PickerViews

PickerView Operations

Theory:

- What are pickerviews
- What are examples of where pickerviews used
- What is a delegate and data source?

Code:

- Displaying a pickerview
- Adding data to the pickerview
- Detecting when value in pickerview changes
- Getting the user's selection from the pickerview
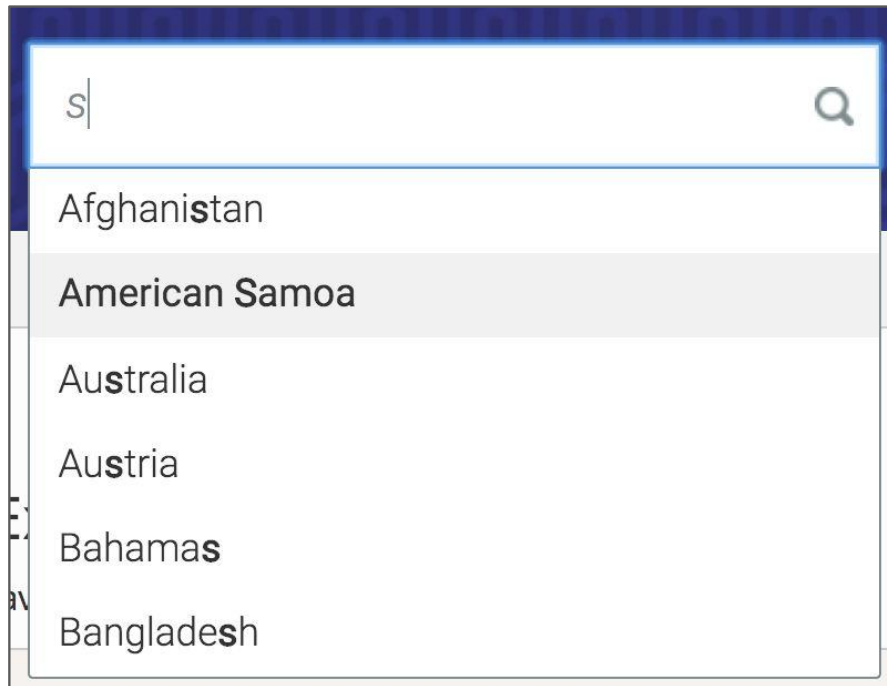
PickerViews

PickerViews:

- Used to display data **and** let the user **select** data

- Similar to a "drop down" menu in web
- Spinner in Android

Dropdown menus vs. Pickers

Dropdown menu

Picker Menu

Pickers vs TableViews

TableViews are another UI control that let user select data.

But a picker and a table view are different

Picker vs. TableView - example

Pickerview

Android:
Spinner

Tableview

Android:
ListView

Built-in vs. Custom Pickers

Built in PickerViews:

- Date / Time Picker

Custom Picker

- Uses your own data

Coding a Picker View

Example App

User can select an item from PickerView

Click button, item is displayed in results label
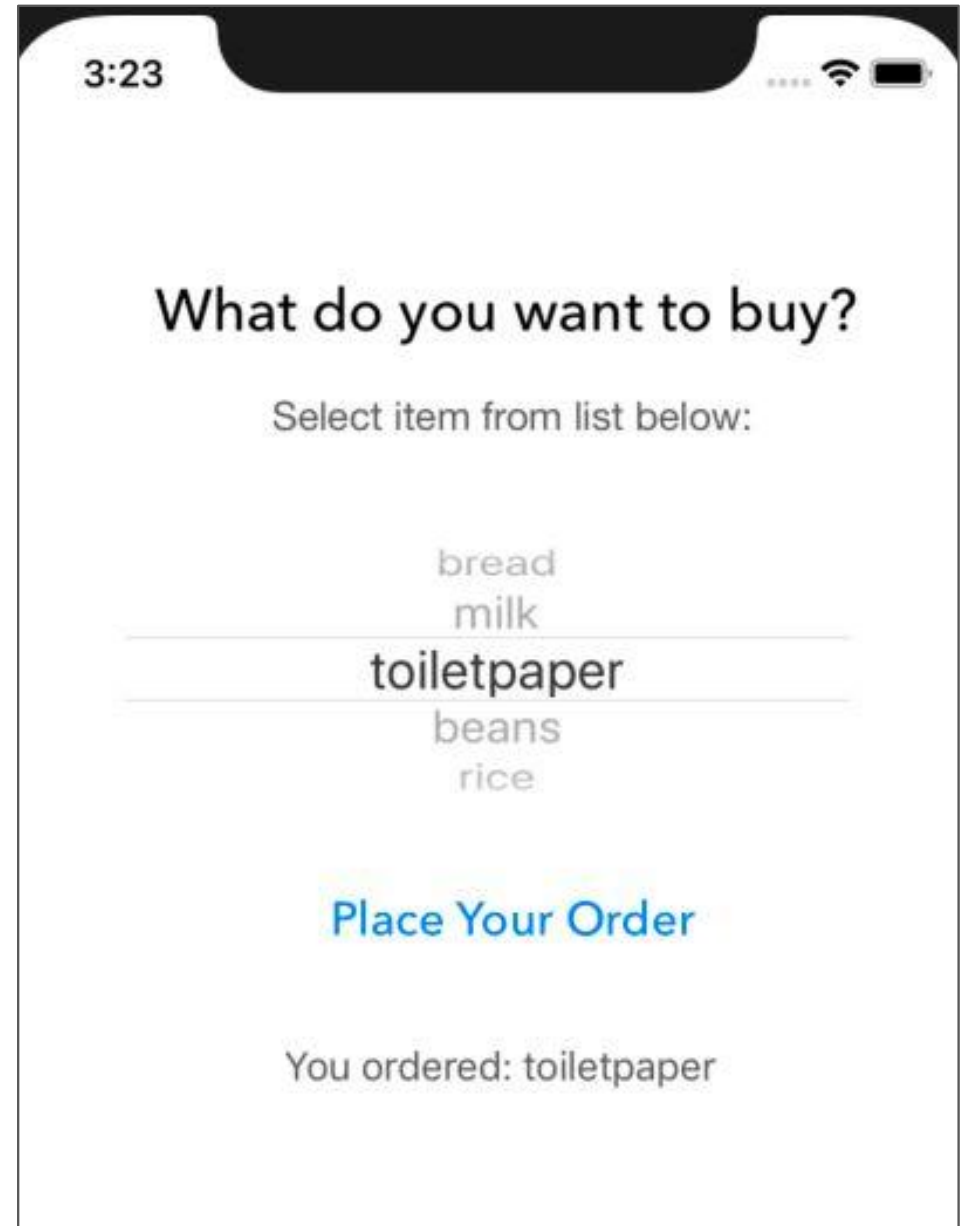
Recipe Alert: How to Add a PickerView

1. Choose a Picker type (Date vs. Custom)
2. In Main.storyboard, add the Picker to your layout
3. In ViewController.swift:
   a. Add outlet for the picker view
   b. Setup class to conform to PickerView **protocols**
   c. Set up the **mandatory PickerView API functions**
      i. `numberOfComponents`
      ii. `numberOfRowsInComponent`
      iii. `titleForRow row:`
   d. Define what data will go in the PickerView
   e. Connect data to the PickerView
4. Done!

1. Choose a Picker Type
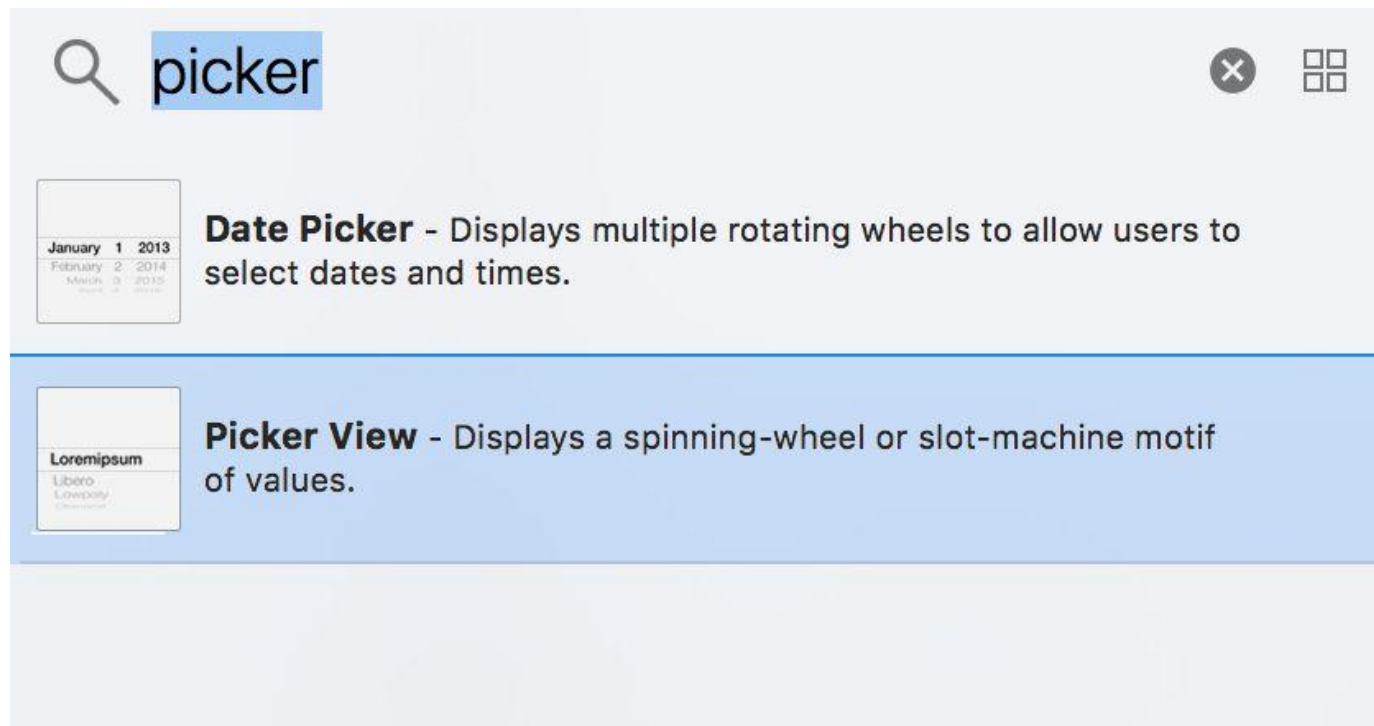
2. Add PickerView and make outlets
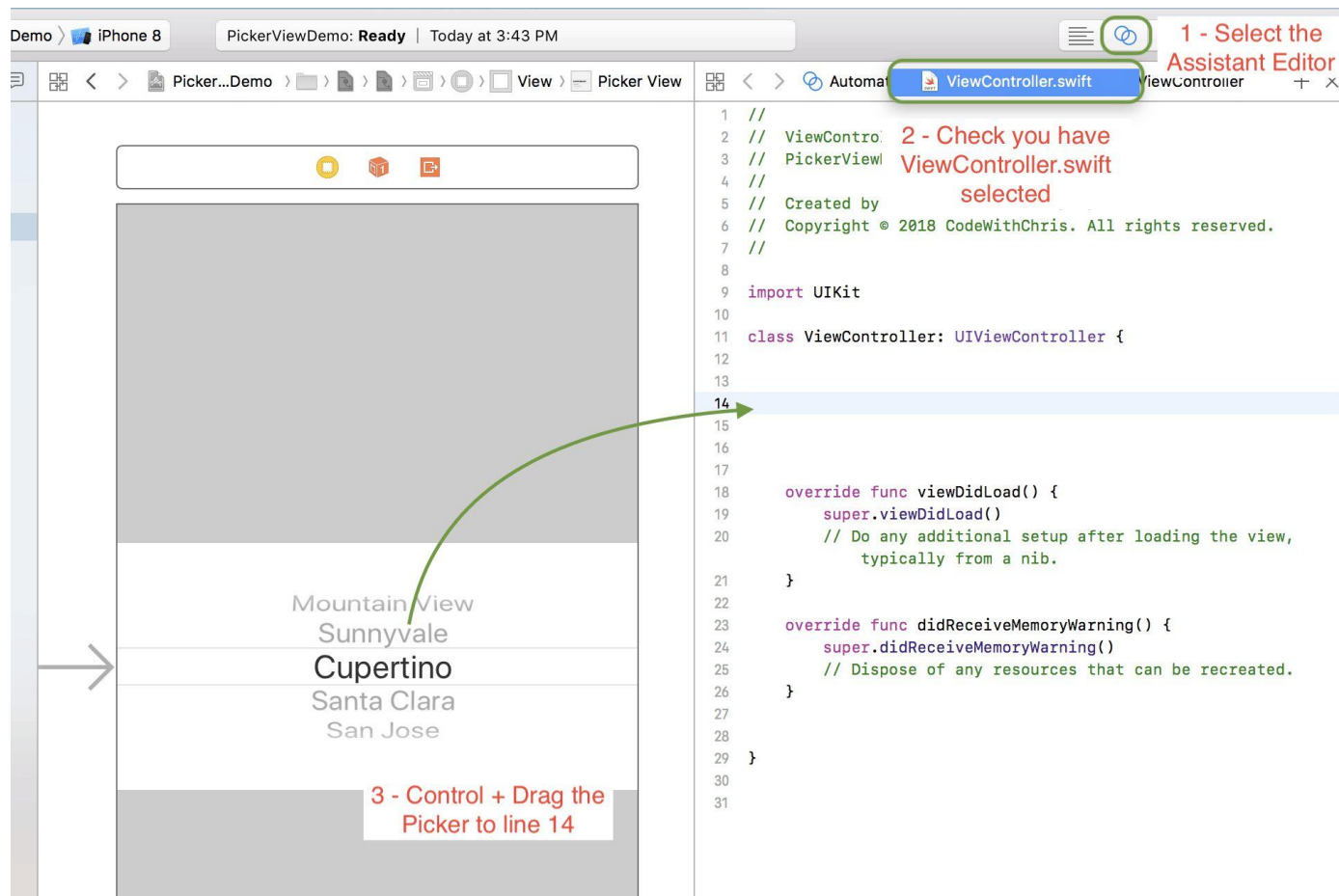
3. Check that class conforms to protocols

Required protocols

```swift
class ViewController: UIViewController, UIPickerViewDelegate, UIPickerViewDataSource
{

    @IBOutlet weak var picker: UIPickerView!

    override func viewDidLoad() {
        super.viewDidLoad()

        // Do any additional setup after loading the view, typically from a nib.
        self.picker.delegate = self
        self.picker.dataSource = self

    }
}
```

Pickerview outlet

required for protocols to work properly

3. PickerView Protocols

Any screen that includes a PickerView must conform to the UiPickerViewDataSource and UIPickerViewDelegate protocols.

These can be selected using XCode's autocomplete feature:
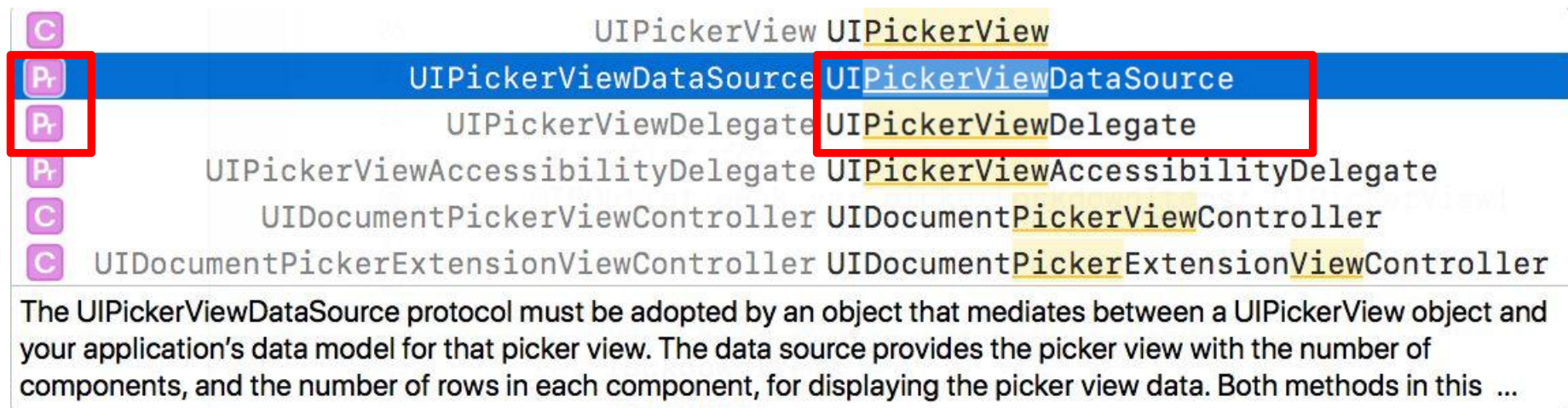
4. Add mandatory PickerView functions

```swift
// Number of columns of data you want to show in the pickerview
func numberOfComponents(in pickerView: UIPickerView) -> Int {
    return 1
}

// The total number of items in your PickerView
func pickerView(_ pickerView: UIPickerView, numberOfRowsInComponent component:
Int) -> Int {
    return 1;
}

// What data should be displayed in each row of the picker
func pickerView(_ pickerView: UIPickerView, titleForRow row: Int, forComponent
component: Int) -> String? {
    return "abcd"
}
```

5. Create a data source for your PickerView

India
United States
United Kingdom
Canada
Australia
UAE
Russia

```
let countries = [
    "United States",
    "United Kingdom",
    "Canada",
    "Australia",
    "UAE",
    "Russia"
]
```

1. If this is what you want in the PickerView then….

2. This is what your data source looks like

6. Add mandatory PickerView functions

```swift
let countries = ["United States","United Kingdom","Canada","Australia","UAE"]

// Number of columns of data you want to show in the pickerview
func numberOfComponents(in pickerView: UIPickerView) -> Int {
    return 1
}
```

Change this if you want to have a multicolumn pickerview

```swift
// The total number of items in your PickerView
func pickerView(_ pickerView: UIPickerView, numberOfRowsInComponent component: Int) -> Int {
    return countries.length;
}
```

Updated to be same number as items in your data source

```swift
// What data should be displayed in each row of the picker
func pickerView(_ pickerView: UIPickerView, titleForRow row: Int, forComponent component: Int) -> String? {
    return countries[row]
}
```

Updated code to show the current item in the data source