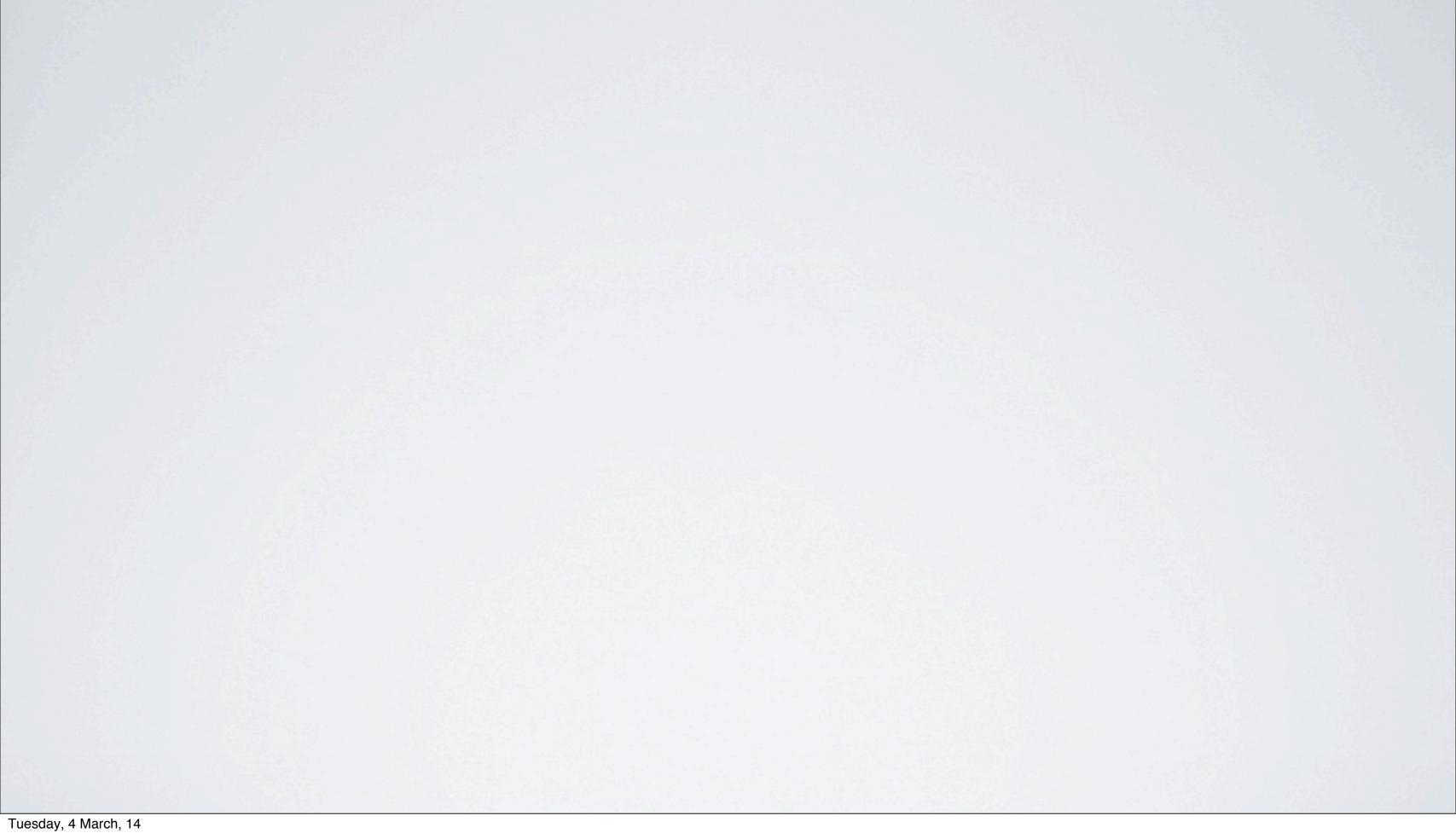
iPhone App Creation

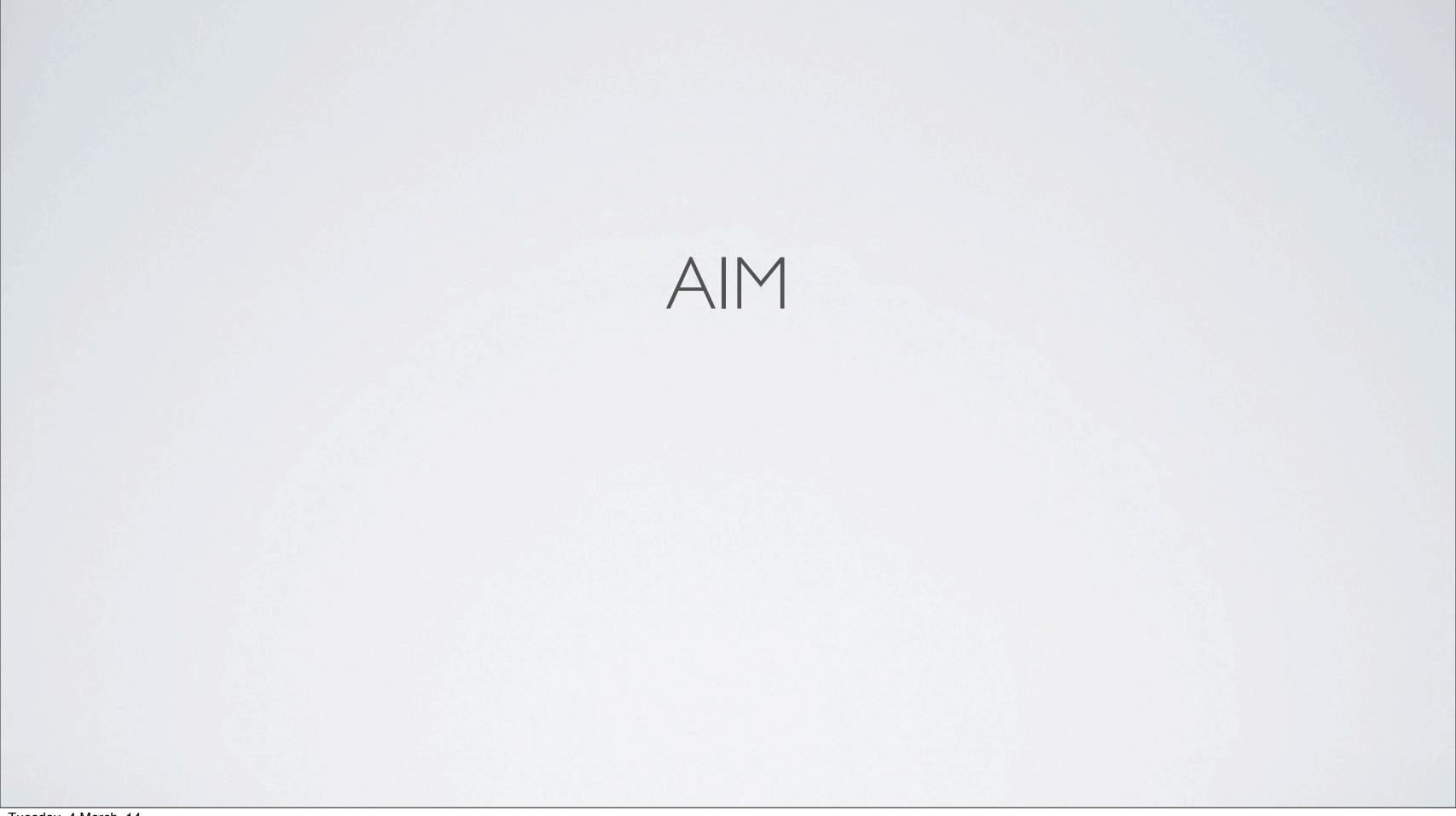
LESSON 01

INSTRUCTOR: JESSE SCOTT



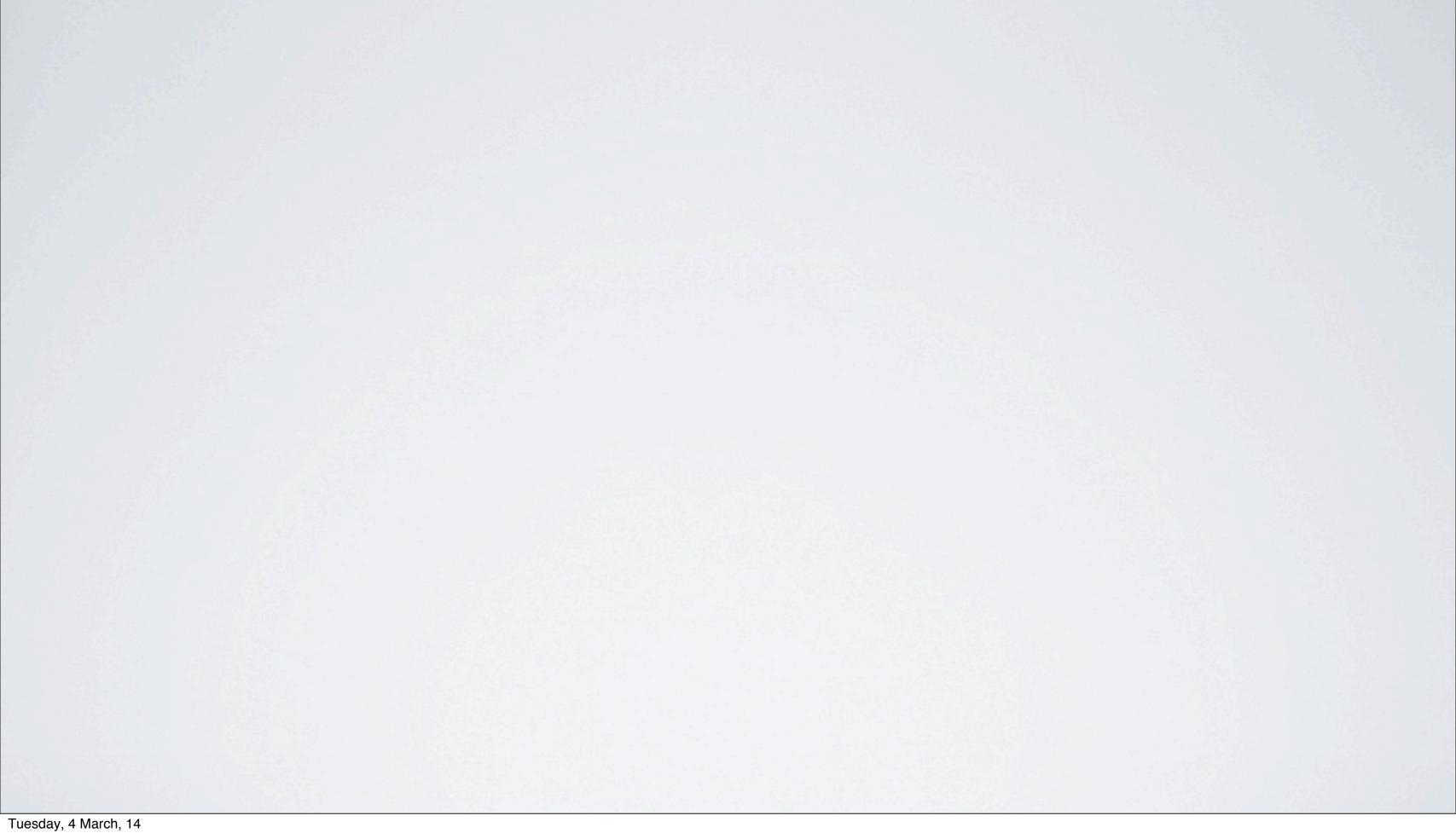






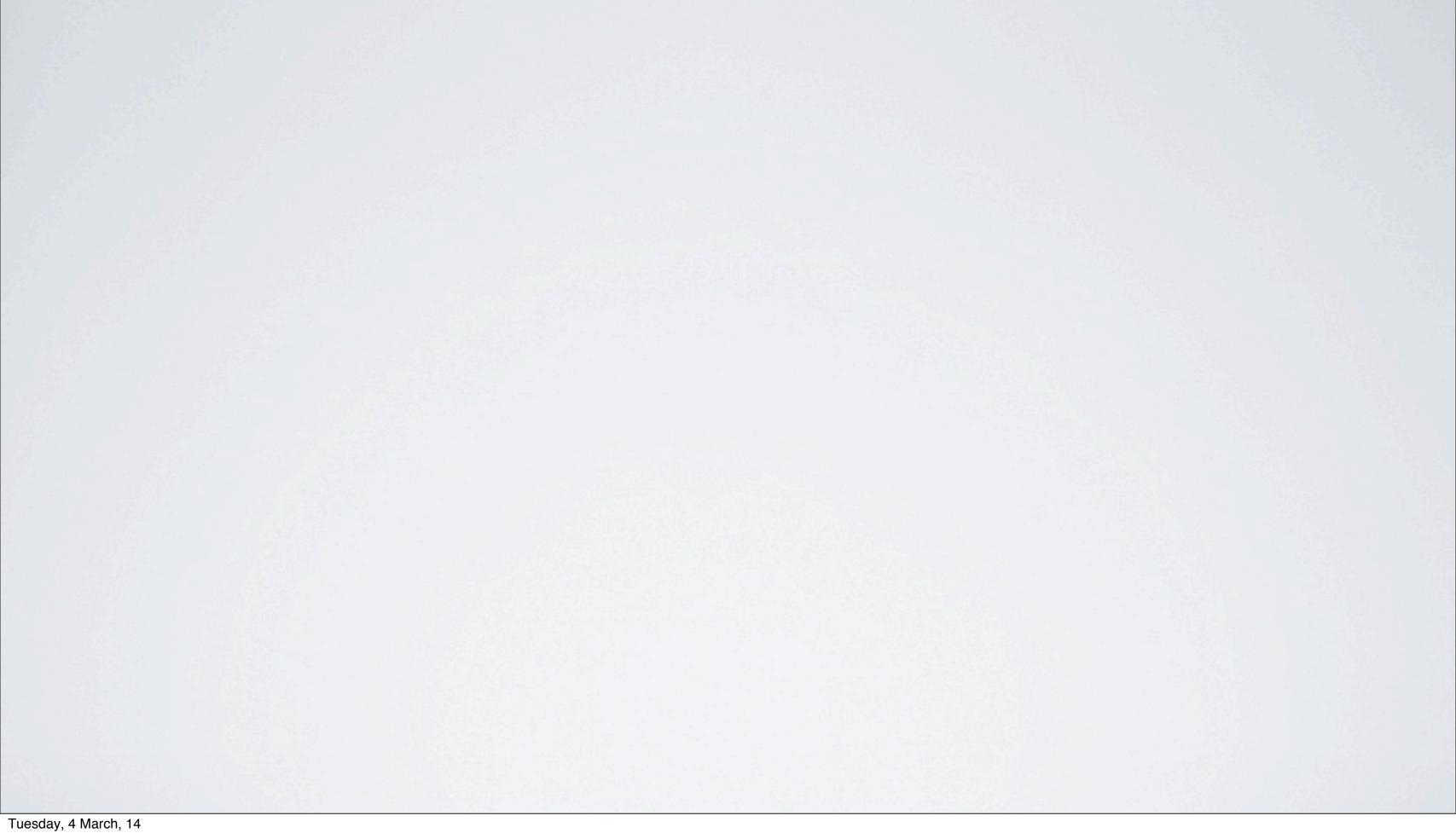
LEARN TO CREATE IPHONE APPS

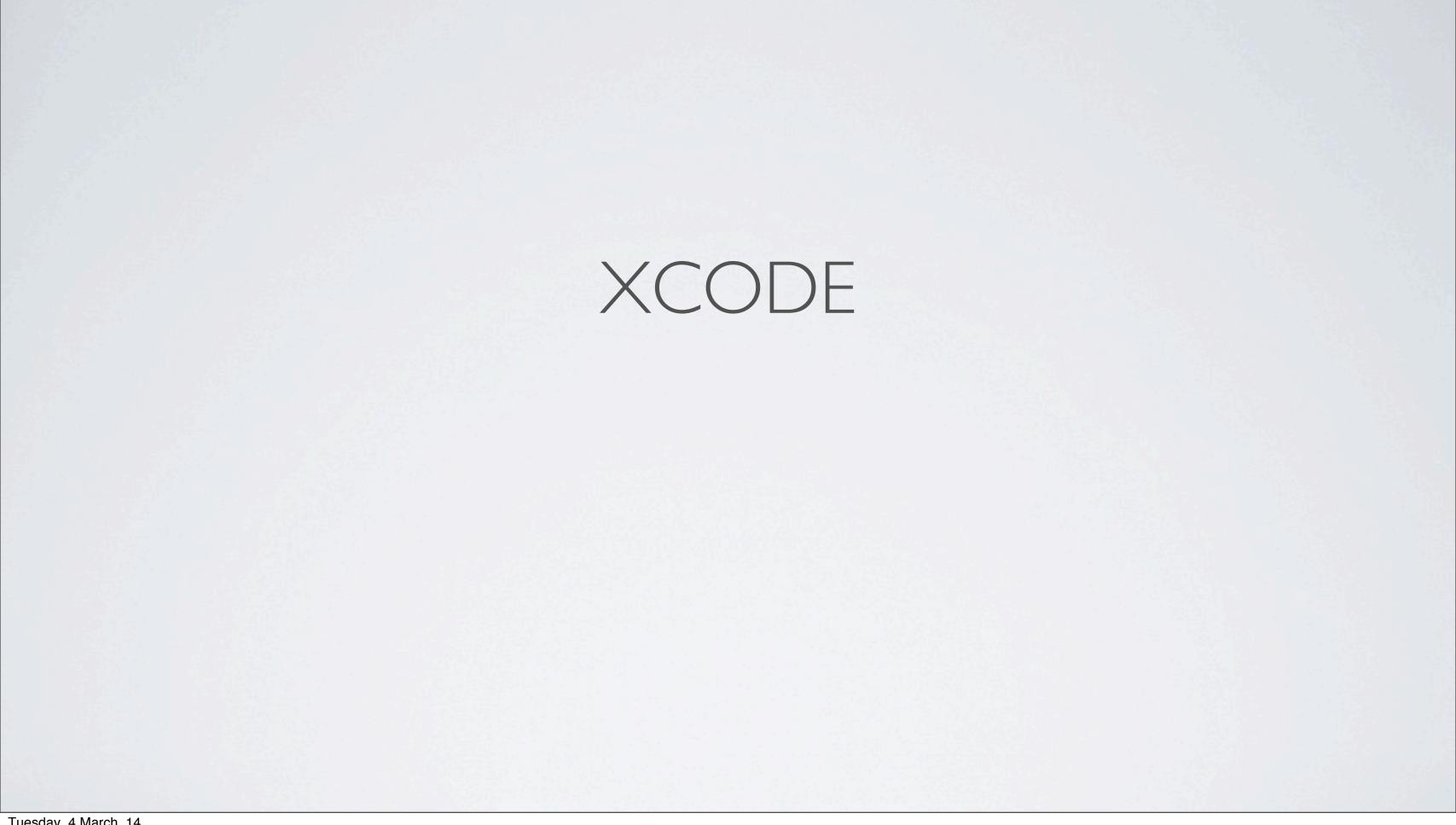
- * Using iOS Software Development Kit & Cocoa Touch Framework
- * XCode IDE
- * Objective-C Programming Language
- * Interface Builder and Recommended Resources
- * Program Design / Software Architecture
- * Human Interface Guidelines
- * App Flow / Interaction Design



HISTORY

- * Objective-C language built in 1980
- * NeXT Software licences it in 1988, Apple buys NeXT in 1996
- * Objective-C 2.0 released in 2007
- * iPhone released in 2007
- * Apple releases iOS Software Development Kit in 2008
- * iPad released in 2010





VERSIONS

* 5.0.2

iOS SDK * 7.0.3

WELCOME SCREEN

Start A New Project

Connect To A Repository

XCode User Guide

Apple Developer Portal

Recent

PROJECTTEMPLATE

iOS

- * Application
- * Framework
- * Other

OSX

- * Application
- * Framework
- * Plug-Ins
- * Other

PROJECT OPTIONS

Product Name

* What The XCode File & Folder Structure Will Be Named

Organization Name

* This Should Be You (or your company)

Company Identifier

* Reverse Domain Name Structure

PROJECT OPTIONS CONTD.

Devices

* iPhone / iPad / iPod Touch

Storyboards

* Allows You To Diagram Application Flow Visually

ARC

* Saves You From Tedious Memory Management

Unit Tests

* Helps In Profiling / Analyzing Your Code

NAVIGATOR AREA - GROUPS

Project Group

* Holds All Your Created Files (Resources, Icons, Classes, Property Lists, etc.)

Framework Group

* Holds ObjC Foundation, Cocoa Touch, etc.

Product Group

* Holds Actual Compiled iOS Executable File

NAVIGATOR AREA - VIEWS

Project View

* File Explorer

Symbol View

* Class/Method Structure

Search View

* Search for matching function, etc.

Issue View

* Browse Compile-time Errors and Warnings

Debug View

* Advanced Stack Calls...

Breakpoint View

* Lists Intentional Pause-Points In Your Program

Log View

* History Of Build Logs & Console Outputs

EDITOR AREA

Jump Bars

* Allows You To Jump Through File Tree

Standard Editor

* Allows You To Type/Paste/Drag Code

Assistant Editor

* Allows You To Edit Two Files Simultaneously

Version Editor

* Allows You To Browse Past Versions Of Your Code

UTILITY AREA

Inspector Area

* Displays File Properties

Library Area

- * File Template Library displays templates of classes and files
- * Code Snippet Library displays common code blocks
- * Object Library displays collection of user interface objects
- * Media Library displays graphics, icons, audio files in your project

DEBUG AREA

Variable Area

* Displays Values Of Variables By Scope

Console Area

* Displays NSLog Output, Compile Time, Errors

TOOLBAR AREA

Run

* Builds, Compiles, Launches Your App

Stop

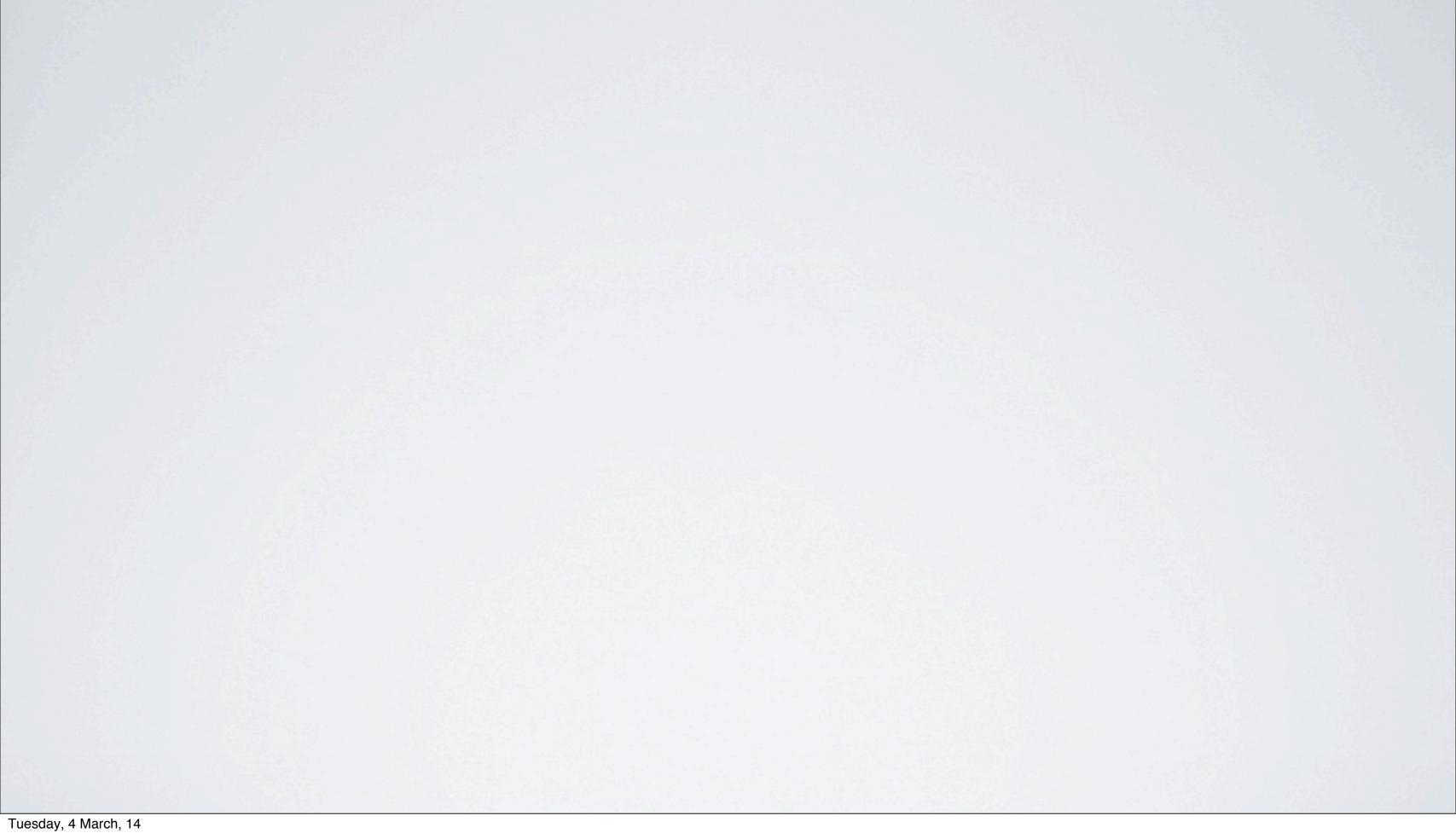
* Stops The Execution Of Your App

Scheme

* Determines The Active Build Target Of Your App

Show/Hide

* Editors, Debug, Utility, Navigator





INTRODUCTORY NOTES

OOP vs Procedural

* Skipping Much Procedural In Favour Of Efficiency

Learn By Typing

* Excessive Commenting Of Your Code

Gradual Learning

* Will Uncover Features / Techniques As We Go

#I HELLO WORLD

Program Structure

* Colour Coding - Comments / Reserved Names / Strings

Import

* Tells Your App Of The Existence Of Other Classes

main()

* Primary Entry Point In All C / C++ / ObjC Applications

@autorelease

* Allows Proper Memory Management

* Semi-Colon's Terminate Statements

#2 DISPLAYING STRINGS

@''''

* Denotes NSString

\n

* Newline Character Sequence

Comments

* Single-line & Multi-line

#3 DISPLAYING VARIABLES

```
Declaring

* type name;

Assigning

* name = value;

Types

* int (%i) float (%f) double (%e) char (%c) String (%@)
```

#4 FUNCTIONS

```
Declare
  * type name() {
      // code here
   - or -
   * type name(type parametre) {
       // code here
Call
  * name();
 - or -
   * name(parametre);
```

#4 FUNCTIONS

```
eg.

int myFunction() {

// code

return int;
}
```

