Assignment 6

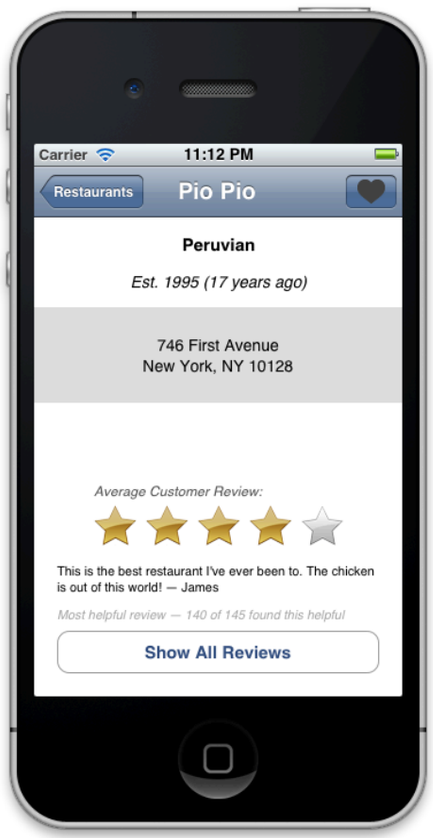
IBActions

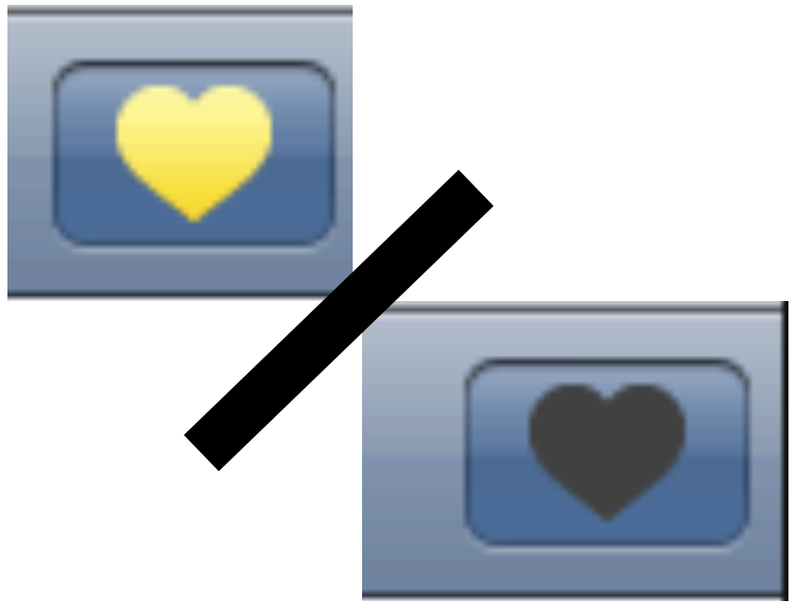
How do we make buttons do things other than change views?

- IBActions
- An IBAction is like an IBOutlet
- When a user manipulates a control on screen (like a button) a special method on the ViewController, called an IBAction, will be called.
- All IBAction methods look like this:
  - `- (IBAction)doSomething:(id)sender;`

Goal

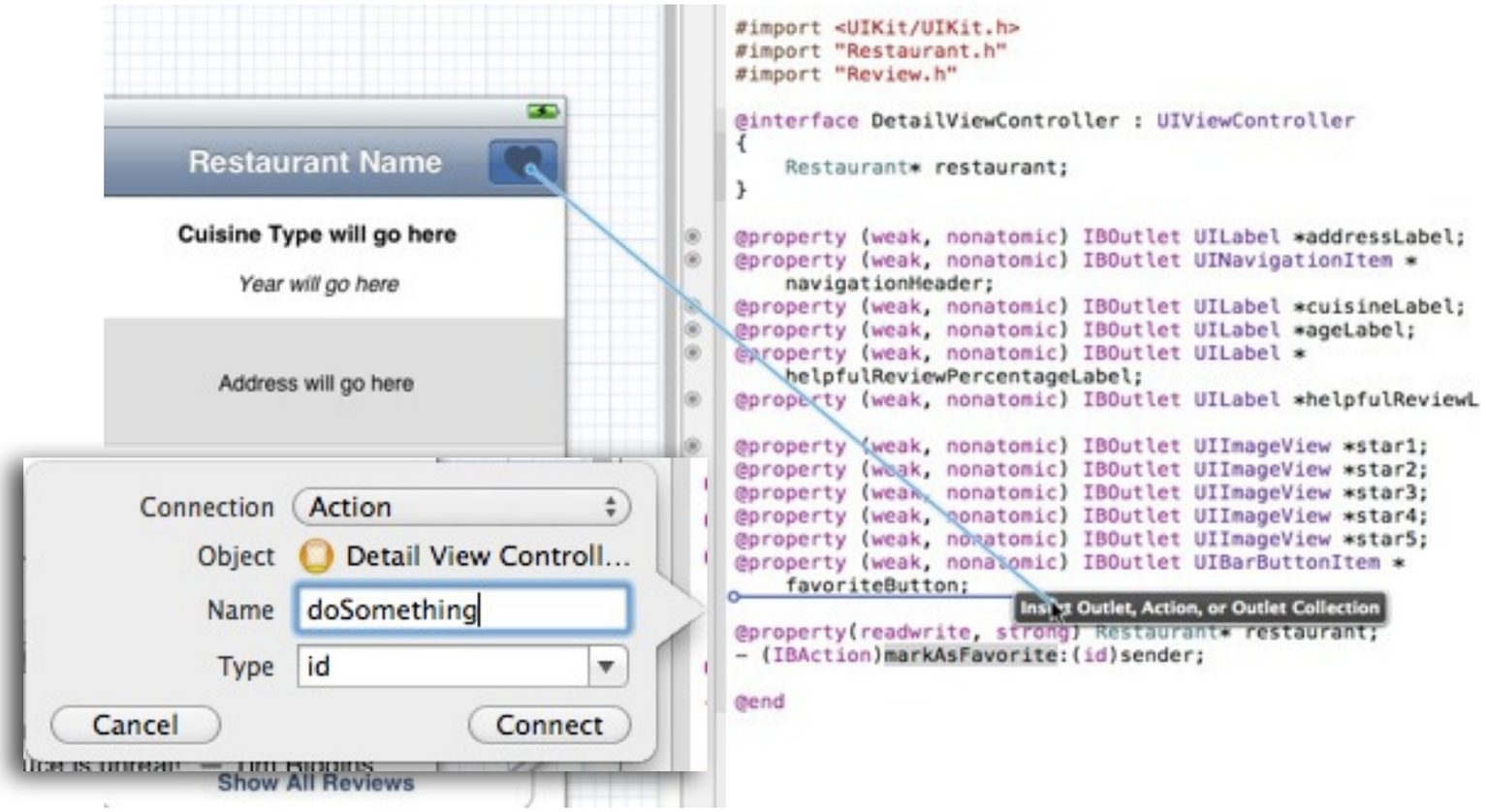
- Create a favorites button, that when pressed toggles a new property on the restaurant, isFavorite
- Ensure that the favorite button's image always is in sync with isFavorite.





Creating an IBAction

- With the assistant view showing on the storyboard



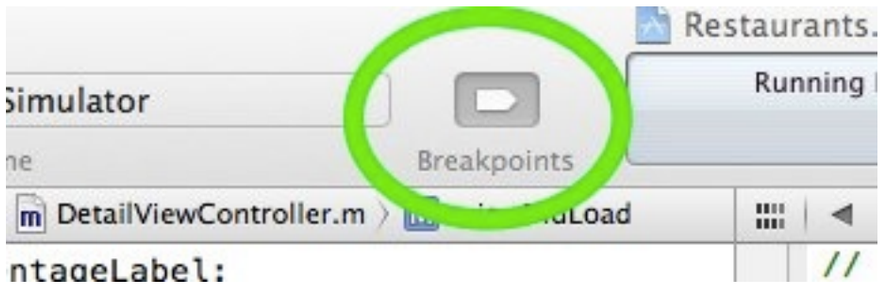
- Drag from the control to the code.
- This will insert a method (and method signature) in the class.

Debugging with Breakpoints

- A breakpoint pauses your code at a specific point.
- When your code is stopped, you can check the value of a variable at that moment.
- This is helpful in loops and complex functions, and when something isn't working right but you can't see why.

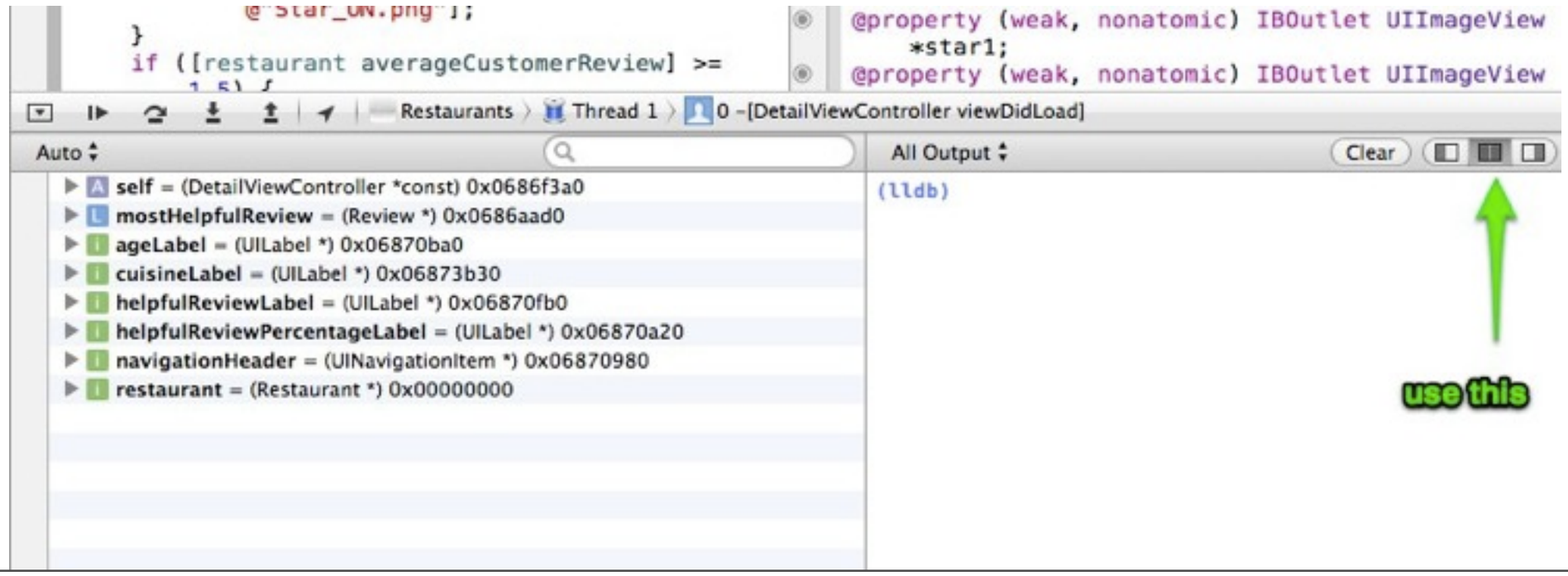
Creating a Breakpoint

- Just click in the left rail
- To remove a breakpoint, drag it out of the rail
- To temporarily disable it, click it (it will toggle on and off)
- To disable all breakpoints, click the Breakpoints button at the top of Xcode.



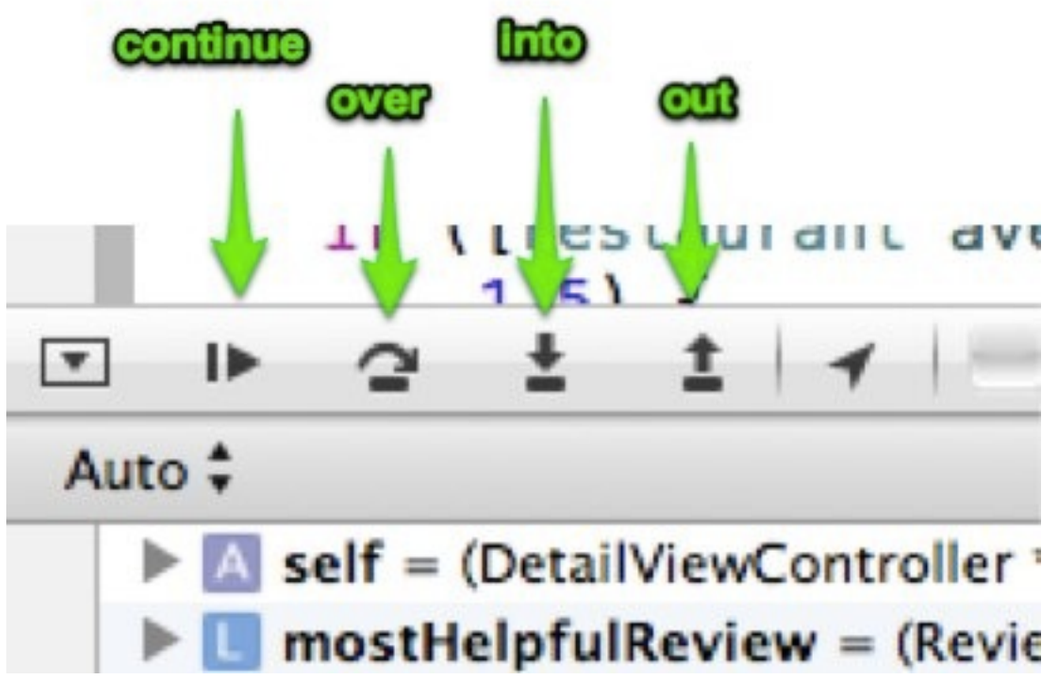
Using a Breakpoint

- Run your application normally.
- When you hit a breakpoint, your application will pause, and Xcode will come to the front, with a green marker where it has paused.
- You can now see the contents of your variables below. 0x0 indicates nil, all other numbers indicate it has a value.
- You can expand values to inspect deeper



Stepping

- You can unpause by hitting the continue button.
- (The code will pause again if it hits another breakpoint)
- If you wish to go line by line, use the step over button.
- You can use the step in and out buttons to navigate your code into a method and out of one.



Task 1: Create a Button

- The correct button to drop in is the UIBarButtonItem
- Download the 2 heart images from the blog, and drag them into your project (make sure you copy resources)
- Set the image of the button to the grey heart

Task 2: Create the IBAction

- CTRL-Drag from the button to the .h or .m file to create a new method
- Name the method “markAsFavorite”

Task 3: Create the Property

- Add a BOOL property to Restaurant called isFavorite.

Task 4: Create the Property

- Add a BOOL property to Restaurant called isFavorite.

Task 5: Toggle isFavorite in the button

- When the user taps isFavorite, change the restaurant's isFavorite property to be the opposite of it's current state.
- Tip: Use the ! operator to reverse a BOOL (true becomes false)
- Run your code. Your button should now be tappable. But is it hitting your action?

Task 6: Add an outlet for the button to control the image

- CTRL-Drag again, but this time set up an IBOutlet called favoriteButton.

Task 7: Change the image to match the state of the button

- In markFavorite:, set the image of the favoriteButton to be heart-selected.png when the restaurant is a favorite, heart.png when it is not.
- Tip: Try an if-else statement for this.
- Run your code and check for doneness. The image should toggle when tapping the button.
- After setting a favorite, go to the restaurant list and return to the Detail View. What's wrong?

Task 8: Set the image correctly on ViewDidLoad:

- Check the state of the isFavorite flag when the view loads and set the button's image accordingly.