Introducing SF Symbols

Paolo Mazzetti, Design Technologies
Tom Adriaenssen, UIKit
Thalia, Bon, Vincenzo...
Brewing tips
Hi Paolo, the trick to a successful brewing lies in having the leaves completely wet before

Lance Wilson, Loïc
Name for Caffè App
This point illustrates that the name also needs to make sense for customers. While dogs do...

Vince, Mark & Patrick
✓ Check out this lasagnal!
Folks, I got it. The action shot clearly shows the layers are even and soggy enough.

Stephen, Tom Adriàenssen
Re: Tea Export Flow
Hey gang, I think this will work fine. I'd suggest changing the Brew button in the export options...

Jamie, Philip, David
✓ Cake section needs more cake
Paolo: I think we need to some “spiffing up” of the cake section of this Today’s page, or per...

Keri
✓ Surfing spots
I was just checking out Santa Cruz to see if the water is too cold. Turns out Half Moon bay is...
Thalia, Bon, Vincenzo...

Brewing tips
Hi Paolo, the trick to a successful brewing lies in having the leaves completely wet before

Lance Wilson, Loïc
Name for Caffè App
This point illustrates that the name also needs to make sense for customers. While dogs do...

Vince, Mark & Patrick
Check out this lasagna!!
Folks, I got it. The action shot clearly shows the layers are even and soggy enough.
<table>
<thead>
<tr>
<th>Location</th>
<th>Photos</th>
</tr>
</thead>
<tbody>
<tr>
<td>San Jose</td>
<td>1,671</td>
</tr>
<tr>
<td>Cupertino</td>
<td>354</td>
</tr>
<tr>
<td>Trip</td>
<td>946</td>
</tr>
<tr>
<td>November</td>
<td>2,169</td>
</tr>
<tr>
<td>September</td>
<td>2,268</td>
</tr>
</tbody>
</table>

7,143 Photos
Today
Done
Design Review

Search

Add To
Edit
- Text Spacing
- Case-Sensitive Layout
- Alternative Stylistic Sets
  - Straight-sided six and nine
  - Open four
  - Vertically centered colon
  - Vertically compact forms
  - High legibility
  - One storey a
  - Soft numerals
  - Calculator
- Circle Symbols
- Square Symbols
- Solid Symbols
- Small Symbols
- Large Symbols
- Lowercase Alignment
- Contextual Alternates
- Lower Case
- Upper Case
Introducing SF Symbols

With over 1,500 configurable symbols, SF Symbols is designed to integrate seamlessly with San Francisco, the system font for Apple platforms. SF Symbols comes in a wide range of weights and scales that automatically align with text labels, and supports Dynamic Type and the Bold Text accessibility feature. You can also design custom symbols with the same design characteristics and accessibility features.

Learn more >

Download SF Symbols beta

Requires macOS 10.14.4 or later
Navigation Bar and Toolbar Icons

Use the following icons in navigation bars and toolbars. For developer guidance, see `UIBarButtonItem`.

**TIP**
You can use text instead of icons to represent items in a navigation bar or toolbar. For example, Calendar uses “Today,” “Calendars,” and “Inbox” in the toolbar. You can also use a fixed space element to provide padding between navigation and toolbar icons.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Meaning</th>
<th>API</th>
</tr>
</thead>
<tbody>
<tr>
<td>📌</td>
<td>Action (Share)</td>
<td>Shows a modal view containing share extensions, action extensions, and tasks, such as Copy, Favorite, or Find, that are useful in the current context.</td>
<td>action</td>
</tr>
<tr>
<td>📜</td>
<td>Add</td>
<td>Creates a new item.</td>
<td>add</td>
</tr>
<tr>
<td>📕</td>
<td>Bookmarks</td>
<td>Shows app-specific bookmarks.</td>
<td>bookmarks</td>
</tr>
<tr>
<td>📷</td>
<td>Camera</td>
<td>Takes a photo or video, or shows the Photo Library.</td>
<td>camera</td>
</tr>
<tr>
<td>🚧</td>
<td>Cancel</td>
<td>Closes the current view or ends edit mode without saving changes.</td>
<td>cancel</td>
</tr>
<tr>
<td>✉️</td>
<td>Compose</td>
<td>Opens a new view in edit mode.</td>
<td>compose</td>
</tr>
<tr>
<td>✅</td>
<td>Done</td>
<td>Saves the state and closes the current view, or exits edit mode.</td>
<td>done</td>
</tr>
<tr>
<td>✏️</td>
<td>Edit</td>
<td>Enters edit mode in the current context.</td>
<td>edit</td>
</tr>
</tbody>
</table>
Today’s special: mint tea with nutty aftertaste
Today’s special: mint tea with

Hot Belgian chocolate with nutty aftertaste
Today's special: mint tea with nutty aftertaste.

Hot Belgian chocolate with nutty aftertaste.
Weight/Scale Variations

Symbols are supported in up to nine weights and three scales.

For optimal layout with text and other symbols, vertically align symbols with the adjacent text.

Side Bearings

Horizontal padding on the left and right side of each symbol can be adjusted by modifying the width of the blue rectangles. Modifications are automatically applied proportionally to all scales and weights.

Exporting

Symbols should be outlined when exporting to ensure the design is preserved when submitting to Xcode.

Template v.0.1

Generated from star.circle

Typeset at 100 points
<table>
<thead>
<tr>
<th>Weight/Scale Variations</th>
<th>Ultralight</th>
<th>Thin</th>
<th>Light</th>
<th>Regular</th>
<th>Medium</th>
<th>Semibold</th>
<th>Bold</th>
<th>Heavy</th>
<th>Black</th>
</tr>
</thead>
</table>

For optimal layout with text and other symbols, vertically align symbols with the adjacent text.

**Side Bearings**

Horizontal padding on the left and right side of each symbol can be adjusted by modifying the width of the blue rectangles. Modifications are automatically applied proportionally to all scales and weights.

**Exporting**

Symbols should be outlined when exporting to ensure the design is preserved when submitting to Xcode.

---

**Template v.0.1**

Generated from star.circle

Typeset at 100 points
Weight/Scale Variations

Symbols are supported in up to nine weights and three scales.

For optimal layout with text and other symbols, vertically align symbols with the adjacent text.

Side Bearings

Horizontal padding on the left and right side of each symbol can be adjusted by modifying the width of the blue rectangles. Modifications are automatically applied proportionally to all scales and weights.

Exporting

Symbols should be outlined when exporting to ensure the design is preserved when submitting to Xcode.
Symbols are supported in up to nine weights and three scales. For optimal layout with text and other symbols, vertically align symbols with the adjacent text. Side bearings, or horizontal padding on the left and right side of each symbol, can be adjusted by modifying the width of the blue rectangles. Modifications are automatically applied proportionally to all scales and weights. Symbols should be outlined when exporting to ensure the design is preserved when submitting to Xcode.
Symbols are supported in up to nine weights and three scales. For optimal layout with text and other symbols, vertically align symbols with the adjacent text.

Horizontal padding on the left and right side of each symbol can be adjusted by modifying the width of the blue rectangles. Modifications are automatically applied proportionally to all scales and weights.

Symbols should be outlined when exporting to ensure the design is preserved when submitting to Xcode.
Design Variations

Symbols are supported in up to nine weights and three scales.

For optimal layout with text and other symbols, vertically align symbols with the adjacent text.

Side Bearings

Horizontal padding on the left and right side of each symbol can be adjusted by modifying the width of the blue rectangles. Modifications are automatically applied proportionally to all scales and weights.

Exporting

Symbols should be outlined when exporting to ensure the design is preserved when submitting to Xcode.
Symbols are supported in up to nine weights and three scales. For optimal layout with text and other symbols, vertically align symbols with the adjacent text.

Horizontal padding on the left and right side of each symbol can be adjusted by modifying the width of the blue rectangles. Modifications are automatically applied proportionally to all scales and weights.

Symbols should be outlined when exporting to ensure the design is preserved when submitting to Xcode.
Summary

New library of symbols to match San Francisco
Vector-based, weights, and scales
Browse using the SF Symbols app
Use them in your specs, reference them by name
Customize them using the SVG template
Building Your App with Symbols
Update speed:
// You're eager to use one of these 1500+ new system symbols
// It's easy!

slowerImageView.image = UIImage(systemName: "tortoise")
fasterImageView.image = UIImage(systemName: "hare")
// You're eager to use one of these 1500+ new system symbols
// It's easy!

slowerImageView.image = UIImage(systemName: "tortoise")
fasterImageView.image = UIImage(systemName: "hare")
Today’s special: mint tea with

Hot Belgian chocolate with nutty aftertaste
Today’s special: mint tea with

Hot Belgian chocolate with nutty aftertaste
// And you want to use the image your designer made?
// It's just like it was:

let cupImage = UIImage(named: "teacup")
// And you want to use the image your designer made?
// It's just like it was:

let cupImage = UIImage(named: "teacup")
Rehydrate!
"envelope"
"envelope"

Non symbol

UIImage(named:)
Symbols are always defined in an asset catalog.
Symbols are always defined in an asset catalog.
Symbols are always defined in an asset catalog.
cupView.image = UIImage(named: "teacup")
iOS 12 or earlier

Rehydrate!
Rehydrate!

iOS 13 or later
Today's special: mint tea with nutty aftertaste

Hot Belgian chocolate with nutty aftertaste
Today’s special: mint tea with

Hot Belgian chocolate with
nutty aftertaste
// Our designer specified to use a large configuration:

let configuration = UIImage.SymbolConfiguration(scale: .large)
imageView.preferredSymbolConfiguration = configuration
// Our designer specified to use a large configuration:

```swift
let configuration = UIImage.SymbolConfiguration(scale: .large)
imageView.preferredSymbolConfiguration = configuration
```
let configuration = UIImage.SymbolConfiguration(scale: .large)
imageView.preferredSymbolConfiguration = configuration
Rehydrate!
Rehydrate!
// Symbols are typographic in nature, and it reflects
// in how you configure them

// typographic points, not screen points
let headlineConfig = UIImage.SymbolConfiguration(pointSize: 23.0f, weight: .medium)

// you can always add a scale
let annotationConfig = UIImage.SymbolConfiguration(pointSize: 17.0f, scale: .small)
symbol point sizes

!=

screen point sizes
Typographical symbol point sizes ≠ screen point sizes

Dimensional
// Symbols are typographic in nature, and it reflects
// in how you configure them

// dynamic type text styles
let footnoteConfig = UIImage.SymbolConfiguration(textStyle: .footnote)

// ... you can always add a scale
let introConfig = UIImage.SymbolConfiguration(textStyle: .body, scale: .large)
/ Symbols are typographic in nature, 
// and it reflects in how you configure them

titleLabel.font = style.titleFont()

let configuration = UIImage.SymbolConfiguration(font: titleLabel.font, scale: .large)
titleImage.preferredSymbolConfiguration = configuration
// Symbols are typographic in nature,
// and it reflects in how you configure them

titleLabel.font = style.titleFont()

let configuration = UIImage.SymbolConfiguration(font: titleLabel.font, scale: .large)
titleImage.preferredSymbolConfiguration = configuration
// It's also possible to combine configurations

let boldLargeConfig = UIImage.SymbolConfiguration(weight: .bold, scale: .large)
let smallConfig = UIImage.SymbolConfiguration(scale: .small)

let config = boldLargeConfig.applying(smallConfig)

// config will be .bold, .small
// It's also possible to combine configurations

let boldLargeConfig = UIImage.SymbolConfiguration(weight: .bold, scale: .large)
let smallConfig = UIImage.SymbolConfiguration(scale: .small)

let config = boldLargeConfig.applying(smallConfig)

// config will be .bold, .small
The noise level is pretty low here.

Jamie there makes the best tea. Period.

I got a free coffee on my birthday :)

I love how they redecorated the place!

Their chai lattes are very, very good.

Power at every table? Perfect for Xcode.
I got a free coffee on my birthday :)

Jamie there makes the best tea. Period.

I love how they redecorated the place!

Their chai lattes are very, very good.

Power at every table? Perfect for Xcode.
I got a free coffee on my birthday :)

I love how they redecorated the place!

Their chai lattes are very, very good.

Power at every table? Perfect for Xcode.

The noise level is pretty low here.

Jamie there makes the best tea. Period.
Great!
I love the baristas here! They are always so friendly and welcoming. Service is fast, and good. Best place!

Good coffee. But a 15 minute wait?!

They allow my dog Bob inside. Nice, uh?

There's a certain stink there. Meh.
You won't even be able to read this.

There's a certain stink there. Meh.

They allow my dog Bob inside. Nice, uh?

Good coffee. But a 15 minute wait?!

There's a jukebox with awesome music.

OMG I LOVE THEIR MUFFINS!!!

I love the baristas here! They are always so friendly and welcoming. Service is fast, and good. Best place!
Really Nice.
// You can inspect the baseline values on images
// Expressed in screen points from the bottom of the image

let chevron = UIImage(systemName: "person.square")

// Not all images have baselines!
if let offset = chevron.baselineOffsetFromBottom {
    // ...
}

// You can inspect the baseline values on images
// Expressed in screen points from the bottom of the image

let chevron = UIImage(systemName: "person.square")

// Not all images have baselines!
if let offset = chevron.baselineOffsetFromBottom {
    // ...
}

CGFloat?
Today:
Today:
Today: ☁️*****
Today: -3.5 +4.5
// You can add baselines to any image
// Important: baseline offsets are in screen points.

let avatarImage: UIImage?

if let loadedImage = AvatarCache.load(forUser: user.id) {
    // give the avatar a baseline
    avatarImage = loadedImage.withBaselineOffsetFromBottom(3.5f)
}
else {
    avatarImage = UIImage(systemName: "person.square")
}
let avatarImage: UIImage?

if let loadedImage = AvatarCache.load(forUser: user.id) {
    // give the avatar a baseline
    avatarImage = loadedImage.withBaselineOffsetFromBottom(3.5f)
} else {
    avatarImage = UIImage(systemName: "person.square")
}
I've been a regular since they started using Apple Pay. It makes ordering *so* much easier. Good job, folks.

If you bring your own cup, you get a discount. I mean, that's such a good idea!

Steve, the guy that used to make iced coffees in their Death Valley branch works here now. They are amazing.

Look up their app in the App Store. Well made, easy to browse and order.

The food is organic, and affordable!
I've been a regular since they started using Apple Pay. It makes ordering *so* much easier. Good job, folks.

If you bring your own cup, you get a discount. I mean, that's such a good idea!

Steve, the guy that used to make iced coffees in their Death Valley branch works here now. They are amazing.

Look up their app in the App Store. Well made, easy to browse and order.

The food is organic, and affordable!
I've been a regular since they started using Apple Pay. It makes ordering *so* much easier. Good job, folks.

If you bring your own cup, you get a discount. I mean, that's such a good idea!

Steve, the guy that used to make iced coffees in their Death Valley branch works here now. They are amazing.

Look up their app in the App Store. Well made, easy to browse and order.

The food is organic, and affordable!
I've been a regular since they started using Apple Pay. It makes ordering *so* much easier. Good job, folks.

If you bring your own cup, you get a discount. I mean, that's such a good idea!

Steve, the guy that used to make iced coffees in their Death Valley branch works here now. They are amazing.

Look up their app in the App Store. Well made, easy to browse and order.

The food is organic, and affordable!
I've been a regular since they started using Apple Pay. It makes ordering *so* much easier. Good job, folks.

If you bring your own cup, you get a discount. I mean, that's such a good idea!

Steve, the guy that used to make iced coffees in their Death Valley branch works here now. They are amazing.

Look up their app in the App Store. Well made, easy to browse and order.

The food is organic, and affordable!
Today’s special: mint tea with nutty aftertaste

Hot Belgian chocolate with nutty aftertaste
Today’s special: mint tea with
Hot Belgian chocolate with nutty aftertaste
They have daily varying teas, including staff favorites and likes.

- Jimmy's Lattes
- Caffe Macs Apple Park
- Fresh Pot
- Cats'n'Caffe
- No-Tea-Bag Teas
- A3 Hot Beverages
- No-Tea-Bag Teas
- Cats'n'Caffe
- Fresh Pot
- Caffe Macs Apple Park
- Jimmy's Lattes

Their chai lattes are very, very good.
They have daily varying teas, including staff favorites and likes.

Jimmy's Lattes

Caffe Macs Apple Park

Fresh Pot

Cats'n'Caffe

No-Tea-Bag Teas

A3 Hot Beverages

Sed pharetra euismod hendrerit.

Their chai lattes are very, very good.
// System buttons have a default configuration of: .body, .large

let watchImage = UIImage(systemName: "stopwatch")

let startButton = UIButton.system(image: watchImage,
    target: self,
    action: #selector(startTimer))
let watchImage = UIImage(systemName: "stopwatch")

let startButton = UIButton.system(image: watchImage,
    target: self,
    action: #selector(startTimer))
Jimmy's Lattes

Caffe Macs Apple Park

Fresh Pot

Cats'n'Caffe

No-Tea-Bag Teas

A3 Hot Beverages

Sed pharetra euismod hendrerit.

Their chai lattes are very, very good.
They have daily varying teas, including staff favorites and likes.
They have daily varying teas, including staff favorites and likes.
Buttons have a default configuration of `.medium`

use `setPreferredSymbolConfiguration()` to change

```swift
let checkButton = UIButton(type: .custom)
checkButton.setImage(UIImage(systemName: "check"), forState: .normal)

let config = UIImage.SymbolConfiguration(textStyle: .caption, scale: .small)
button.setPreferredSymbolConfiguration(config, forState: .normal)
```
// Buttons have a default configuration of .medium
// use setPreferredSymbolConfiguration() to change

let checkButton = UIButton(type: .custom)
checkButton.setImage(UIImage(systemName: "check"), forState: .normal)

let config = UIImage.SymbolConfiguration(textStyle: .caption, scale: .small)
button.setPreferredSymbolConfiguration(config, forState: .normal)
/ Buttons have a default configuration of .medium
// use setPreferredSymbolConfiguration() to change

let checkButton = UIButton(type: .custom)
checkButton.setImage(UIImage(systemName: "check"), forState: .normal)

let config = UIImage.SymbolConfiguration(textStyle: .caption, scale: .small)
button.setPreferredSymbolConfiguration(config, forState: .normal)
Screenshot of "maps" app with two buttons on each screen, one a text button, one a glyph.

Done
Screenshot of "maps" app with two buttons on each screen, one a text button, one a glyph.
Screenshot of "maps" app with 2 buttons on each screen, one a text button, one a glyph.
Image

.renderingMode = .automatic
Non-symbol Image

*automatic* = *alwaysOriginal*

Symbol Image

*automatic* = *alwaysTemplate*
let cupImage = UIImage(named: "teacup")
let receipt = UIGraphicsImageRenderer(size: receiptSize).image {
    // ...
    cupImage.draw(in: CGRect(origin: .zero, size: image.size))  // forced size
    // ...
}
let cupImage = UIImage(named: "teacup")
let receipt = UIGraphicsImageRenderer(size: receiptSize).image {
    // ...
    cupImage.draw(in: CGRect(origin: .zero, size: image.size))  // forced size
    // ...
}
let cupImage = UIImage(named: "teacup")
let receipt = UIGraphicsImageRenderer(size: receiptSize).image {
    // ...
    cupImage.draw(at: .zero) // use natural size of the image
    // ...
}
let cupImage = UIImage(named: "teacup")
let cupImage = UIImage(named: "teacup")
let cupImage = UIImage(named: "teacup")

let configuration = UIImage.SymbolConfiguration(pointSize: 34.0f, weight: .bold)

let largeTeacupImage = cupImage.withConfiguration(configuration)
let cupImage = UIImage(named: "teacup")

let configuration = UIImage.SymbolConfiguration(pointSize: 34.0f, weight: .bold)
let largeTeacupImage = cupImage.withConfiguration(configuration)
let cupImage = UIImage(named: "teacup")

let configuration = UIImage.SymbolConfiguration(pointSize: 34.0f, weight: .bold)
let largeTeacupImage = cupImage.withConfiguration(configuration)
They have daily varying teas, including staff favorites and likes.
They have daily varying teas, including staff favorites and likes.
They have daily varying teas, including staff and.
let string = NSMutableAttributedString(string: "I just symbol images!",
                                    attributes: [.foregroundColor: UIColor.label])

let heartImage = UIImage(systemName: "heart.fill")
let heartAttachment = NSTextAttachment(image: heartImage)
let heartString = NSAttributedString(attachment: heartAttachment)

string.insert(heartString, at: 7)
let string = NSMutableAttributedString(string: "I just symbol images!", attributes: [.foregroundColor: UIColor.label])

let heartImage = UIImage(systemName: "heart.fill")
let heartAttachment = NSTextAttachment(image: heartImage)
let heartString = NSAttributedString(attachment: heartAttachment)

string.insert(heartString, at: 7)
Their chai lattes are very, very good.

They have daily varying teas, including staff and.
Their chai lattes are very, very good.

They have daily varying teas, including staff and.
// Tinting images
// -- apply a color to the whole image

let heartImage = UIImage(systemName: "heart.fill")
let redHeartImage = heartImage.withTintColor(.redColor)
// Tinting images
// -- apply a color to the whole image

let heartImage = UIImage(systemName: "heart.fill")
let redHeartImage = heartImage.withTintColor(.redColor)
// Tinting images
// -- apply a color to the whole image

let heartImage = UIImage(systemName: "heart.fill")
let redHeartImage = heartImage.withTintColor(.redColor, renderingMode: .alwaysOriginal)
// Tinting images
// -- apply a color to the whole image

let heartImage = UIImage(systemName: "heart.fill")
let redHeartImage = heartImage.withTintColor(.redColor, renderingMode: .alwaysOriginal)
Prefer horizontal and vertical centering

Build layouts smallest to largest

Be flexible: image sizes can change

Caching images rarely helps performance

Avoid rasterizing!
<table>
<thead>
<tr>
<th>Event</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessibility Lessons</td>
<td>Tuesday, 9:00</td>
</tr>
<tr>
<td>Creating Great Localized Experiences with Xcode 11</td>
<td>Tuesday, 10:00</td>
</tr>
<tr>
<td>Implementing Dark Mode on iOS</td>
<td>Wednesday, 10:00</td>
</tr>
</tbody>
</table>
## More Information


<table>
<thead>
<tr>
<th>Lab</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text, Fonts, and SF Symbols Lab</td>
<td>Thursday, 12:00</td>
</tr>
<tr>
<td>App Icon and SF Symbols Design Lab</td>
<td>Friday, 1:30</td>
</tr>
</tbody>
</table>