



ADVANCED iOS APPLICATION DEVELOPMENT

MPCS 51032 • SPRING 2020 • SESSION 1D

ATTRIBUTED STRINGS

NSATTRIBUTEDSTRING



Boring

- UILabel text can be set for the entire string

NSATTRIBUTEDSTRING

- Attributed string

- An association of characters and their attributes

- Available since iOS5

- `NSAttributedString` object

- Manages character strings and associated sets of attributes

Class

NSAttributedString

A string that has associated attributes (such as visual style, hyperlinks, or accessibility data) for portions of its text.

SDKs

iOS 3.2+

macOS 10.0+

tvOS 9.0+

watchOS 2.0+

Framework

Foundation

[On This Page](#)

[Overview](#) ⓘ

[Topics](#) ⓘ

[Relationships](#) ⓘ

[See Also](#) ⓘ

Overview

An NSAttributedString object manages character strings and associated sets of attributes (for example, font and kerning) that apply to individual characters or ranges of characters in the string. An association of characters and attributes is called an attributed string. The cluster's two p

NSATTRIBUTEDSTRING

- Applies to individual characters or ranges of characters in the string
 - `NSMakeRange(0, 15)`
- `NSMutableAttributedString` for modifiable attributed strings

Class

NSAttributedString

A string that has associated attributes (such as visual style, hyperlinks, or accessibility data) for portions of its text.

SDKs

iOS 3.2+

macOS 10.0+

tvOS 9.0+

watchOS 2.0+

Framework

Foundation

On This Page

[Overview](#) ⓘ

[Topics](#) ⓘ

[Relationships](#) ⓘ

[See Also](#) ⓘ

Overview

An NSAttributedString object manages character strings and associated sets of attributes (for example, font and kerning) that apply to individual characters or ranges of characters in the string. An association of characters and their attributes is called an attributed string. The cluster's two public

NSATTRIBUTEDSTRING



- That's more like it!

NSATTRIBUTEDSTRING

SUBTITLE

- An attributed string identifies attributes by name
 - Uses NSDictionary object to store a value under the given name
- You can assign any attribute name/value pair you wish to a range of characters

Using Attributed Strings with Labels

Create a UILabel, change the color and round the corners of the label. Font and color pro

```
let helloLabel = UILabel(frame: CGRect(x: 0, y: 0, width: 600, height: 100))
helloLabel.backgroundColor = UIColor.yellow
helloLabel.layer.masksToBounds = true
helloLabel.layer.cornerRadius = 10.0
helloLabel.textAlignment = NSTextAlignment.center
helloLabel.text = "Hello Label!"
```

Hello Label!

Let the label use an attributed string instead

```
var attributedString = NSMutableAttributedString(string: "Fancy Hello World!",
                                                attributes: [
                                                    NSAttributedStringEncoding.font:UIFont(name:
```

```
helloLabel.attributedText = attributedString
```

Fancy Hello World!

NSATTRIBUTEDSTRING

```
let helloLabel = UILabel(frame: CGRect(x: 0, y: 0, width: 600, height: 100))
helloLabel.backgroundColor = UIColor.yellow
helloLabel.layer.masksToBounds = true
helloLabel.layer.cornerRadius = 10.0
helloLabel.textAlignment = NSTextAlignment.center
helloLabel.text = "Hello Label!"
```



Hello Label!

NSATTRIBUTEDSTRING

Let the label use an attributed string instead

```
var attributedString = NSMutableAttributedString(string: "Fancy Hello World!",  
                                              attributes: [  
                                                NSAttributedStringEncodingKey.font:UIFont(name: "Courier", size: 50.0)!])  
helloLabel.attributedText = attributedString
```

Fancy Hello World!

NSATTRIBUTEDSTRING

SUBTITLE

- If you are using attributed strings with the Core Text framework, you can also use the attribute keys defined by that framework
 - iOS X, standard attribute keys are described in the “Constants” section of NSAttributedString UIKit Additions Reference
 - OS X, standard attribute keys are described in the “Constants” section of NSAttributedString AppKit Additions Reference.

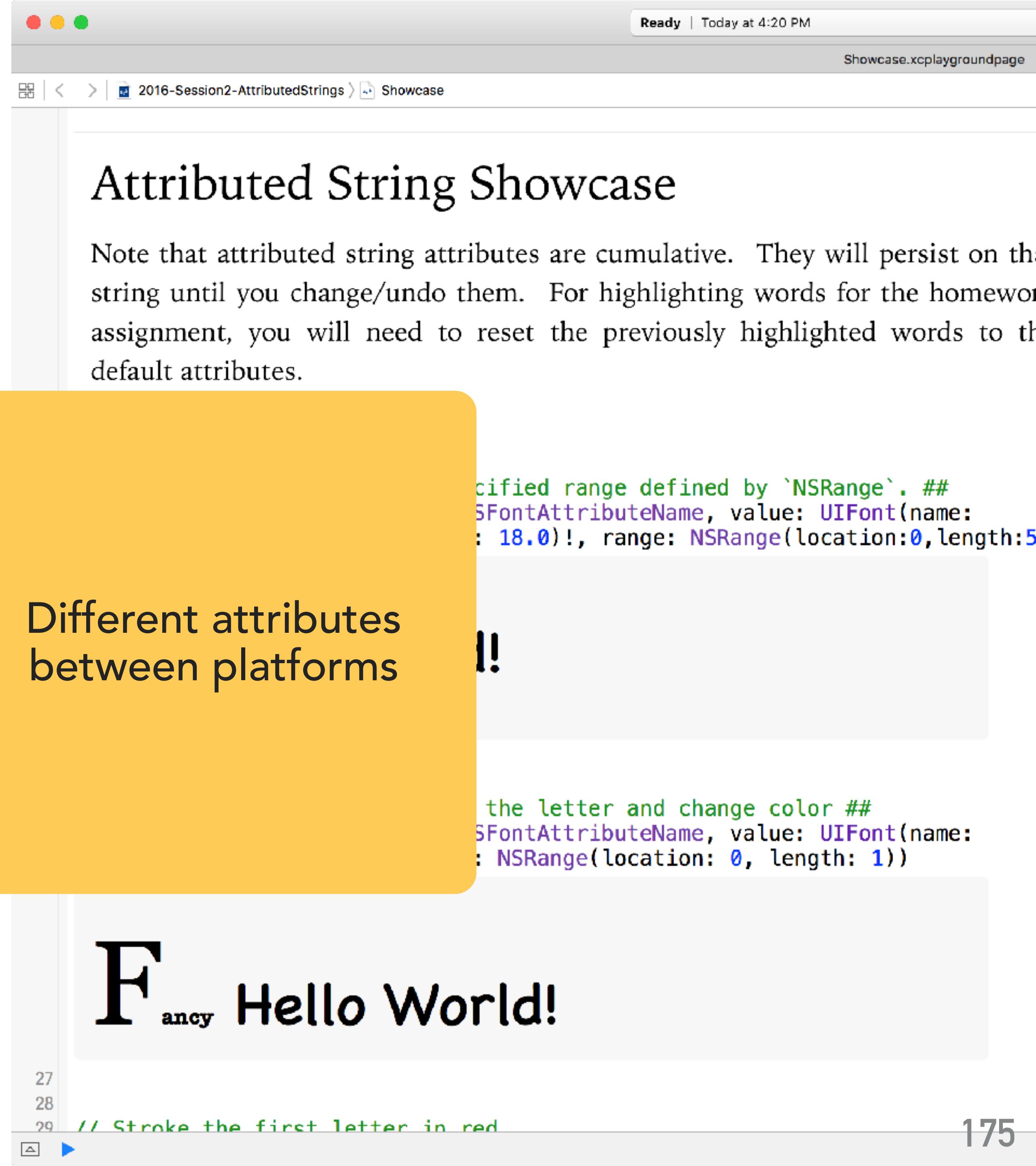
Attributed String Showcase

Note that attributed string attributes are cumulative. They will persist on the string until you change/undo them. For highlighting words for the homework assignment, you will need to reset the previously highlighted words to their default attributes.

Different attributes between platforms

```
specified range defined by `NSRange`. ##  
SFFontAttributeName, value: UIFont(name:  
: 18.0)!, range: NSRange(location:0, length:5)  
  
!!  
  
the letter and change color ##  
SFFontAttributeName, value: UIFont(name:  
: NSRange(location: 0, length: 1))
```

Fancy Hello World!



A screenshot of an Xcode playground window titled "Showcase.xcplaygroundpage". The status bar at the top right shows "Ready | Today at 4:20 PM". The main area displays the code for an attributed string. The code includes several NSMutableAttributedString methods to apply different font attributes to specific ranges of text. The resulting output is visible below the code, showing the word "Hello" with its first letter "H" in a large, bold, black font, and the word "World!" in a smaller, regular black font. The playground interface shows code completion suggestions and a run button.

```
27  
28  
29 // Stroke the first letter in red
```

NSATTRIBUTEDSTRING

Attributed String Programming Guide



Table of Contents

- Introduction
- Attributed Strings
- Creating Attributed Strings in Cocoa
- ▶ Accessing Attributes
- ▶ Changing an Attributed String
- Drawing Attributed Strings
- ▶ RTF Files and Attributed Strings
- ▶ Formatted Documents and Attributed Strings
- Word and Line Calculations in Attributed Strings
- Standard Attributes
- Revision History
- Index

[Next](#)

Introduction to Attributed String Programming Guide

Attributed String Programming Guide describes the attributed string objects, instantiated from the `NSAttributedString` class or the `CFAttributedString` Core Foundation opaque type, which manage sets of text attributes, such as font and kerning, that are associated with character strings or individual characters.

Who Should Read This Document

You should read this document if you need to work directly with attributed string objects.

Organization of This Document

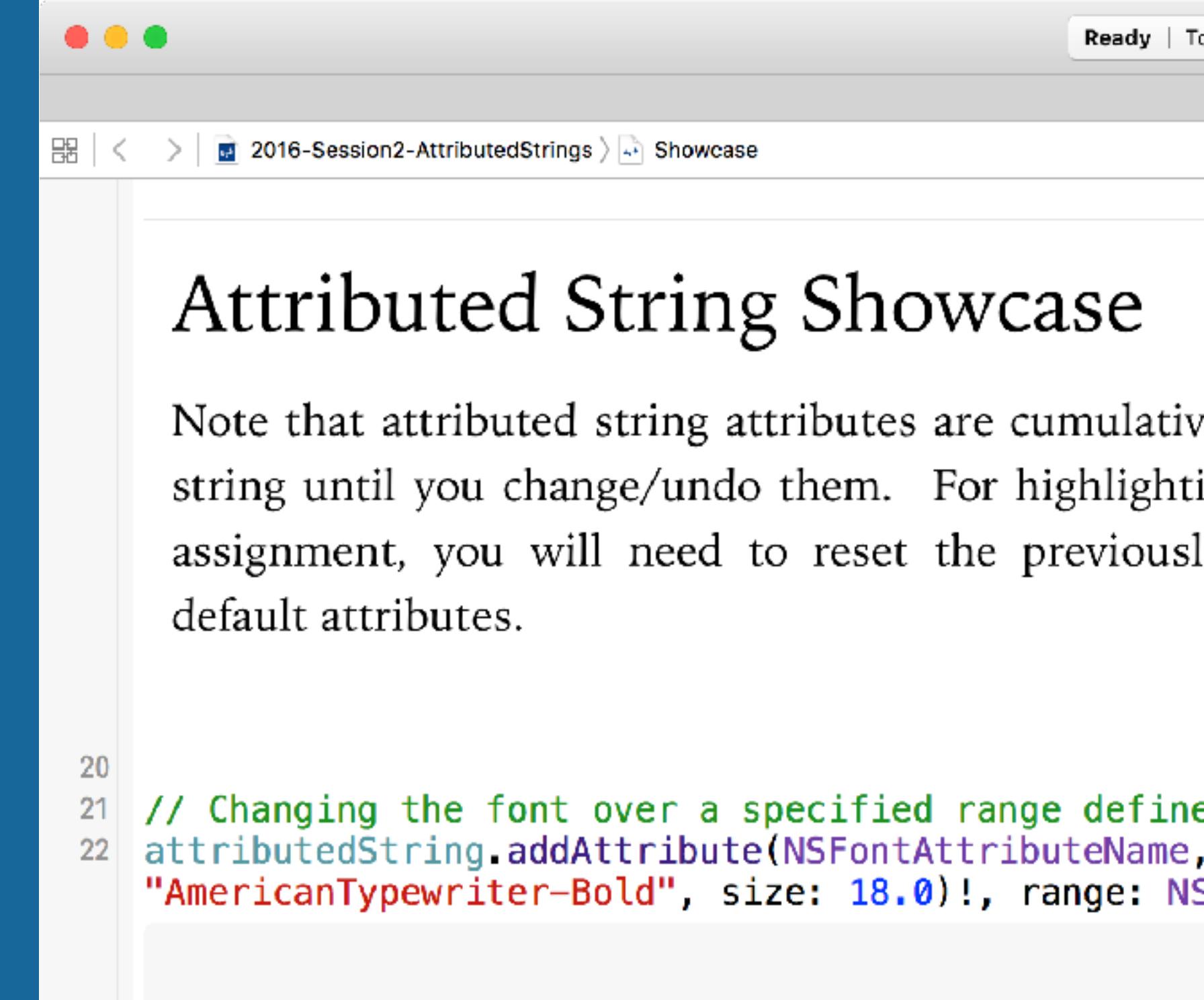
OBJECTIVE-C REFERENCE

- `NSAttributedString`
- `NSMutableAttributedString`

NSATTRIBUTEDSTRING

SAMPLE ATTRIBUTES

- NSString *const NSFontAttributeName;
- NSString *const NSParagraphStyleAttributeName;
- NSString *const NSForegroundColorAttributeName;
- NSString *const NSBackgroundColorAttributeName;
- NSString *const NSLigatureAttributeName;
- NSString *const NSBaselineOffsetAttributeName;
- NSString *const NSStrikethroughStyleAttributeName;
- NSString *const NSSrokeColorAttributeName;
- NSString *const NSSrokeWidthAttributeName;
- NSString *const NSShadowAttributeName;



The screenshot shows a Xcode interface with a file named "2016-Session2-AttributedStrings>Showcase". The code in the editor is as follows:

```

20
21 // Changing the font over a specified range defined
22 attributedString.addAttribute(NSFontAttributeName,
  "AmericanTypewriter-Bold", size: 18.0)!, range: NSMakeRange(10, 10)
23
24 // Stroke the first letter in red
25 attributedString.addAttribute(NSStrokeColorAttributeName, color: NSColor.red,
  range: NSMakeRange(0, 1))
26
27 // Change the background color of the entire string
28 attributedString.addAttribute(NSBackgroundColorAttributeName, color: NSColor.cyan,
  range: NSMakeRange(0, attributedString.length))
29
30 // Change the foreground color of the entire string
31 attributedString.addAttribute(NSForegroundColorAttributeName, color: NSColor.purple,
  range: NSMakeRange(0, attributedString.length))
32
33 // Change the baseline offset of the entire string
34 attributedString.addAttribute(NSBaselineOffsetAttributeName, value: 10.0,
  range: NSMakeRange(0, attributedString.length))
35
36 // Add a ligature to the entire string
37 attributedString.addAttribute(NSLigatureAttributeName, value: true,
  range: NSMakeRange(0, attributedString.length))
38
39 // Add a strikethrough to the entire string
40 attributedString.addAttribute(NSStrikethroughStyleAttributeName, value: 1,
  range: NSMakeRange(0, attributedString.length))
41
42 // Add a stroke to the entire string
43 attributedString.addAttribute(NSStrokeColorAttributeName, color: NSColor.red,
  range: NSMakeRange(0, attributedString.length))
44
45 // Add a shadow to the entire string
46 attributedString.addAttribute(NSShadowAttributeName, color: NSColor.black,
  range: NSMakeRange(0, attributedString.length))

```

**Warning: In Swift 4,
attributed objects have been
replaced with a struct
NSAttributedStringKey**

ATTRIBUTED STRINGS IN SWIFT

ATTRIBUTED STRINGS IN SWIFT

Create attributes array

```
let attributes = [NSAttributedStringKey.backgroundColor: UIColor.red,  
                  NSAttributedStringKey.foregroundColor: UIColor.white,  
                  NSAttributedStringKey.font: UIFont(name: "Georgia", size: 50.0)!  
]  
  
var fancyAttributedString = NSMutableAttributedString(string: "Super Fancy", attributes: attributes)  
helloLabel.attributedText = fancyAttributedString
```

Create attributed string

Super Fancy

Set label property
(not .text)

NSATTRIBUTEDSTRINGS

```
let attributes = [NSAttributedStringKey.backgroundColor: UIColor.red,  
                  NSAttributedStringKey.foregroundColor: UIColor.white,  
                  NSAttributedStringKey.font: UIFont(name: "Georgia", size: 50.0)!  
]  
var fancyAttributedString = NSMutableAttributedString(string: "Super Fancy", attributes: attributes)  
helloLabel.attributedText = fancyAttributedString
```



Super Fancy

- Set `attributedText` property on UILabel to use an attributed string instead of `text`

ATTRIBUTED STRINGS IN SWIFT

The screenshot shows a Mac OS X desktop with an Xcode playground window titled "Showcase.xcplaygroundpage". The window title bar includes "Ready | Today at 4:33 PM" and standard OS X window controls. The main area displays the following content:

Attributed String Showcase

Previous

```
7
8 import UIKit
9
10 // Create an attributed string
11 var attributedString = NSMutableAttributedString()
12
13 // Create an attributed string with fancy text
14 attributedString = NSMutableAttributedString(string: "Fancy Hello World!",  
UIFont(name: "ChalkboardSE-Regular", size: 50.0)!)]
```

A yellow callout bubble in the bottom right corner contains the word "Playground". Below the playground window, the text "Fancy Hello World!" is displayed in a large, bold, black font.

ATTRIBUTED STRINGS

IN

OBJECTIVE-C



NSATTRIBUTEDSTRINGS

```
{...}

- (void)writePrettyLabel
{
    NSMutableAttributedString *prettyString = [[NSMutableAttributedString alloc]
                                              initWithString:@"The Fox Jumped Over The Box!"];

    // Set font, notice the range is for the whole string
    UIFont *font = [UIFont fontWithName:@"Helvetica-Bold" size:18];
    [prettyString addAttribute:NSFontAttributeName value:font range:NSMakeRange(0, 10)];

    // Set background color, again for entire range
    [prettyString addAttribute:NSMutableBackgroundColorAttributeName
                           value:[UIColor colorWithRed:0.103 green:0.305 blue:0.492 alpha:1.0
                           range:NSMakeRange(5, 15)]];

    // Assign the attributed String to the @attributedText property of the UILabel
    self.prettyLabel.attributedText = prettyString;
}

@end
```

CREATE ATTRIBUTED STRING

SET ATTRIBUTES

SET LABEL PROPERTY

NSATTRIBUTEDSTRINGS

```
infoString=@"This is an example of Attributed String";  
  
NSMutableAttributedString *attString=[[NSMutableAttributedString alloc] initWithString:infoString];  
NSInteger _stringLength=[infoString length];  
  
UIColor *_black=[UIColor blackColor];  
UIFont *font=[UIFont fontWithName:@"Helvetica-Bold" size:30.0f];  
[attString addAttribute:NSFontAttributeName value:font range:NSMakeRange(0, _stringLength)];  
[attString addAttribute:NSForegroundColorAttributeName value:_black range:NSMakeRange(0, _stringLength)];
```

This is an example of Attributed String.

NSATTRIBUTEDSTRINGS

```
UIColor *_red=[UIColor redColor];
UIFont *font=[UIFont fontWithName:@"Helvetica-Bold" size:72.0f];
[attString addAttribute:NSFontAttributeName value:font range:NSMakeRange(0, _stringLength)];
[attString addAttribute:NSSrokeColorAttributeName value:_red range:NSMakeRange(0, _stringLength)];
[attString addAttribute:NSSrokeWidthAttributeName value:[NSNumber numberWithFloat:3.0] range:NSMakeRange(0,
_stringLength)];
```

This is an
example of
Attributed String.

NSATTRIBUTEDSTRINGS

```
UIColor *_red=[UIColor redColor];
UIFont *font=[UIFont fontWithName:@"Helvetica-Bold" size:72.0f];
[attString addAttribute:NSFontAttributeName value:font range:NSMakeRange(0, _stringLength)];
[attString addAttribute:NSSrokeColorAttributeName value:_red range:NSMakeRange(0, _stringLength)];
[attString addAttribute:NSSrokeWidthAttributeName value:[NSNumber numberWithFloat:-3.0] range:NSMakeRange(0,
_stringLength)];
```

**This is an
example of
Attributed String.**

NSATTRIBUTEDSTRINGS

```
UIColor *_red=[UIColor redColor];
UIColor *_green=[UIColor greenColor];
UIFont *font=[UIFont fontWithName:@"Helvetica-Bold" size:72.0f];
[attString addAttribute:NSFontAttributeName value:font range:NSMakeRange(0, _stringLength)];
[attString addAttribute:NSMutableForegroundColorAttributeName value:_green range:NSMakeRange(0, _stringLength)];
[attString addAttribute:NSMutableStrokeColorAttributeName value:_red range:NSMakeRange(0, _stringLength)];
[attString addAttribute:NSMutableStrokeWidthAttributeName value:[NSNumber numberWithFloat:-3.0] range:NSMakeRange(0,
_stringLength)];
```

This is an
example of
Attributed String.

NSATTRIBUTEDSTRINGS

```
UIColor *_green=[UIColor greenColor];
UIFont *font=[UIFont fontWithName:@"Helvetica-Bold" size:72.0f];

NSShadow *shadowDic=[[NSShadow alloc] init];
[shadowDic setShadowBlurRadius:5.0];
[shadowDic setShadowColor:[UIColor grayColor]];
[shadowDic setShadowOffset:CGSizeMake(0, 3)];

[attString addAttribute:NSFontAttributeName value:font range:NSMakeRange(0, _stringLength)];
[attString addAttribute:NSSForegroundColorAttributeName value:_green range:NSMakeRange(0, _stringLength)];
[attString addAttribute:NSShadowAttributeName value:shadowDic range:NSMakeRange(0, _stringLength)];
```

This is an
example of
Attributed String.

NSATTRIBUTEDSTRINGS

```
UIColor *_red=[UIColor redColor];
UIFont *font=[UIFont fontWithName:@"Helvetica-Bold" size:72.0f];
[attString addAttribute:NSFontAttributeName value:font range:NSMakeRange(0, _stringLength)];
[attString addAttribute:NSMutableForegroundColorAttributeName value:_red range:NSMakeRange(0, _stringLength)];

[attString addAttribute:NSMutableKernAttributeName value:[NSNumber numberWithInt:5] range:NSMakeRange(0, _stringLength)];
```

This is an example of Attributed String.

NSATTRIBUTEDSTRINGS

```
UIColor *_red=[UIColor redColor];
UIFont *font=[UIFont fontWithName:@"Helvetica-Bold" size:30.0f];
[attString addAttribute:NSFontAttributeName value:font range:NSMakeRange(0, _stringLength)];
[attString addAttribute:NSForegroundColorAttributeName value:_red range:NSMakeRange(0, _stringLength)];
[attString addAttribute:NSMutableAttributedString underlineStyle value:[NSNumber numberWithInt:2] range:NSMakeRange(0, _stringLength)];
```

This is an example of Attributed String.

```
UIColor *_blue=[UIColor blueColor];
UIColor *_blueL=[UIColor colorWithRed:0 green:0 blue:0.5 alpha:0.7];
UIFont *font=[UIFont fontWithName:@"Helvetica-Bold" size:30.0f];

[attString addAttribute:NSFontAttributeName value:font range:NSMakeRange(0, _stringLength)];
[attString addAttribute:NSForegroundColorAttributeName value:_blue range:NSMakeRange(0, _stringLength)];
[attString addAttribute:NSMutableAttributedString backgroundColor value:_blueL range:NSMakeRange(0, 20)];
```

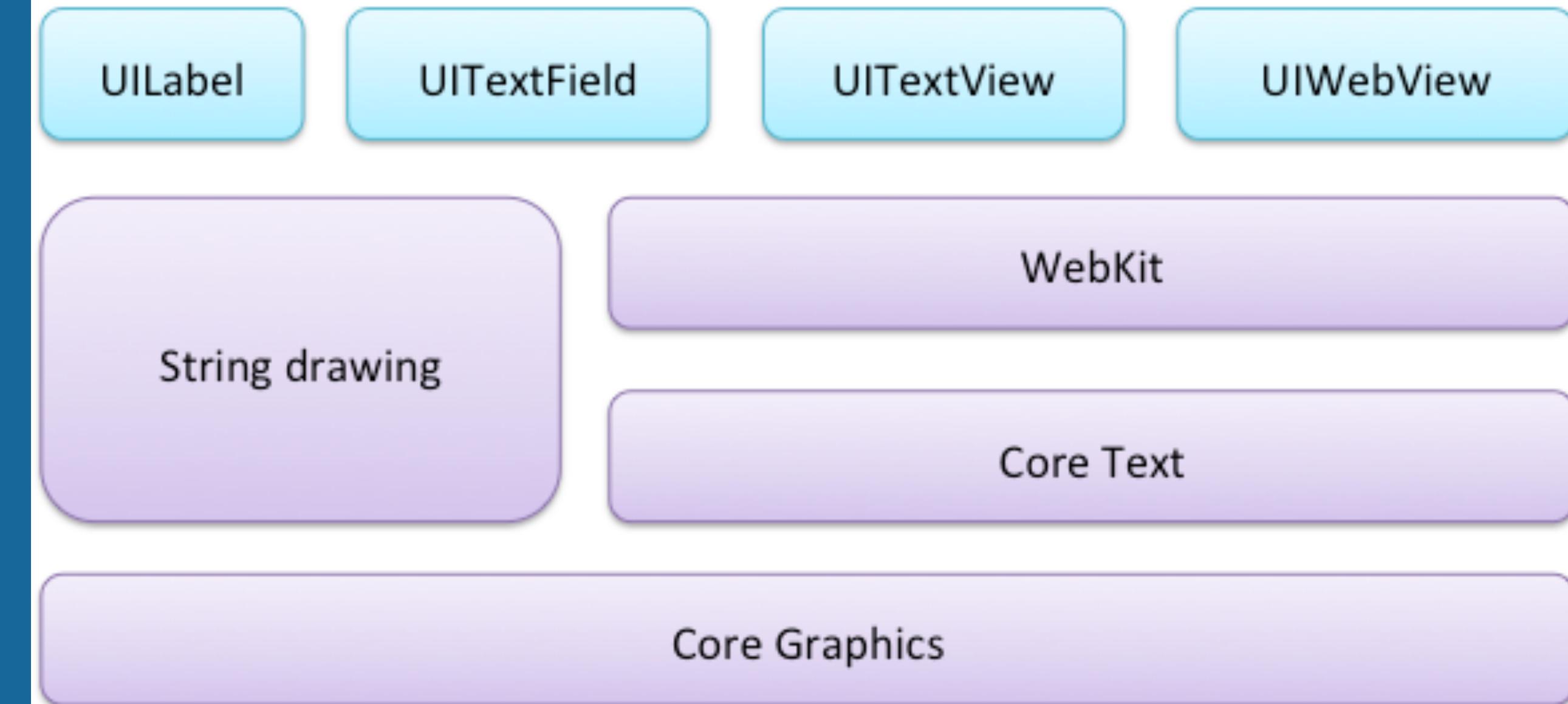
This is an example of Attributed String.

A BIT OF TEXT KIT

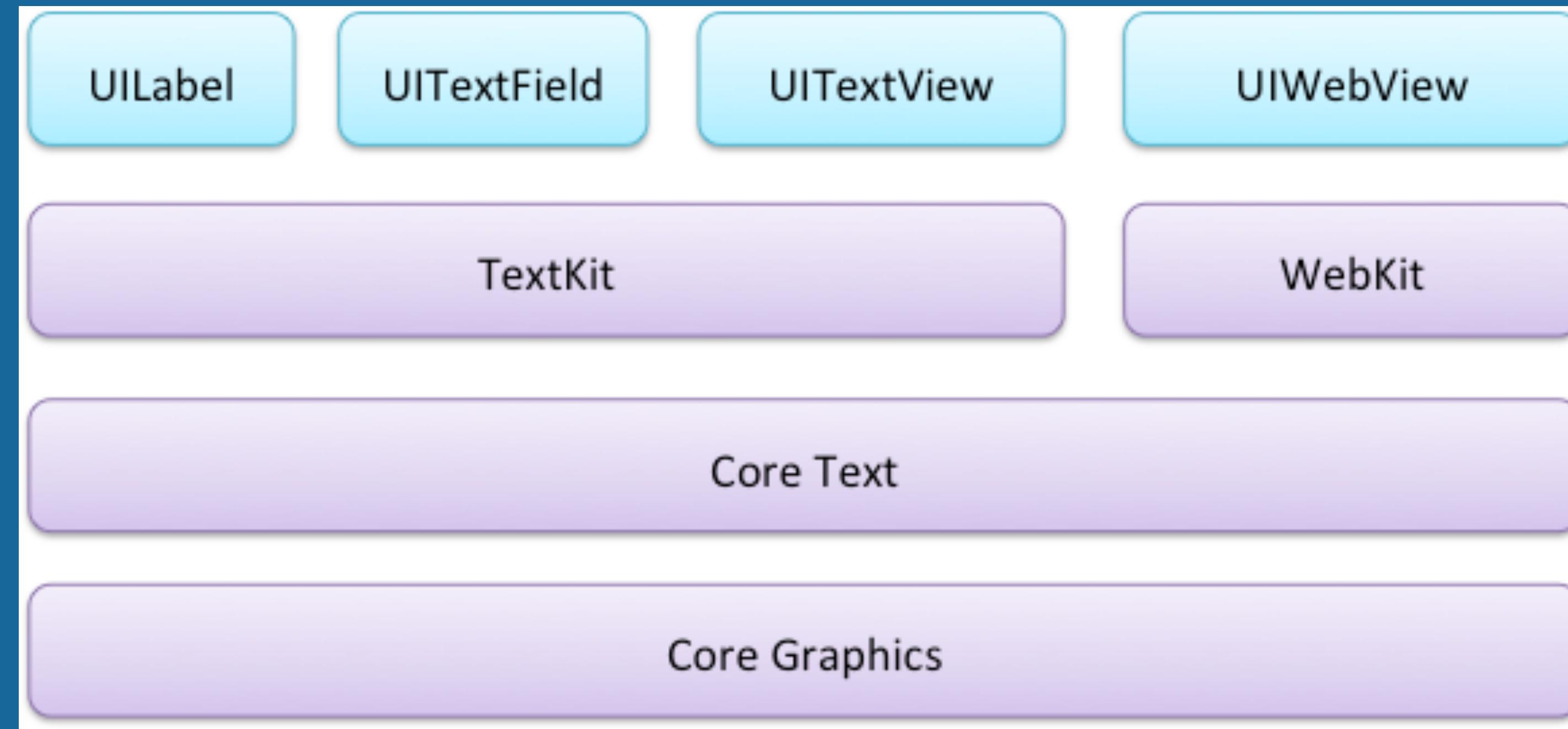
TEXT KIT

SUBTITLE

- Text in iOS
 - iOS5- UIWebView
 - iOS6 NSAttributedString support
 - iOS7 TextKit
- Support the text heavy redesign of iOS7



TEXT KIT

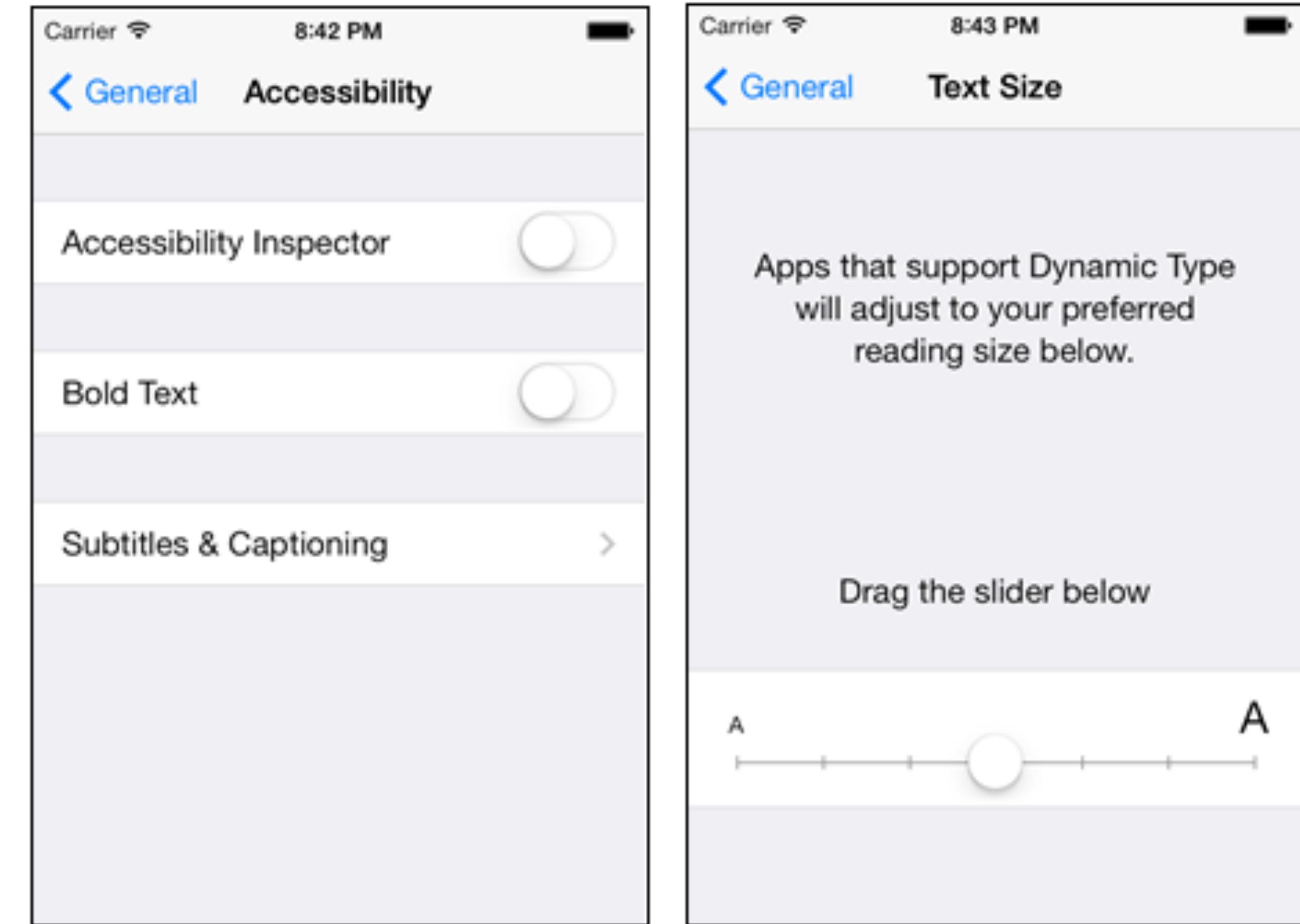


- Built on top of Core Text framework
 - Notoriously difficult API to work with

TEXT KIT

SUBTITLE

- Text Kit signature features
 - Dynamic type
 - Letterpress effect
 - Exclusion Path
 - Dynamic text formatting and storage

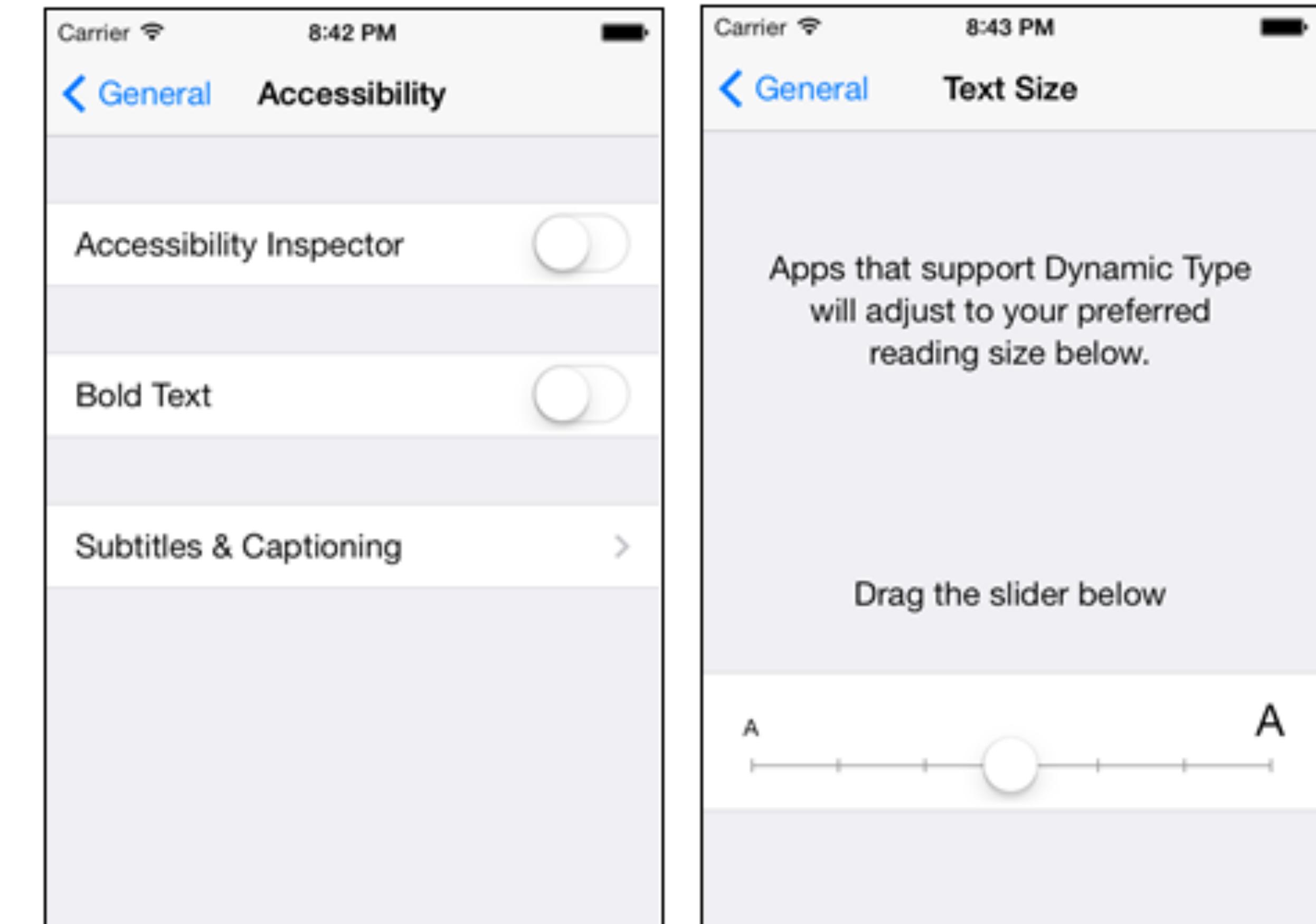


TEXT KIT

SUBTITLE

- DynamicType
 - When using default text, user can resize globally

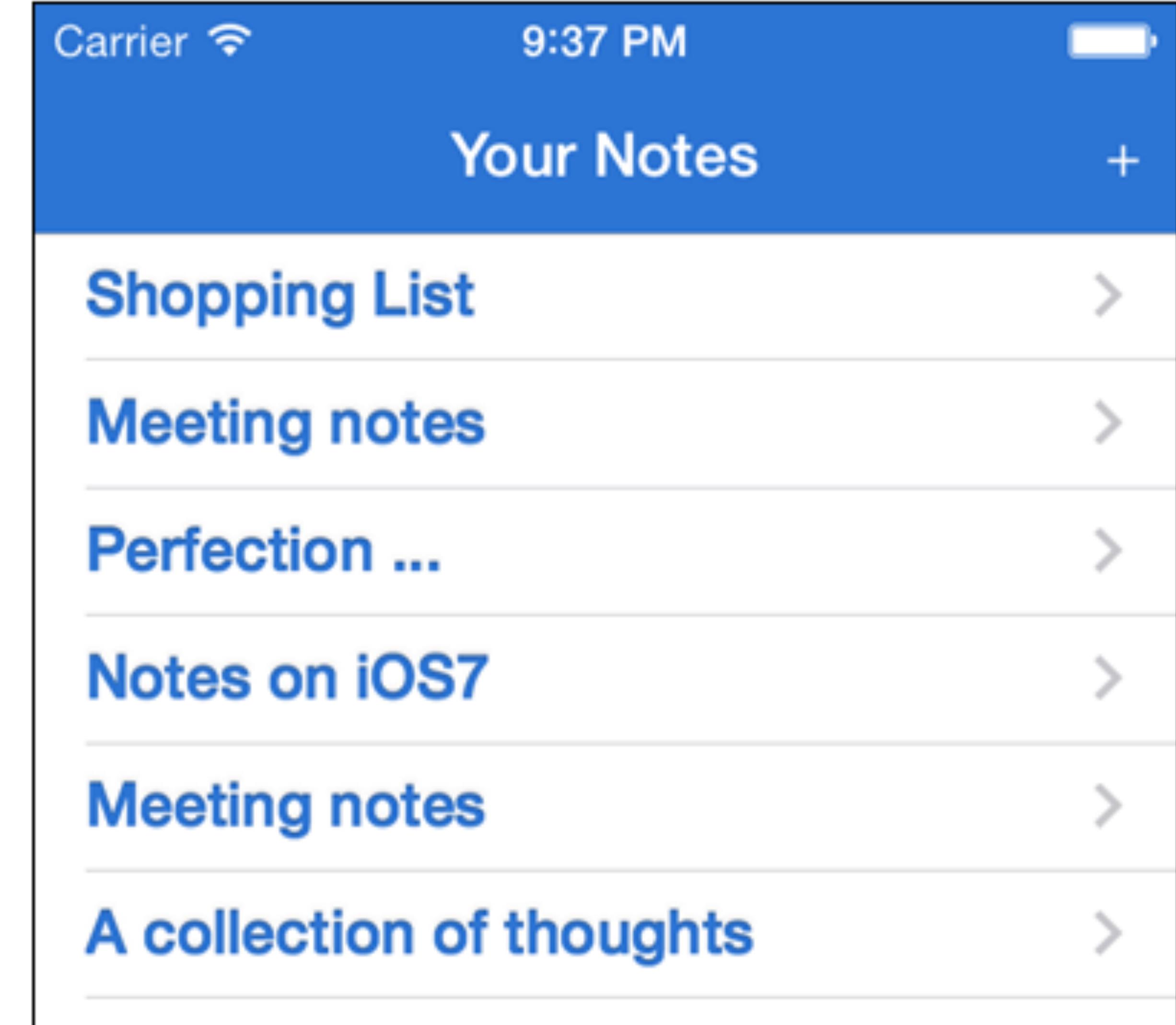
Users can ruin your interface



TEXT KIT

SUBTITLE

- Letterpress effect
 - Letterpress effects add subtle shading and highlights to text that give it a sense of depth — much like the text has been slightly pressed into the screen



TEXT KIT

Reminders

NSTextEffectLetterpressStyle

TEXT KIT

SUBTITLE

- Exclusion Path
 - Flowing text around images or other objects is a standard feature of most word processors
 - Text Kit allows you to render text around complex paths and shapes through exclusion paths



- TextKit tutorial
 - <http://www.raywenderlich.com/77092/text-kit-tutorial-swift>

Text Kit Tutorial in Swift



Gabriel Hauber on September 30, 2014

Update 12/12/14: Updated for Xcode 6.1.1.

Note from Ray: This is a Swift update to a popular Objective-C tutorial on our site, released as part of the [iOS 8 Feast](#). Update by Gabriel Hauber, [Original post](#) by Tutorial Team member [Colin Eberhardt](#). Enjoy!

The way that iOS renders text continues to grow more powerful over the years as Apple adds more features and capabilities. The release of iOS 7



ADVANCED iOS APPLICATION DEVELOPMENT

MPCS 51032 • SPRING 2020 • SESSION 1D