iOS Foundations II
Session 4

- Passing data to View Controllers
- UITextField
- Plists

Plists

- Apple-flavored XML
- Keys must be Strings
- Values must be NSCoding compliant
- Load from your bundle or from the web
- Root-level object is typically a Dictionary (can also be an Array)

Plists - usage

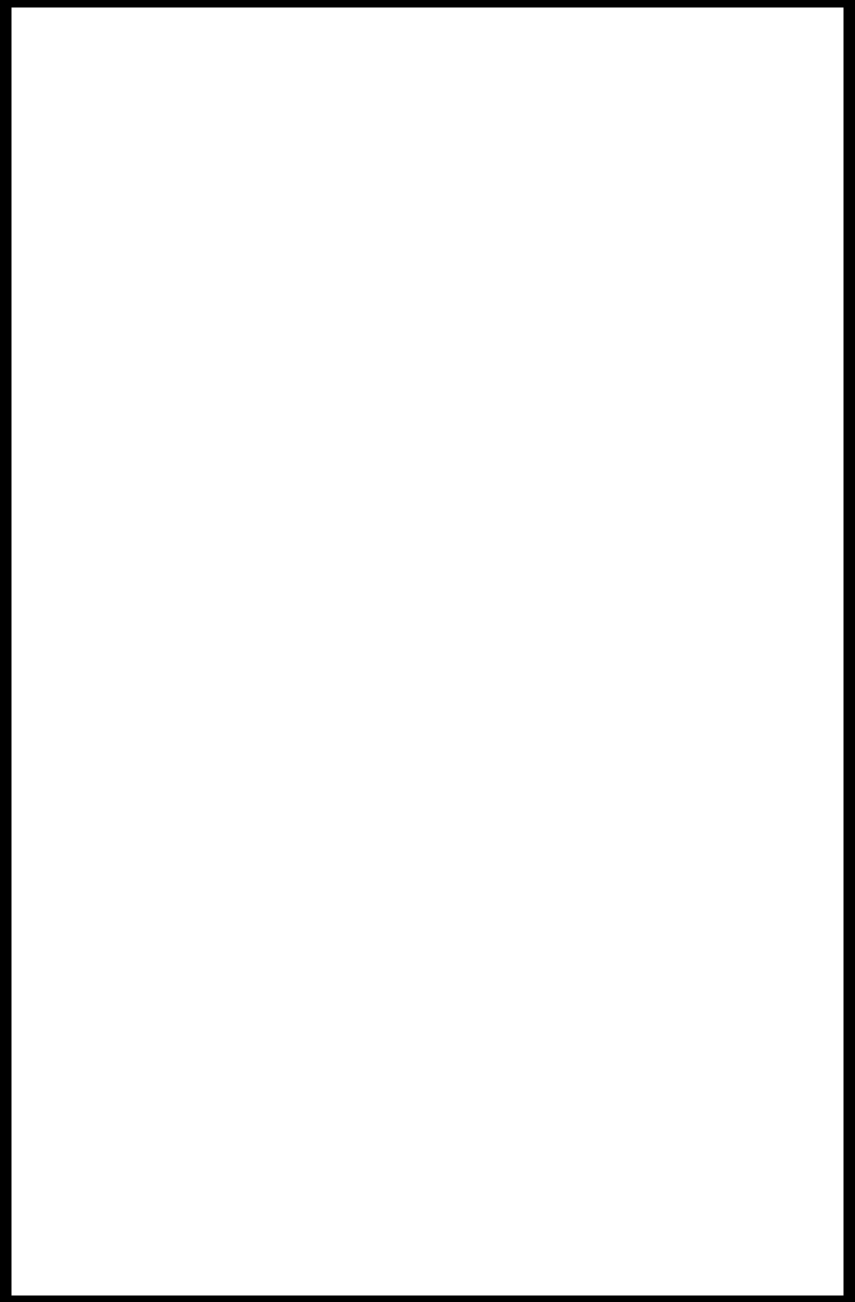
- Get the URL of the plist file
- Read the plist into an Array of Dictionary using class methods
- Use conditional unwrapping/casting to parse the data
- Use objectForKey: or subscripting: array[n]
- Use a for loop to parse a collection

Passing data to View Controllers

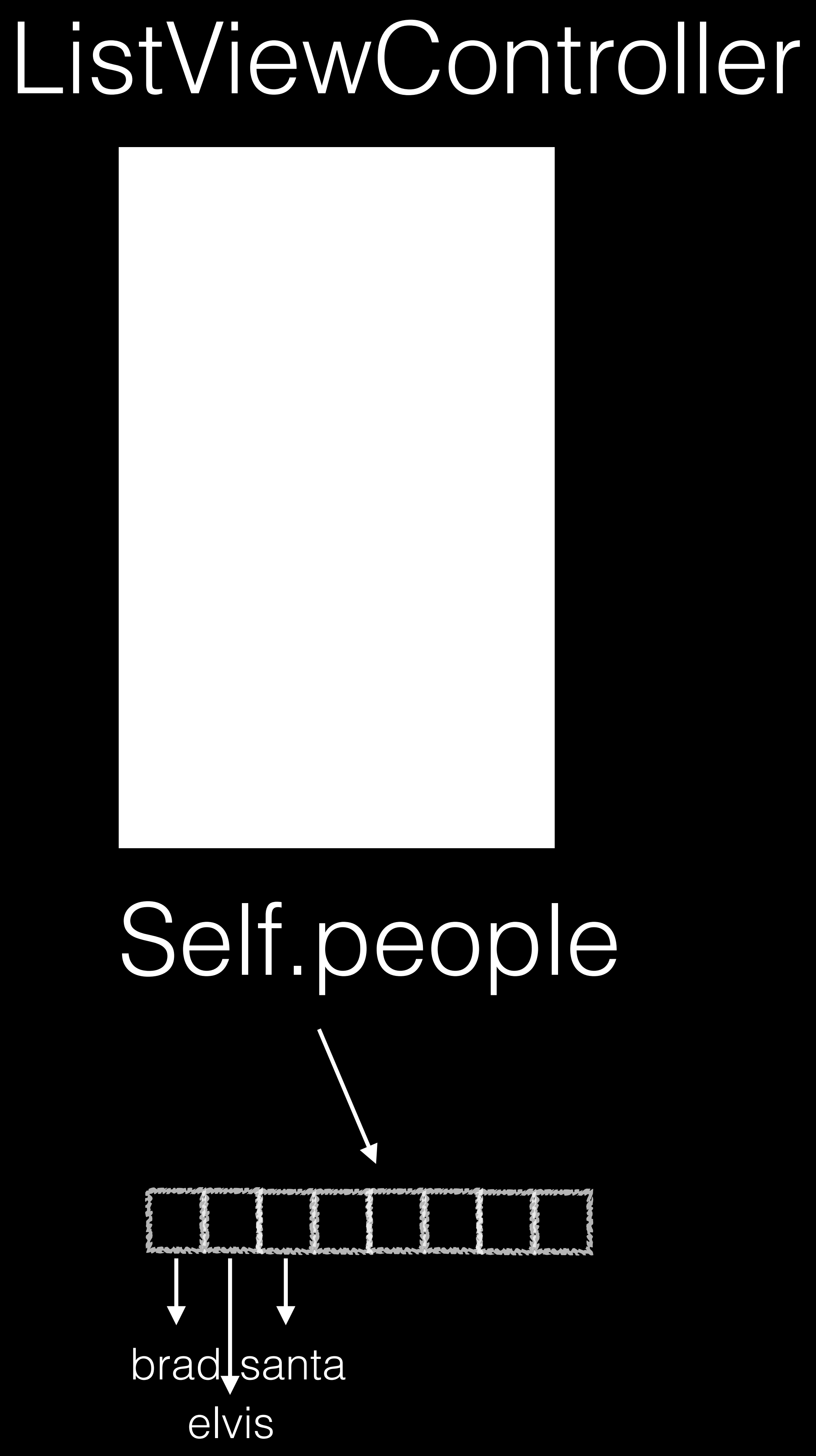
- There are many patterns we can use to pass data around our app:
  - Delegation
  - Notification Center
  - Singletons
- But for now, we can do something as simple as passing a reference directly to the new view controller.

Passing data to View Controllers.
Breaking it down, yo

ListViewController



ListViewController creates an array of Person objects and sets it people property ’s value to that array



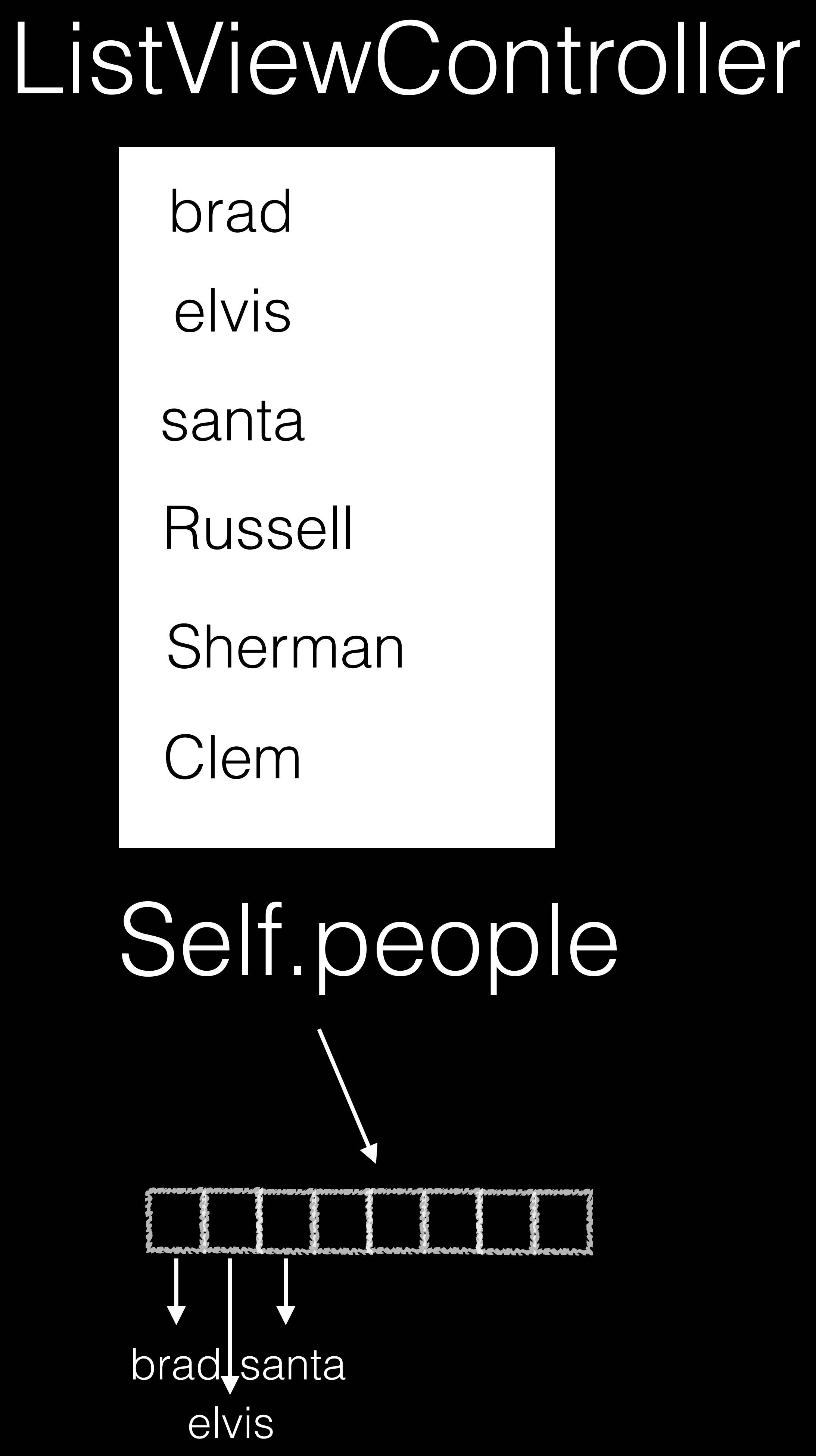
ListViewController

Self.people

brad santa

elvis

ListViewController, as the datasource of the tableview, uses the self.people array as the backing array for the tableview



ListViewController

brad

elvis

santa

Russell

Sherman

Clem

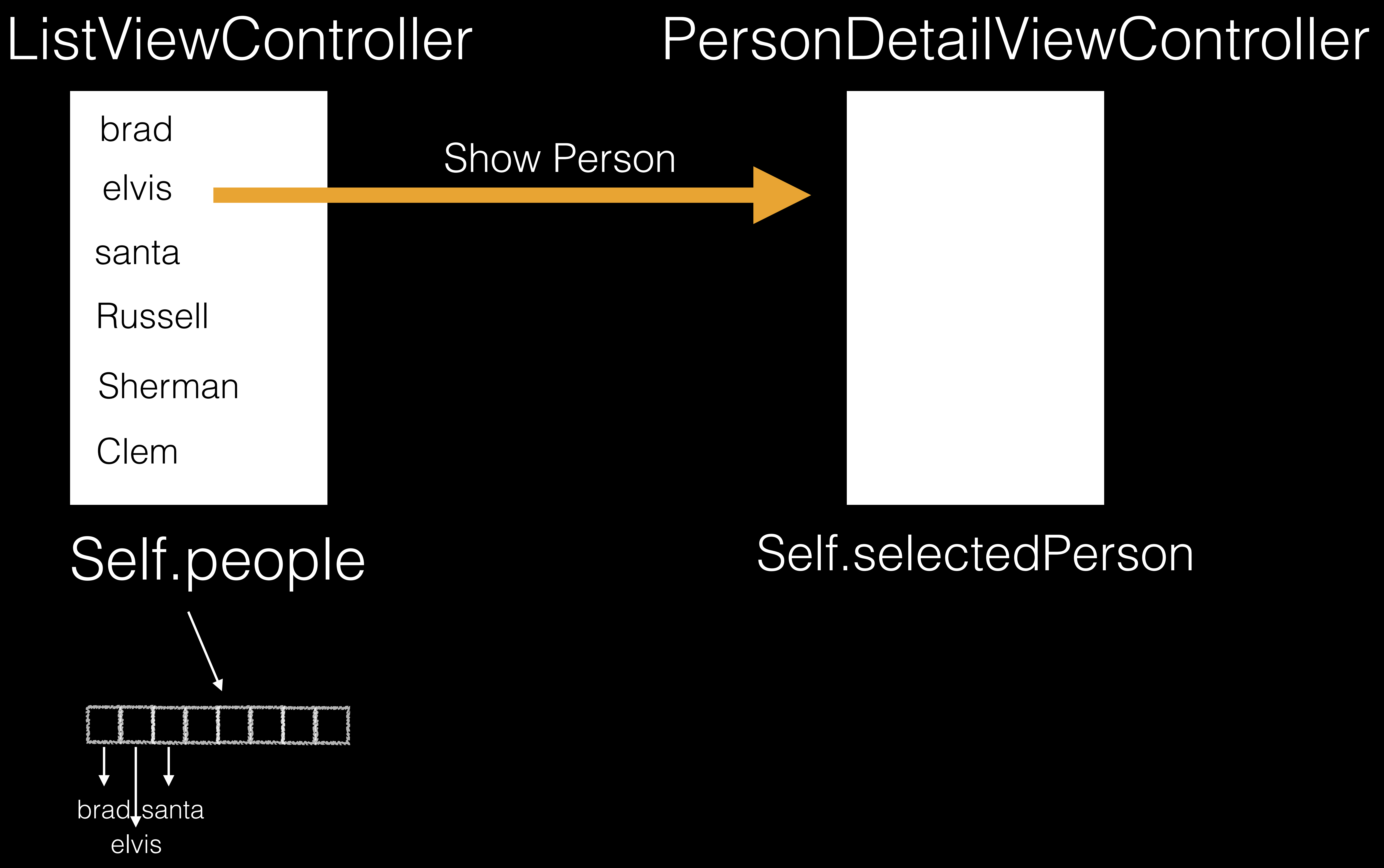
Self.people

brad

santa

elvis

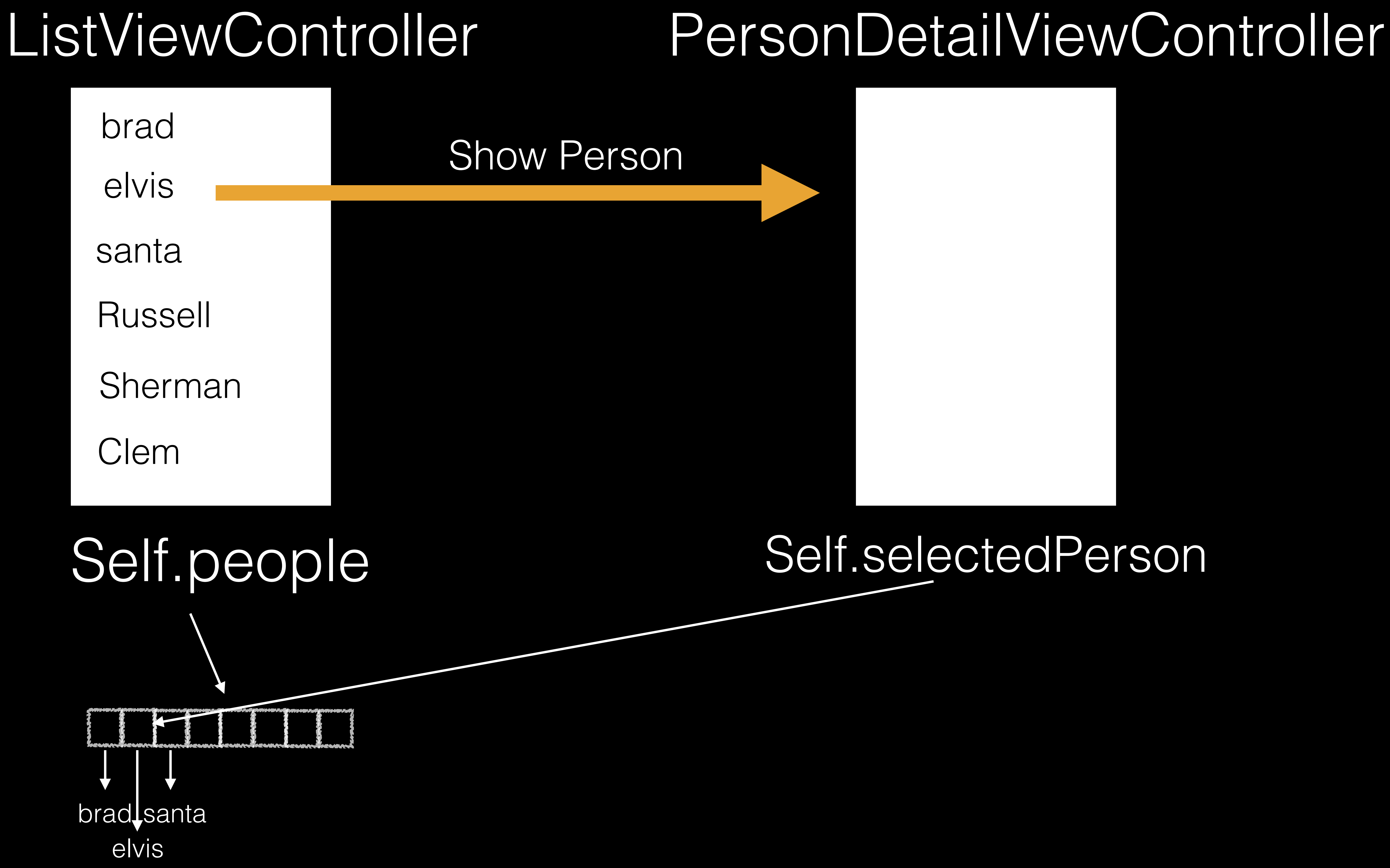
The User clicks on a cell, triggering our ShowPerson Segue. The PersonDetailViewController is initialized.



In Prepare for Segue, the ListViewController intercepts the segue and grabs a reference to the destination view controller, which is the PersonDetailViewController.

```
override func prepareForSegue(segue: UIStoryboardSegue!, sender: AnyObject!) {

    if segue.identifier == "ShowPerson" {

        var personDetailViewController = segue.destinationViewController as
  PersonDetailViewController

        //now we can prepare this view controller to be displayed with what
  ever data we need to pass to it.
    }
}
```

ListViewController sets PersonDetailViewController’s selectedPerson property to reference the person that was clicked from the list.



ListViewController

PersonDetailViewController

brad

elvis

santa

Russell

Sherman

Clem

Show Person

Self.people

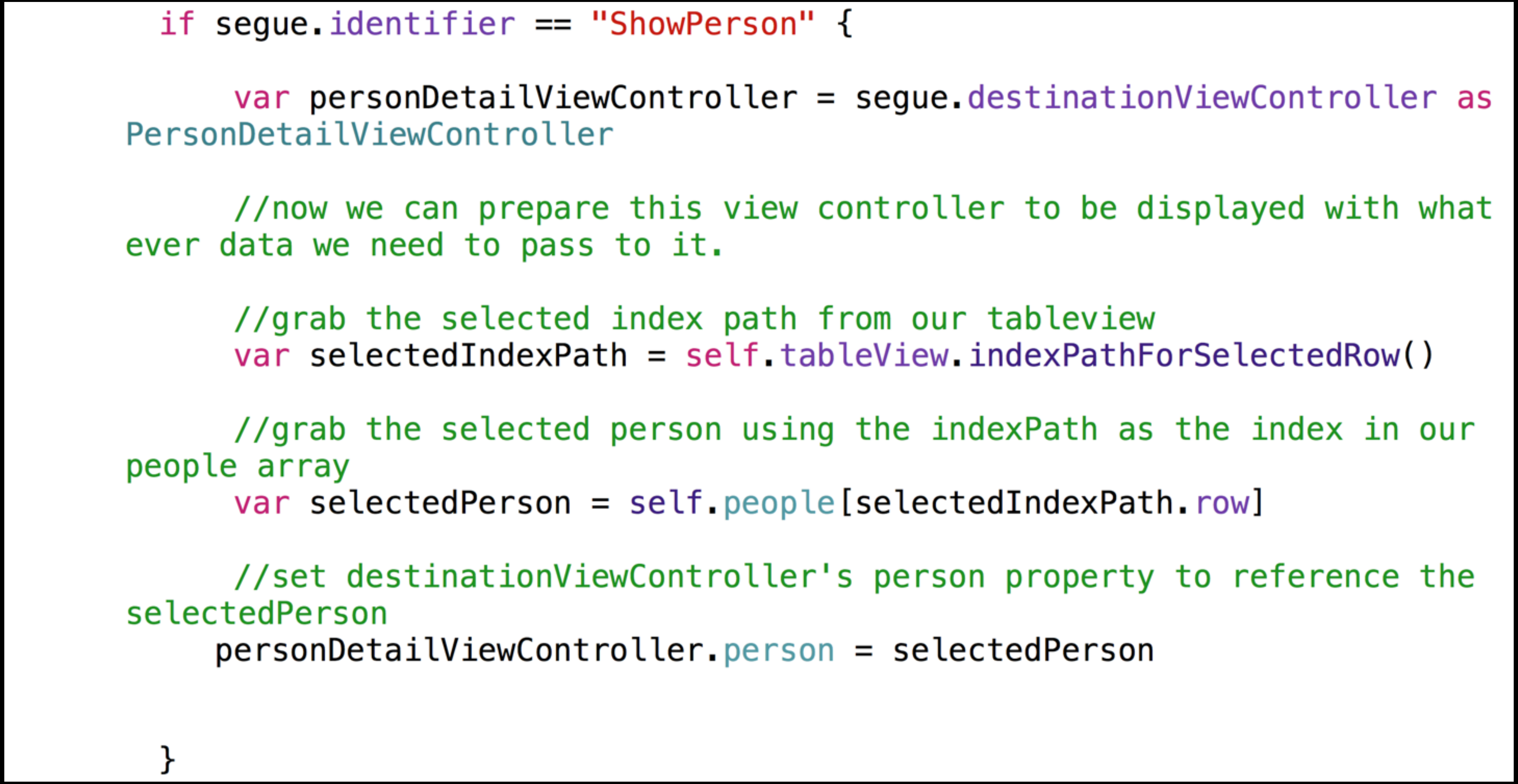
Self.selectedPerson

brad

santa

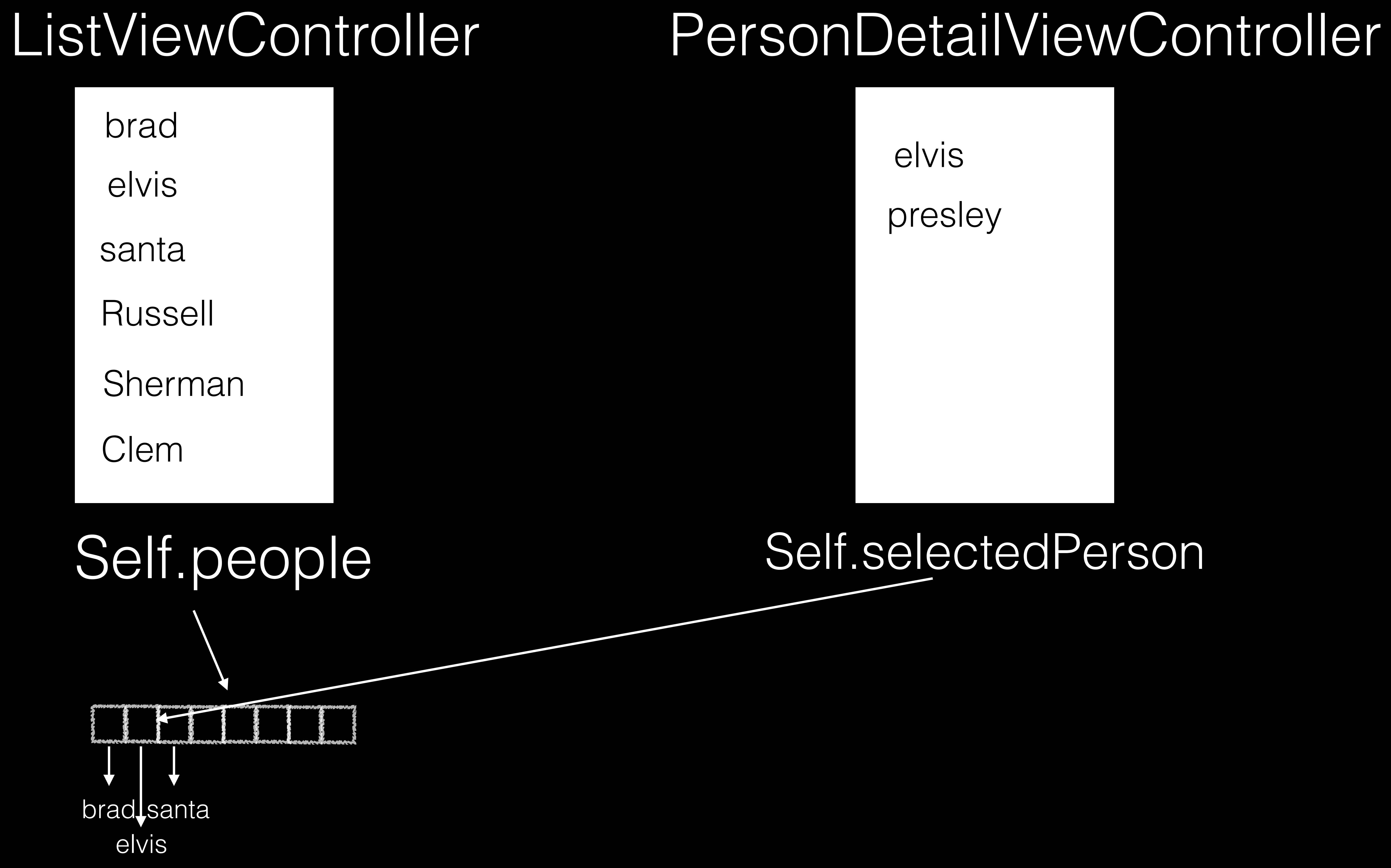
elvis

ListViewController sets PersonDetailViewController’s selectedPerson property to reference the person that was clicked from the list.



```
  if segue.identifier == "ShowPerson" {

      var personDetailViewController = segue.destinationViewController as
PersonDetailViewController

      //now we can prepare this view controller to be displayed with what
ever data we need to pass to it.

      //grab the selected index path from our tableview
      var selectedIndexPath = self.tableView.indexPathForSelectedRow()

      //grab the selected person using the indexPath as the index in our
people array
      var selectedPerson = self.people[selectedIndexPath.row]

      //set destinationViewController's person property to reference the
selectedPerson
     personDetailViewController.person = selectedPerson


  }
```

Now PersonDetailViewController can use it’s selectedPerson property to fill out his interface with that Person’s name properties. Any changes he makes to this Person object apply to the person object inside the original array!



ListViewController

brad

elvis

santa

Russell

Sherman

Clem

Self.people

PersonDetailViewController

elvis

presley

Self.selectedPerson

brad santa

elvis

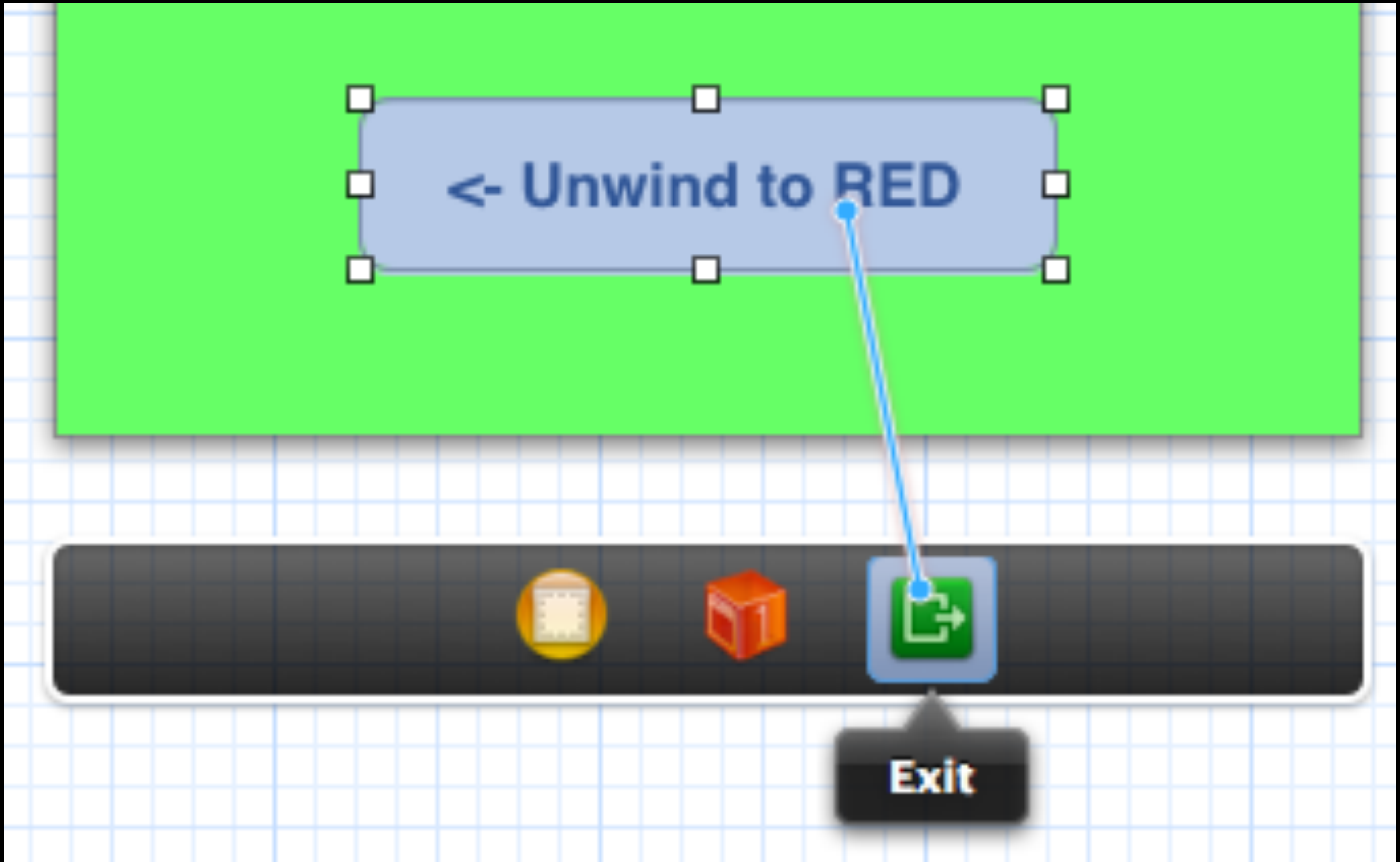
iOS Foundations II
Session 4.1

- Unwind Segues
- UITextFieldDelegate
- Key-Value Observing (NSNotificationCenter)
- UITextField Notifications

Unwind Segues

- UISegue for dismissing or popping the top view controller
- Connected from action trigger (button) to Exit icon
- Calls prepare for segue from sourceVC
- Dest. must first have an @IBAction that takes a UISegue arg

Unwind Segues



UITextFieldDelegate

- UITextField asks it's delegate before processing a key press
- Used for managing editing and changing the textField's text
- Useful for restricting certain characters (e.g. whitespace in a url)

UITextFieldDelegate

Managing Editing

- textFieldShouldBeginEditing:
- textFieldDidBeginEditing:
- textFieldShouldEndEditing:
- textFieldDidEndEditing:

`UITextFieldDelegate`

Editing the Text Field’s Text

-textField:shouldChangeCharactersInRange:replacementString:

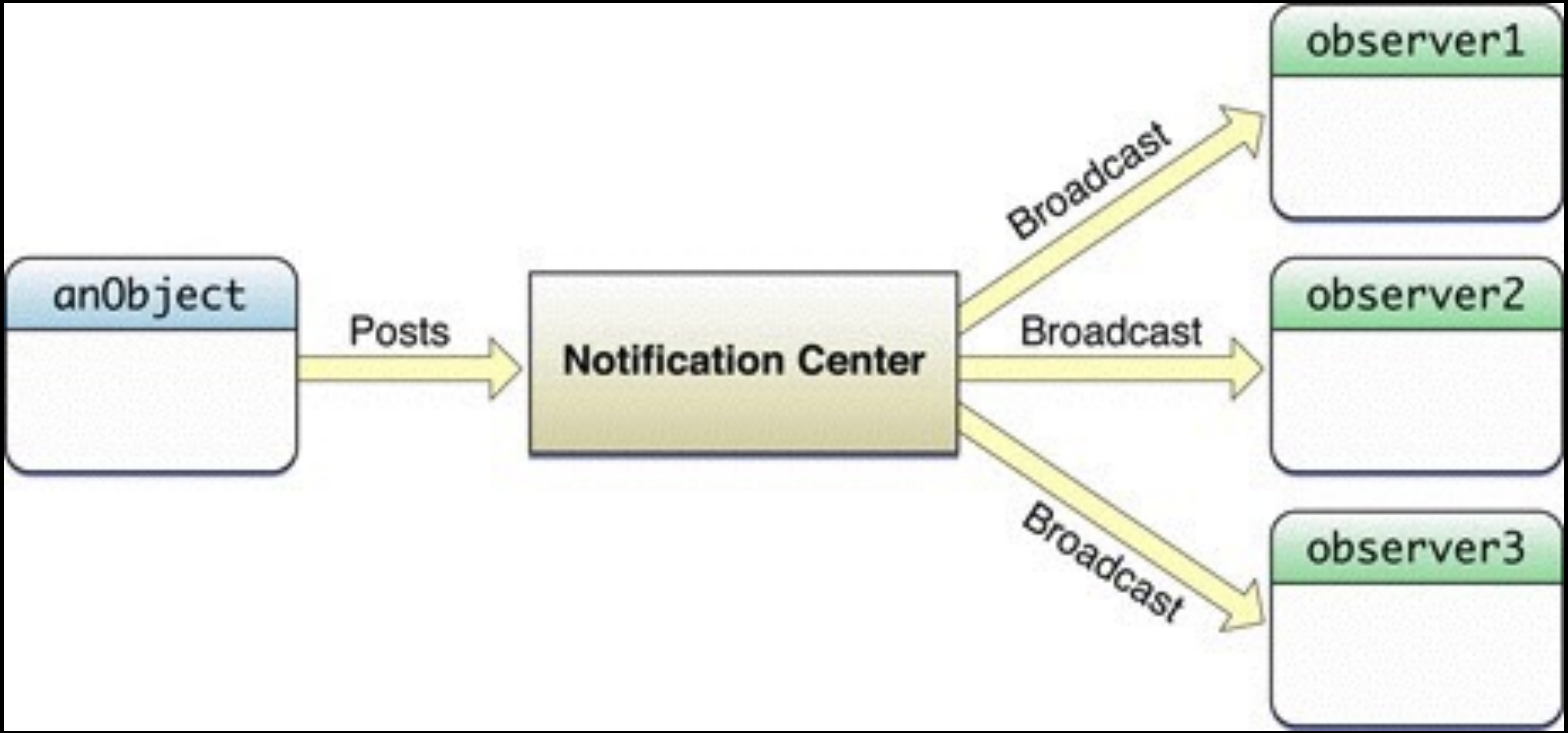
-textFieldShouldClear:

-textFieldShouldReturn:

Key-Value-Observing (KVO)

- Subscribe to (observe) a key path to respond to changes & events
- Publish to a key-path to broadcast an event or change
- Notifications are one-to-many
- Delegates are typically one-to-one

NSNotificationCenter



UITextFieldTextDidBeginEditingNotification

Notifies observers that an editing session began in a text field. The affected text field is stored in the `object` parameter of the notification. The `userInfo` dictionary is not used.

`UITextFieldTextDidChangeNotification`

Notifies observers that the text in a text field changed. The affected text field is stored in the `object` parameter of the notification.

`UITextFieldTextDidEndEditingNotification`

Notifies observers that the editing session ended for a text field. The affected text field is stored in the `object` parameter of the notification. The `userInfo` dictionary is not used.