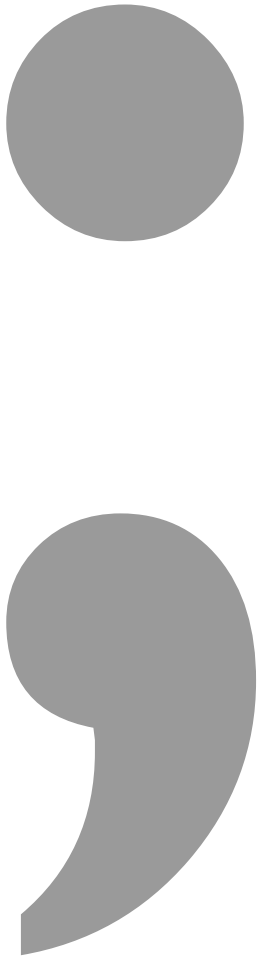
Code Basics

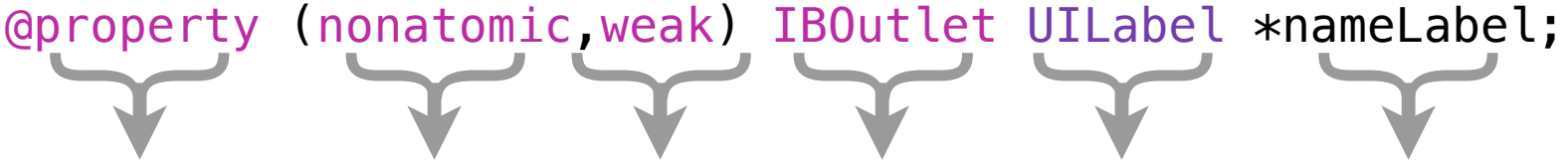
Every Line of Code Ends with…



(a semi-colon)

Declaring Properties

Syntax



Naming Conventions

Apple says to use Camel Case

```
int thisIsCamelCase = 1;
```

Other Cases

```
int ThisIsWordCase = 1;
int Thisissentencecase = 1;
int THISISUPPERCASE = 1;
int thisislowercase = 1;
int some_languages_suggest_underscores = 1;
```

Tip: use object type in name

```
@property (strong, nonatomic) NSString *nameString;
@property (strong, nonatomic) NSArray  *nameArray;
@property (strong, nonatomic) UILabel  *nameLabel;
```

Declaring Methods

Syntax

```
- (void)doSomethingCool;
```

Declaring Methods

Syntax

```
- (void)doSomethingCool;
- (float)squareWithNumber:(float)number;
```

Declaring Methods

Syntax

```
- (void)doSomethingCool;
- (float)squareWithNumber:(float)number;
- (IBAction)myButtonPressed:(id)sender;
```

Available to Interface Builder

Method Name

Untyped Object Parameter

Naming Conventions

Apple says to use Camel Case

Use verbs for method names and be clear

```
- (Activity *)fetchActivityForDate:(NSDate *)activityDate;
```

Two Words About AutoComplete…

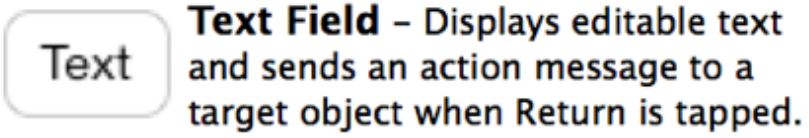
USE IT!!!

- Tab through to accept the text
- Pay attention to the parameter types is asking for and give it what it wants!
- If it's not trying to AutoComplete, something's probably wrong with your code, so go back and fix it first!
- Be careful to choose the right AutoComplete

Controls
UITextField

UITextField

What events might you want to know?



Events

- Editing Did Begin
- Text Did Change
- Editing Did End

…but also…

- Should Change Characters
- Should Return

UITextField Instance Methods

```
- (void)textFieldDidBeginEditing:(UITextField *)textField;
- (void)textFieldDidEndEditing:(UITextField *)textField;
- (BOOL)textFieldShouldReturn:(UITextField *)textField {
    [textField resignFirstResponder];
    return YES;
}
- (BOOL)textField:(UITextField *)textField
    shouldChangeCharactersInRange:(NSRange)range
    replacementString:(NSString *)string {
        NSCharacterSet *numbersCharSet = [[NSCharacterSet
            characterSetWithCharactersInString:@“1234567890”]
            invertedSet];
        NSString *filteredString = [[string
            componentsSeparatedByCharactersInSet:
            numbersCharSet] componentsJoinedByString:@""];
        return [string isEqualToString:filteredString];
}
```

Where’s Did Change??

```
- (IBAction)textFieldDidChange:(UITextField *)textField;
```

Delegates

Usually, an object houses it's own code

Sometimes, it's easier for an object to "delegate" it's code to another object (often a parent)

```
@interface MainViewController :
    UIViewController <UITextFieldDelegate>
_myTextField.delegate = self; // OR by wiring it up in IB
```

So, by declaring **and** setting a delegate, it says:

- Look for my code over here
- AutoComplete method names (yay)!
- Warn if any required methods are missing

HelloWorld
+Text

HelloWorld+TextField Instructions

1. Add a text field and a new button
2. The button changes the label to the value of the text field, but only accepts vowels
3. The Done button dismisses the keyboard

```
- (BOOL)textFieldShouldReturn:(UITextField *)textField {
    [textField resignFirstResponder];
    return YES;
}
- (BOOL)textField:(UITextField *)textField
     shouldChangeCharactersInRange:(NSRange)range
     replacementString:(NSString *)string {
        NSCharacterSet *numbersCharSet = [[NSCharacterSet
            characterSetWithCharactersInString:@“1234567890”]
            invertedSet];
        NSString *filteredString = [[string
            componentsSeparatedByCharactersInSet:
            numbersCharSet] componentsJoinedByString:@""];
         return [string isEqualToString:filteredString];
}
```

Git

Why Version Control?

- Create a history of your changes
- Rollback to a point that works
- Work on new features without damaging main project
- Have a backup, if using a remote server
- Collaborate with others by sharing same code
- For potential employers to evaluate

But we'll worry about that stuff later. For now…

- **It's where you turn in homework**

Common Git Terms

Repository - where files & history are stored

Local/Remote - location of a repo

Fork - creating your separate copy of a repo

Clone - creating a local copy of a repo

Commit - saving and documenting changes

Push - moves all changes to the remote repo

+++