

Building Apps with Dynamic Type

Session 245

Clare Kasemset, Software Engineering Manager
Nandini Sundar, Software Engineer

Agenda

What is Dynamic Type?

What's new in iOS 11

Guidelines and API

Demos with a sample app

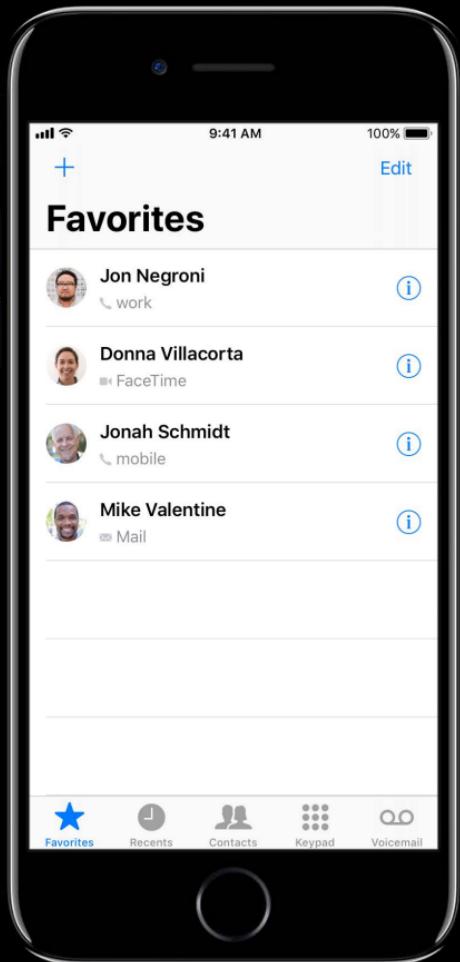
What is Dynamic Type?

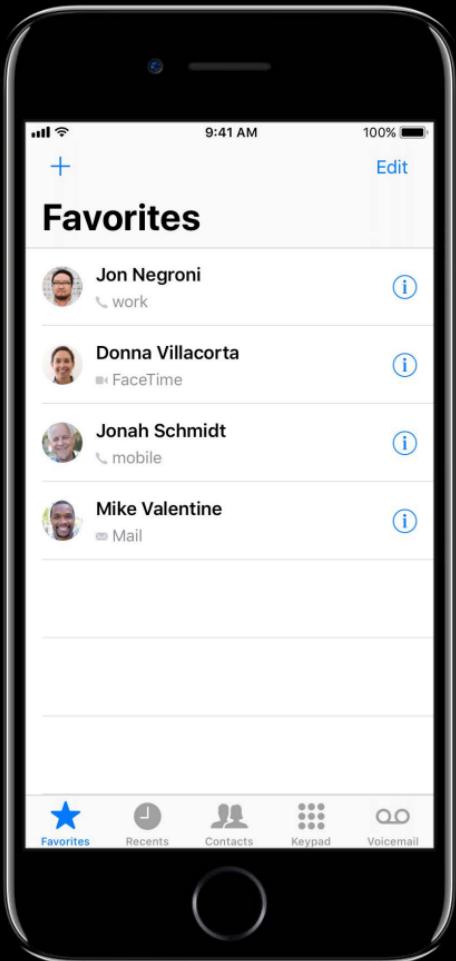
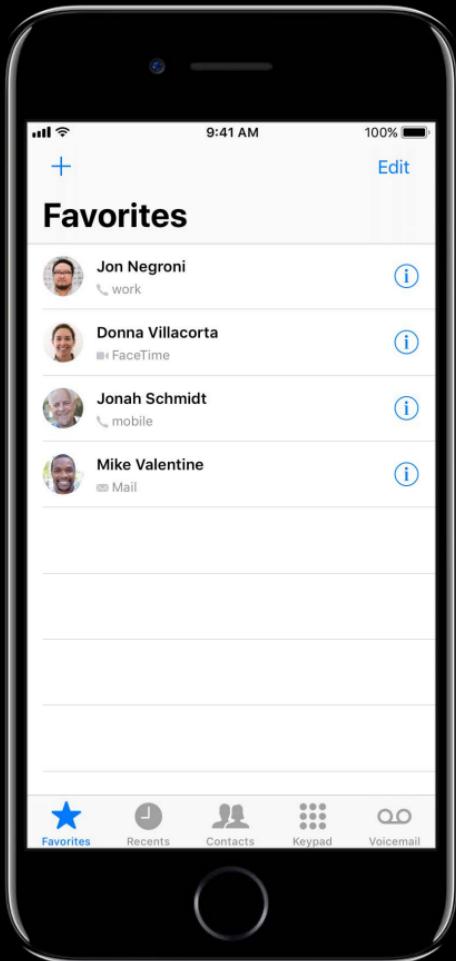
다이나믹 타입이란?

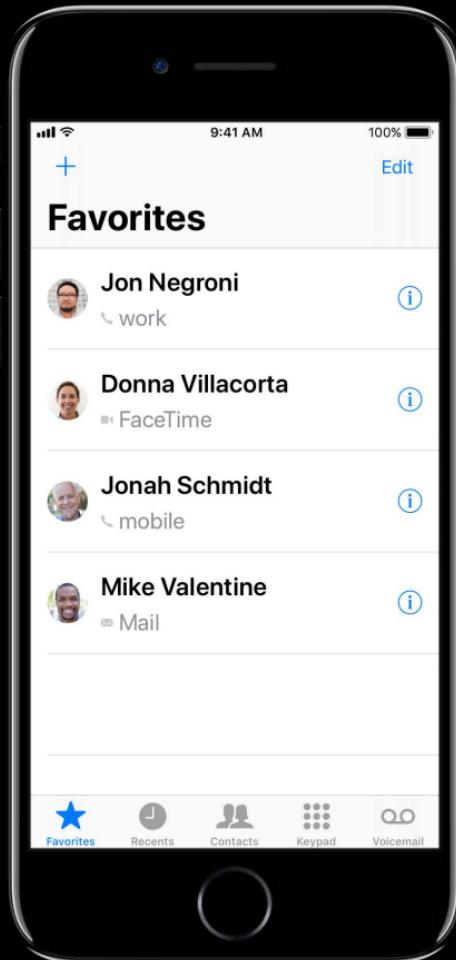
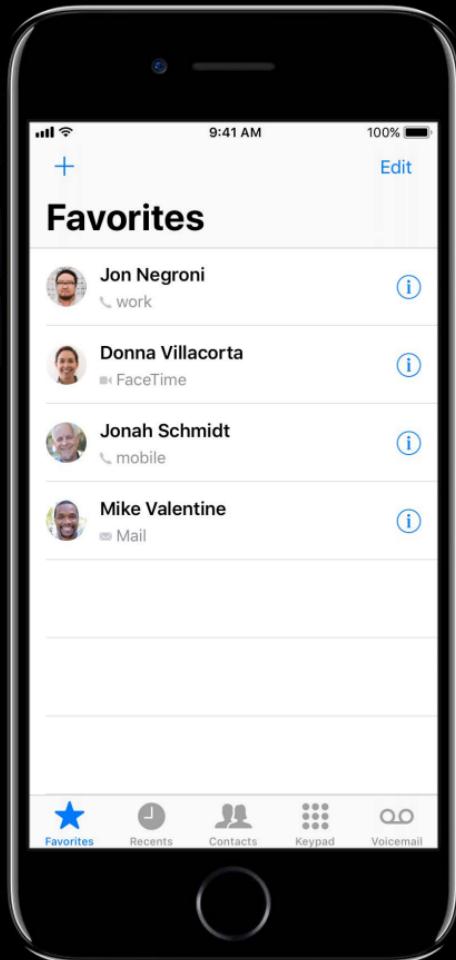
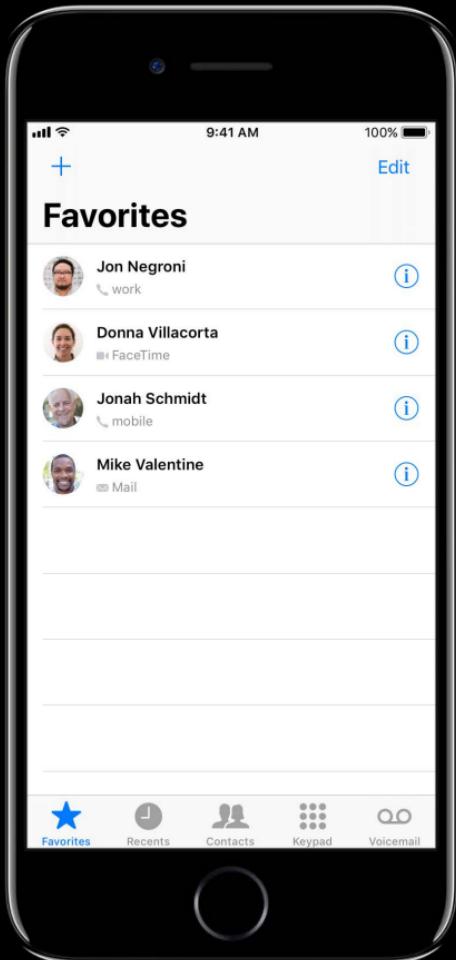
사용자가 글자 크기를 조절할 수 있는 것

- 누군가는 더 큰 글자를 선호한다

또는 작은







ପ୍ରକାଶ ଲେଖକ



노안

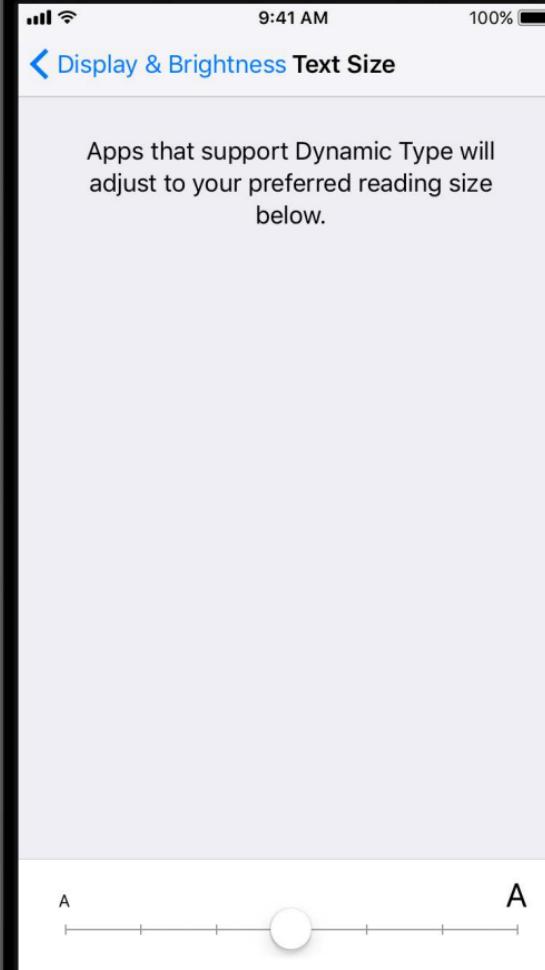


선호의 문제가 아니라 누군가에게는 필수적이다.

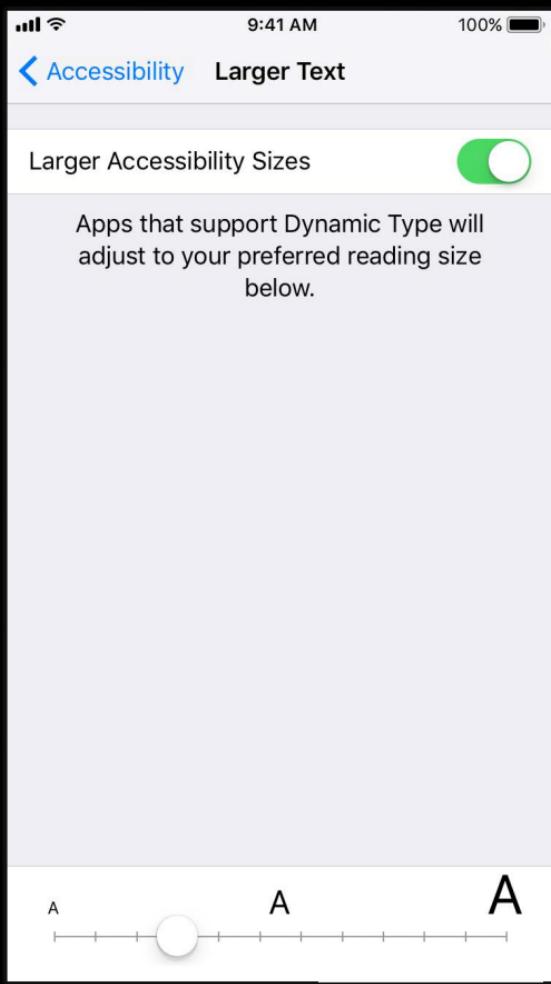


다이나믹 타입은 설정 앱에서 찾을 수 있다.
디스플레이와 밝기

7단계가 있으며 기본값은 중간이다.

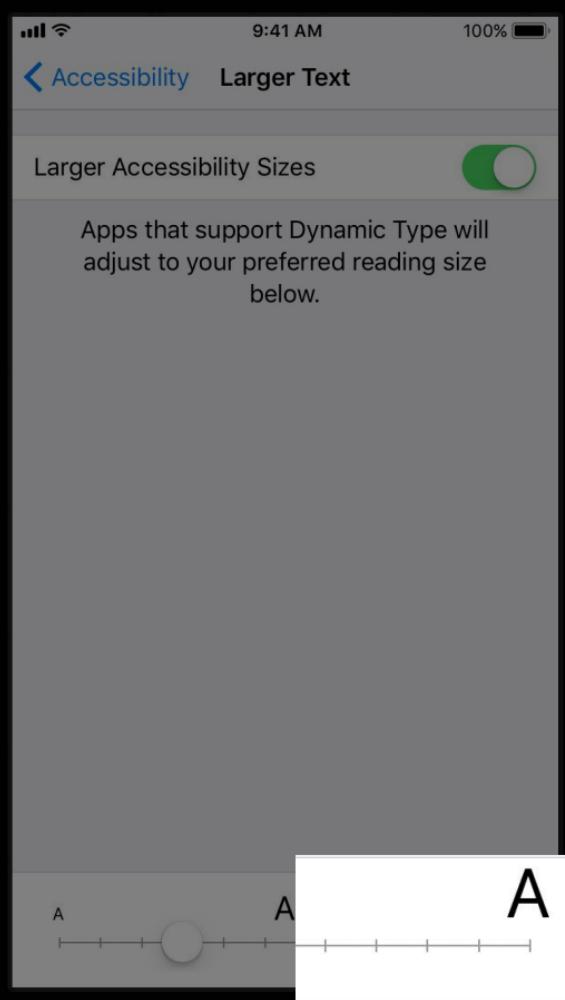


그리고 접근성 설정에서도 찾을 수 있다

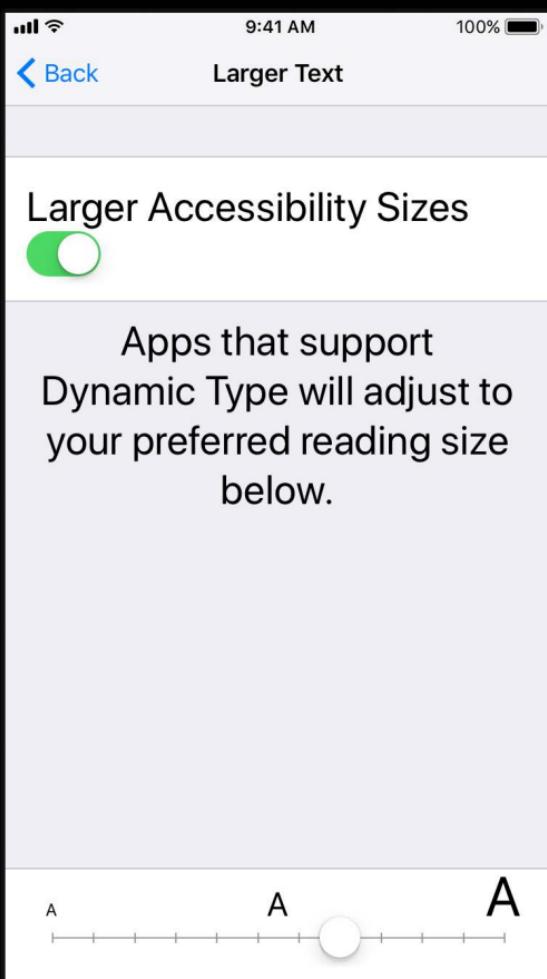


여기서는 5단계 더 커울 수 있다

시력이 낫은 사용자를 위해 추가 제공된다.

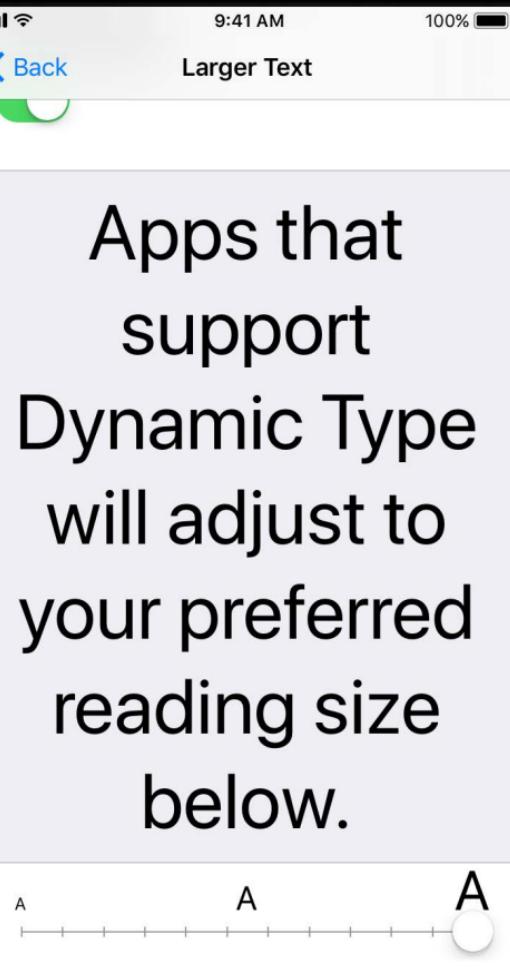


기본
크기
가지가
이정도라면



흐릿한 더 큰 글씨로 가독성이 있다.

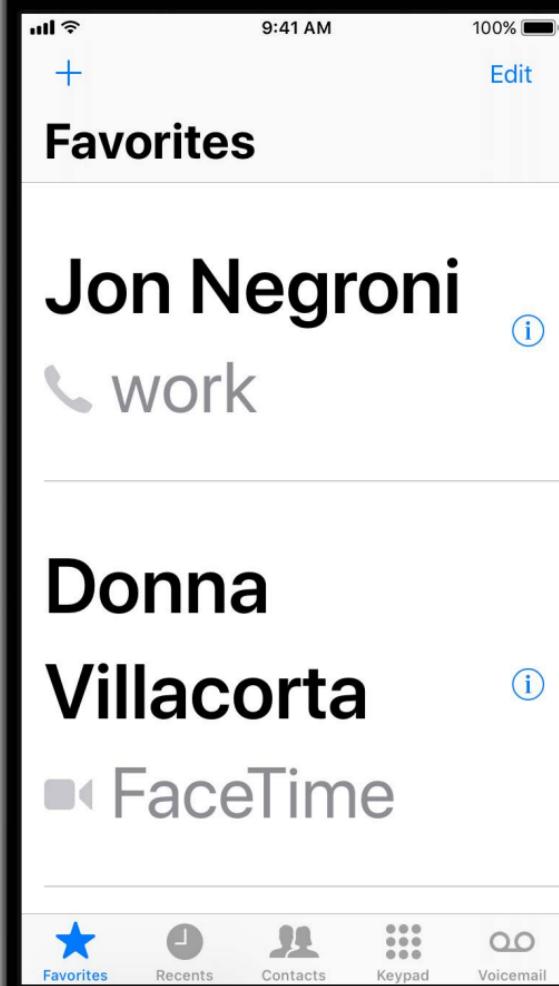
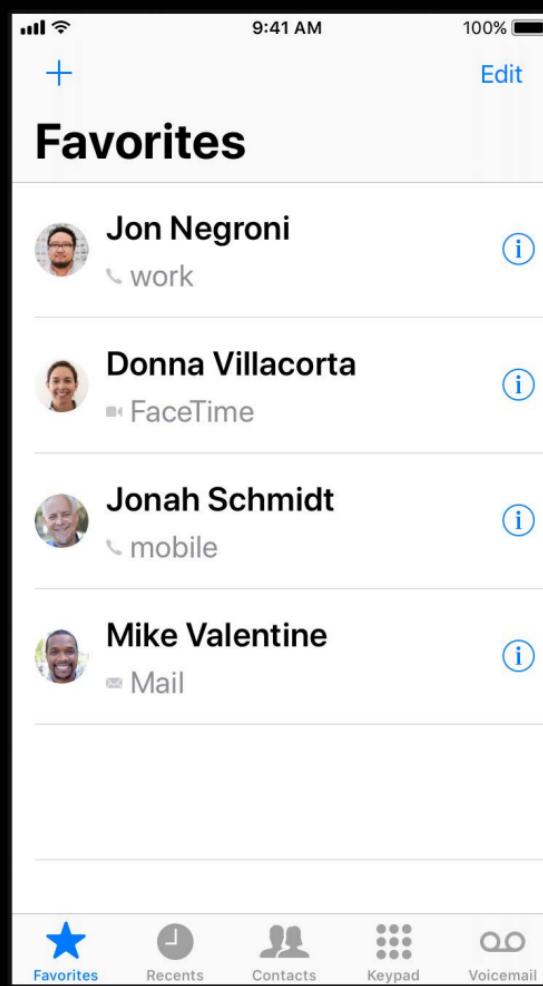
이 가능성이 정밀한 유형화기 확장해 보자.



원쪽은

일반 설정에서
문자 크기를 화대로
한 것

오른쪽은 정상
화면

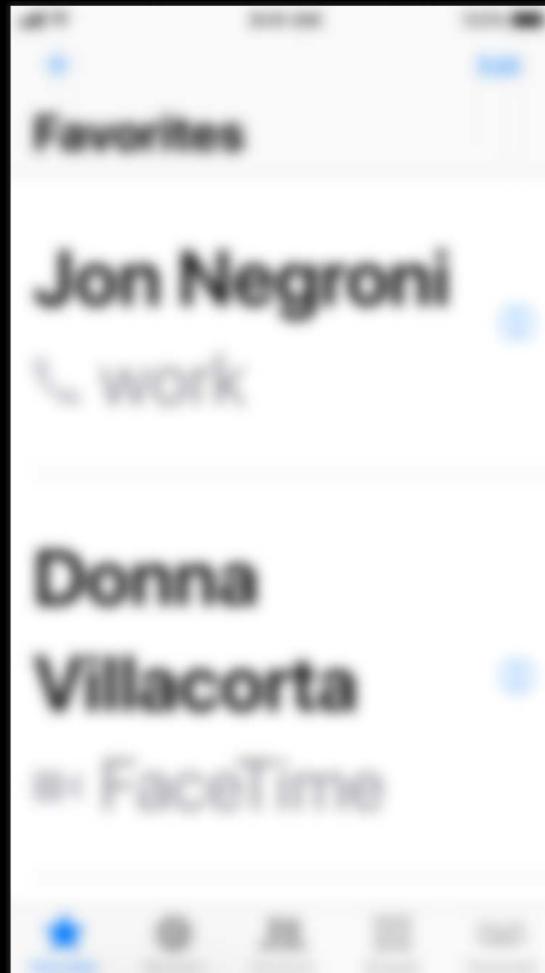
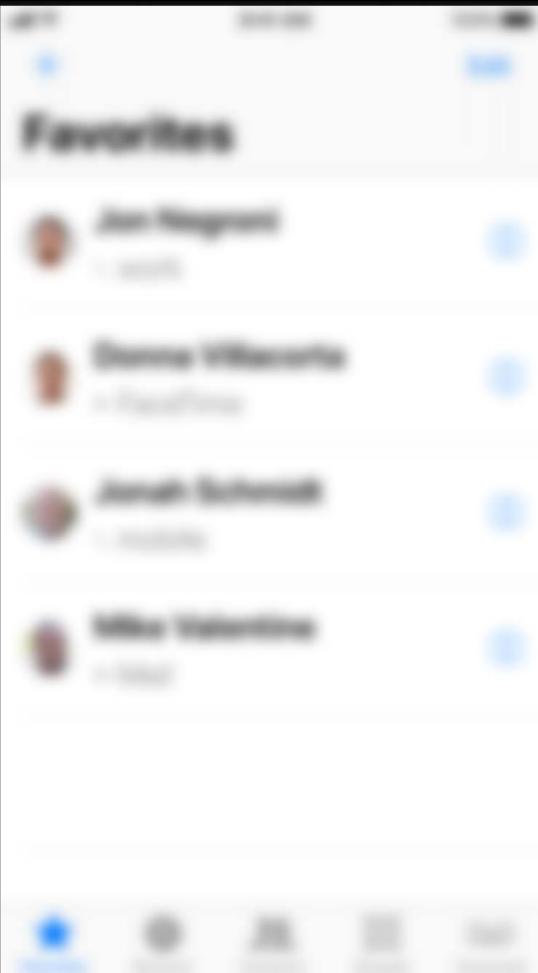


저서역 이영자 관정
에서 보자.

원쪽은 거의 읽을수 없다.

오른쪽은 훨씬
읽을 수 있다.

글자크기는 가독성에
엄청난 영향을 준다.



iOS 11 | 새 기능
다양한 디자인 템플릿을
선택해 보세요.

What's New in iOS 11

Goals

106 || ၁၅။ ဒေသကြောင်း မြန်မာ့ အတွက် ဖြစ်ပါသည်.

Goals

Text is large enough for the user to read

글자는 반드시 봄을 수 있을 만큼 커야 한다.

Goals

Text is large enough for the user to read

Text is fully readable

글자는 잘려서 읽을수 없으면 안된다.

人

정지경

Goals

Text is large enough for the user to read

Text is fully readable

App UI looks beautiful

어떤 글자 크기에서도 UI가 맘마저 앓아야 한다.

Goals

Text is large enough for the user to read

Text is fully readable

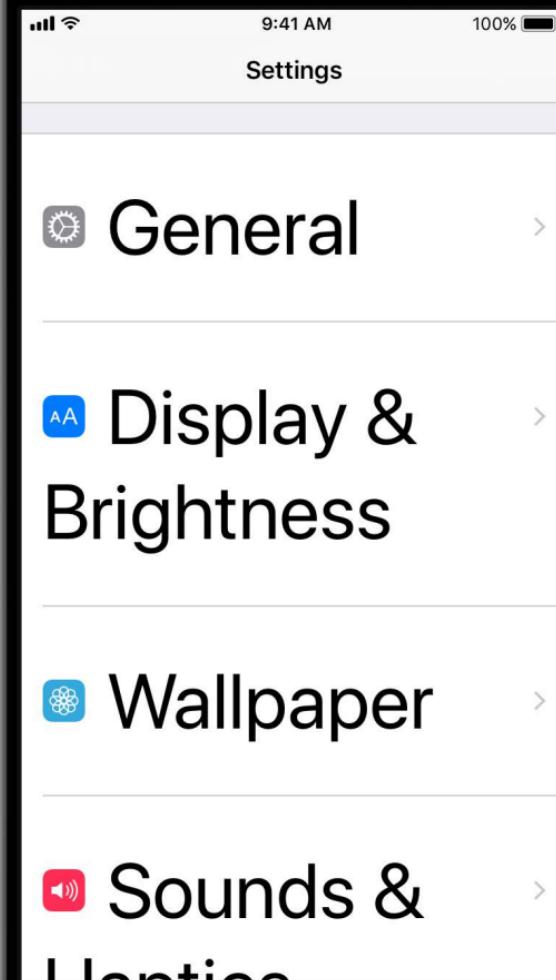
App UI looks beautiful

디자인은 사용자에게
매우 중요합니다

Design For Everyone

WWDC 2017

여기 설정 앱~
보면
어떤 글자 크기에서도
먼저기 찾는다.

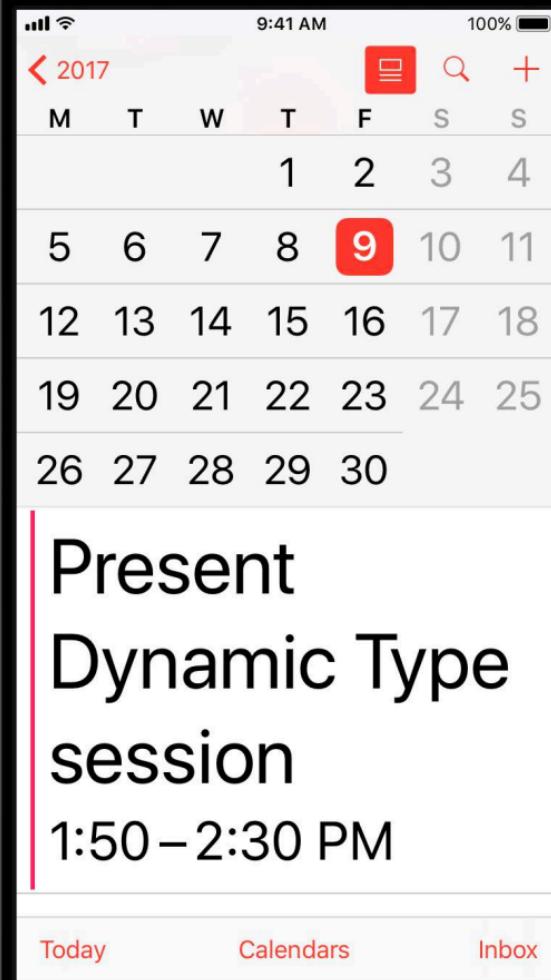
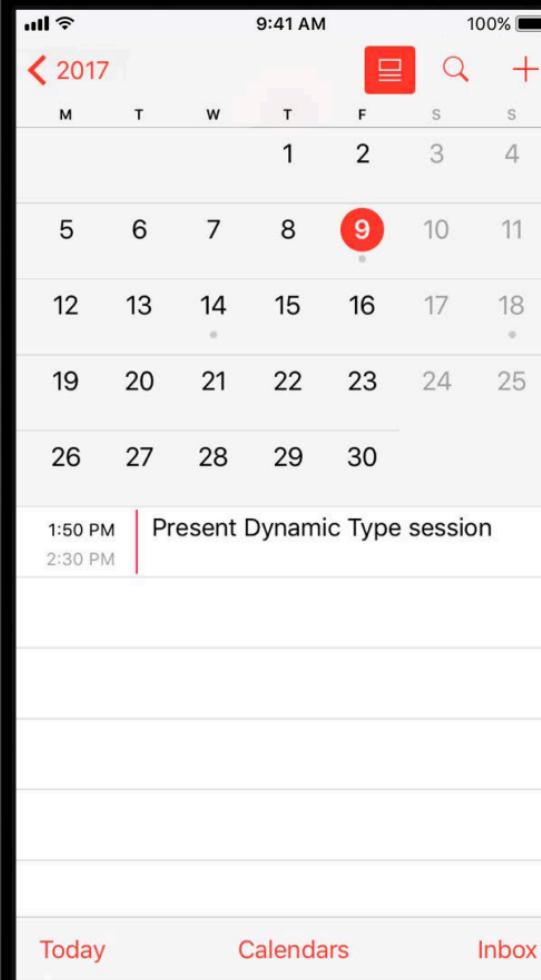


다른 예식인 달력이다.

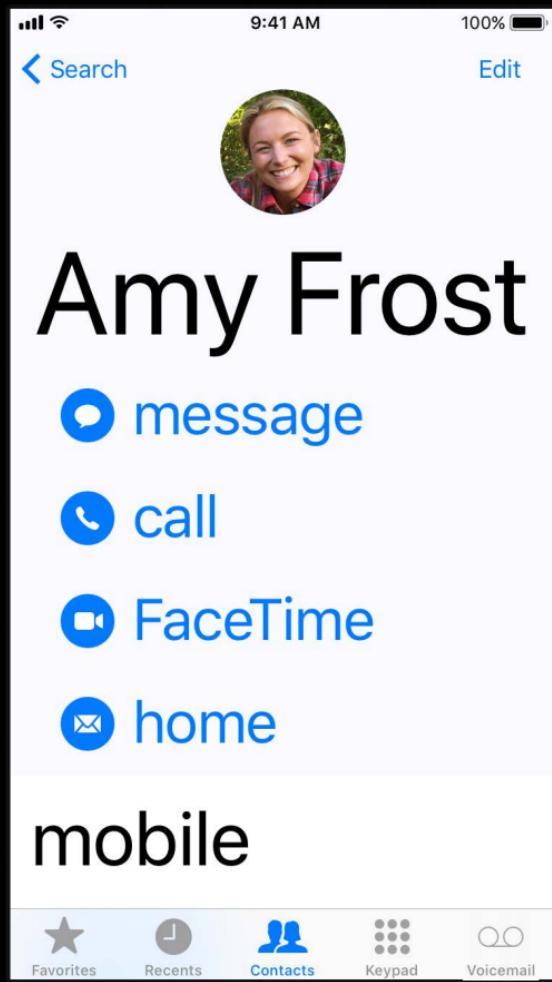
이행은 시간이

더 큰 규모에서는

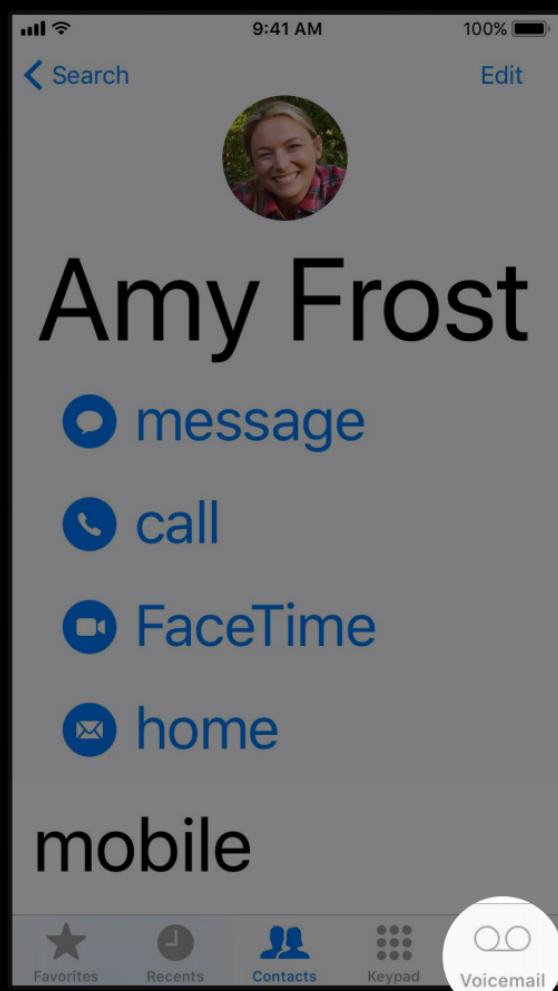
다르게 위치한다.



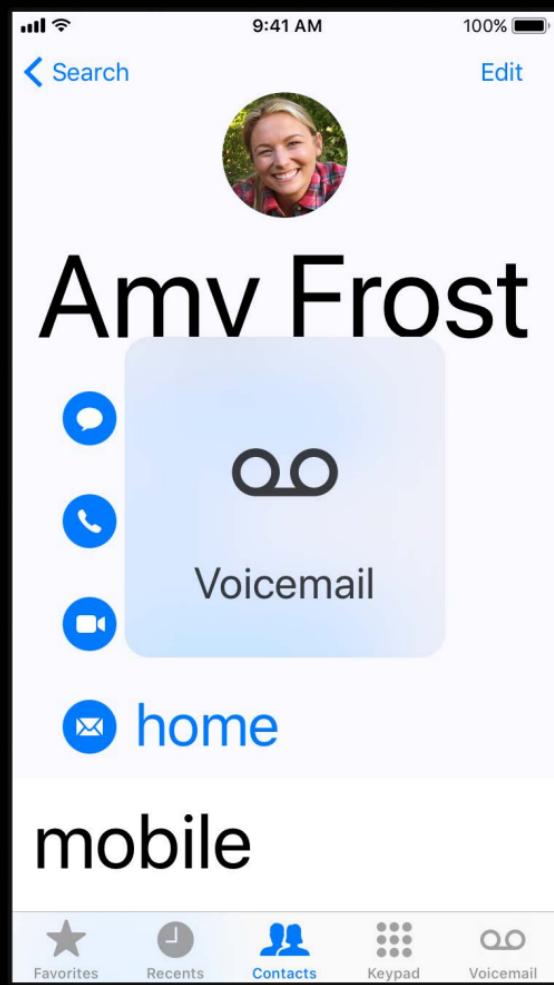
여기서 템플릿의 글자 크기가 고정된다.



Ios 11.0.1
방화면 길거 누군가



해당 Item이 크게 보여집니다.



Guidelines and API

Guidelines and API

Scaling font sizes 화면에 맞게 크기 조절

Accommodating large text 큰 글자를 허용하기 위해 확장하는 법

Table views 가장 일반적인 table view 기법

Images 다이나믹, 태팅에서 이미지를 어렵게 만들고 싶을 때

Scaling Font Sizes

글자크기로
제어하는 가장 쉬운 방법은

내장 text style을
이용하는 것이다.

오른쪽은 기본 폰트의 경우이다.

Title1: System 28pt

Title2: System 22pt

Title3: System 20pt

Headline: System 17pt

Body: System 17pt

Callout: System 16pt

Subhead: System 15pt

Footnote: System 13pt

Caption1: System 12pt

Caption2: System 11pt

Text Styles

기본의 푸른색이니 크기이다.
↑
24

Standard Sizes

Title1
Title2
Title3
Headline
Body
Callout
Subhead
Footnote
Caption1
Caption2

Accessibility Sizes

Text Styles

인쇄 || 이전에는

Body style만

설정은 크게 바꿨다.

Standard Sizes

Title1
Title2
Title3
Headline
Body
Callout
Subhead
Footnote
Caption1
Caption2

Accessibility Sizes

Body

Text Styles

NEW

Ios 11에서는

제작

모든 디자인에 적용한다.

Ios 11에서는 스타일은 사용하지
모두 영향을 않는다.

Standard Sizes

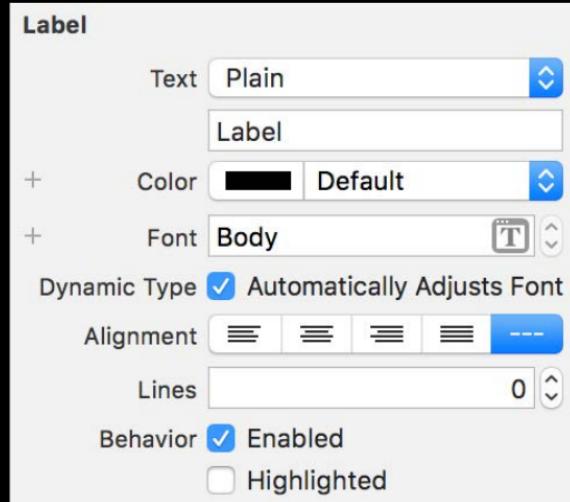
Title1
Title2
Title3
Headline
Body
Callout
Subhead
Footnote
Caption1
Caption2

Accessibility Sizes

Title1
Title2
Title3
Headline
Body
Callout
Subhead
Footnote
Caption1
Caption2

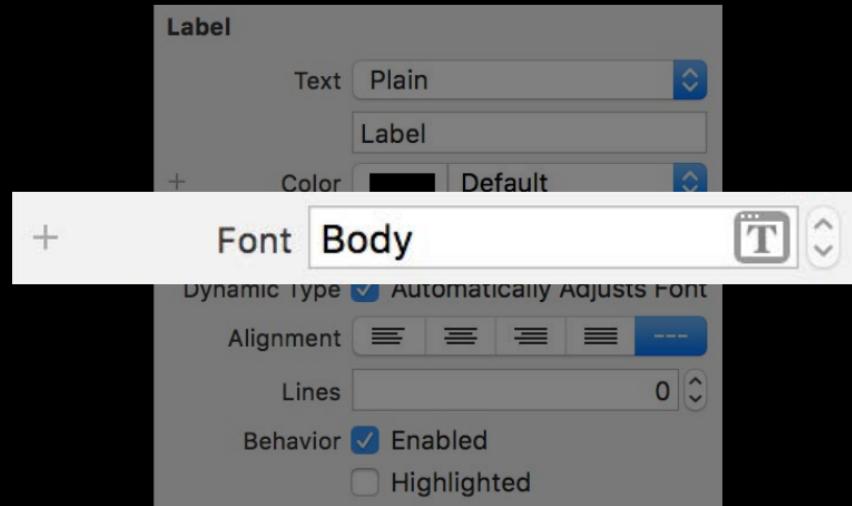
Text Styles

IB의 Style은 변경 가능하다.



Text Styles

보통 System 17과 같은 크기를 기대한다.



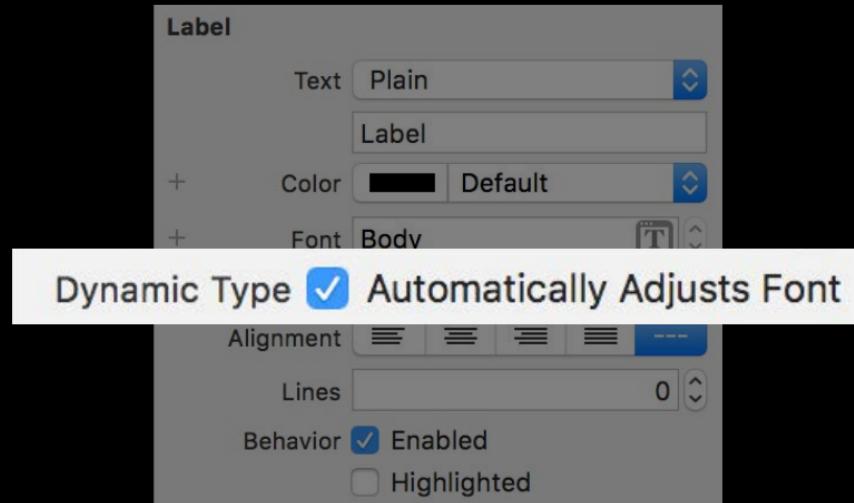
Text Styles

NEW

Ios 11에서 이 편집이 추가되었다.

체크가 되어있으면 크기 조정을 허용한다.

기본값은 체크 해제 상태이다.



Text Styles

```
label.font = UIFont.preferredFont(forTextStyle: .body)  
label.adjustsFontForContentSizeCategory = true
```

코드로 적용하는 것 또한 가능하다.

Text Styles

```
label.font = UIFont.preferredFont(forTextStyle: .body)  
label.adjustsFontForContentSizeCategory = true
```

Text Styles

```
label.font = UIFont.preferredFont(forTextStyle: .body)  
label.adjustsFontForContentSizeCategory = true
```

Custom Fonts

text style 을 사용할 수 있다면?

custom font 를 적용하는 것이다.

Title Font

Body Font

Custom Fonts

자연스럽게
제작된
Custom fonts도 자연스럽게
제작할 수 있습니다.

NEW

Title Font

Body Font

Title Font

Body Font

Custom Fonts

기존의 폰트 대신에 사용자 정의 폰트 사용

```
label.font = customFont
```

Custom Fonts

NEW

글꼴을
조작 가능한
코드 가이드
제공 가능하다.

```
label.font = UIFontMetrics.default.scaledFont(for: customFont)
```

Custom Fonts

NEW

기본 설정은 Body style을 기준으로 한 것이다.

특정 스타일의 폰트 크기로 조절 할 수 있다.

```
titleLabel.font = UIFontMetrics(forTextStyle: .title1).scaledFont(for: customFont)
```

Web Views

웹 뷰에서 조작도 가능하다.

아이폰의 크기를 지원하는 만큼 오직 apple 기기에서만 가능하다.

```
body {  
    font: -apple-system-body; // available on Apple devices only  
}  
  
h1 {  
    font-size: 1.3rem;  
}
```

Web Views

```
body {  
    font: -apple-system-body; // available on Apple devices only  
}  
  
h1 {  
    font-size: 1.3rem;  
}
```

Web Views

다른 폰트로 지원하지면
fall back 폰트를 설정 해야한다.

```
body {  
    font: -apple-system-body; // available on Apple devices only  
}  
  
h1 {  
    font-size: 1.3rem;  
}
```

이제 더 큰 글자에서 앱 제작/판권 등
어떻게 가능성이 있다?

Accommodating Large Text

Fitting Large Text on Screen

문자가 가로사이즈에 맞다고 치자

Lorem ipsum dolor sit amet

Fitting Large Text on Screen

Scale font size

글자를
커우자
마자
문제가
생겼다.

Lorem ipsum dolor sit amet

Fitting Large Text on Screen

Constrain trailing edge

+ trailing 제약을 추가했다.

여전히 문제가 있다. 문자가 너무 잘린다.



Lorem ipsum...

Fitting Large Text on Screen

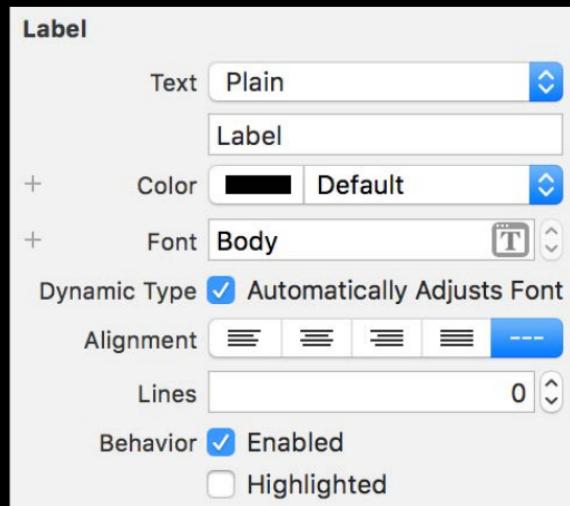
Wrap to multiple lines

여러줄로
내용을
다음과
같이 표시합니다.

**Lorem ipsum
dolor sit amet**

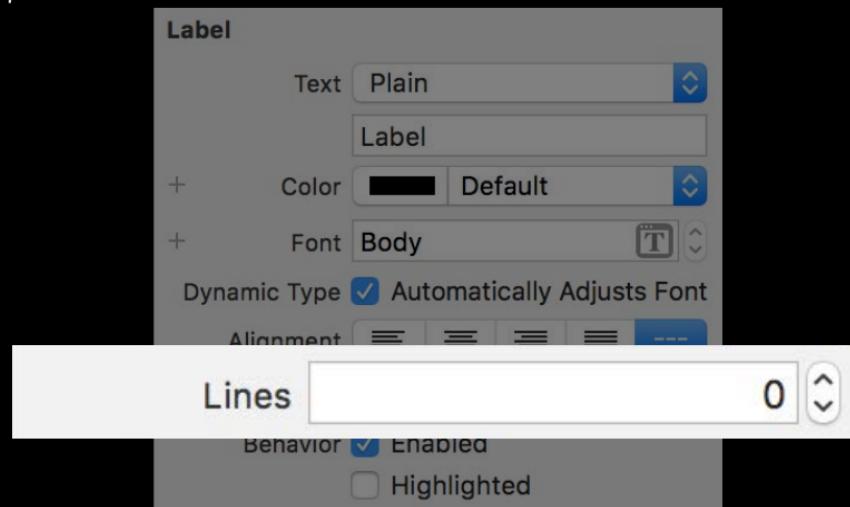
Wrap to Multiple Lines

IB에서 하는 방법은



Wrap to Multiple Lines

Lines 2
0으로 적용해 주는 것이다.



Wrap to Multiple Lines

```
label.numberOfLines = 0
```

Avoid Constant Values Based on Default Text Size

40만큼 제약을 걸었는지와 기본 폰트에서는 문제 없게 보인다.

First Label
Second Label **I**

Avoid Constant Values Based on Default Text Size

다이나믹 터치어/서는 이런 문제가 생길 수도 있다

First Label
Second Label I

First Label
Second Label I

Avoid Constant Values Based on Default Text Size

First Label
Second Label I

First Label
Second Label I

우리나라 원하는 것은
이것입니다.

NEW

Auto Layout System Spacing Constraints

IOS 11에서는 새로운 AutoLayout의 `width`.

두 `Baseline` 사이에 적절한 공간을 결정한다. 폰트에 따라

```
secondLabel.firstBaselineAnchor.constraintEqualToSystemSpacingBelow(  
    firstLabel.lastBaselineAnchor, multiplier: 1.0)
```

Auto Layout System Spacing Constraints

NEW

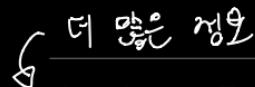
얕은 고정의 풀리기로
multiplier 를 조절한다

```
secondLabel.firstBaselineAnchor.constraintEqualToSystemSpacingBelow(  
    firstLabel.lastBaselineAnchor, multiplier: 1.0)
```

Auto Layout System Spacing Constraints

NEW

```
secondLabel.firstBaselineAnchor.constraintEqualToSystemSpacingBelow(  
    firstLabel.lastBaselineAnchor, multiplier: 1.0)
```



Scaled Values

오토레이아웃을 사용안하는 뷰나 코드도 처리 가능하다.

```
frame.origin.y += 40.0
```

Scaled Values

text 사이즈에 따라 float 값(?) 스케일링 조절 했던 것 같아.

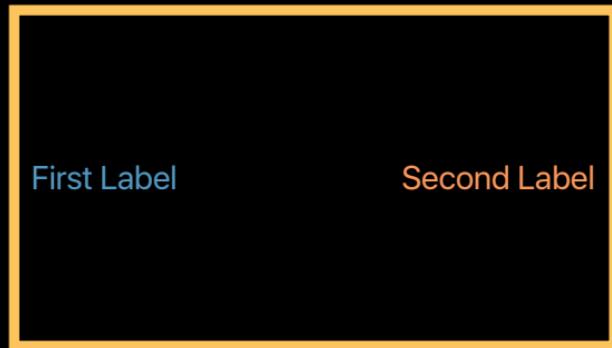
NEW

```
frame.origin.y += UIFontMetrics.default.scaledValue(for: 40.0)
```

Side-By-Side Text

다른 시나리오를 아시우마

두개의 라벨이 있다.



First Label

Second Label

Side-By-Side Text

Scale font size

크기를 커우면 잘됨이 발생한다.

First... Seco...

Side-By-Side Text

Wrap to multiple lines

Lines 를 조절 해봅시다.

- 아까전 3가지 목표를 되세계봅시다
- 이온수 액을 밤과 커야한다
- 잘건우篷이 없어야 한다.
- 제이아웃이 아름다워야 한다



Side-By-Side Text

Stack vertically

✓ - Stacking 이 더 좋아보인다

쪽을 좌우 전환²을 사용하는게

더 좋아보인다.

First Label
Second Label

1/20 오후 9:41
스마트 블로그
모두 사용한다.

Some things you can ask me:



Phone

"Call Brian"



FaceTime

"FaceTime Lisa"



Apps

"Launch Photos"



Messages

"Tell Susan I'll be right there"



Calendar

"Set up a meeting at 9"



Phone

"Call Brian"



FaceTime

"FaceTime Lisa"



Apps

"Launch Photos"

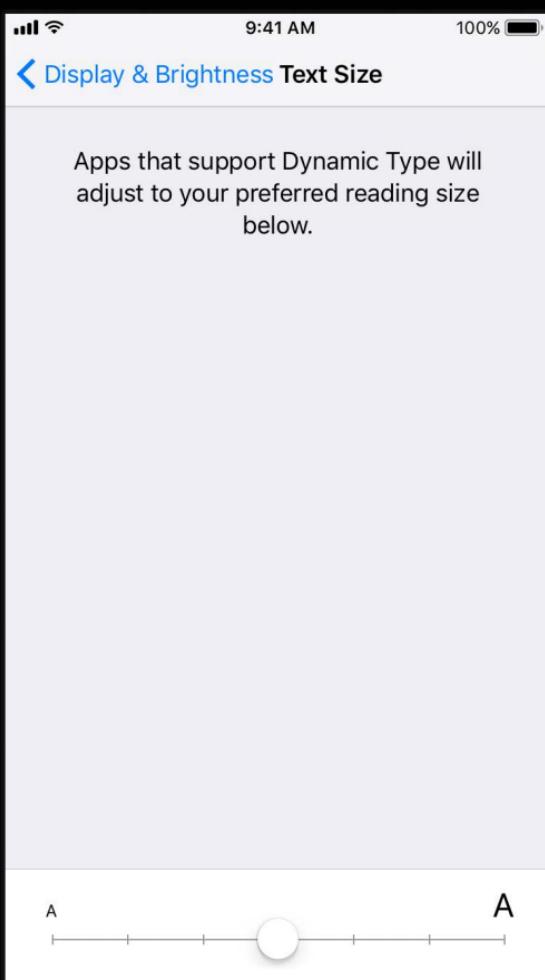


Messages

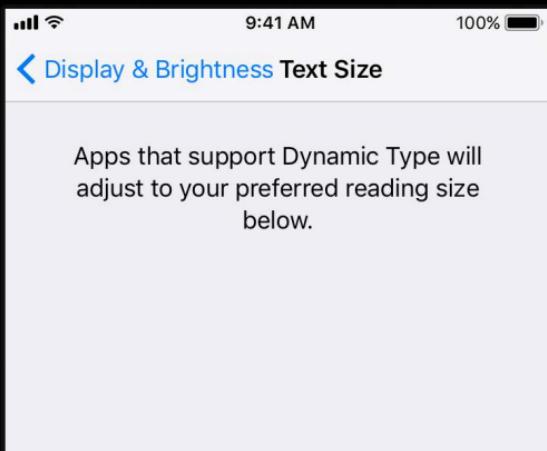
"Tell Susan I'll be right



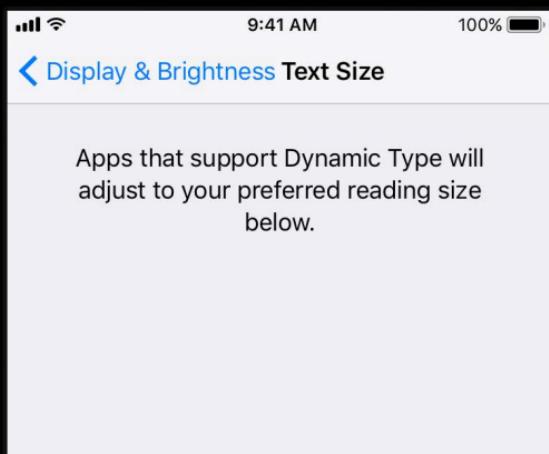
କାଣ୍ଡା ମାତ୍ରା ଏଇହାଙ୍କି କିମ୍ବା ଆଖିଲୁ
ଥାଇଗାନ୍ତିର



0 | 한글 헤드라인.



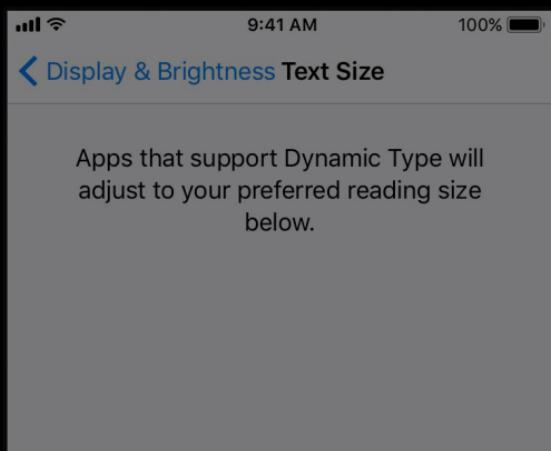
traitCollection.preferredContentSizeCategory



```
traitCollection.preferredContentSizeCategory
```

```
UIApplication.shared.preferredContentSizeCategory
```





`traitCollection.preferredContentSizeCategory`

`UIApplication.shared.preferredContentSizeCategory`

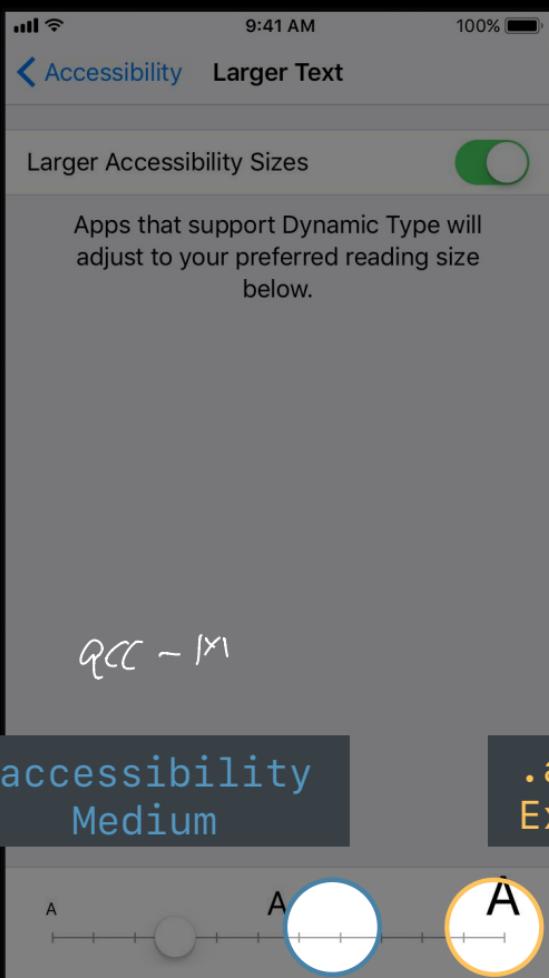
`.extraSmall`



`.large`

`3X`

`.extraExtraExtraLarge`



NEW

Make Layout Decisions Based on Text Size

iOS에는 사용할 function이 많지
않아서는 안쓰고 대체하는 편이
훨씬 좋을 수 있음.

```
if traitCollection.preferredContentSizeCategory.isAccessibilityCategory {  
    // Vertically stack  
} else {  
    // Lay out side by side  
}
```

Make Layout Decisions Based on Text Size

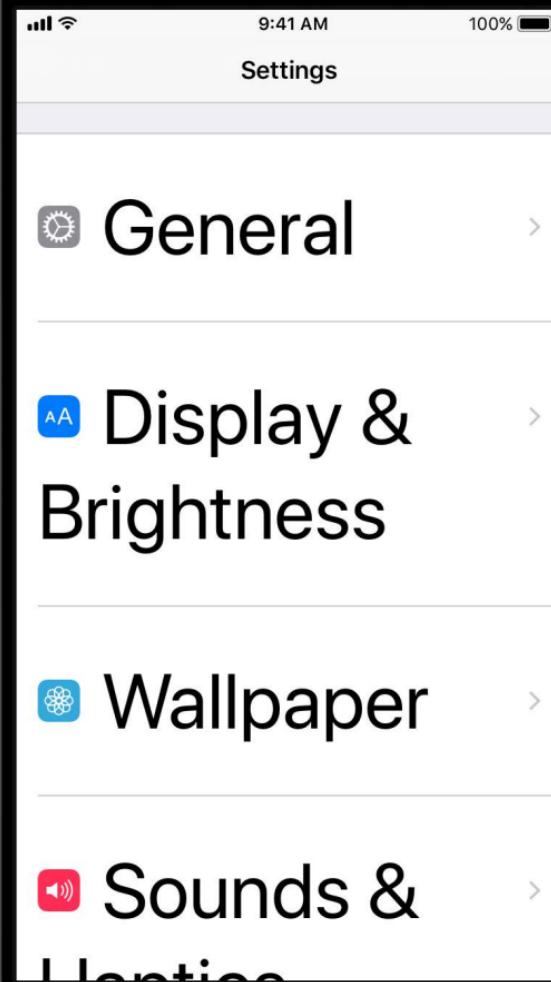
또는 암세값을 이용해 수도 있다.

NEW

```
if traitCollection.preferredContentSizeCategory > .extraExtraLarge {  
    // Vertically stack  
} else {  
    // Lay out side by side  
}
```

Table Views

UITableView
 UITableView
 UITableView
 UITableView



detail은
나중에
✓ Stack
all의 정의

Table View

| | |
|------------|-------------------|
| Text label | Detail text label |

Table View

Text label

Detail text label

Text label

Detail text label

Text label

Detail text label

| | |
|------------|-------------------|
| Text label | Detail text label |

Default Table View Behaviors in iOS 11

NEW

Default Table View Behaviors in iOS 11

NEW

Standard table view cells adapt layout for Dynamic Type

다이나믹 타입 & Cell 높이를 가지고 흥미롭

Default Table View Behaviors in iOS 11

NEW

Standard table view cells adapt layout for Dynamic Type

Cell heights are based on their content

Self-Sizing Table View Cells

Self-Sizing Table View Cells

Table view asks each cell to provide its own size

Self-Sizing Table View Cells

Table view asks each cell to provide its own size

Provide an estimated row height for off-screen cells

세일포사이즈에 따른 셀의 경우 먼저 넓은 크기를 물어본다.

Self-Sizing Table View Cells

Self-Sizing Table View Cells

Enable self-sizing for cells if needed

ios||은 세(프 사이즈) default 옵션.

Self-Sizing Table View Cells

만약 고정크기를 사용한다면 이정기 틀 수 있다.

Enable self-sizing for cells if needed

```
tableView.rowHeight = UITableViewAutomaticDimension  
tableView.estimatedRowHeight = <a reasonable estimate>
```

Self-Sizing Table View Cells

Enable self-sizing for cells if needed

```
tableView.rowHeight = UITableViewAutomaticDimension  
tableView.estimatedRowHeight = <a reasonable estimate>
```

If applicable, do the same for section headers and footers

Self-Sizing Table View Cells

Enable self-sizing for cells if needed

```
tableView.rowHeight = UITableViewAutomaticDimension  
tableView.estimatedRowHeight = <a reasonable estimate>
```

If applicable, do the same for section headers and footers

```
tableView.estimatedSectionHeaderHeight = <a reasonable estimate>  
tableView.sectionHeaderHeight = UITableViewAutomaticDimension  
tableView.estimatedSectionFooterHeight = <a reasonable estimate>  
tableView.sectionFooterHeight = UITableViewAutomaticDimension
```

header footer ↴

Self-Sizing Custom Cells

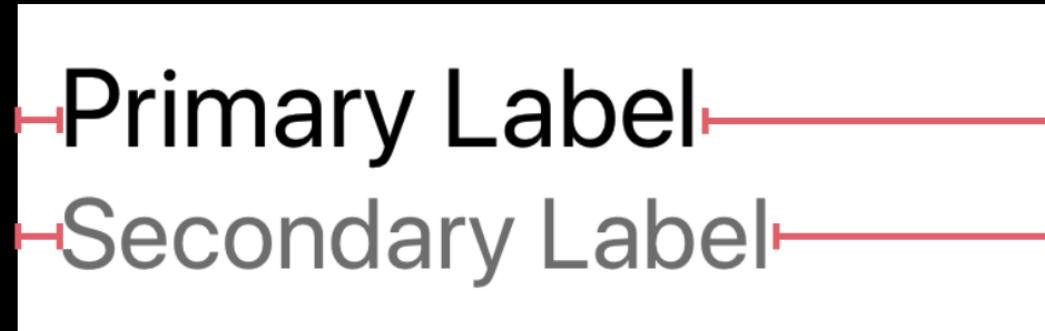
011/2 24

Primary Label
Secondary Label

Self-Sizing Custom Cells

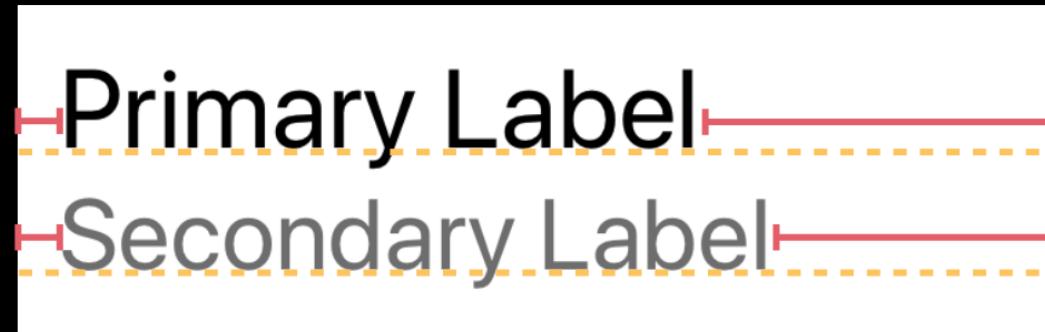
leading, trailing 제약이 있음.

```
contentView.layoutMarginsGuide.leadingAnchor.constraint(equalTo: primaryLabel.leadingAnchor)
```



Self-Sizing Custom Cells

```
contentView.layoutMarginsGuide.leadingAnchor.constraint(equalTo: primaryLabel.leadingAnchor)
```

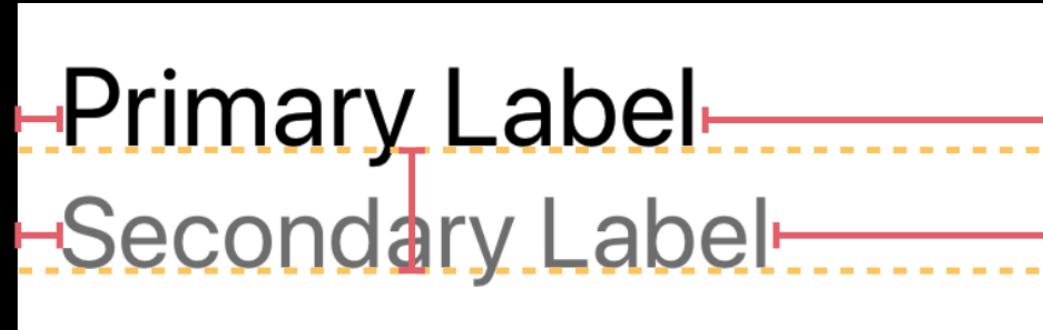


Self-Sizing Custom Cells

```
secondaryLabel.firstBaselineAnchor.constraintEqualToSystemSpacingBelow(  
primaryLabel.lastBaselineAnchor, multiplier: 1.0)
```

수직으로 서로 헛어뜨렸다. (아까 알았던 방법)

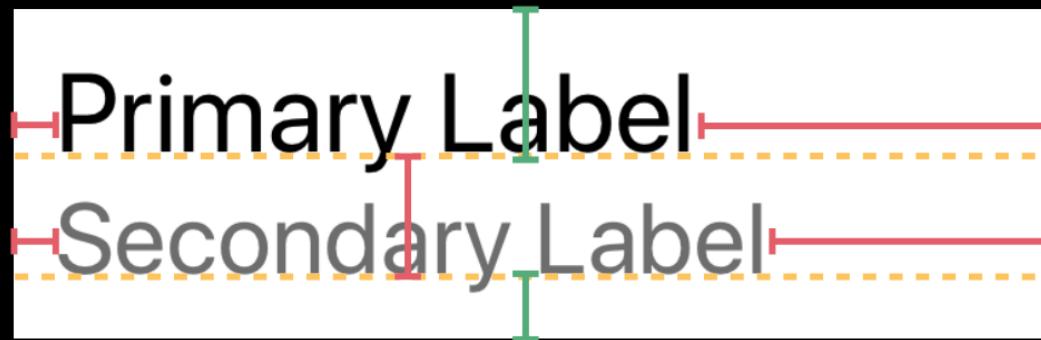
이제 끝인가?



Self-Sizing Custom Cells

```
primaryLabel.firstBaselineAnchor.constraintEqualToSystemSpacingBelow(  
    contentView.topAnchor, multiplier: 1.0),  
contentView.bottomAnchor.constraintEqualToSystemSpacingBelow(  
    secondaryLabel.lastBaselineAnchor, multiplier: 1.0)
```

상하에 대응
제약이 추가로
설정된다.
그때는 세 가지
일 수 있다



Self-Sizing Custom Cells

Self-Sizing Custom Cells

Also possible with manual layout

ручной
установкой
размеров

(not autolayout)

Self-Sizing Custom Cells

Also possible with manual layout

- Override `sizeThatFits` to return correct height

Self-Sizing Custom Cells

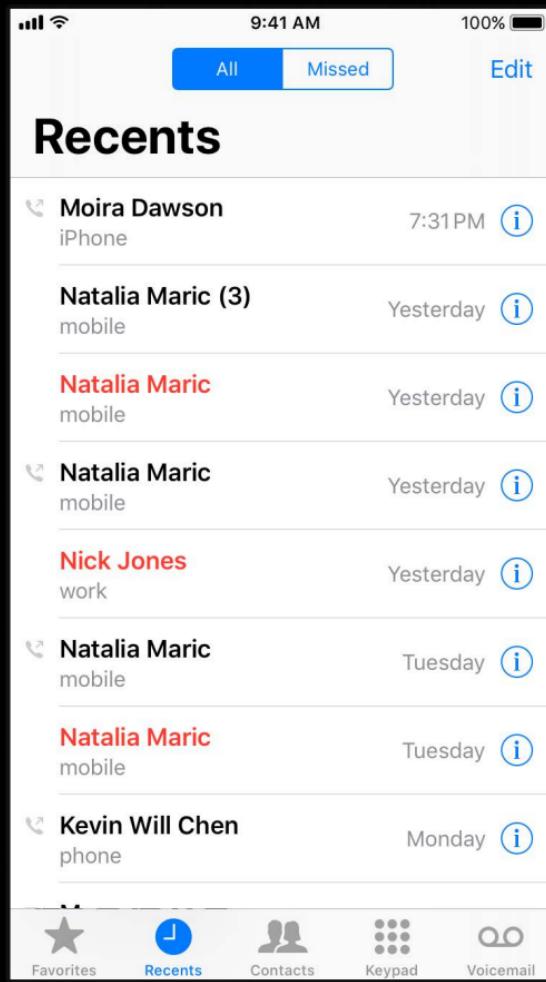
Also possible with manual layout

- Override `sizeThatFits` to return correct height
- Use `contentView.bounds.size.width` to determine available width

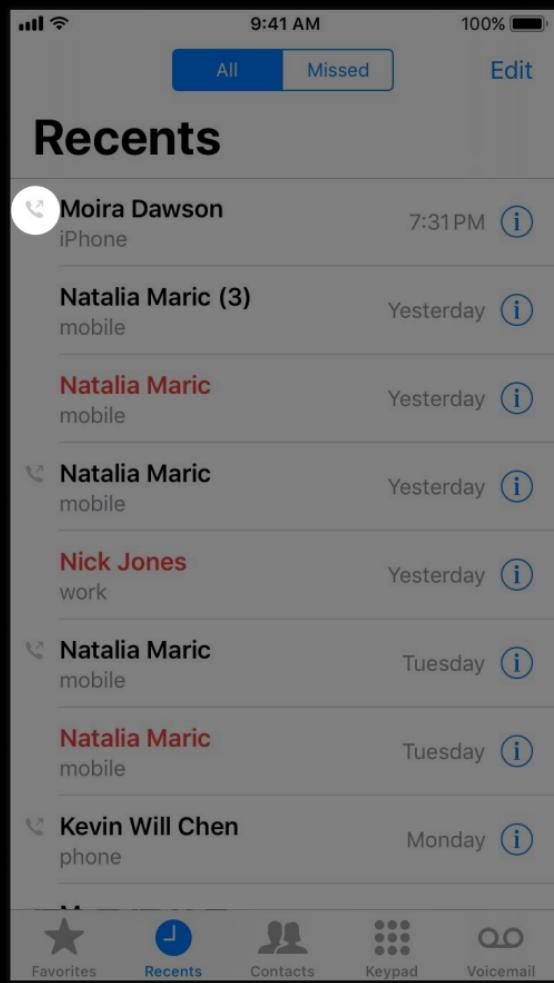
Images

이미지는 텍스트가 아니지만
길을 수 있어야 하는 상황이 많다

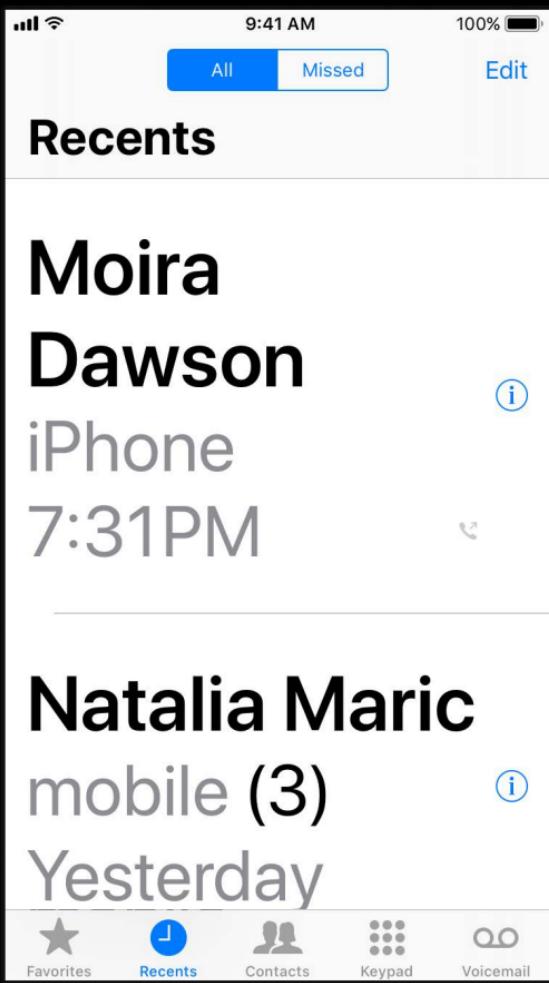
전화 목록.

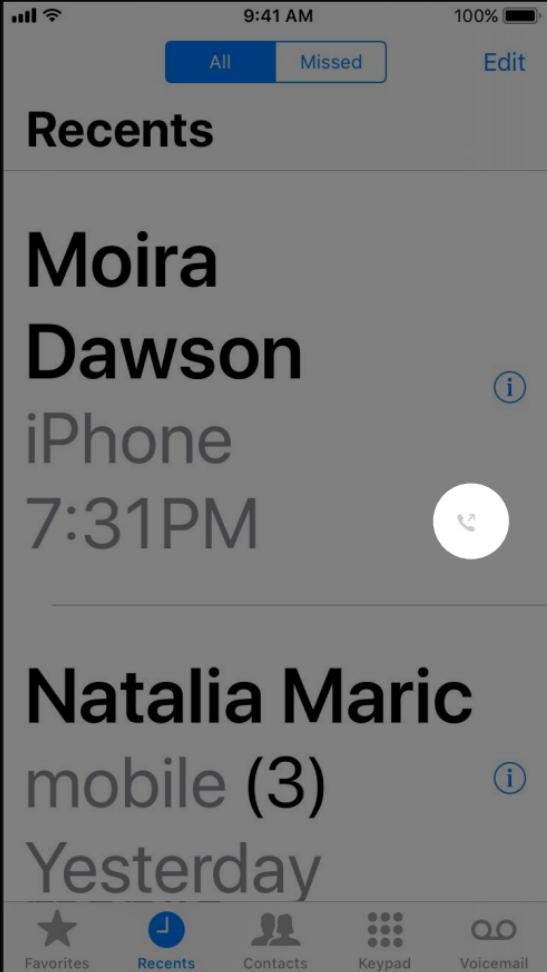


0|0|2|3
동자 여보는 알 수 있다

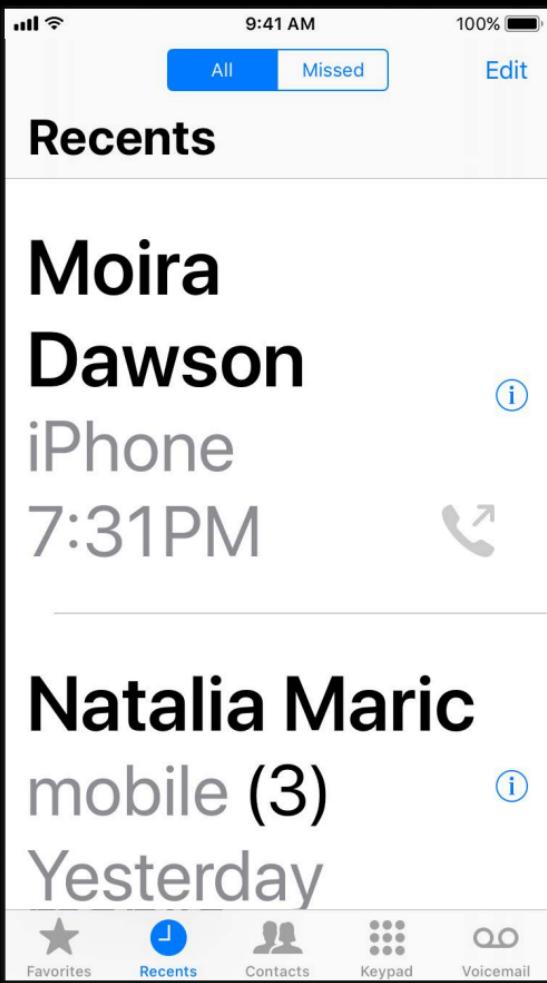


하지만 text 쪽에만 듣기면
이미지는 예상과 같다





이전의 Scale을
늘려야 한다.



Allow Images to Scale Up

NEW

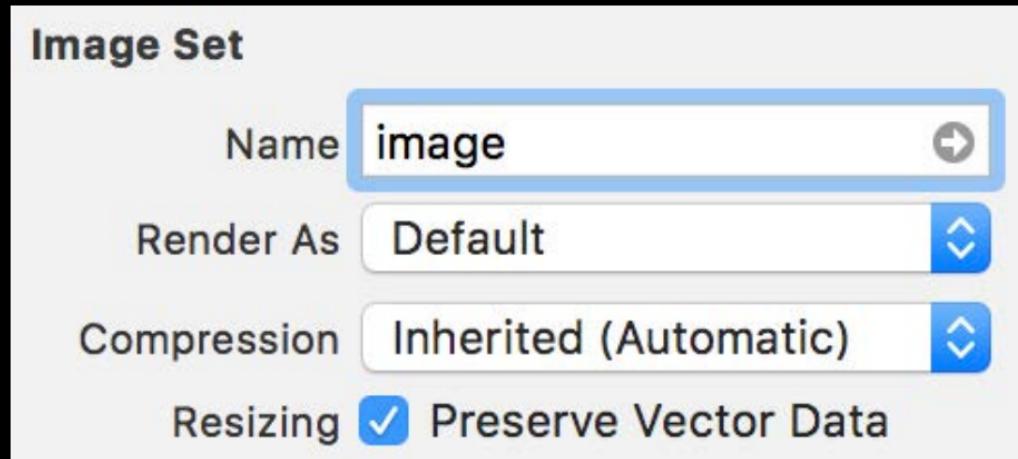
Provide PDF at 1x scale

Vector PDF 는 1x scale 이다.
그러면 크기로 늘려도 안 깨진다.

Allow Images to Scale Up

NEW

Provide PDF at 1x scale



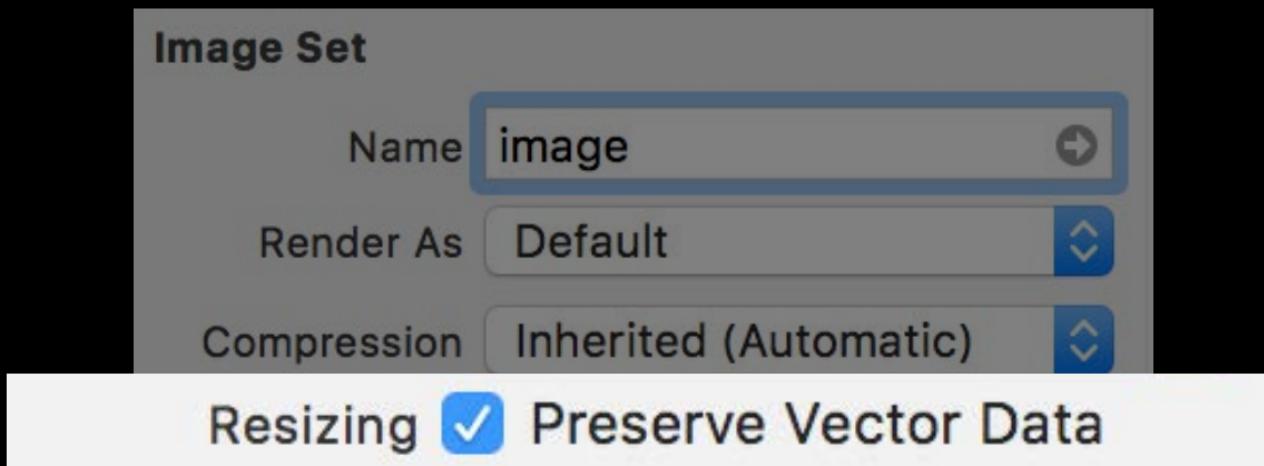
NEW

Allow Images to Scale Up

이제 **제공 안되어 있으면** vector는 레스터화 된다.

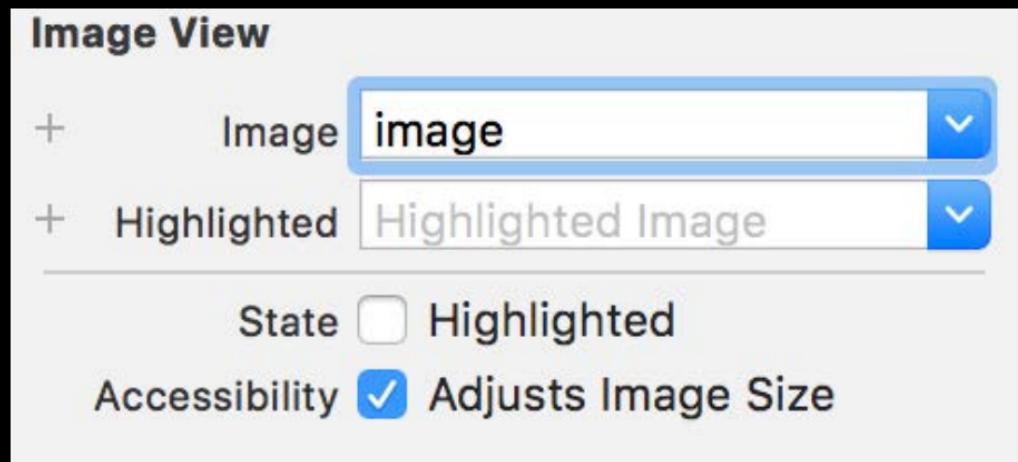
Provide PDF at 1x scale

제공하면 어느 크기는 잘 지원될 것이다.



Allow Images to Scale Up

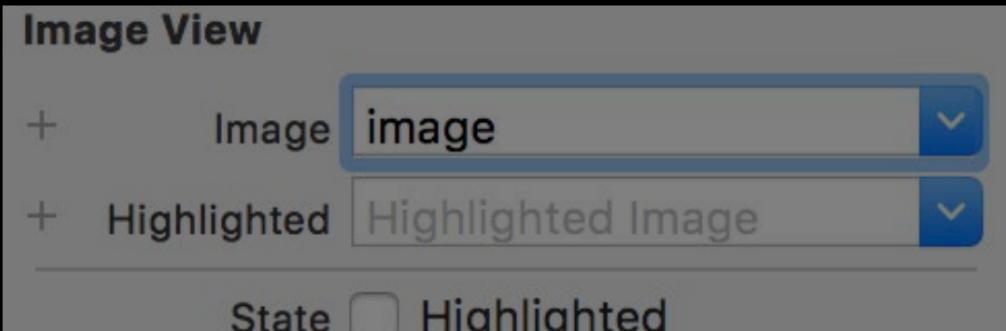
NEW



Allow Images to Scale Up

NEW

그리고 이미지를 탐색할 때
이미지를 확장합니다.



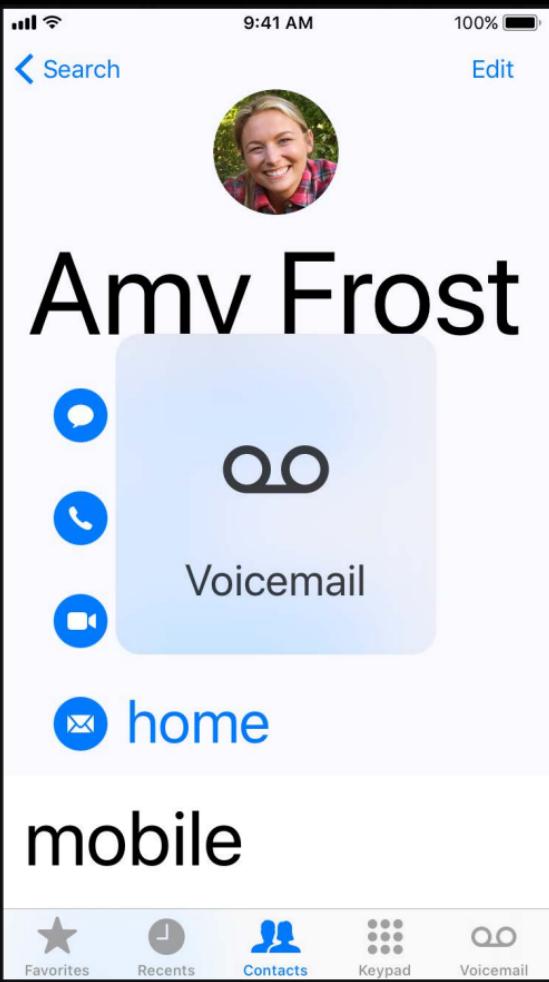
Accessibility Adjusts Image Size

Allow Images to Scale Up

NEW

```
// UIAccessibilityContentSizeCategoryImageAdjusting protocol  
// Scale the image for the 5 largest text sizes (accessibility sizes)  
// Works for UIImageView, UIButton, and NSTextAttachment  
  
imageView.adjustsImageSizeForAccessibilityContentSizeCategory = true
```

bar item은
같이 누른 때
이거나 터치하는 시점에서.



Allow Bar Item Images to Scale Smoothly

NEW

Allow Bar Item Images to Scale Smoothly

NEW

If PDF, use Preserve Vector Data checkbox in asset catalog

Vector 이미지를 쓸 때资产каталог에 벡터로 넣어

Allow Bar Item Images to Scale Smoothly

NEW

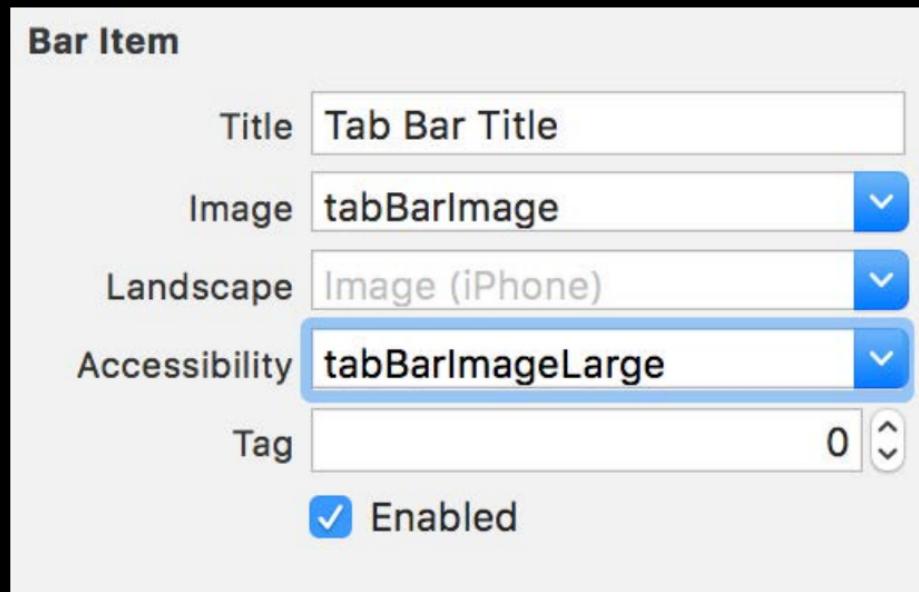
If PDF, use Preserve Vector Data checkbox in asset catalog

If not PDF, provide a larger version (75 x 75 points)

벡터가 아니라면 큰 이미지를 제공하자

Allow Bar Item Images to Scale Smoothly

NEW

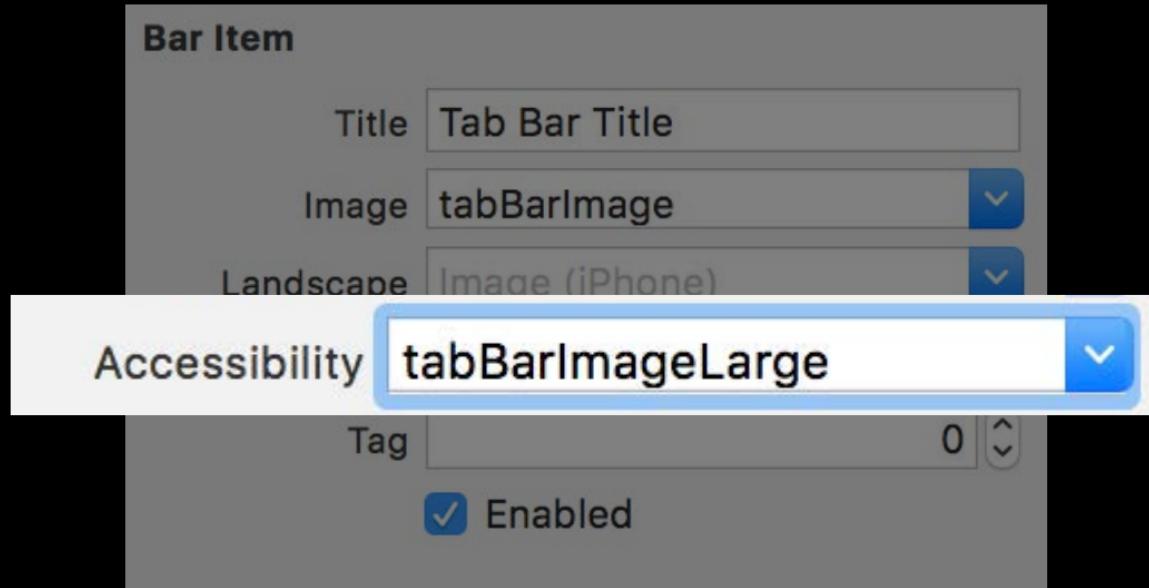


Allow Bar Item Images to Scale Smoothly

NEW

Vector
이미지

TabBar
Image
Large



Allow Bar Item Images to Scale Smoothly

NEW

753 21%

```
// Available on UIBarButtonItem  
barButtonItem.largeContentSizeImage = largerImage
```

Demo

Adapting your app for Dynamic Type

Nandini Sundar, Software Engineer

More Examples in Sample Code

Interface Builder examples

Wrapping text around images

Scrolling when necessary

And more!

Summary

Summary

Easy to support Dynamic Type with iOS 11 API

Summary

Easy to support Dynamic Type with iOS 11 API

Supporting Dynamic Type is good for your users

More Information

<https://developer.apple.com/wwdc17/245>

Related Sessions

Design For Everyone WWDC 2017

What's New in Accessibility WWDC 2017

Media and Gaming Accessibility WWDC 2017

Auto Layout Techniques in Interface Builder WWDC 2017

Labs

Accessibility and Dynamic Type Lab

Technology Lab C

Fri 2:30PM–4:00PM

