

#WWDC18

# AVSpeechSynthesis

## Making iOS talk

Session 236

Chris Fleizach, iOS Accessibility

# Agenda

What and why

AVSpeechSynthesis basics

Choosing the right voice

Rate, pitch, and volume

Attributed strings

# AVSpeechSynthesis

API for computer synthesized speech

Uses

- Announcements
- Non-sighted interfaces
- Education apps
- Many more...

Example—Audio updates during a workout

# AVSpeechSynthesis and Accessibility

Powerful tool for helping many users

- Cognitive
- Speech vocalization
- Non-sighted

Not a replacement for VoiceOver

- Speech can overlap with VoiceOver's
- Won't be available to Braille devices
- Make your app accessible instead using UIAccessibility

# Basics

Create AVSpeechSynthesizer

Ensure it's retained until speech is done

- (Speech will be cancelled in the synthesizer is deallocated)

```
let synthesizer = AVSpeechSynthesizer()
```

# Basics

Create an utterance

Dispatch to synthesizer

```
let utterance = AVSpeechUtterance(string: "Hello")  
synthesizer.speak(utterance)
```

# AVSpeechSynthesis and Audio Sessions

AVAudioSession automatically activated on speak()

To mix with other audio, use

```
AVAudioSession.sharedInstance().setCategory(AVAudioSessionCategoryPlayback,  
with: .mixWithOthers)
```

To duck other audio, use

```
AVAudioSession.sharedInstance().setCategory(AVAudioSessionCategoryPlayback, with: .duckOthers)
```

# Callbacks

Delegate methods inform about the life cycle of an utterance

`AVSpeechSynthesizerDelegate` defines optional methods

- Speech started
- Speech finished
- Character range will be spoken
- Speech paused
- Speech continued



```
synthesizer.delegate = self
```

```
func speechSynthesizer(_ synthesizer: AVSpeechSynthesizer, didStart utterance: AVSpeechUtterance) {  
    print("Speech started")  
}
```

```
func speechSynthesizer(_ synthesizer: AVSpeechSynthesizer, didFinish utterance: AVSpeechUtterance) {  
    print("Speech finished")  
}
```

```
func speechSynthesizer(_ synthesizer: AVSpeechSynthesizer, willSpeakRangeOfSpeechString characterRange: NSRange, utterance: AVSpeechUtterance) {  
    guard let rangeInString = Range(characterRange, in: utterance.speechString()) else { return }  
    print("Will speak: \(utterance.speechString[rangeInString])")  
}
```

***Demo***



Speak

VPN 9:41 AM 100%

# Choosing the Right Voice

Built in voices (one for each supported language)

Siri voices not available through API

Users can download higher quality voices

# Choosing the Right Voice

Select with an identifier or a language

Selecting by language will select the users default voice

```
let utterance = AVSpeechUtterance(string: "Hello")
// Select an English (US) voice preferred by user (if no preference, default is used)
utterance.voice = AVSpeechSynthesisVoice(language: "en-US")

// Select the first voice
let allVoices = AVSpeechSynthesisVoice.speechVoices()
utterance.voice = AVSpeechSynthesisVoice(identifier: allVoices[0].identifier)
```

# Choosing the Right Voice

## Languages supported

---

Arabic	Cantonese (Hong Kong)	Czech	Danish	Dutch	English (US)
English (UK)	English (Australia)	English (Ireland)	English (South Africa)	Finnish	Flemish (Belgium)
French (France)	French (Canada)	German	Greek	Hebrew	Hindi
Hungarian	Indonesian	Italian	Japanese	Korean	Mandarin (Mainland China)
Mandarin (Taiwan)	Norwegian	Polish	Portuguese	Portuguese (Brazil)	Romanian
Russian	Slovak	Spanish (Mexico)	Spanish (Spain)	Swedish	Thai
Turkish					

---

# Speech Rate

Controls the rate of speech [0-1]

- Speaking Rate is scaled from  $\cong 0x \rightarrow 1x$  with values [0, .5]
- Speaking Rate is scaled from  $1x \rightarrow 4x$  with values [.5, 1]

```
let utterance = AVSpeechUtterance(string: "Hello")
utterance.rate = 0.75 // faster
utterance.rate = 0.25 // slower
utterance.rate = AVSpeechUtteranceDefaultSpeechRate
utterance.rate = AVSpeechUtteranceMaximumSpeechRate
```

# Pitch and Volume

Set properties on AVSpeechUtterance

- Pitch [0-1]
- Volume [0-1]

```
let utterance = AVSpeechUtterance(string: "Hello")
utterance.pitch = 1 // high pitched voice
utterance.volume = 0.25 // lower speech volume, does not affect system volume
```



# Attributed Strings

Customize behavior of generated speech with attributes

## IPA Notation

- International Phonetic Alphabet
- Uses—Specialized names, proper nouns, business names...
- Available—en-US, en-AU, en-GB, de-DE, es-ES, es-MX, fr-CA, fr-FR, it-IT, ja-JP
- Example—iPhone = 'aɪ.ˈfoʊn

# IPA Notation

How do you generate IPA notation?

Settings > General > Accessibility > Speech > Pronunciations

Phrase  
iPhone

Substitution



Dictate or spell out how you want the phrase to be pronounced.

Languages >  
English (US)



Phrase  
iPhone

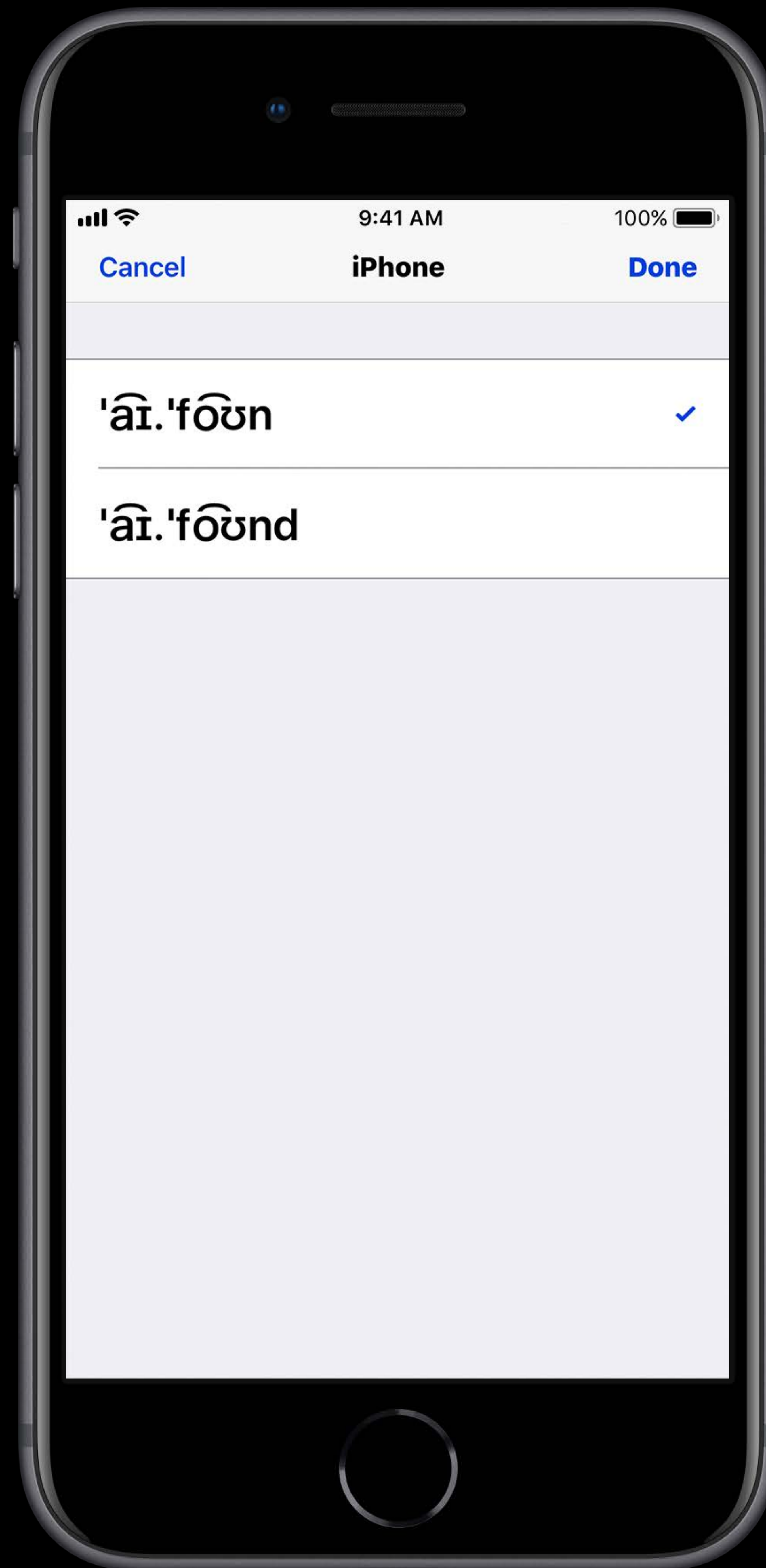
Substitution



Dictate or spell out how you want the phrase to be pronounced.

Languages  
English (US)





9:41 AM

100%

Cancel

iPhone

Done

'aɪ. 'fəʊn



'aɪ. 'fəʊnd

Phrase  
iPhone

Substitution   
'aɪ.'foʊn

Dictate or spell out how you want  
the phrase to be pronounced.

Languages >  
English (US)

Voices >  
All

Ignore case

# IPA Notation

Use attributed string API with IPA pronunciation

```
let attributedString = NSMutableAttributedString(string: "Hello iPhone")
attributedString.addAttribute(.accessibilitySpeechIPANotation, value: "'âɪ.'fôʊn", range:
NSRange(location: 6, length: 6))

let utterance = AVSpeechUtterance(attributedString: attributedString)
```

# Summary

Augment your app experience by adding speech at the right time

Multiple languages and voices are available

Customize pronunciation with IPA notation



# More Information

<https://developer.apple.com/wwdc18/236>

 **WWDC18**