabel*) getNumPressesLabel
{
return numPressesLabel;
}
-(IBAction)buttonPressed:(id)sender
{
numPresses++;
numPressesLabel.text = [[NSString alloc] initWithFormat:@"%d",numPresses];
}
- (void)viewDidLoad
{
[super viewDidLoad];
// Do any additional setup after loading the view, typically from a nib.
numPresses=0;
numPressesLabel.text = [[NSString alloc] initWithFormat:@"%d",numPresses];
}
- (void)didReceiveMemoryWarning
{
[super didReceiveMemoryWarning];
// Dispose of any resources that can be recreated.
}
@end
This “sender” is a pointer to the button object that was pressed. We can query that object. For example [sender currentTitle] returns the title of the button. In our current program this would be the NSString @”PRESS”.
Quick Help
No Quick Help
Search Documentation
Objects
Label
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
1 2
Segmented Control – Displays multiple segments, each of which functions as a discrete button.
tutorial1

The second is how to construct the NSString @”The last button pressed was n” where n is either 0 or 1.
See if you can figure out! Help will be provided on request.

Zip up your project folder and move it to your Sakai dropbox for cs121.