

Foundations

Foundations are the basis for all components and patterns throughout TPL. They provide built-in brand and accessibility standards.



Our Resources

[TPL Figma Library](#)

[TPL Storybook](#)

[iOS-TPL](#)

[TPL GitHub](#)

[Our Figma Team](#)

[Design Systems Jira](#)

We will be releasing more components and patterns on an ongoing basis. [We welcome your feedback](#) on what is currently available and what you'd like to see next.

Accessibility

The New York Times brings truth to the world for readers of all abilities. Written in collaboration with the Inclusive Design Squad.



1-pager: Accessibility opportunities in H1 2024

Make an impact with these four practical ways to meaningfully improve the WCAG conformance of your products.



Sign in to your Google Account

You must sign in to access this content

[Sign in](#)

Overview

All product designers, front-end engineers, and product owners at the Times are expected to adhere to standard accessibility practices. Creating products that account for the following disabilities should be in scope by default:

- Blindness
- Low vision
- Deafness
- Hearing impairments
- Motor impairments
- Temporary/situational impairments

The New York Times conforms to the Web Content Accessibility Guidelines ([WCAG](#)) 2.1 AA standards.

[Times Product Language's Foundations](#) provide accessible tokens and components to build with. Product creators are responsible for understanding the unique needs of their products and practicing the following guidance:

Interaction

Reading Order

With a logical sequence of elements, readers who use assistive technologies ca...

Links and Hit Area

Links need to center around clear and concise language, but you can adjust their ...

Media

Text on Photos and Videos

You can make text on top of complicated backgrounds reliably readable with a few ...

Audio and Video Supplements

Adding additional content preserves the Time's high-quality journalism for all readers.

Alt Text

Writing 1-3 sentences per published image is our digital norm.

Layout

Layout

Readers adapt their devices to their needs in order to experience the Times as best as ...

Resources



A11y Design Office Hours

Mobile A11y Standards

Web A11y Standards

Further Reading

What is Accessibility?

Accessibility is the practice of planning, designing, and coding digital products to provide equivalent experiences to readers with disabilities.

Web, iOS, and Android are designed to afford high quality experiences for people across the wide range of auditory, cognitive, neurological, physical, speech, and visual abilities. While some needs are addressed by platforms automatically (like providing native screen reader software), many crucial considerations must still be made directly by product creators (like specifying a logical [reading order](#) of content on a page).

Accessibility can benefit people without disabilities. For example, considerations made for people with hearing impairments may also benefit people who are aging or are in a loud environment.

Accessibility is also a legal requirement. The Americans with Disabilities Act (ADA) was signed into law on July 26, 1990, securing increased access and opportunity for people with disabilities across community life ([EEOC](#)). The ADA now enforces equal access in the digital sphere, typically utilizing [WCAG](#) as their standard.



This is our 1.0 release of Accessibility Guidelines. We welcome your feedback on what is currently available and what you'd like to see next.



Interaction

Reading Order

With a logical sequence of elements, readers who use assistive technologies can experience all of the Times.

Important Things Earlier

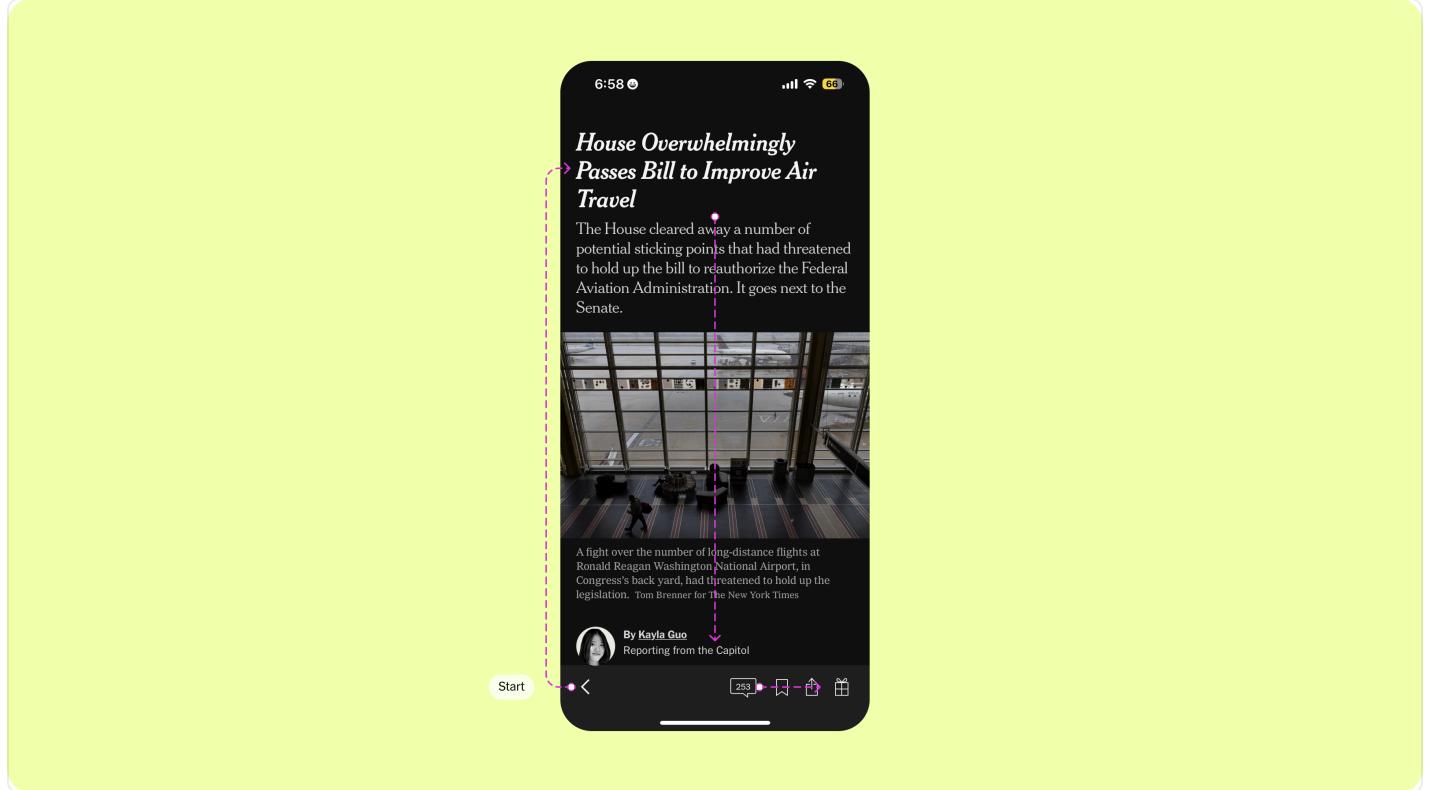
Start with the assumption that English language readers will traverse a screen from left-to-right top-to-bottom—this is how [@keyboard navigation](#) and [@screen readers](#) work by default. Things nearer to the top of the screen are more likely to be encountered and not skipped.

[@wayfinding](#) is almost always the most important thing to offer first; readers should know where they are before engaging with content. For example: Navigation, back buttons, and close buttons are usually the first elements in the reading order. According to your design intentions, place content you wish readers to engage with most next in the reading order.

If beneficial to the visual design, elements may be overridden to appear on the screen in a different location in the reading order than expected; this is common for sticky elements like nav bars.



Use [Autoflow](#), the pencil tool, or any other means in Figma to annotate your desired reading order.



The back button has been overridden to be first in the reading order despite being at the bottom of the screen. The rest of the story content follows.

Heading Hierarchy

Blind readers most commonly scan The New York Times by headings. Because headings typically offer a stronger visual weight they are also ideal for helping low vision readers scan a page in a [layer cake method](#). Headings may still be visually hidden if beneficial to the visual design.

Headings have several levels that can be nested—in digital products these are called H1, H2, H3, and so forth. Be sure to never skip headings levels. Like the index of a book, each heading should relate to the heading level above it. For example, all H3s underneath an H2 called “Opinion” should be the titles of Opinion stories.

Wednesday, July 26, 2023
Today's Paper

U.S. INTERNATIONAL CANADA ESPAÑOL 中文

GIVE THE TIMES Account ▾ Nasdaq -0.12% ▾

The New York Times

LIVE Federal Reserve Meeting 14m ago Hunter Biden Case 27m ago

U.S. Says Main Thrust of Ukraine's Counteroffensive Has Begun

Artillery duels flared along the southern front, as troops trained by NATO countries pressed forward with Western tanks and armored vehicles.

See more headlines

Vladimir Putin's war in Ukraine brought him closer to some African leaders. Now it appears to be pushing them apart.

6 MIN READ

President Biden ordered the U.S. to share evidence of Russian war crimes with the International Criminal Court in The Hague.

4 MIN READ

LIVE 24m ago Judge Puts Hunter Biden Plea on Hold After Questions on Deal's Scope

The judge delayed a decision, demanding that the two sides make changes in the deal clarifying the limits of its broad immunity from prosecution.

See more updates

Fed Raises Rates After a Pause and Leaves Door Open to More

Federal Reserve officials lifted borrowing costs by a quarter-point after pausing in June. They could rise more, but the central bank is not ready to commit.

5 MIN READ

Big consumer companies keep raising prices, complicating the Fed's efforts to tame inflation.

2 MIN READ

Niú York On patrol near the front line.

Tyler Hicks/The New York Times

In Rare Move, Japanese Pop Star Comes Out Publicly as Gay

"I don't want people to struggle like me," said Shinjiro Atae, making an announcement that is extremely unusual in conservative Japan.

6 MIN READ

My friend is mistreating her nanny. Should I say something?

5 MIN READ

A retired star pitcher helps athletes get through the yips.

6 MIN READ

Opinion

THOMAS B. EDSALL
Democrats and Republicans Are Living in Different Worlds

10 MIN READ

ADAM SHINAR
In Israel, the Worst May Be Yet to Come

6 MIN READ

Amir Cohen/Reuters

PETER COY
Companies Aren't Eager to Cut Their Prices as Costs Fall

4 MIN READ

Federal funds target rate

Year	Federal funds target rate (%)
2000	6.0
2001	1.75
2002	1.0
2003	0.25
2004	1.0
2005	5.25
2006	5.5
2007	2.0
2008	0.25
2009	0.0
2010	0.25
2011	0.5
2012	0.75
2013	0.0
2014	0.25
2015	0.5
2016	0.75
2017	1.0
2018	2.0
2019	2.5
2020	0.25
2021	0.5
2022	0.25
2023	5.5

By Karl Russell

0.25-point increase

RECESSIONS

2000 '05 '10 '15 '20

H1: Home page [Visually hidden]

H2: Live Stories [Visually hidden]

- Federal Reserve Meeting
- Hunter Biden Case

H2: Top Stories about Ukraine War [Visually hidden]

- U.S. Says Main Thrust of Ukraine's Counteroffensive Has Begun
- Vladimir Putin's war in Ukraine brought him closer to some African leaders. Now it appears to be pushing them apart.
- President Biden ordered the U.S. to share evidence of Russian war crimes with the International Criminal Court in The Hague.

H2: Top Story about Hunter Biden [Visually hidden]

- Judge Puts Hunter Biden Plea on Hold After Questions on Deal's Scope

H2: Top Stories about U.S. Economy [Visually hidden]

- Fed Raises Rates After a Pause and Leaves Door Open to More

- Big consumer companies keep raising prices, complicating the Fed's efforts to tame inflation.

...

H2: Feature Stories [Visually hidden]

- In Rare Move, Japanese Pop Star Comes Out Publicly as Gay
- My friend is mistreating her nanny. Should I say something?
- A retired star pitcher helps athletes get through the yips.

H2: Opinion

- Democrats and Republicans Are Living in Different Worlds
- Companies Aren't Eager to Cut Their Prices as Costs Fall

...

Example heading hierarchy for the Home page.

Rows and Lists

Sometimes content benefits from [@visual chunking](#) but is either not important enough to belong in the heading hierarchy or causes far too many headings to be in the heading hierarchy (a rule of thumb is 20 headings maximum). This is an ideal situation for rows or lists.

Use rows to present a series of text elements that act as one link. Use bulleted or numbered lists to present a series of non-interactive text elements.

Many rows of Stories, for example, may better be represented as rows of links rather than headings.

Here's what we're covering:

- This is the big test': U.S. officials describe a new Ukrainian effort to sever Russia's hold on the south.
- U.S. officials describe a 3-point rationale for Ukraine's new push.
- Russia launches another barrage of missiles across Ukraine.

Her

Here's what we're covering: </h2>

- "This is the big test": U.S. officials describ nizto sever Russia's hold on the south.
- U.S. officials describe a 3-point rationale ... for Ukraine's new push.
- Russia launches another barrage of ... missiles across Ukraine.

Many rows of Stories may also better be represented as a bulleted list of links rather than headings.

FAQ

What's the difference between reading order and focus/tab order?

Focus/tab order is the sequence in which *interactive elements* (things that can be clicked such as buttons, links, checkboxes, etc) are traversed on a screen. Reading order is the sequence in which *everything* is traversed on a screen—even non-interactive elements.

Focus/tab order should always follow the same direction as reading order. Therefore achieving a good reading order also yields a good focus/tab order.

→ [Links and Hit Area](#)

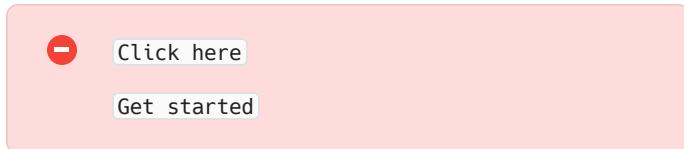
Links and Hit Area

Links need to center around clear and concise language, but you can adjust their hit area as you see fit.

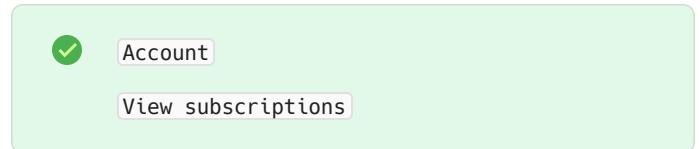
Accessible Names

Links take readers to a different place. In order for the reader to know where they're going, especially screen reader users who commonly navigate by jumping between links, all context must lie within the link's `@accessible name`.

All text inside a link automatically becomes the `@accessible name` and clickable (though this can be adjusted—see the following section to learn more).



These are poor `@accessible names` because they do not offer enough information to the reader.



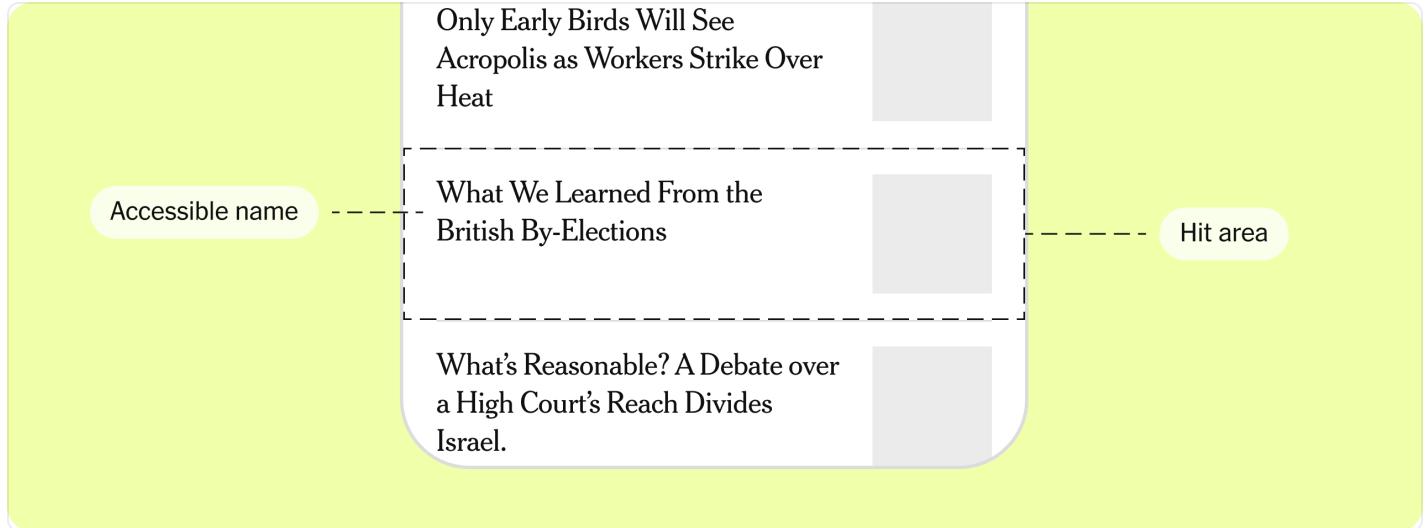
These are good `@accessible names` because they describe a destination or action.

It's also cumbersome for text to be overly long. The headline of a story might make a great link, for example, but including the author, date, and summary is likely too wordy.

Hit Area

By default all of the content inside a link becomes the `@accessible name` and clickable, but sometimes it's beneficial for the `@hit area` to be different.

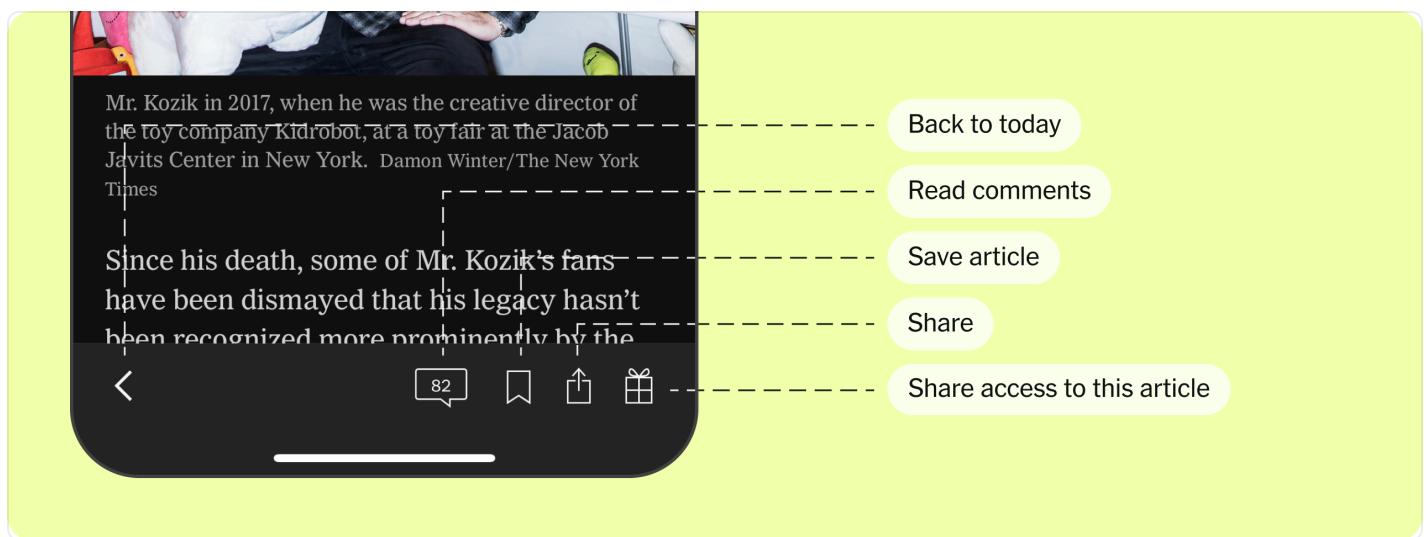
For example, an entire card containing a thumbnail and headline could be interactive but only the headline should be the `@accessible name` (see [Link Box](#)). Be sure to annotate the changed `@hit area` and what should be used as the `@accessible name`.



The headline makes a good [@accessible name](#) while the entire area of the card makes a good [@hit area](#); because the image is decorative, no [alt text](#) is necessary.

Links Without Text

Sometimes a non-text element such as an icon or SVG should act as a link. They should still have a visually hidden [@accessible name](#) that can be read aloud to screen reader users. Make sure to annotate what this [@accessible name](#) should be.



Icons that act as links should always have an [@accessible name](#).

Media

Text on Photos and Videos

You can make text on top of complicated backgrounds reliably readable with a few simple techniques.

Contrast Standards

All text, including text on top of photos, videos, and other media, should meet [WCAG contrast requirements](#):

- **4.5:1** : For regular text less than 24px
- **3:1** : For regular text 24px or larger, or semibold text 19px or larger

This can be challenging because the background is typically less predictable and not a solid color.

Figma Plugin

Text on photos or videos should be tested with the [Contrast Figma plugin](#). The Contrast plugin samples several swatches of the background media and checks if they surpass the required contrast with the foreground text.

The New York Times strives to adhere to WCAG color contrast standards as rigorously as possible. Please aim for at least 80% of the swatches produced by Contrast to pass AA standards.

When in doubt, you can always message #subx-design-systems in Slack for a second opinion.



[Download the Contrast plugin here](#). Make sure to toggle the “Enable Smart Sample for layer” option.



Contrast X

Select Scan

Handgloves

#F8F8F8 11 Samples

Smallest Contrast Ratio	18.3 : 1
Normal Text	AA AAA
Large Text	AA AAA
Graphics	AA

Enable Smart Sample for layer ?

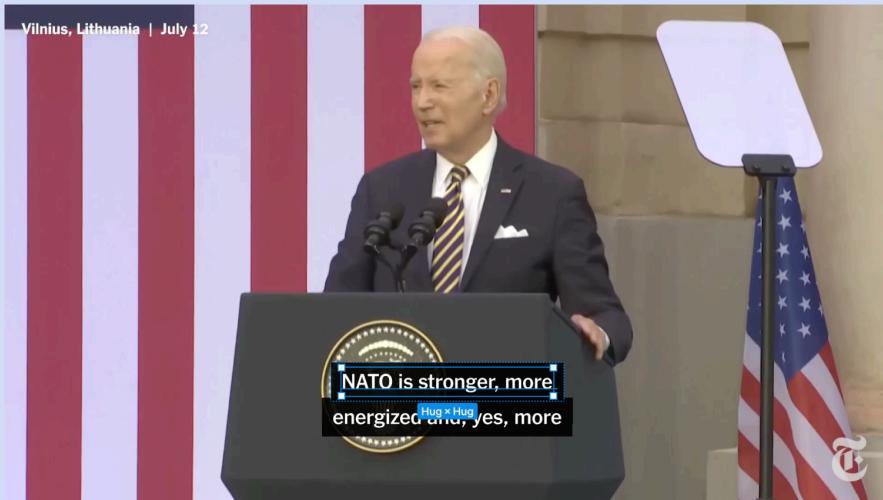
Use APCA algorithm Beta ?

The text "1976" should meet Normal Text AA contrast.

Background Behind Text

Using a solid or semi-opaque background behind the text is an unobtrusive technique to guarantee adequate contrast. It's especially useful for video, whose background may constantly change.

90% is a good starting point for background opacity. 54% is the lowest background opacity when overlaying white text on top of black, and vice versa.



Contrast X

Select Scan

Handgloves

#FFFFFF 32 Samples

Smallest Contrast Ratio	17.44 : 1
Normal Text	AA AAA
Large Text	AA AAA
Graphics	AA

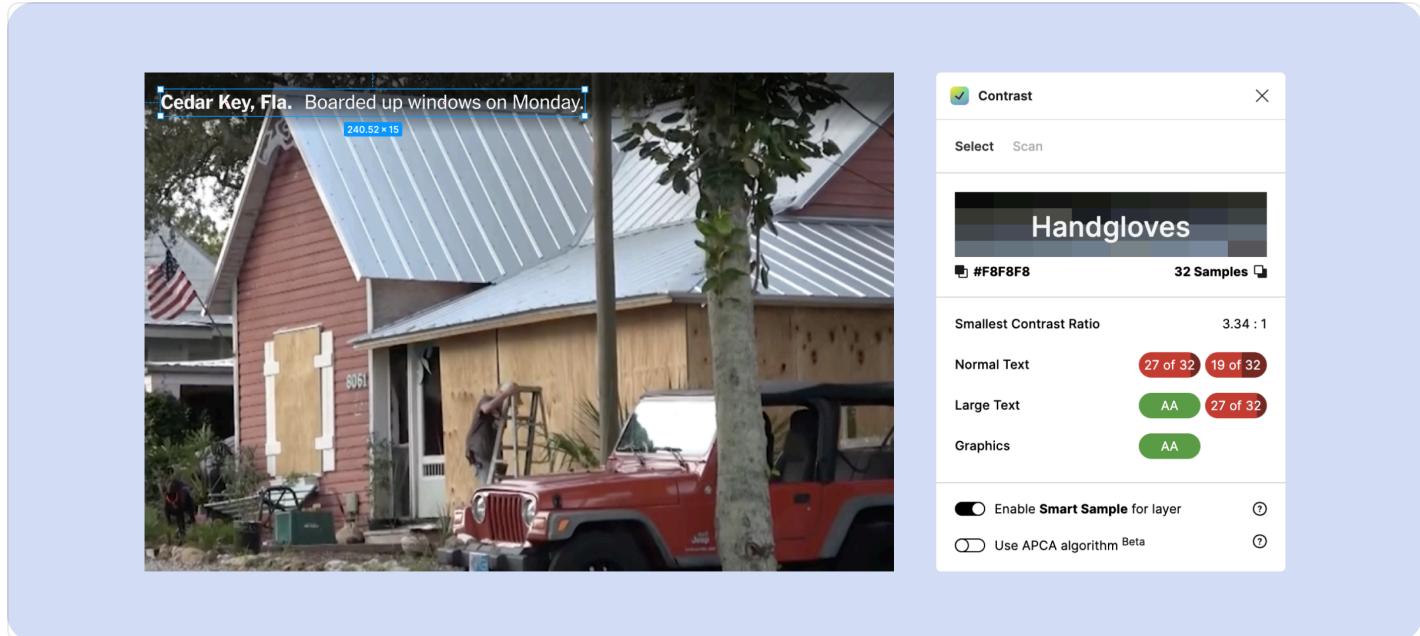
Enable Smart Sample for layer ?

Use APCA algorithm Beta ?

Example with a 90% opacity background behind text.

Gradient Behind Text

Applying a gradient can also improve color contrast with text and add drama to a design. Gradients work especially well with photography.



27 out of 32 (84%) of swatches pass WCAG AA, making it acceptable.

Media with Solid Background

Photos and sometimes videos may naturally feature a portion with a consistent hue. Dark areas in the media are good for white text and vice versa.

The screenshot shows a news article from Wesleyan University's website. The headline reads "Wesleyan University Ends Legacy Admissions". A contrast analysis tool is overlaid on the page, specifically the "Contrast" feature of the "Handgloves" extension. The tool shows the following results:

Category	AA	AAA
Normal Text	AA	AAA
Large Text	AA	AAA
Graphics	AA	

Below the table, there are two settings: "Enable Smart Sample for layer" and "Use APCA algorithm Beta".

Dark grass and recoloring improve the contrast of the white text.

← Links and Hit Area

→ Audio and Video Supplements

Audio and Video Supplements

Adding additional content preserves the Time's high-quality journalism for all readers.

Transcripts

Audio-only and video-only content should be paired with a transcript of both what is said aloud and what is non-verbally communicated. A reader who is deaf, for example, should be able to understand the spoken content and any important context required.

Example: A podcast includes a description of new features in a recent software release. It involves two speakers informally discussing the new and updated features and describing how they are used. One of the speakers works from a list of questions that was used to outline the discussion prior to recording. After the recording is complete, the outline is then edited and supplemented to match the dialogue etc. The resulting transcript is then made available on the speakers Web site along with the audio-only file. The text alternative that identifies the audio only content reads, "Episode 42: Zap Version 12 (text transcript follows)" and the link to the transcript is provided immediately following the audio-only content.

[Credit: G158: W3C]

Captions

All forms of video at the Times, including live video, should offer captions. Captions synchronously convey both dialogue and important sounds through text on top of the video.



Captions are available by clicking the "CC" button.

Audio Description

Prerecorded video should offer audio description that provides context to what's happening in between dialogue. Readers with low vision who aren't able to perceive visual details should still be able to understand the full context of video content.

i Audio description is not required for live video.

Cinemagraphs

When [@cinemagraphs](#) communicate key information, such as a technique for making a recipe or a data chart, make sure the information is also communicated with text. Readers who are blind, for example, should be able to understand the information a [@cinemagraph](#) is expressing.

Not all [@cinemagraphs](#) contain key information. If removing the [@cinemagraph](#) does not harm the information being communicated as a whole, it does not need accompanied text. This logic is similar to deciding when an image needs alt text.

bring to a boil. Turn off heat, cover and let sit for 10-12 minutes. Drain, rinse under cold water or transfer to a bowl of ice and water, then peel. Cool in the refrigerator, loosely covered, for 15 minutes.



Step 2

Halve eggs lengthwise, and carefully scoop

The cooking instructions are a suitable text supplement for this cinemagraph.

FAQ

What's the difference between captions and transcripts?

Similar in form, captions and transcripts serve unique purposes. Captions provide text-equivalents in time—synchronously—with video content while transcripts do not.

- For videos with sound, captions are required while transcripts are optional.
- For audio-only content (e.g. podcasts) and video-only content (e.g. silent movie), transcripts are required while captions are optional.

← [Text on Photos and Videos](#)

→ [Alt Text](#)

Alt Text

Writing 1-3 sentences per published image is our digital norm.

Skipping Alt Text

While alt text is generally useful, it may also produce an unnecessarily wordy experience for readers who use screen readers. If an image, video, or other form of media does not communicate essential information to the reader (e.g. article thumbnails, background images), consider skipping the alt text.

Fernando Botero, Artist of Whimsical Rotundity, Is Dead at 91

His voluptuous figures, both in paintings and in sculpture, portrayed the high and mighty as well as everyday people through a distorting prism.

6 MIN READ



Martin Bureau/Agence France-Presse



The Colombian artist Fernando Botero at his studio in Paris in 2001. Martin Bureau/Agence France-Presse

Fernando Botero, Artist of

No alt text

Alt text required

Thumbnails are typically decorative and don't require alt text. Images that add important context need alt text.

Newsroom's Alt Text Practices

When crafting alt text, please refer to the techniques maintained by the Newsroom:

- For images and pictures: [Best Practices for Writing and Editing Alt Text \(Image Descriptions\)](#)
- For charts and data visualizations: [Graphics Department Accessibility Standards: Alt Text](#)

Icon Buttons

[Icon Buttons](#) always require a label, which acts as the alt text. Make sure the label describes what the Icon Button will do rather than the what the icon is.



Updated 42 minutes ago

Jack Ewing



Copy shareable link

Here's the latest on the auto strike.

President Biden on Friday sided with members of the [United Auto Workers](#), who have walked out of three plants in the Midwest amid a contract dispute over pay, pensions and work hours at the three

"Copy shareable link" is more appropriate than "Two links connected", for example.

← [Audio and Video Supplements](#)

→ [Layout](#)

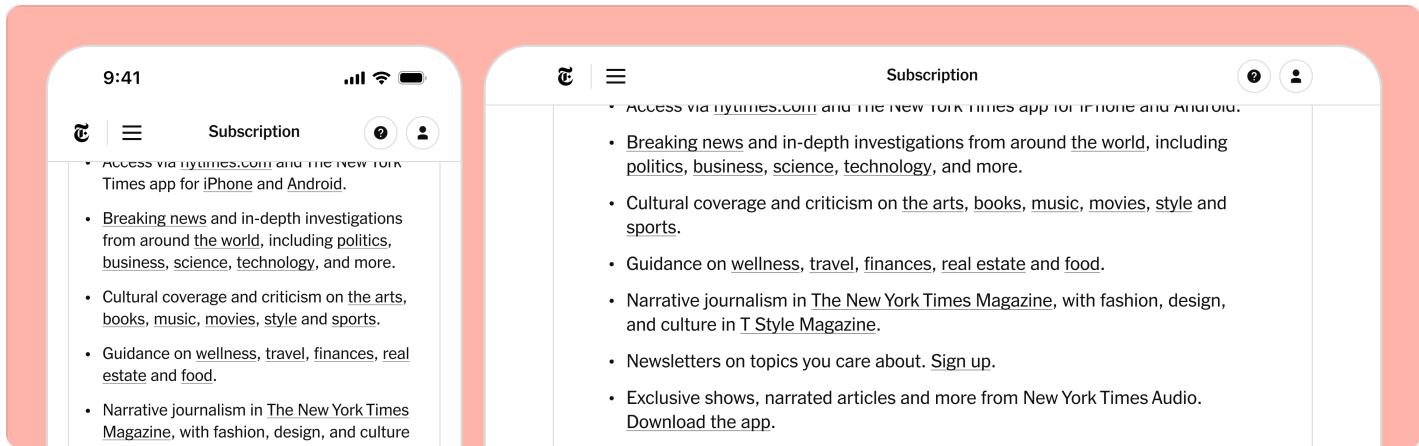


Layout

Readers adapt their devices to their needs in order to experience the Times as best as possible.

Landscape Mode

[Web only] Unless essential to the experience, all screens should work in portrait and landscape orientation. Readers may strongly prefer an orientation due to their mobility or may have their device mounted in a particular orientation.



Text Resizing

For readers who prefer larger text, designers should test their designs at a minimum of 200% of the original font size—or double. This is to ensure that we consider necessary layout changes for readers who scale their font size up to or larger than 200% (Apple's Dynamic Type, for example, can scale up to 310%). All of TPL's [typography tokens](#) are responsive by default.

Here's what we're covering:

- 'This is the big test': U.S. officials describe a new Ukrainian effort to sever Russia's hold on the south.
- U.S. officials describe a 3-point rationale for Ukraine's new push.
- Russia launches another barrage of missiles across Ukraine.

Here's what we're covering:

- 'This is the big test': U.S. officials describe a new

Consider which elements also ought to scale with font size, such as bullet points.

320px Minimum Width

Readers experience the Times on a variety of devices and may also adjust their zoom settings to enlarge their screens. 320px is a good minimum screen width as it supports many phone screens as well as a starting viewport width of 1280 CSS pixels at 400% zoom.

Updated Sept. 15, 2020

When Marilyn Monroe moved to Brentwood in 1962, the Los Angeles neighborhood provided the perfect seclusion for the world's most famous woman.

The four-bedroom Spanish colonial-style house was tucked off a quiet street, with a kidney-shaped pool and towering palm trees. The house was known as "Cursum Perficio," which in Latin loosely translates to "I end the journey."

Six months after she moved in, Ms. Monroe died of a drug overdose in her bedroom. She was 36.

Though her time there was short, the Brentwood home has become a quiet monument to her grand life, with fans still leaving flowers

Updated Sept. 15, 2020

When Marilyn Monroe moved to Brentwood in 1962, the Los Angeles neighborhood provided the perfect seclusion for the world's most famous woman.

320px, 100%

1280px, 400%

Resources



TPL

Mobile A11y Standards

The Mobile Standards group is currently defining accessibility standards specifically for iOS and Android. Contact Zev Eisenberg and Cam Pulsford for more information.



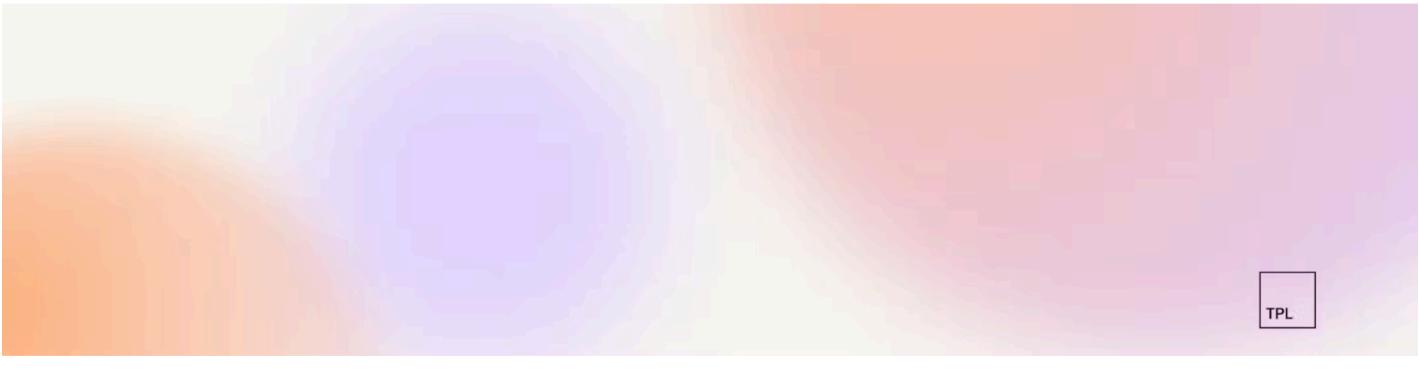
See the [Mobile A11y Standards Document](#)



Sign in to your Google Account

You must sign in to access this content

[Sign in](#)



TPL

Web A11y Standards

The Web Accessibility Checklist was created by the former Accessibility Community of Practice and covers important accessibility considerations specifically for web. Contact David Leininger for more information.



See the [Web Accessibility Checklist](#)



Sign in to your Google Account

You must sign in to access this content

[Sign in](#)

Further Reading

Internal Information

[2022 Accessibility Audit Themes, Fires/Exposure, & Prioritization](#): Summary of the results of our 2022 audit which looked at the NYT homepage and several story pages.

[2019 Accessibility Audit](#): Summary of audit that spans most of the NYT ecosystem of digital products.

[Inclusive Design Squad Audit](#): Internal audit of our a11y and DEI initiatives in product design which contains links to other teams' tactical work.

[Accessibility Team Coda](#)

Internal Guides

[NYT Web Accessibility Checklist](#): The most commonly used guide by engineers and designers at NYT.

[Accessibility Checklist by T Brand](#): How T Brand designers check for accessibility at every stage of their process, starting from the beginning.

[A11y in Practice by T Brand](#): Thorough guide on how/why T Brand designers incorporate accessibility into their process; includes guides on alt text, font resizing, heading hierarchy, and more.

[Graphics Department Accessibility Standards: Alt Text](#): How the Graphics Department decides when, where, and how to write good alt text for images and graphics.

[Voiceover documentation by NYT Audio](#): Rough draft of how the NYT Audio designs and documents screen reader experiences

[Graphics & Accessibility](#): Accessibility tips for creating visual assets.

[How to Write Alt Text](#): In progress guide on writing editorial alt text including examples.

Tools

Voiceover documentation in Figma: Excellent example of how to document the Voiceover experience in an actual design; thank you Jamie Liao and Audio!

- VO documentation for Personal Space
- VO documentation for Initial Onboarding
- VO documentation for Newsletters in the App

Contrast: Figma plugin for calculating color contrast; free to use)

Stark: Comprehensive Figma plugin that helps with common a11y problems; limited tools with free account

Include-Accessibility Annotations: Streamlines + provides education on process of annotating designs for accessibility.

External Learning

Basic Web Accessibility Checklist: by Deque, a simple 2 page PDF on the most important accessibility considerations for digital designers

A11y Project Checklist: Straightforward, tactical checklist for designers and engineers to conform to WCAG AA standards

Fireside chat with Jarvis Sam, former VP of DEI at Nike: Discussion of sympathy vs. empathy, radical candor, and the business tactics of DEI efforts.

A Designer's Guide to Documenting Accessibility & User Interactions: By Stephanie Walter, a comprehensive guide for designers on how/what to annotate for accessibility concerns.

Accessibility Guidance x WCAG 2.2

Accessibility Guidn...				
Principle	Guideline	Success Criteria	Accessibility Guidnace	Notes
▼ Perceivable 29	▼ Text Alternatives	1	1.1.1 Non-text Content (Level A) Links and Hit Area Alt Text	
	▼ Time-based Media	9	1.2.1 Audio-only and Video-only (Prerecorded) (Level A) Audio and Video Supplements	
			1.2.2 Captions (Prerecorded) (Level A) Audio and Video Supplements	
			1.2.3 Audio Description or Media Alternative (Prerecorded) (Level A) Audio and Video Supplements	
			1.2.4 Captions (Live) (Level AA) Audio and Video Supplements	
			1.2.5 Audio Description (Prerecorded) (Level AA) Audio and Video Supplements	
			1.2.6 Sign Language (Prerecorded) (Level AAA) Not mandatory	

		1.2.7 Extended Audio Description (Prerecorded) (Level AAA)	Not mandatory	
		1.2.8 Media Alternative (Prerecorded) (Level AAA)	Not mandatory	
		1.2.9 Audio-only (Live) (Level AAA)	Not mandatory	
▼ Adaptable 6	6	1.3.1 Info and Relationships (Level A)	Reading Order	
		1.3.2 Meaningful Sequence (Level A)	Reading Order	
		1.3.3 Sensory Characteristics (Level A)	Built into TPL Reading Order Links and Hit Area	
		1.3.4 Orientation (Level AA)	Layout	
		1.3.5 Identify Input Purpose (Level AA)	Reading Order Built into TPL	
		1.3.6 Identify Purpose (Level AAA)	Not mandatory	
▼ Distinguishable 13	13	1.4.1 Use of Color (Level A)	Not accounted for Built into TPL	Partially accounted for in Color guidance
		1.4.2 Audio Control (Level A)	Not accounted for Built into TPL	Ought to be built into future component
		1.4.3 Contrast (Minimum) (Level AA)	Text on Photos and Videos Built into TPL	
		1.4.4 Resize text (Level AA)	Built into TPL Layout	
		1.4.5 Images of Text (Level AA)	Reading Order Alt Text	
		1.4.6 Contrast (Enhanced) (Level AAA)	Not mandatory	
		1.4.7 Low or No Background Audio (Level AAA)	Not mandatory	
		1.4.8 Visual Presentation (Level AAA)	Not mandatory	
		1.4.9 Images of Text (No Exception) (Level AAA)	Not mandatory	
		1.4.10 Reflow (Level AA)	Layout	
		1.4.11 Non-text Contrast (Level AA)	Built into TPL	
		1.4.12 Text Spacing (Level AA)	Built into TPL	
		1.4.13 Content on Hover or Focus (Level AA)	Built into TPL Reading Order	Could add more note to Reading Order
▼ Operable 34	▼ Keyboard Accessible 4	2.1.1 Keyboard (Level A)	Built into TPL Reading Order	

	2.1.2 No Keyboard Trap (Level A)	Reading Order	Could add more note to Reading Order
	2.1.3 Keyboard (No Exception) (Level AAA)	Not mandatory	
	2.1.4 Character Key Shortcuts (Level A)	Not accounted for	
▼ Enough Time 6	2.2.1 Timing Adjustable (Level A)	Not accounted for	
	2.2.2 Pause, Stop, Hide (Level A)	Not accounted for Built into TPL	Ought to be in future Motion guidelines
	2.2.3 No Timing (Level AAA)	Not mandatory	
	2.2.4 Interruptions (Level AAA)	Not mandatory	
	2.2.5 Re-authenticating (Level AAA)	Not mandatory	
	2.2.6 Timeouts (Level AAA)	Not mandatory	
▼ Seizures and Physical Reactions 3	2.3.1 Three Flashes or Below Threshold (Level A)	Not accounted for Audio and Video Supplements	Ought to be in future Motion guidelines
	2.3.2 Three Flashes (Level AAA)	Not mandatory	
	2.3.3 Animation from Interactions (Level AAA)	Not mandatory	
▼ Navigable 13	2.4.1 Bypass Blocks (Level A)	Not accounted for Built into TPL	Ought to be built into future component
	2.4.2 Page Titled (Level A)	Reading Order	Could add more note to Reading Order
	2.4.3 Focus Order (Level A)	Reading Order	
	2.4.4 Link Purpose (In Context) (Level A)	Built into TPL Links and Hit Area	
	2.4.5 Multiple Ways (Level AA)	Not accounted for	
	2.4.6 Headings and Labels (Level AA)	Reading Order	
	2.4.7 Focus Visible (Level AA)	Built into TPL	
	2.4.8 Location (Level AAA)	Not mandatory	
	2.4.9 Link Purpose (Link Only) (Level AAA)	Not mandatory	
	2.4.10 Section Headings (Level AAA)	Not mandatory	
	2.4.11 Focus Not Obscured (Minimum) (Level AA)	Built into TPL	

		2.4.12 Focus Not Obscured (Enhanced) (Level AAA)	Not mandatory
		2.4.13 Focus Appearance (Level AAA)	Not mandatory
▼ Input Modalities 8	8	2.5.1 Pointer Gestures (Level A)	Not accounted for
		2.5.2 Pointer Cancellation (Level A)	Built into TPL
		2.5.3 Label in Name (Level A)	Built into TPL
		2.5.4 Motion Actuation (Level A)	Not accounted for
		2.5.5 Target Size (Enhanced) (Level AAA)	Not mandatory
		2.5.6 Concurrent Input Mechanisms (Level AAA)	Not mandatory
		2.5.7 Dragging Movements (Level AA)	Built into TPL
		2.5.8 Target Size (Minimum) (Level AA)	Built into TPL
▼ Understandable 21	6	3.1.1 Language of Page (Level A)	Not accounted for
		3.1.2 Language of Parts (Level AA)	Not accounted for
	6	3.1.3 Unusual Words (Level AAA)	Not mandatory
		3.1.4 Abbreviations (Level AAA)	Not mandatory
		3.1.5 Reading Level (Level AAA)	Not mandatory
		3.1.6 Pronunciation (Level AAA)	Not mandatory
	6	3.2.1 On Focus (Level A)	Built into TPL
		3.2.2 On Input (Level A)	Built into TPL
		3.2.3 Consistent Navigation (Level AA)	Not accounted for
		3.2.4 Consistent Identification (Level AA)	Built into TPL
		3.2.5 Change on Request (Level AAA)	Not mandatory
		3.2.6 Consistent Help (Level A)	Not accounted for
▼ Input Assistance 9	9	3.3.1 Error Identification (Level A)	Built into TPL
		3.3.2 Labels or Instructions (Level A)	Reading Order
		3.3.3 Error Suggestion (Level AA)	Not accounted for
			Ought to be built into future component

		3.3.4 Error Prevention (Legal, Financial, Data) (Level AA)	Not accounted for Built into TPL	Ought to be built into future component
		3.3.5 Help (Level AAA)	Not mandatory	
		3.3.6 Error Prevention (All) (Level AAA)	Not mandatory	
		3.3.7 Redundant Entry (Level A)	Not accounted for Built into TPL	Ought to be built into future component
		3.3.8 Accessible Authentication (Minimum) (Level AA)	Not accounted for	
		3.3.9 Accessible Authentication (Enhanced) (Level AAA)	Not mandatory	
▼ Robust	3	4.1.1 Parsing (Level A)	Built into TPL	

Accessibility Guidance x Apple HIG

Accessibility Guidan...

Guidance	Sub-guidance	Criteria	Accessibility Gi
▼ Interaction 15	▼ Gestures 6	Don't override the platform gestures. People expect gestures that target system features — like swiping down to reveal Notification Center — to work regardless of the app they're using.	Not accounte
		Prefer simplified gestures for common interactions. Complex gestures such as multifinger or multihand gestures, long presses, or gestures that require repeated movements can be challenging for many people. Using the simplest gestures possible improves the experience for everyone who interacts with your app.	Not accounte
		Provide alternative ways to perform gesture-based actions. Include an option for people who may not be able to perform a specific gesture. For example, if people can use a gesture to delete a row in a table, you can also provide a way to delete items through an edit mode or by offering a Delete button in an item detail view.	Not accounte
		When possible, make your app's core functionality accessible through more than one type of physical interaction. For example, Camera on iPhone and iPad lets people take a photo by tapping the onscreen button or by pressing the device's volume down button. In addition to making photo-capture more convenient for everyone, these alternative interactions provide options to people who might have limited grip strength or dexterity.	Not accounte
		If you define custom gestures, be sure to support assistive technologies that give people alternative ways to interact with your app. For example, with Pointer Control, people can use a wrist, index finger, or head-based pointer; with Dwell Control, they can use only their eyes to select and activate objects. One way to support technologies like VoiceOver, Dwell Control, and Switch Control is to implement custom actions; for developer guidance, see UIAccessibilityCustomAction .	Not accounte

		<p>Make drag and drop accessible in your iOS or iPadOS app. When you use the accessibility APIs to identify drag sources and drop targets in your app, assistive technologies can help people drag and drop items. For developer guidance, see accessibilityDragSourceDescriptors and accessibilityDropPointDescriptors.</p>	Not accounte
▼ Buttons and controls	5	<p>Give all controls and interactive elements a hit target that's large enough. For example, on touchscreen devices, a hit target needs to measure at least 44x44 pt; in visionOS, place controls so that their centers are at least 60 pt apart. People with limited mobility need larger hit targets to help them interact with your app. It can be frustrating to interact with too-small controls in any platform, even when people use a pointer.</p> <p>Characterize the accessibility of custom elements. You can use system APIs to tell assistive technologies how a component behaves. For example, using button or NSAccessibilityButton to characterize a view as a button means that VoiceOver speaks the view's description followed by the word <i>button</i>, which tells people that the view behaves like a button.</p> <p>Use a consistent style hierarchy to communicate the relative importance of buttons. In iOS, iPadOS, and tvOS, for example, you can use the visually prominent filled style for the button that performs the most likely action in a view, using less prominent styles — such as gray or plain — for buttons that perform less important actions. (For developer guidance, see UIButton.Configuration.) In visionOS, system-provided buttons generally include a visible background by default. In iOS, iPadOS, visionOS, and for some buttons in macOS, people can also turn on Button Shapes to make it easier to distinguish active buttons from surrounding content.</p> <p>Prefer the system-provided switch component. SwiftUI provides a switch that indicates its state by the position of its knob and its fill color. For some people, however, the addition of labels makes it easier to perceive whether a switch is on or off. When you use system-provided switches, iOS, iPadOS, tvOS, visionOS, and watchOS automatically display on/off glyphs within them when people turn on On/Off Labels.</p> <p>Consider giving links a visual indicator in addition to color, such as an underline. It's fine to use color to identify a link, but if you use it as the only indicator, people — such as those with color blindness or cognitive or situational attention impairments — may not be able to perceive the distinction.</p>	Built into TPL Not accounte
▼ User inputs	3	<p>Let people input information by speaking instead of typing or gesturing. Adding a dictation button in a text entry field lets people choose speech as their preferred input method. If you create a custom keyboard, be sure to include a microphone key for dictation.</p> <p>Support Siri or Shortcuts for performing important tasks by voice alone. To learn more about helping people use Siri interactions in your app, see Siri.</p> <p>When possible, don't prevent people from selecting plain text. Many people rely on using selected text as input for translations and definitions.</p>	Built into TPL Not accounte
▼ Haptics	1	<p>Support the system-defined haptics where available. Many people rely on haptics to help them interact with apps when they can't see the display. For example, system apps play haptics to notify people when a task has succeeded or failed or when an event is about to happen. Be sure to use the system-defined haptics consistently in your app so that you don't confuse people. For guidance, see Playing haptics.</p>	Not accounte

▼ Voiceover	12	▼ Content Descriptions	5	<p>Provide alternative descriptions for all images that convey meaning. If you don't describe the meaningful images in your content, you prevent VoiceOver users from fully experiencing your app. To create a useful description, start by reporting what would be self-explanatory to someone who is able to see the image. Because VoiceOver reads the text surrounding the image and any captions, focus your description on information that's conveyed by the image itself.</p>	Alt Text
				<p>Make infographics fully accessible. Provide a concise description of the infographic that explains what it conveys. If people can interact with the infographic to get more or different information, you need to make these interactions available to VoiceOver users, too. The accessibility APIs provide ways to represent custom interactive elements so that assistive technologies can help people use them.</p>	Reading Order Alt Text
				<p>When an image is purely decorative and isn't intended to communicate anything important, hide it from assistive technologies. Making VoiceOver describe a purely decorative image can waste people's time and add to their cognitive load without providing any benefit.</p>	Alt Text
				<p>Give each page a unique title and provide headings that identify sections in your information hierarchy. When people arrive on a page, the title is the first piece of information they receive from an assistive technology. To help people understand the structure of your app, create a unique title for each page that succinctly describes its contents or purpose. Similarly, people need accurate section headings to help them build a mental map of the information hierarchy of each page.</p>	Reading Order
				<p>Help everyone enjoy your video and audio content. When you provide closed captions, audio descriptions, and transcripts, you can help people benefit from audio and video content in ways that work for them.</p> <p><i>Closed captions</i> give people a textual equivalent for the audible information in a video. You can also use closed captions to provide multiple translations for the same content, letting the system choose the version that matches the device's current settings. Because closed captions aren't always available, it's important to provide subtitles, too.</p> <p><i>Audio descriptions</i> provide a spoken narration of important information that's presented only visually.</p> <p>A <i>transcript</i> provides a complete textual description of a video, covering both audible and visual information, so that people can enjoy the video in different ways.</p> <p>For developer guidance, see Selecting Subtitles and Alternative Audio Tracks.</p>	Audio and Vic Supplements
▼ Navigation	7			<p>Make sure VoiceOver users can navigate to every element. VoiceOver uses accessibility information from UI elements to help people understand the location of each element and what it can do. System-provided UI components include this accessibility information by default, but VoiceOver can't help people discover and use custom elements unless you provide the information. For developer guidance, see Accessibility modifiers.</p>	Reading Order Built into TPL Not accounted

Improve the VoiceOver experience by specifying how elements are grouped, ordered, or linked. Proximity, alignment, and other contextual cues can help sighted people perceive the relationships among visible elements, but these cues don't work well for VoiceOver users. Examine your app for places where relationships among elements are visual only, and describe these relationships to VoiceOver.

For example, the layout below relies on proximity and centering to imply that each phrase is a caption for the image above it. However, if you don't tell VoiceOver that each image needs to be grouped with its phrase, VoiceOver reads, "A large container holding a variety of mangoes. A large container holding many green artichokes. Mangoes come from trees that belong to the genus Mangifera. Artichokes come from a variety of a species of thistle." This happens because VoiceOver reads elements from top to bottom by default. For developer guidance, see [shouldGroupAccessibilityChildren](#) and [accessibilityTitleUIElement](#).

Tell VoiceOver when visible content or layout changes. An unexpected change in content or layout can be very confusing to VoiceOver users, because it means that their mental map of the content is no longer accurate. It's crucial to report visible changes so that VoiceOver and other assistive technologies can help people update their understanding of the content. For developer guidance, see [UIAccessibility.Notification](#) (UIKit) or [NSAccessibility.Notification](#) (AppKit).

Built into TPL

Not accounted for

Help people predict when a control opens a different webpage or app. An unexpected change in context can cause confusion and require people to suddenly rebuild their mental model of the current experience. One way to draw attention to a potential change in context is append an ellipsis to a button's title. Throughout the system, an ellipsis trailing the title is the standard way for a button to communicate that it opens another window or view in which people can complete the action. For example, Mail in iOS and iPadOS appends an ellipsis to the Move Message button, signaling that a separate view opens, listing the destinations people can choose.

Built into TPL

Not accounted for

Provide alternative text labels for all important interface elements.

Alternative text labels aren't visible, but they let VoiceOver audibly describe app elements, making navigation easier for people with visual disabilities. System-provided controls have useful labels by default, but you need to create labels for custom elements. For example, if you create an accessibility element that represents a custom rating button, you might supply the label "Rate."

Links and Hit

Alt Text

Support the VoiceOver rotor when necessary. VoiceOver users can use a control called the *rotor* to navigate a document or webpage by headings, links, or other section types. The rotor can also bring up the braille keyboard. You can help VoiceOver users navigate among related items in your app by identifying these items to the rotor. For developer guidance, see [UIAccessibilityCustomRotor](#) and [NSAccessibilityCustomRotor](#).

Reading Order

In iPadOS, macOS, and visionOS, make sure people can use the keyboard to navigate and interact with all components in your app. Ideally, people can turn on Full Keyboard Access and perform every task in your experience using only the keyboard. In addition to [accessibility keyboard shortcuts](#), the system defines a large number of other [keyboard shortcuts](#) that many people use all the time. To support everyone, it's important to avoid overriding any system-defined keyboard shortcuts in your app. For guidance, see [Keyboards](#).

Not accounted for

In iOS, iPadOS, tvOS, visionOS, and watchOS, use Dynamic Type and test that your app's layout adapts to all font sizes. Dynamic Type lets people pick the font size that works for them. Verify that your design can scale and that both text and glyphs are legible at all font sizes. On iPhone or iPad, for example, turn on Larger Accessibility Text Sizes in Settings > Accessibility > Display & Text Size > Larger Text, and make sure your app remains comfortably readable. You can download the Dynamic Type size tables in [Apple Design Resources](#) for each platform.

Layout
Not accounte
Built into TPL

As font size increases, keep text truncation to a minimum. In general, aim to display as much useful text in the largest accessibility font size as you do in the largest standard font size. Avoid truncating text in scrollable regions unless people can open a separate view to read the rest of the content. You can prevent text truncation in a label by configuring it to use as many lines as needed to display a useful amount of text; for developer guidance, see [numberOfLines](#).

Not accounte
Layout

Consider adjusting layout at large font sizes. When font size increases in a horizontally constrained context, inline items and container boundaries can crowd text, making it less readable. For example, if you display text inline with secondary items — such as glyphs or timestamps — the text has less horizontal space. At large font sizes, an inline layout might cause text to truncate or result in overlapping text and secondary items. In this case, consider using a stacked layout where the text appears above the secondary items. Similarly, multiple columns of text can become less readable at large font sizes because each column constrains horizontal space. In this case, consider reducing the number of columns when font size increases to avoid text truncation and improve overall readability. For developer guidance, see [isAccessibilityCategory](#).

Layout

Increase the size of meaningful interface icons as font size increases. If you use interface icons to communicate important information, make sure they are easy to view at larger font sizes, too. When you use [SF Symbols](#), you get icons that scale automatically with Dynamic Type size changes.

Built into TPL

Maintain a consistent information hierarchy regardless of the current font size. For example, keep primary elements toward the top of a view even when the font size is very large, so that people don't lose track of these elements.

Reading Orde

Prefer regular or heavy font weights in your app. Consider using Regular, Medium, Semibold, or Bold font weights, because they are easier to see. Avoid Ultralight, Thin, and Light font weights, which can be more difficult to see.

Built into TPL

Ensure your app responds correctly and looks good when people turn on bold text. In iOS, iPadOS, tvOS, visionOS, and watchOS, people turn on the bold text accessibility setting to make text and symbols easier to see. In response, your app needs to make all text bolder and give all glyphs an increased stroke weight. The system fonts and SF symbols automatically adjust to the bold text accessibility setting.

Not accounte
Built into TPL

Make sure custom fonts are legible. Custom typefaces can sometimes be difficult to read. Unless your app has a compelling need for a custom font, such as for branding purposes or to create an immersive gaming experience, it's usually best to use the system fonts. If you do use a custom font, make sure it's easy to read, even at small sizes.

Built into TPL

Avoid full text justification. The whitespace created by fully justified text can create patterns that make it difficult for many people to read and focus on the text. Left justification — or right justification in right-to-left languages — provides a framing reference for people with learning and literacy challenges, such as dyslexia.

Built into TPL

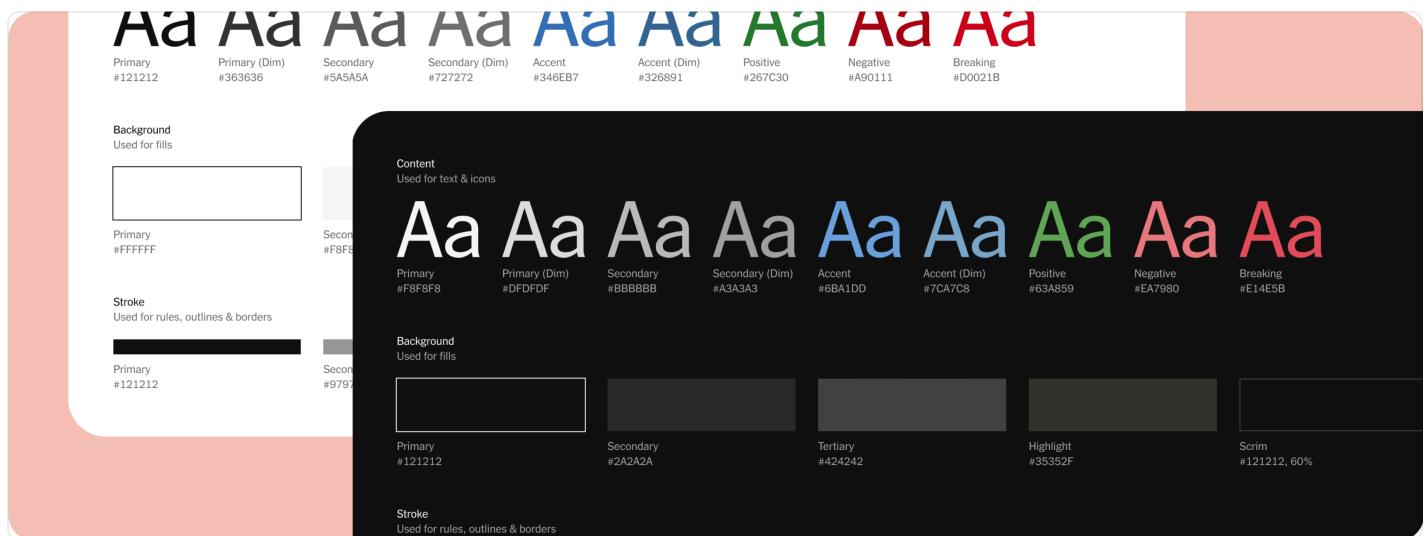
▼ Color and effects	6	6	Don't rely solely on color to differentiate between objects or communicate important information. If you use color to convey information, be sure to provide text labels or glyph shapes to help everyone perceive it.	Built into TPL
			Prefer system colors for text. When you use system colors in text, it responds correctly to accessibility settings such as Invert Colors and Increase Contrast.	Built into TPL
			Avoid using color combinations as the only way to distinguish between two states or values. Many colorblind people find it difficult to distinguish blue from orange; other problematic combinations are red and green, red and black, and either red or green combined with gray. When it makes sense to use a combination of colors to communicate states or values, include additional visual indicators so everyone can perceive the information. For example, instead of using red and green circles to indicate offline and online, you could use a red square and a green circle. Some image-editing software includes tools that can help you proof for colorblindness.	Built into TPL
			Ensure your views respond correctly to Invert Colors. People can turn on Invert Colors when they prefer to view items on a dark background. In the Smart Invert mode of Invert Colors, images, video, and full-color icons (such as app icons and nontemplate images) don't invert, and dark UI stays dark. Test your app or game to find places where you might need to prevent an image — like a photo in a custom view — from inverting.	Built into TPL Not accounte
			Use strongly contrasting colors to improve readability. Many factors affect the perception of color, including font size and weight, color brightness, screen resolution, and lighting conditions. When you increase color contrast of visual elements like text, glyphs, and controls, you can help more people use your app in more situations. To find out if the contrast of adjacent colors in your UI meets minimum acceptable levels, you can use Xcode's Accessibility Inspector or an online color calculator based on the Web Content Accessibility Guidelines (WCAG) color contrast formula. In general, smaller or lighter-weight text needs to have greater contrast to be legible. Use the following values for guidance.	Built into TPL
			Change blurring and transparency when people turn on Reduce Transparency. For example, make areas of blurred content and translucency mostly opaque. For best results, use a color value in the opaque area that's different from the original color value you used when the area was blurred or translucent.	Not accounte
▼ Motion	4	4	Avoid requiring animations unless they're essential for your experience. In general, let people use your app without relying on any animations.	Not accounte Built into TPL
			Play tightened animations when Reduce Motion is on. People can turn on Reduce Motion if they tend to get distracted or experience dizziness or nausea when viewing animations that include effects such as zooming, scaling, spinning, or peripheral motion. In response to this setting, you need to turn off or reduce animations that are known to cause problems (to learn more, see Responsive design for motion). If you use a problematic animation to communicate important information, consider designing a non animated alternative or tightening the physics of the animation to reduce its motion. For example: <ul style="list-style-type: none"> Tighten springs to reduce bounce effects or track 1:1 as a person gestures 	Not accounte Built into TPL



Color

Overview

Color is used to communicate meaning and provide distinction between UI elements. Our system prioritizes accessibility for our readers.



Color values are categorized based on utility: **Content**, **Stroke**, and **Background**. These palettes have been tested to ensure they meet accessibility requirements.

Figma	Web	iOS	Android
<input checked="" type="checkbox"/> Color Styles	<input checked="" type="checkbox"/> JS / CSS-in-JS	<input checked="" type="checkbox"/> SwiftUI Extensions	<input checked="" type="checkbox"/> Kotlin
<input checked="" type="checkbox"/> Color Variables		<input checked="" type="checkbox"/> UIKit Color Extension	<input checked="" type="checkbox"/> XML Resources

Learn more about [using colors as Figma variables](#).

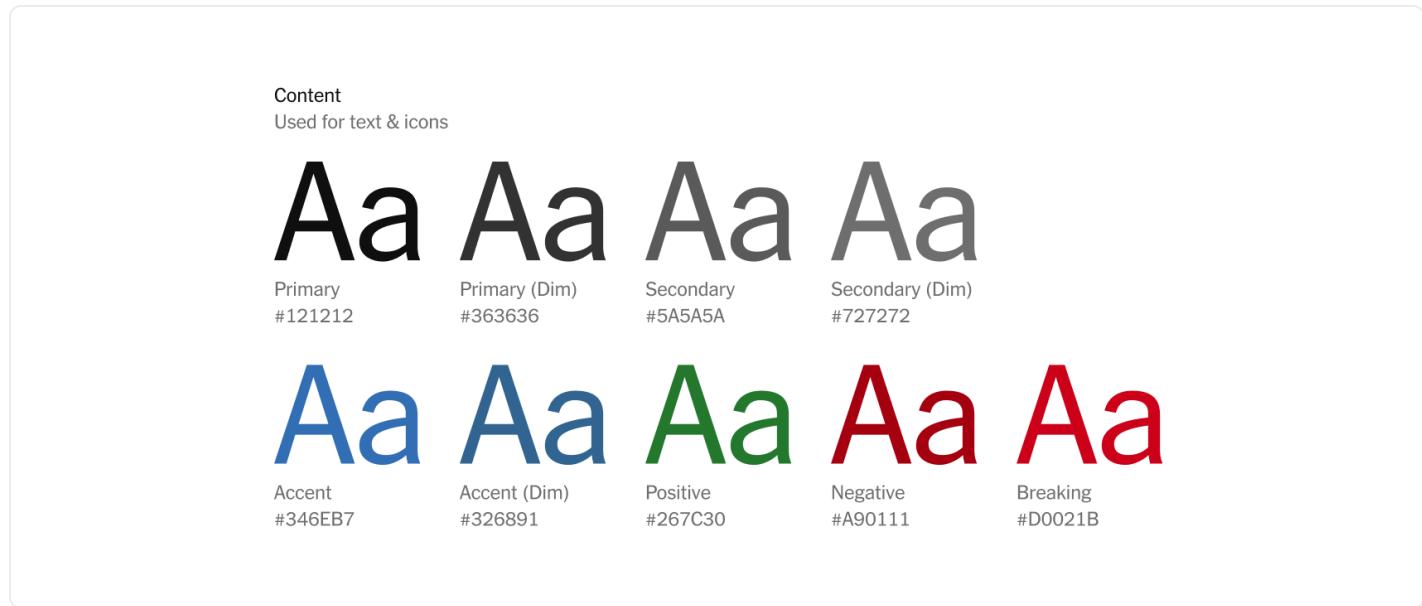
[See in Figma ↗](#)

[See in Storybook ↗](#)

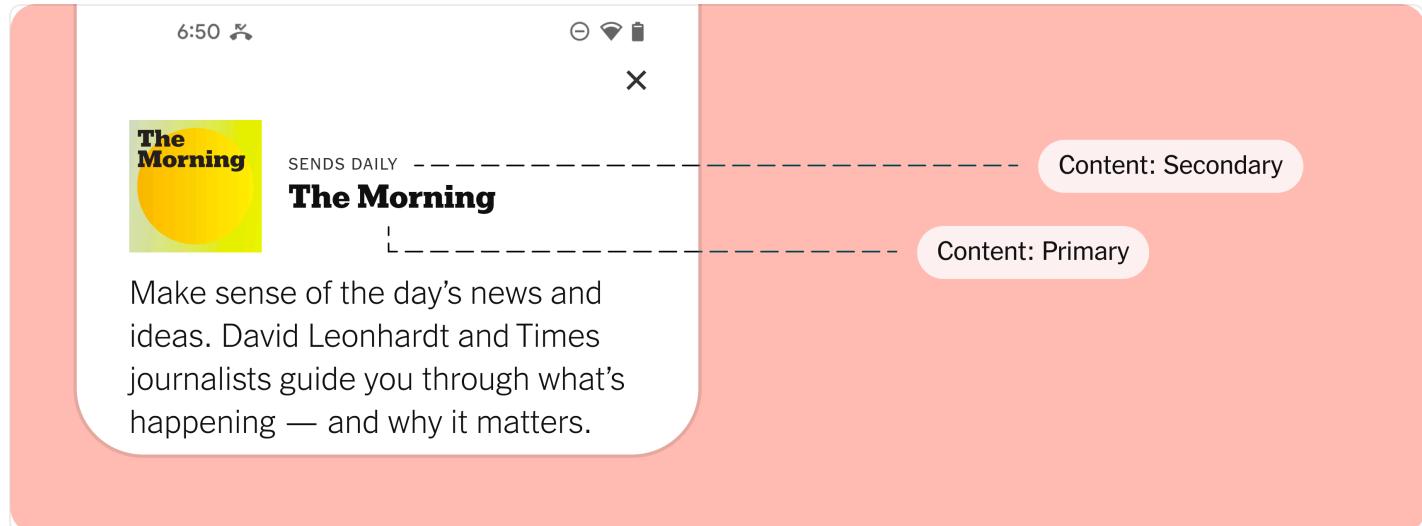
Guidance

For text and icons

The Content palette is used when applying color to **text** and **icons**. To ensure accessibility, use these colors with our [background](#) palette colors.



Content Primary is our default text color and should be used with **Content Secondary** to reinforce hierarchy.



For inline links we recommend using **Content Primary**.

If you are looking for a more eye-catching link color, **Content Accent** and **Content Accent (Dim)** are also available.

Content/Primary

... since 2011, Michael has tracked his steps with a Fitbit. He travels all over the world, including and Reykjavik. He even has a truck named after him in Iceland! He combines walking and jogging with trash.

We chatted about his walk and the lengths he'll go to maintain his daily goal of 20,000 steps (at least).

For strokes

Stroke

Used for rules, outlines & borders



Primary
#121212



Secondary
#979797



Tertiary
#DFDFDF

The stroke palette is used when applying color to strokes on components like **Rule**, for example. **Primary**, **Secondary**, and **Tertiary** are used to control their visual prominence.

A screenshot of a mobile application interface demonstrating stroke colors. The interface includes a top navigation bar with three colored segments (blue, light blue, yellow), a central content area with text, and a right sidebar with a form and buttons. Labels at the bottom indicate the stroke colors used for different elements: 'Stroke: Tertiary' for the bottom-most red bar, 'Stroke: Primary' for the white button with a black border, and 'Stroke: Secondary' for the grey button with a white border.

Play in The New York Times Crossword app >

Skip

app
powered.

We can text you a link to The New York Times app

+1 ▾ Enter phone number

Send download link

We will not save your phone number and it will not be used for anything other than this one-time text. Message and data rates

Stroke: Tertiary

Stroke: Primary

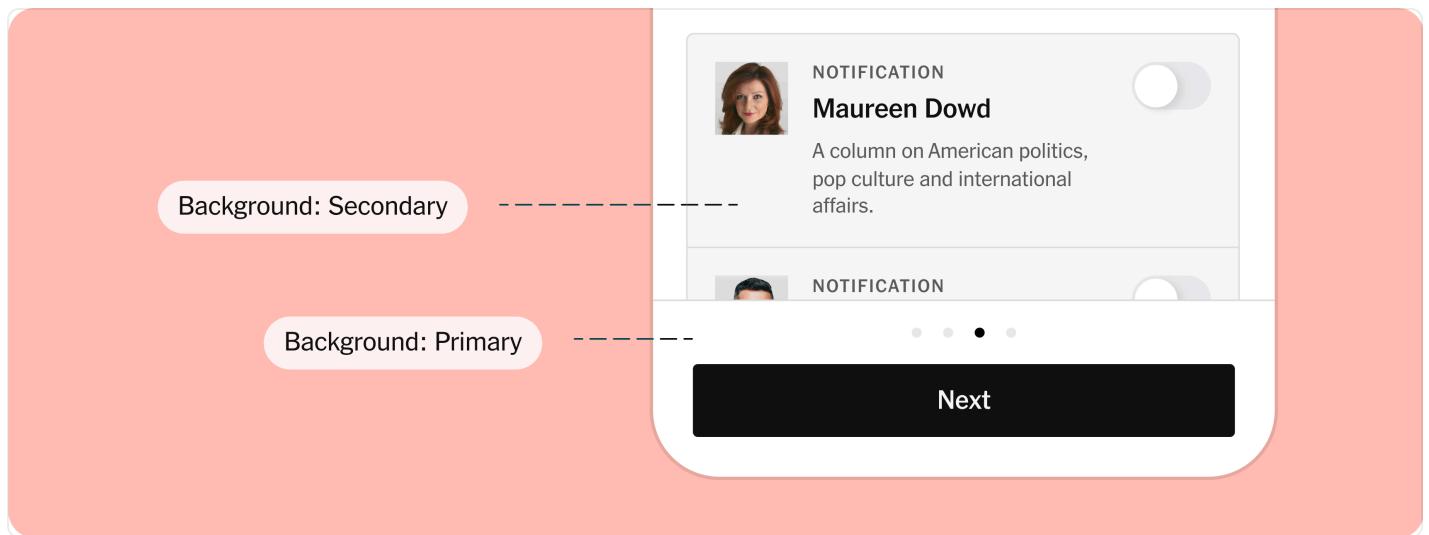
Stroke: Secondary

For backgrounds

The background palette is used when applying a background to a page or component. All our Content colors pass 4.5:1 contrast requirements on **Background Primary** and **Background Secondary**.



Background Primary is our default UI background color.



Background Secondary is used to offset and distinguish an element from Background Primary. **Background Tertiary** is reserved for rare cases and should be used with caution as it does not pass WCAG 2.2 with all content colors. If you are unsure or have questions, [please reach out](#).

Background Highlight is used to highlight text. We've seen this in stories and are curious whether it might be useful in other contexts. **Background Scrim** is semitransparent and intended for overlays and other interactive indicators.

The image shows a mobile screen with a dark header bar. In the top left corner, there's a "Live Chat" button with three dots. Next to it is a row of small circular profile pictures. Below the header, a user profile picture of a woman is shown with the text "1m ago" and "Sydney Ember covering politics". The main content area contains a text message: "Would she put any limits on abortion" followed by "Warren says we can't count on the courts to protect women's reproductive rights." and "Booker now gets a question on". At the bottom of this section, there's a dashed blue line indicating elevation. To the right of this section, there's a news article snippet with the headline "against the airman accused of leaking Pentagon documents." and a "3 MIN READ" link. Below the article is a dark, textured image of what appears to be pine needles or twigs. A white rectangular button labeled "Reply C" is overlaid on this image. Another dashed blue line is visible here, indicating further elevation.

Background: Highlight

Background: Scrim

For elevated backgrounds

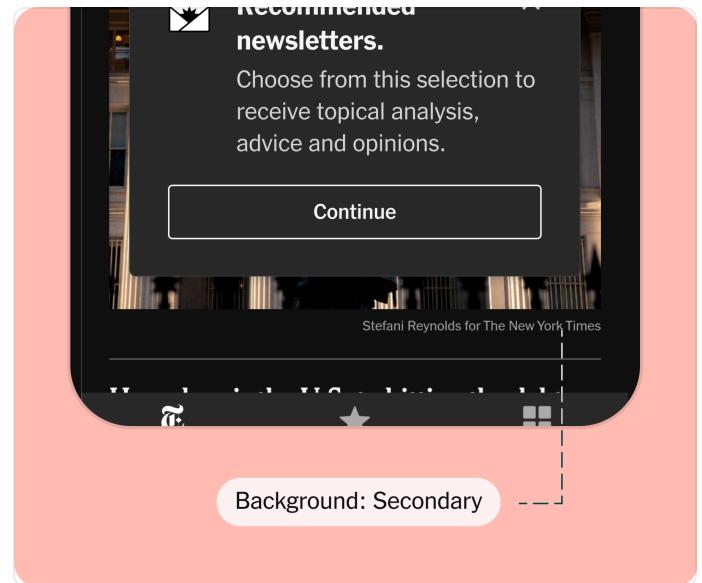
Although we don't yet have comprehensive elevation guidance, TPL can help with elevated backgrounds.

When using the Light theme we recommend **Background Primary** for elevated backgrounds. A drop shadow or Background Scrim can be used to signal elevation further.

The image shows a mobile screen with a light-colored header bar. Below it, there's a section titled "Recommended newsletters." featuring a small thumbnail image and some text. A white rectangular button labeled "Continue" is overlaid on this section. A dashed blue line is at the bottom of this section, indicating elevation. Below this, there's a horizontal navigation bar with several icons. A dashed blue line is also present at the bottom of this bar, indicating further elevation.

Background: Primary

When using the Dark theme, we recommend using **Background Secondary** to indicate elevation. Drop shadows and Background Scrim are still appropriate, but less noticeable.



Testing color contrast

Although our system provides colors that pass contrast requirements, how you use them ultimately determines whether your interface is accessible. Ensure you are using accessible colors using the many resources available online that can be used to ensure that they are being used correctly.

- [WebAIM Contrast Checker](#): A website where you can input specific foreground and background color values to check contrast ratio compliance.
- [A11y - Color Contrast Checker](#): A Figma plug-in that can test colors within your files.
- Storybook's [A11y addon](#) will flag any color contrast violations as well as any other markup issues it can detect.

The WebAIM Contrast Checker screenshot shows the 'Contrast Checker' section with a 'Foreground Color' set to #0000FF and a 'Background Color' set to #FFFFFF. The 'Contrast Ratio' is displayed as 8.59:1. The A11y - Color Contrast Checker Figma plugin screenshot shows a Figma workspace with four blue squares containing white letter 'A's. A floating panel titled 'Color Contrast' displays contrast ratios for various text and background combinations, with some failing to meet WCAG AA standards (indicated by red X's).

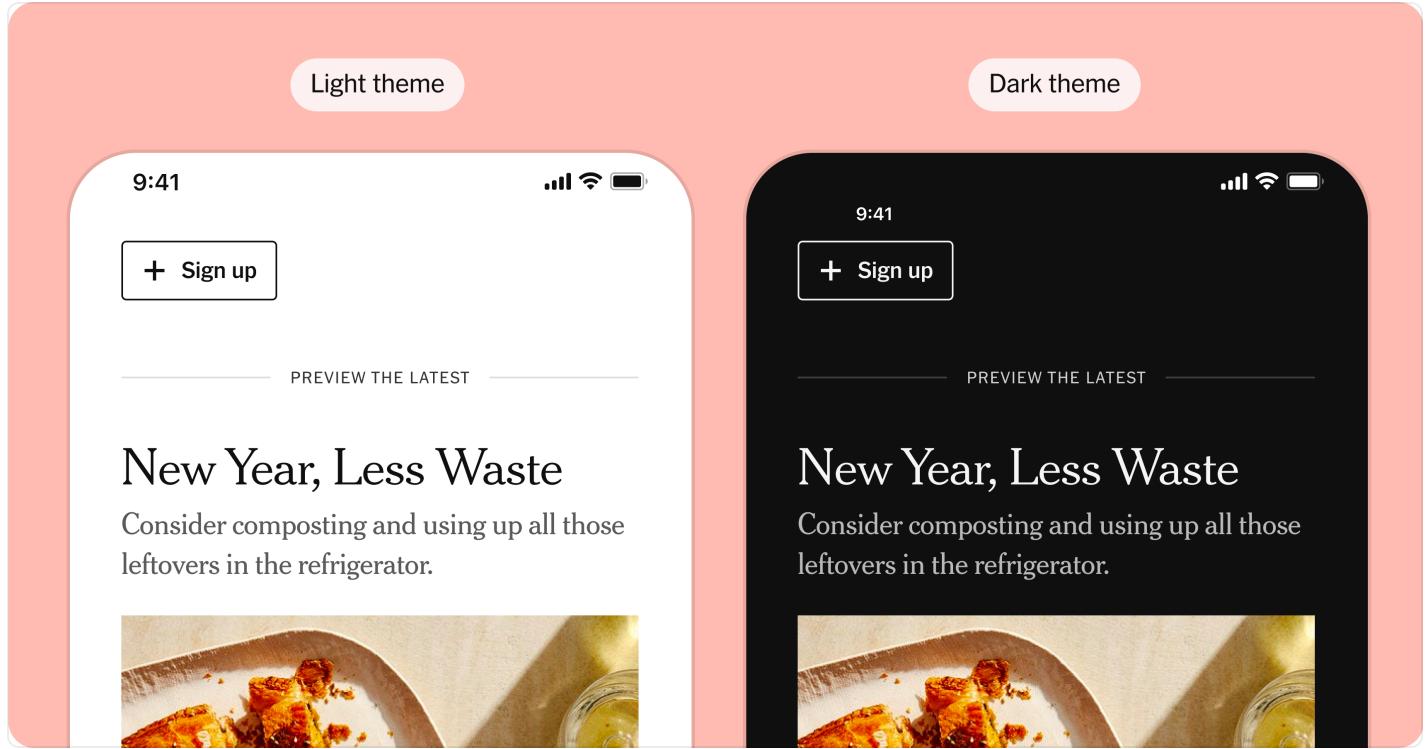
Dark mode

To understand how to control the light or dark appearance of any element, component or page, you need to start with what we call [@Color themes](#), [@ Display modes](#), and [@Color behaviors](#).

@Color theme

Our **Light** and **Dark themes** allows you to design and build for Dark mode.

- Our Light theme colors offer **dark** content and **light** backgrounds (positive contrast polarity).
- Our Dark theme colors offer **light** content and **dark** background colors (negative contrast polarity).



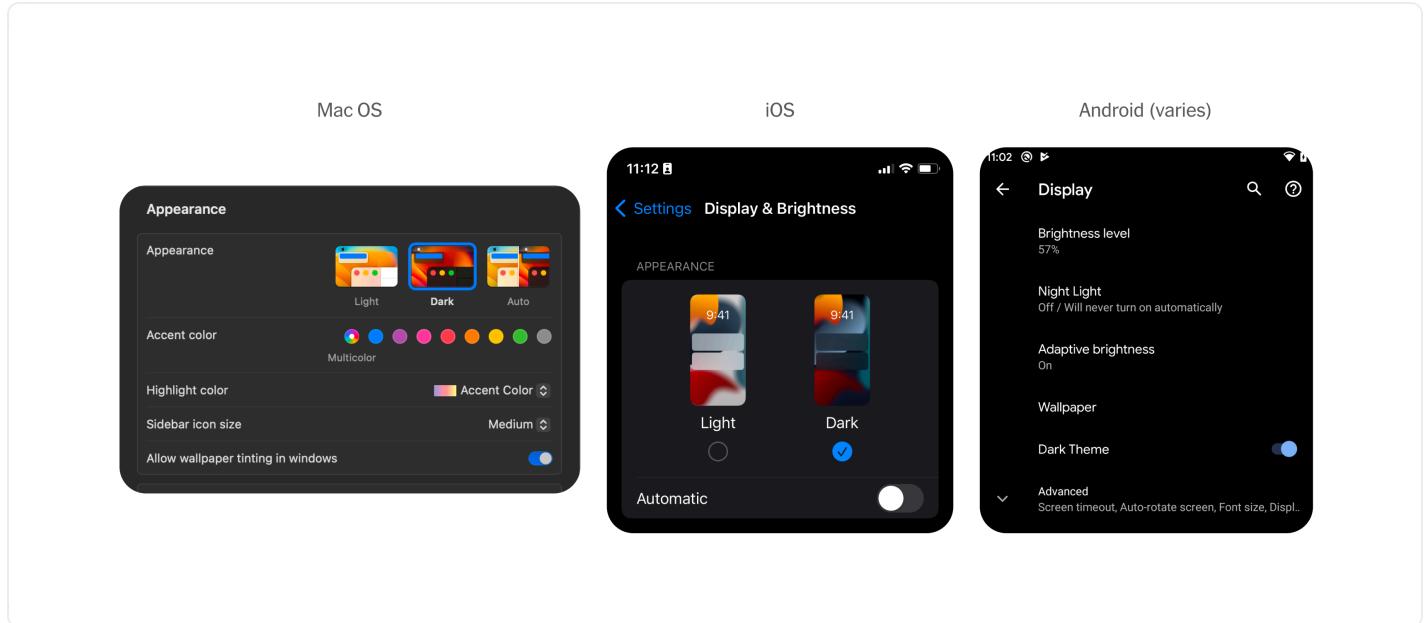
There are two color values associated with every semantic color, one in each theme.

@ Display mode

Our system references device and user settings to determine whether a page, screen or component will appear in **Light** or **Dark mode**.

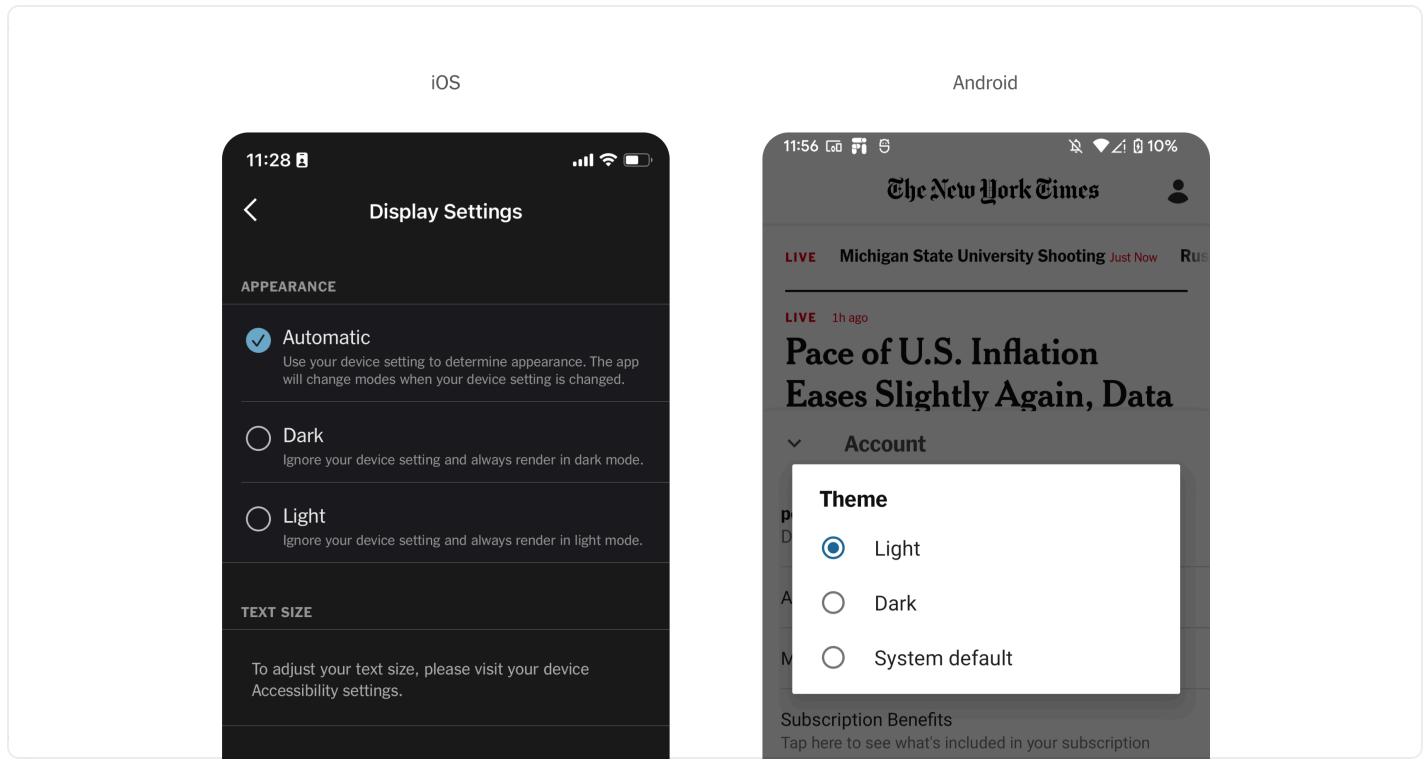
Device setting

Most mobile and desktop devices allow you to set a preference for Light or Dark mode, also called appearance or theme on some devices. In the NYTimes iOS and Android apps, our colors respond to a reader's device setting by default.



App setting

Our apps allow our readers to set a per-app preference, which overrides their device setting.



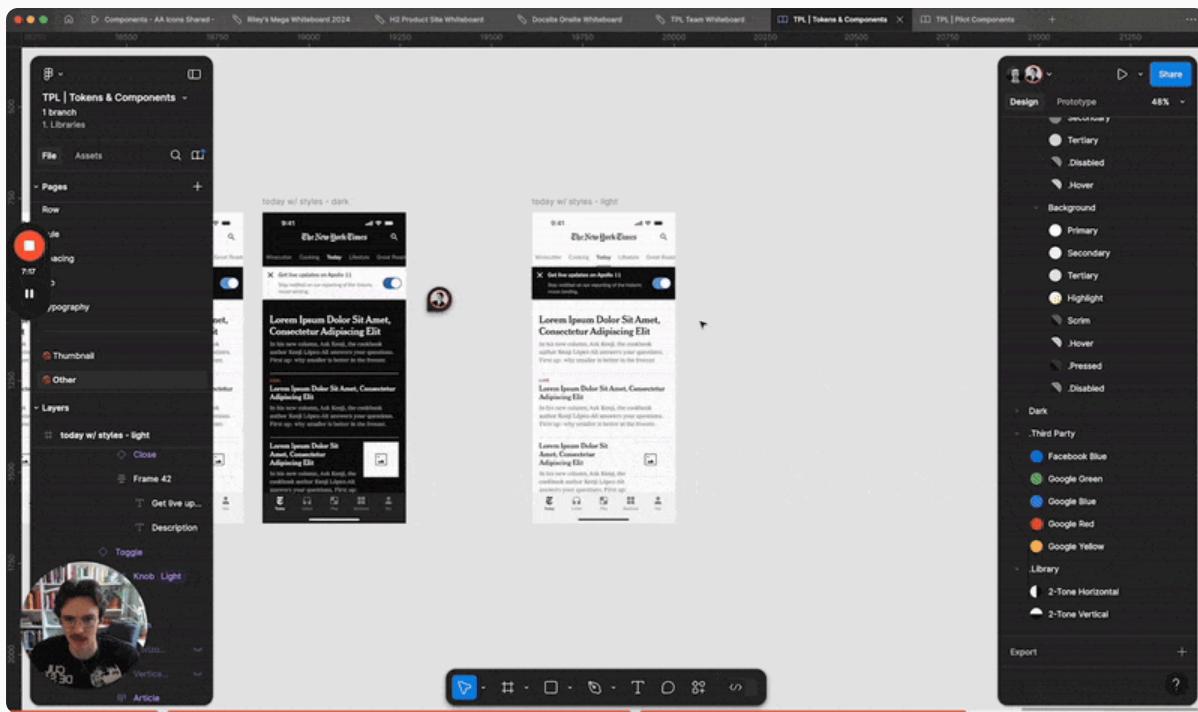
⚠️ Currently, [nytimes.com](#) does not enable Dark mode in web browsers. If you are curious about Dark mode for web, please reach out to us on [Slack](#).

@Color behavior

You can control the appearance of colors using four settings: **User Default**, **User Inverse**, **Always Dark**, and **Always Light**.

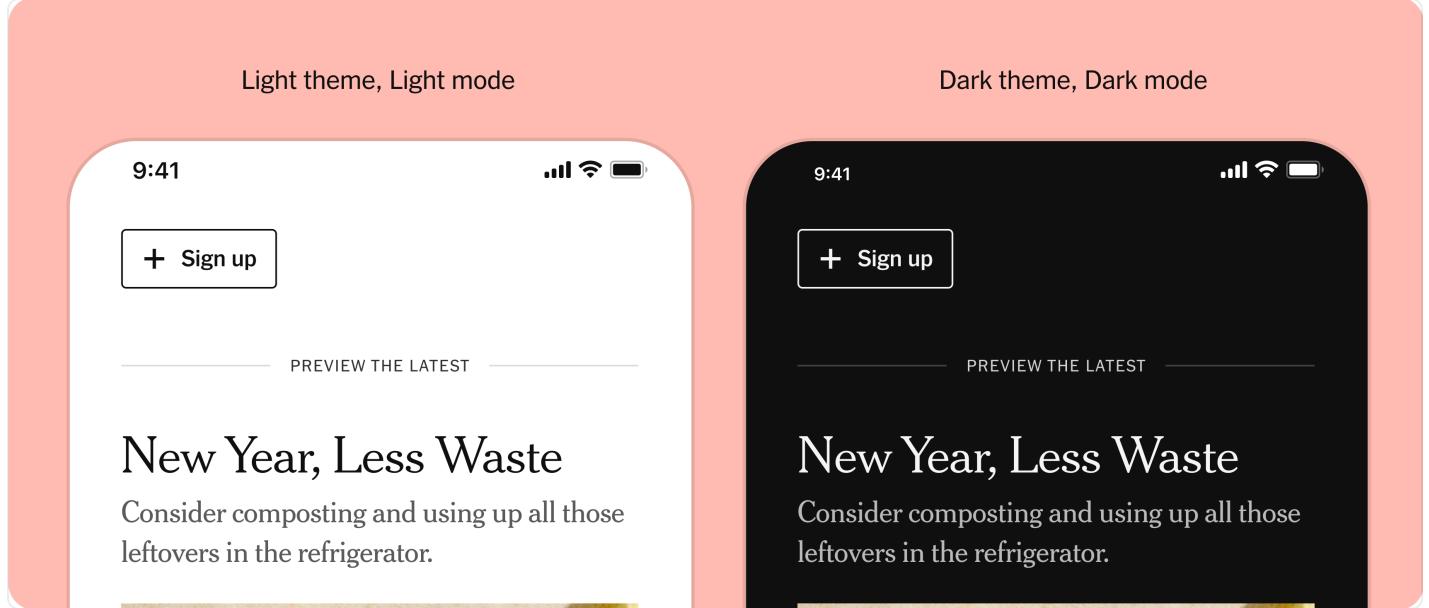


TPL now uses Variables in Figma to spec and hand off Dark Mode and Color Behaviors. Watch our [Variables Tutorial](#) to learn how to take advantage of *one-click dark-mode* ✨



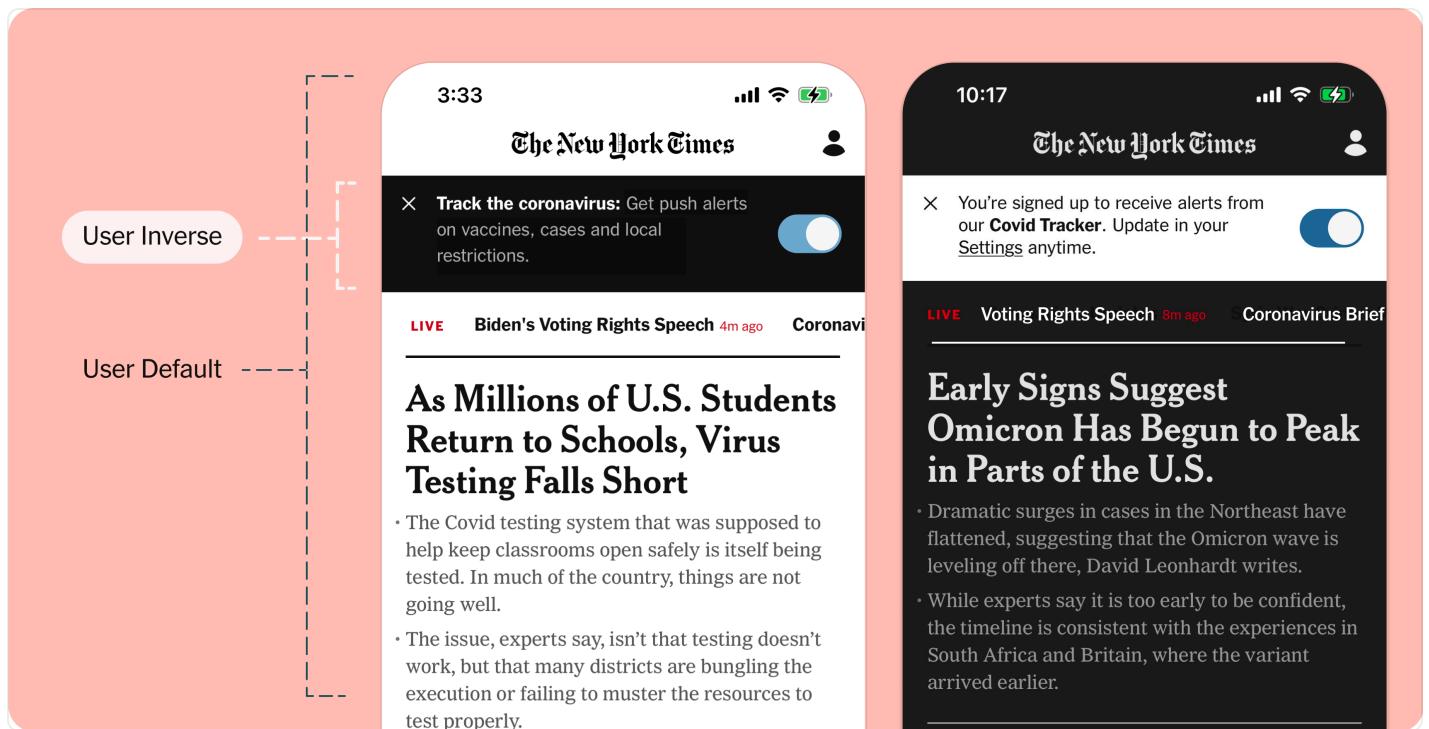
User Default

This setting matches the color theme of your component/feature with the reader's Device or App setting. Content with this property uses the Light theme in Light Mode and the Dark theme in Dark Mode.



User Inverse: Theme and @Display mode are inverted

This setting inverts the color theme of your component/feature from the reader's Device or App setting. Content with this property uses the Dark theme in Light mode, and the Light theme in Dark mode.

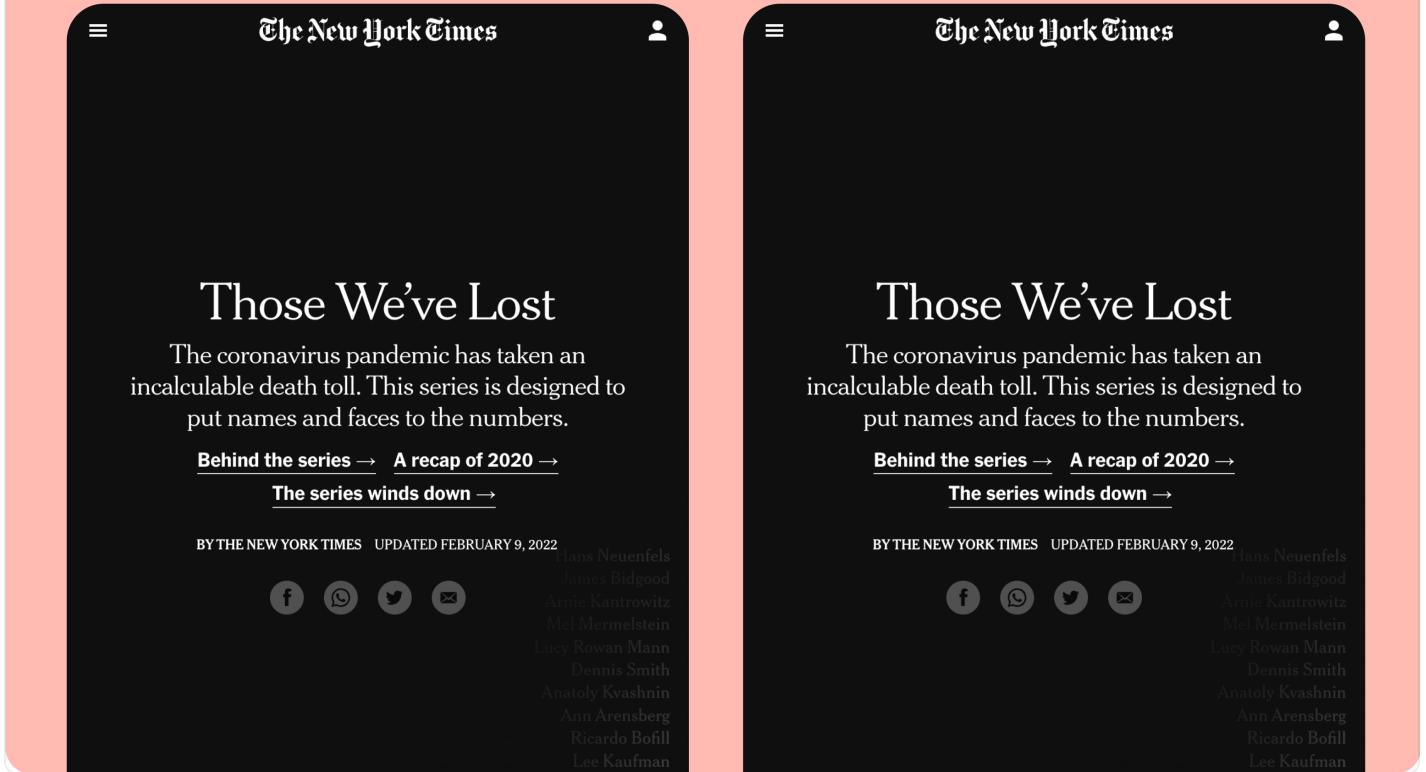


This should be used for elements that are intentionally offset from the content around it, like a Toast or Button. We recommend using it on specific, contained elements of an experience to make the most of the offsetting effect.

Always Dark

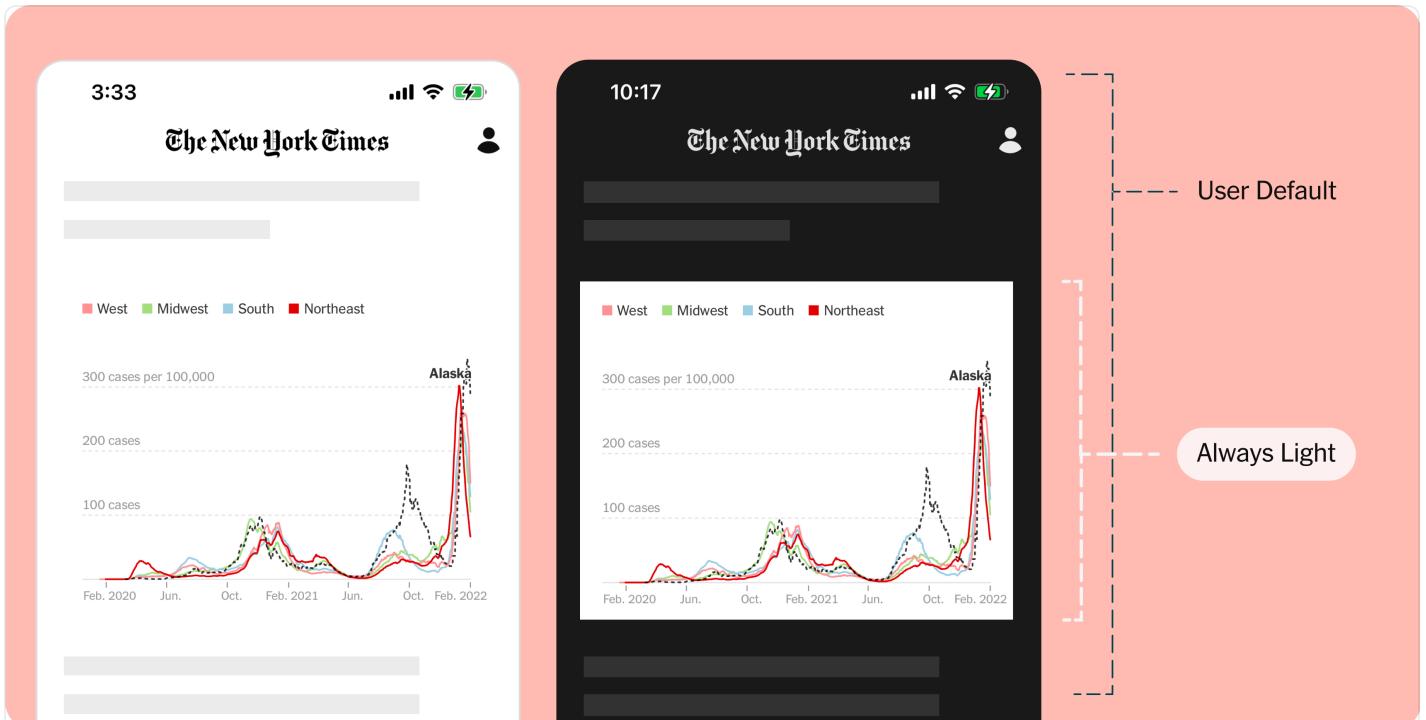
This setting ignores the reader's Device or App setting. Content with this property uses the Dark theme in both Light and Dark modes. Always Dark layouts can be used to help readers focus on immersive content.

Dark theme, mode-agnostic



Always Light

This setting ignores the reader's Device or App setting. Content with this property uses the Light theme in both Light and Dark modes.



⚠️ “Always” properties should be used carefully and in rare cases. Consider how a reader might react to an experience that does not meet their expectation.

Color Term Glossary

Term	Definition
Color theme	A palette of semantic colors. Currently, TPL offers Light theme and Dark theme .
Display mode	This is the system or app setting for screen display. The specific term used to describe these modes varies depending on the browser, device, operating system, etc. TPL uses the terms Light mode and Dark mode .
Color behavior	This is a product-team defined setting that determines which color theme is used in relationship to the user-defined Display Mode.

Still here? Honestly, kinda creepy 😊

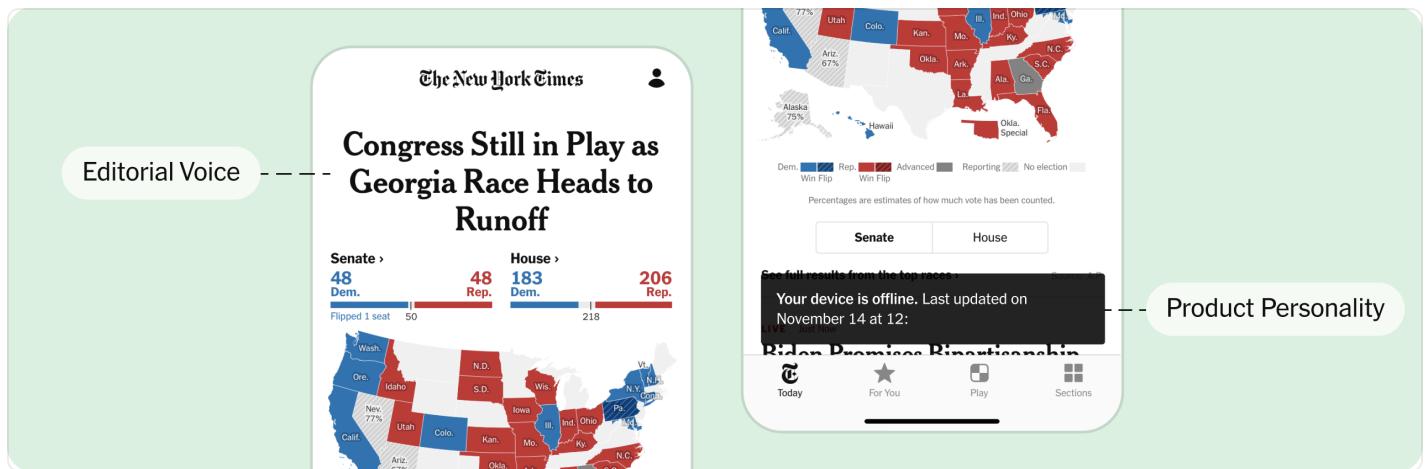
[See in Figma ↗](#)

[See in Storybook ↗](#)

Content

Overview

As The New York Times grows and evolves, the importance of speaking to our readers cohesively and effectively across our experiences will continue to do the same. It's important that we speak to readers with a product personality that is informed by our brand voice and distinct from the voice of our editorial staff.



Our product personality is **personable** and **efficient**. It is meant to warmly and speedily help readers find their way through our product experience and accomplish their desired tasks. Remember, some **moments** warrant more efficiency and others require more warmth.

Support

If you have any questions regarding UX Writing, please don't hesitate to reach out on Slack.

Fundamentals

Following these principles will help you write content cohesively with teams across the product experience. The significance of any one principle will vary based on the [moment](#) at hand. Don't feel forced to follow all of them equally every time.

Empathizing with readers

Employ accessibility best practices. Refer to our [accessibility guidance](#) for more details.

Meet readers where they are. Consider more than just a reader's desired destination — consider how a reader might be feeling at a given moment and use language appropriate for someone in that state.

Consider where a reader has come from and where they want to go. Ask yourself: Does a reader understand what the next step is and what will happen if they take it?

Not everyone is ready to take action. That's okay. Think of the product's personality like a concierge; using warmth when recommending something without being forceful, selling or persuading.

Our readers don't work here. Assume the vast majority of our readers are less familiar with our products, features, and brand than we are.

Leading with simplicity

Don't do too much. Consider the messaging goal of each product moment. If multiple goals exist, prioritize them to ensure the experience makes sense to the reader.

Use simple grammar, syntax and vocabulary.

Avoid jargon. We have over 100 million registered readers — our writing should be digestible to as many of them as possible.

Considering editorial context

Avoid distracting from the editorial voice. Our product's personality is not the reason readers come to our surfaces but it is part of the reason they stay.

Pressure-test language against the full spectrum of what we publish. Ask yourself: does this language feel appropriate alongside various kinds of subject-matter?

Don't hesitate to reach out if you need help. The Times has a dedicated writing team who can provide feedback and guidance as needed.

Moments

We've identified five moments that cover many of the scenarios where we need to communicate with users in different ways, based on reader needs and our goals. We expect these moments will overlap in practice; do your best to blend them together when necessary and [reach out](#) if you need anything!

Wayfinding

Wayfinding moments are when readers are navigating to desired destinations. This commonly applies to persistent section labels and navigational elements, for example.

Subscription benefits	>
Account settings	>
Subscription overview	>
Your Content	
Saved articles	>
Newsletters	>
Get Support	

Guidance

Try to be both brief and descriptive in these moments. Remember that these elements could appear adjacent to our editorial voice where a more direct or formal tone might be warranted.

Management

Managing moments are when users are completing important tasks like transactions or adjusting settings. This commonly applies to transactional processes (subscription, upgrade and cancellation) and turning notifications on or off, for example.

1. Account your@email.com [Edit](#)

2. Payment

Credit or Debit Card
 Apple Pay
 PayPal

[Continue](#)

••• Personable
••• Efficient

Guidance

Focus on helping readers get things done quickly and reliably in these moments.

Acknowledgment

Acknowledging moments are when the product is providing feedback on reader actions and offering direction as needed. This commonly applies to confirmation and error messages, for example.

Logged in



You are logged in as
fred.again@gmail.com

••• Personable
••• Efficient

Guidance

Efficiency is important in these moments. However, we still need to provide enough context and direction for readers to move forward. Consider their feelings of success or frustration.

Recommendation

Recommending moments are when the product offers new topics or content beyond what the reader may be looking for. This commonly applies to newsletter and story recommendations, for example.

A smartphone screen displays a recommended newsletter. The header reads "Recommended Newsletter". The main content is a story by John McWhorter titled "An exploration of race and language shape our politics and culture from a Columbia University linguist." Below the story, it says "Available only for New York Times subscribers." A small profile picture of John McWhorter is shown next to his name, which is bolded. Below his name, it says "SENT TWICE A WEEK". At the bottom, there are two buttons: "Sign up" and "See the latest".

••• Personable
••• Efficient

Guidance

These moments are good opportunities to be more personable as we explain what we are recommending and, potentially, why we are recommending it.

Welcoming

Welcoming moments are when we greet and introduce readers to a subscription or product. This applies to the "Get Started" experience for new subscribers and the welcome screens for new app readers, for example.

Two smartphones are shown side-by-side. The left phone displays a "Get Started" screen with a "T" logo at the top. The text reads "Thank you for subscribing. We've sent an email confirmation to your@email.com". There are two buttons at the bottom: "Set up your new subscription" (in a dark box) and "Take the new subscriber survey". The right phone displays a "Welcome" screen with a "T" logo at the top. The text reads "The best experience is in our app." Below this is a button that says "Return to articles in your personal reading list". Both phones show a timestamp of 9:41 and standard connectivity icons.

••• Personable
••• Efficient

Guidance

These moments serve as an introduction to things that likely are somewhat new or unfamiliar for readers. It may be appropriate to give more information in these moment, and do so warmly.

Don't hesitate to [reach out if you need help](#). We have writers dedicated to Content writing and support who can provide whatever feedback and guidance you are looking for.

Terms and styles

As a supplement to the New York Times Stylebook, this database offers guidance about the terms and styles we use, or do not use, in our product experience.

Term / Style	Guidance
Calls to action	Users can “log in” to our website and apps, “sign up” for newsletters and events and “subscribe” to products. Try not to confuse these actions.
Casing	Use sentence case, rather than title case, in headers and calls to action.
Contractions	Use where appropriate.
Devices	Be mindful that people may be using a variety of devices, including screen readers, to access our content. Avoid device-specific language, such as “click here.”
Directions	Be mindful that people may be using a variety of devices, including screen readers, to access our content. Avoid directional language, such as “see below.”
Emojis	Use these rarely.
Error messages	Prioritize offering the user a way to resolve an issue or accomplish a goal.
Exclamation points	Use these rarely.
Jargon	Avoid jargon. No slang or technical language, either.
New York Times All Access	This is “the bundle.” Includes all digital products in our portfolio: News, Games, Cooking, Audio, Wirecutter, The Athletic. Use this order when describing these products.
New York Times Cooking	Included in All Access and offered as an individual product subscription. Accessible via nytimes.com/cooking and the NYT Cooking app.
New York Times Games	Included in All Access and offered as an individual product subscription. Accessible via nytimes.com/games , in the Play tab of The New York Times app and in the NYT Games app.
New York Times News	Included in All Access. While it is not promoted, it remains available as an individual product subscription. The “Basic Access” subscription name has been retired. To describe its contents, use “investigation, culture, analysis.” Avoid signaling that this is an all-encompassing option.
New York Times Wirecutter	Included in All Access and offered as an individual product subscription. Accessible via www.nytimes.com/wirecutter and The New York Times app.
The Athletic	Included in All Access and offered as an individual product subscription. Accessible via theathletic.com and The Athletic app. Select articles from The Athletic are available on nytimes.com and The New York Times app.

The New York Times

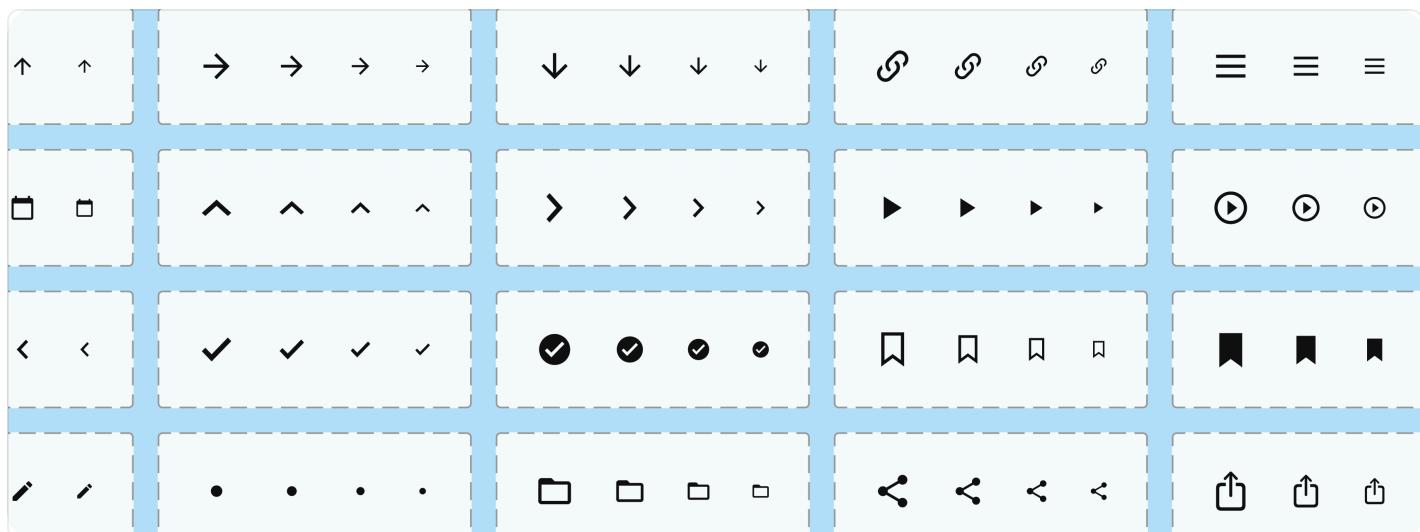
This is our brand. Use the full name wherever possible, but you may shorten to "The Times" or "Times" on our own platforms.

Don't hesitate to reach out if you need help. We have writers dedicated to Content writing and support who can provide whatever feedback and guidance you are looking for.

Icons

Overview

Icons are small, versatile graphics that can be used to communicate information.



Icons fall into three categories:

1. [@Standard](#) icons for signifying actions & objects
2. [@Brand](#) icons for signifying NYT brands
3. [@Third-party](#) icons for signifying outside brands

Icons are commonly used within a component, such as [Buttons](#), making them interactive. Icons should typically be paired with text providing supplemental context. If used on their own they should always be labeled for assistive technologies.

All icons are available in **12px**, **16px**, **20px**, and **24px**. Button sizing corresponds to these increments.



Figma

Web

iOS

Android

Components

React / CSS-in-JS

No

No

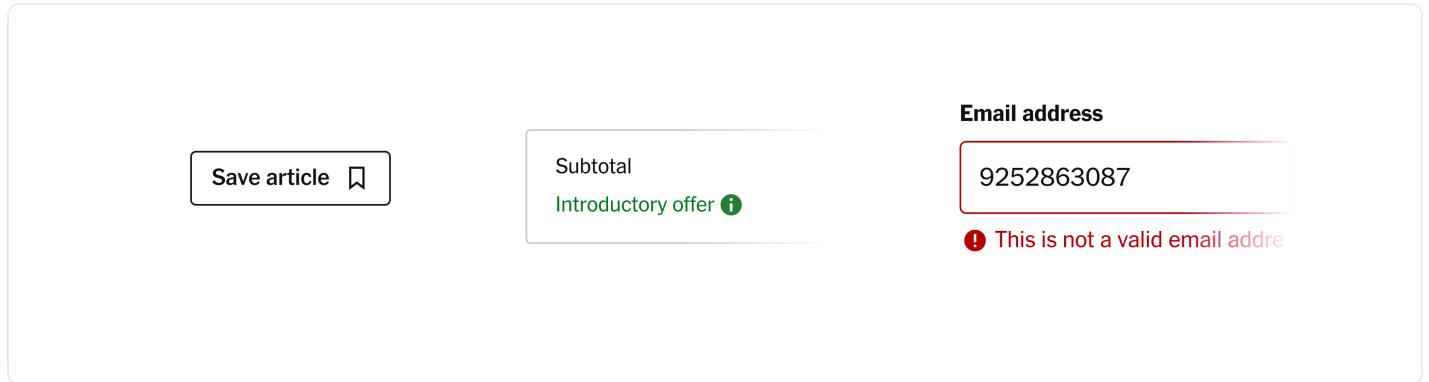
[See in Figma ↗](#)

[See in Storybook ↗](#)

Guidance

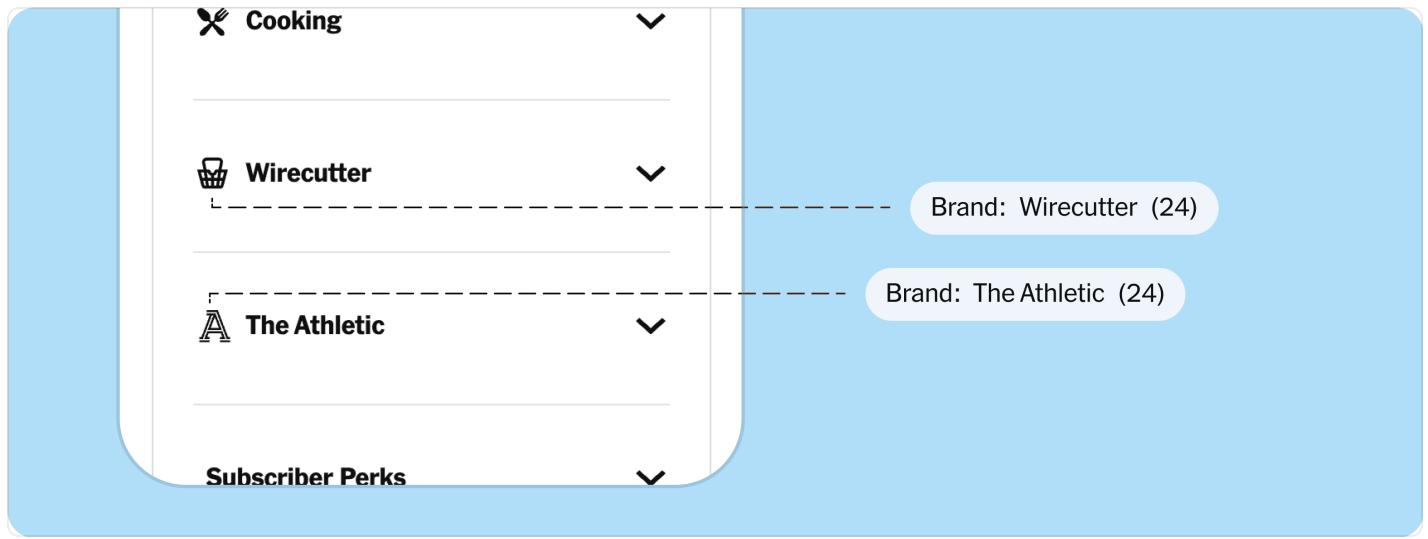
For actions & elements

@Standard icons are designed to express a variety of things. Refer to the name of the icon when deciding if it's appropriate for your use case. For example, `Alert` should be used for urgent use cases and `Info` should be used for less-urgent use cases.



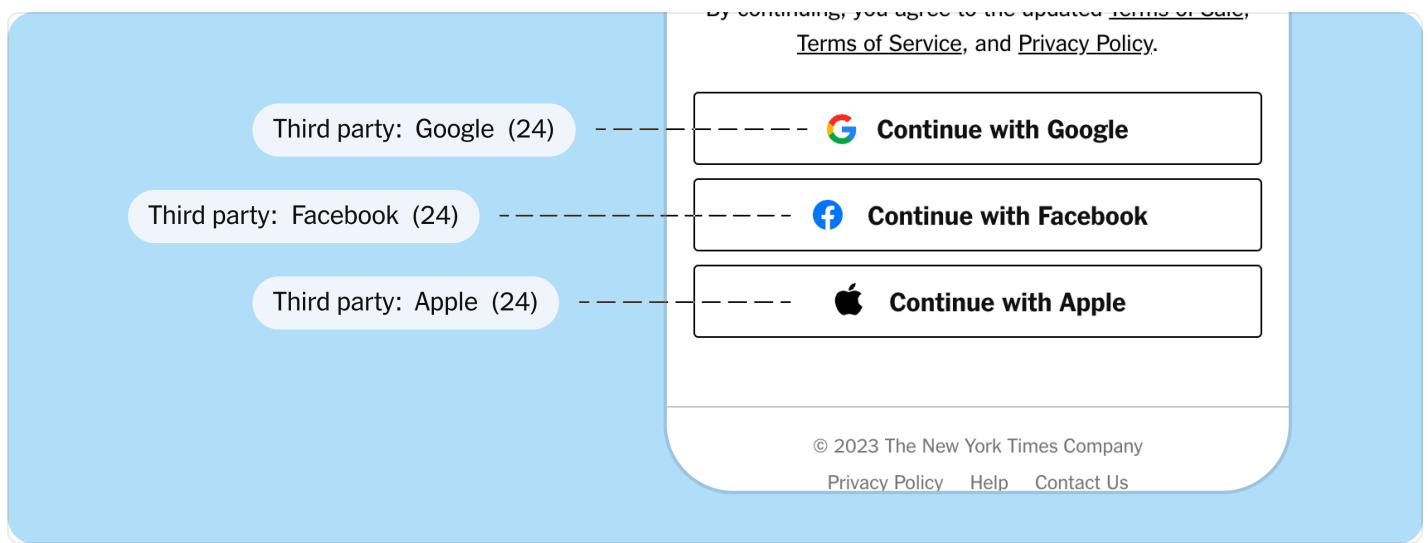
For NYT brands

@Brand icons are designed to indicate one of The New York Times' brands.



For outside brands

@Third-party icons are typically designed by the outside companies they are expressing. They each meet the outside brand's guidelines and should rarely be used outside of a component.



Applying color

By default, Icon matches the color of surrounding text. Most icons can be set to use any [Content color token](#), except for some brand icons (eg: Apple, Facebook, Google, and Super-T).

Accessibility

An Icon can signify many things. For readers who are blind or low vision and unable to visually gather context to interpret the meaning of an Icon, providing a visually hidden label can be crucial.

When an Icon is not paired with text—for example, an [Icon Button](#)—it **must** receive a visually hidden label (using the label prop). This will ensure that the intent of the Icon is communicated clearly to readers who rely on assistive technology.

However, when an Icon is paired with text—for example, in a [Button](#) that also uses text—the Icon does **not** need a visually hidden label. Instead, the Icon should only act as a secondary signifier to the text and should be skipped by assistive

technologies (using the `aria-hidden` attribute).

Specifications

Type	Name	Sizes
Standard	Alert Arrow up Arrow right Arrow down Arrow left Calendar Caret up Caret right Caret down Caret left Checkmark Check circle Close Create Dot Folder Help History Home Info Link Menu More horiz More vert Muted Pause Person Play Play circle Plus Refresh Save Save filled Search Settings Share Share (Android) Share (Apple) Star Unmuted	12, 16, 20, 24
Brand	The Athletic Cooking Games Newspaper Super-T Wirecutter	12, 16, 20, 24
Third-party	Apple Facebook Google	12, 16, 20, 24

You made it just in time for the animal parade!



[See in Figma ↗](#)

[See in Storybook ↗](#)

Messaging

TPL's messaging framework helps you more easily and consistently meet your messaging goals.

| How do I determine the appropriate messaging for x when y happens?

1. First, determine what type of messaging you need to accomplish
2. Then, pick a messaging component from the category that...
 - Makes sense given the user flow & the UI
 - Has the appropriate level of prominence

	Tip	Badge	Dialog	Toast	Sheet
Awareness	✓		✓		✓
Guidance	✓				
Signal		✓		✓	

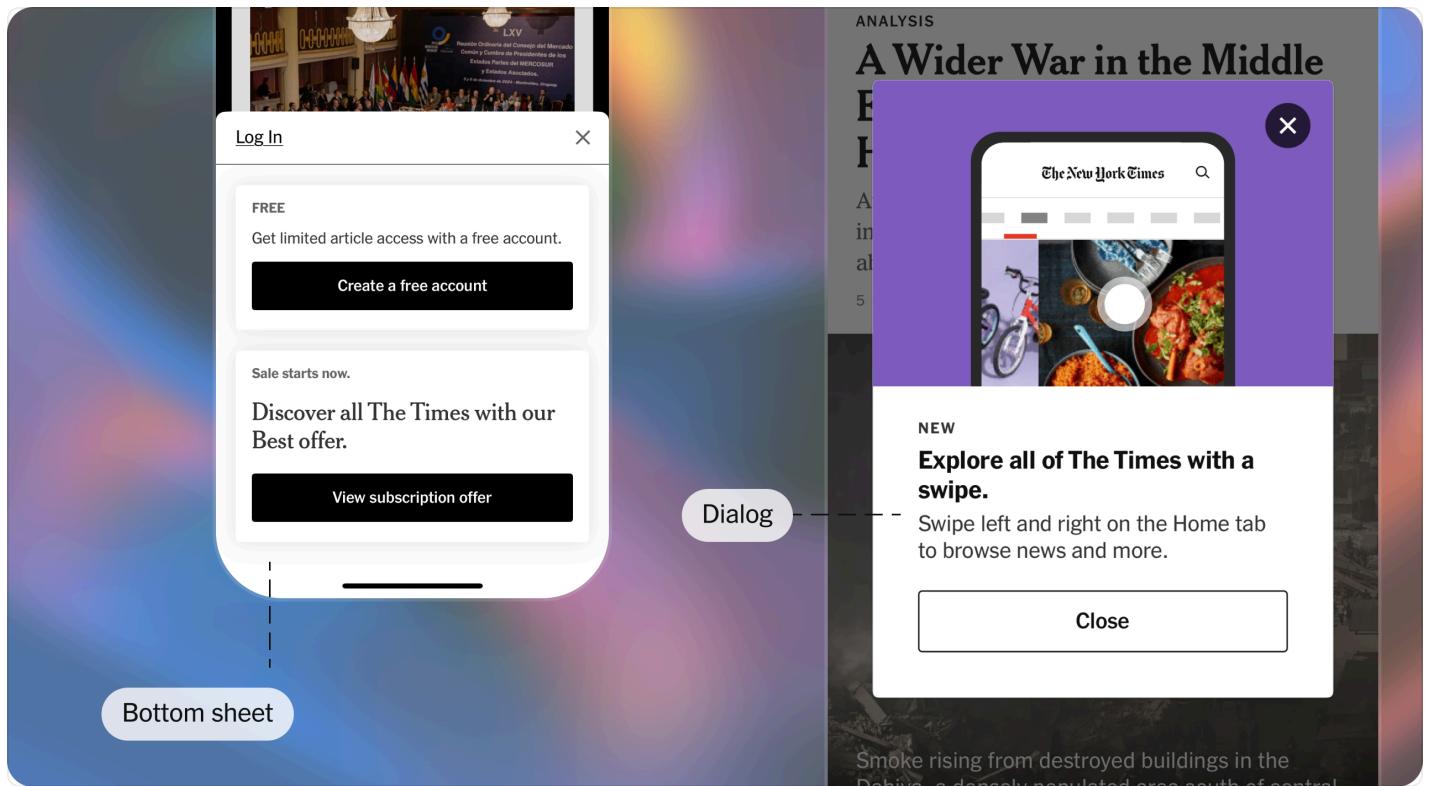
Messaging types

Awareness

Moments when we greet and introduce readers to a new feature or experience. This applies to the "Get Started" experience for new subscribers and the welcome screens for new app readers, for example.



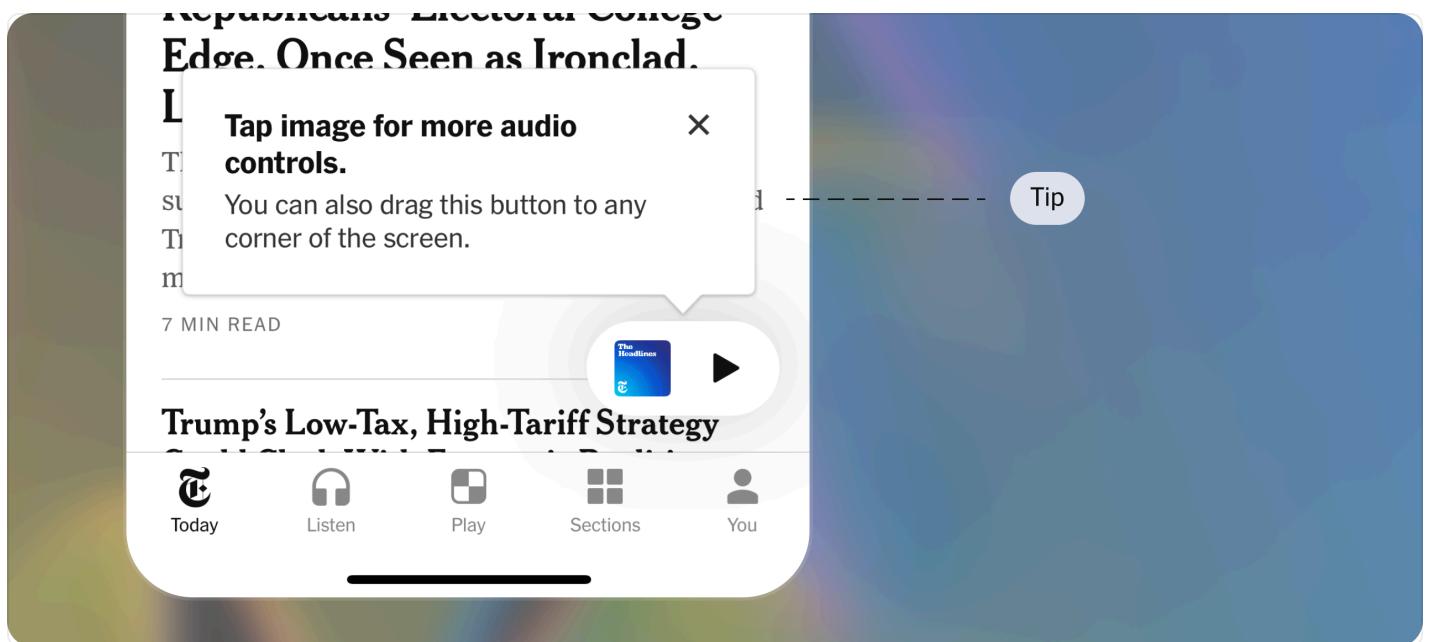
These moments serve as an introduction to things that likely are somewhat new or unfamiliar for readers. It may be appropriate to give more information in these moment, and do so warmly.



Guidance

Moments when readers are navigating to desired destinations, or we are highlighting an action we think they should take. These messages reach the user in the moment, helping guide them to a destination or action (that is a key part of their journey) that they are not already engaging with.

- 💡 Try to be both brief and descriptive in these moments. Remember that these elements could appear adjacent to our editorial voice where a more direct or formal tone might be warranted.

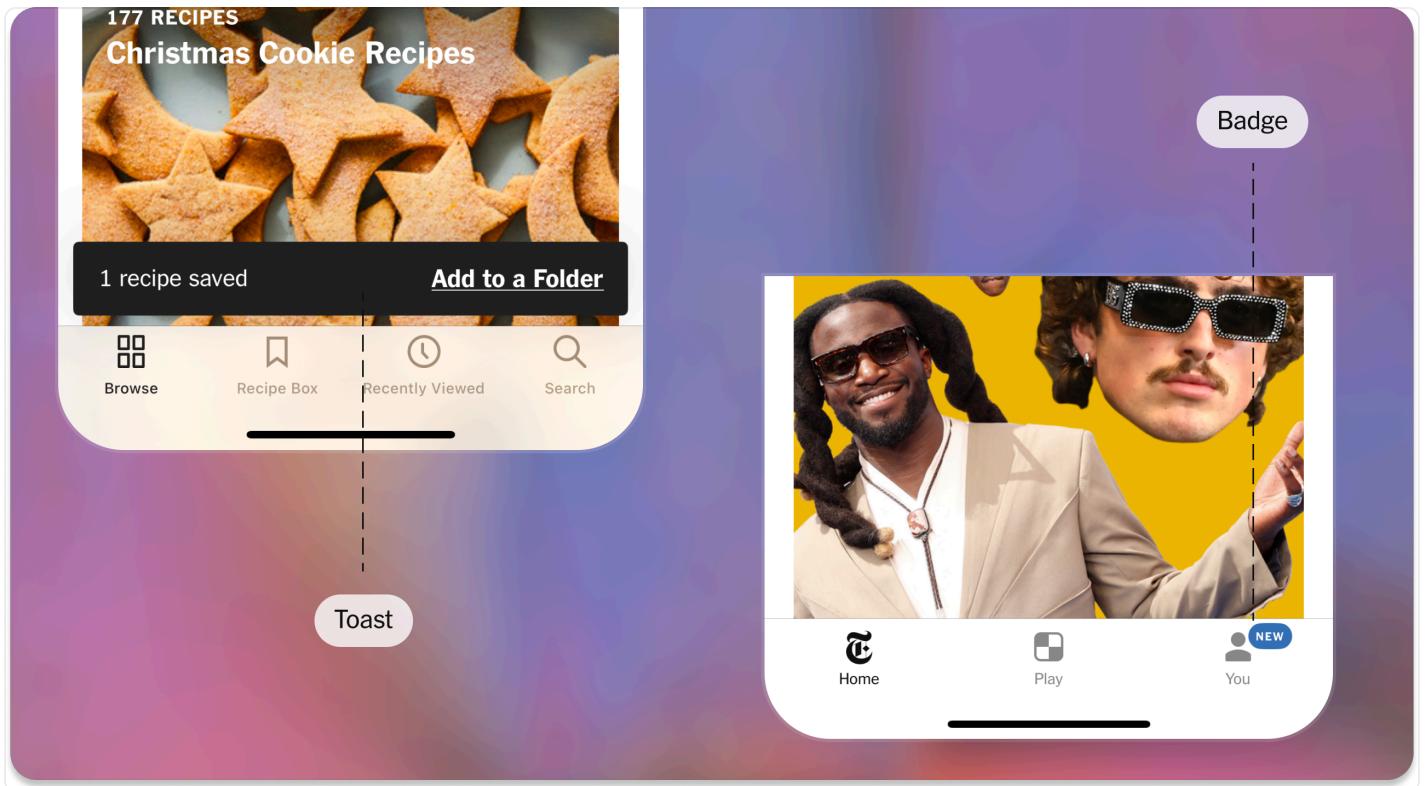


Signal

Reflects a change in content or UX as an outcome of a user action. These are moments when the product is providing feedback on reader actions and offering direction as needed. This commonly applies to confirmation and error messages, for example.



Efficiency is important in these moments. However, we still need to provide enough context and direction for readers to move forward. Consider their feelings of success or frustration.



User Flow Considerations

In order to present app product messages that create clarity, optimize for a user's time, and build trust with our product we must consider:

- Where a user is in their journey with us
- What they have and have not already seen or engaged with across the NYT product experience
- What other messaging is prioritized in their journey (examples: sales, gifting)

Messaging Components

● Dialog

Appears as a modal covering the entire screen. It is often used to promote features or direct users to something new. Often leveraged for the announcement of a new feature or large update to an existing feature that necessitates a splashier tone or strong visual aid.

Prominence: High

● Sheet

Announcement of new experiences or features that are high in subscriber value, *and* that necessitate a splashier tone or strong visual aid. Tied to a broader product marketing campaign and should be reserved for those moments. Most impactful when multiple features are bundled together. Not recommended for use for individual smaller UX updates or features.

Prominence: Medium

Tip

Tip displays a contextual, dismissible message that brings attention to a feature. Consider the user journey—do not display to users who have already engaged with the feature or destination or have been shown the bottom sheet.

Prominence: Low

Toast

Announces a change of status. It is often used to acknowledge or confirm an action, and does not require an action.

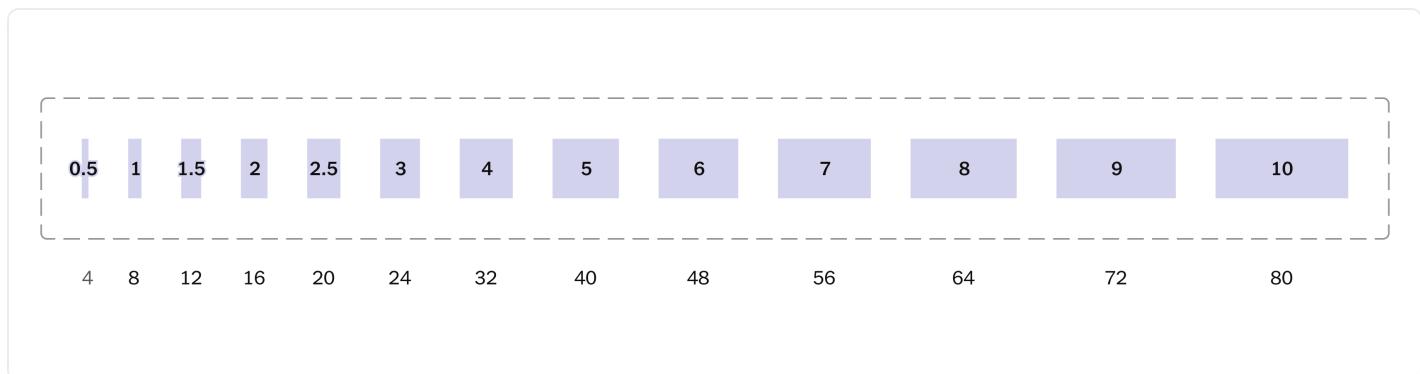
Prominence: Low

Spacing

Overview

Spacing can be used horizontally and vertically to control the visual density or openness of a digital experience. It's also used to create visual relationships between related elements.

Our spacing system is built from a base value of 8px, represented as a 1 token, that can be modified to any other value on [our scale](#), from .5 to 10 (4px to 80px).



Figma

Web

iOS

Android

Components

JS / CSS-in-JS

CGFloat Extension

Kotlin

Variables

XML Resources

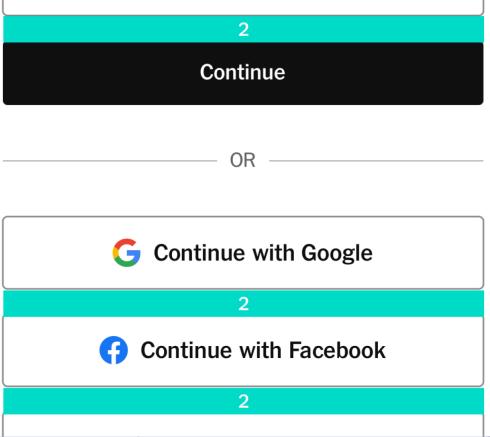
[See in Figma ↗](#)

[See in Storybook ↗](#)

Guidance

Even spacing

Consistent spacing between elements creates visual normalcy that is least disruptive and reduces cognitive load. It also creates a relationship of similarity between the elements that are spaced this way.



OR

Horizontal Spacing (2)

The most crucial business and policy news you need to know from Andrew Ross Sorkin and team.

RECOMMENDED BASED ON YOUR CHOICES



2.5

NYT Cooking App

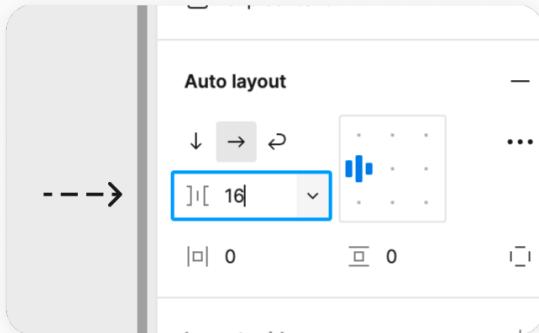
Get more from your recipe box with NYT Cooking.

Download

2.5

Vertical Spacing (2.5)

 Auto-Layout is the best way to create even spacing in Figma. You'll need to manually enter TPL spacing values into the "Space between" field.



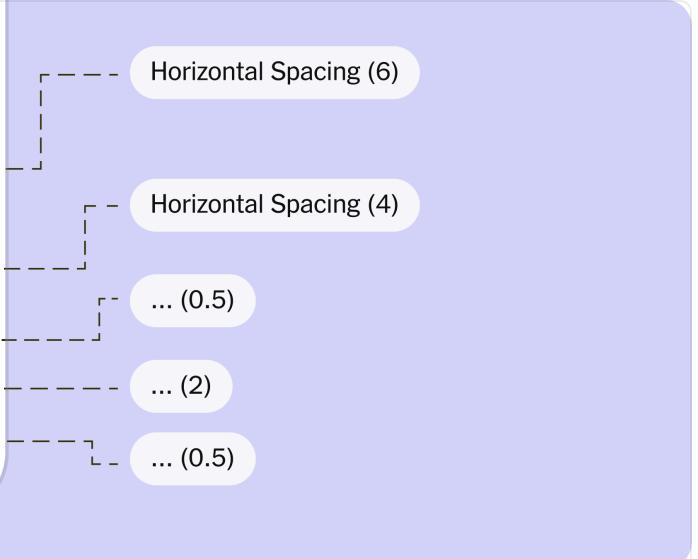
Good news! We have piloted offering spacing values as Variables in Figma. If you have feedback, [Slack us!](#)

Irregular spacing

Irregular spacing creates more nuanced relationships between elements than [even spacing](#). Although more visually complex, it suggests which elements are related (closer together) and which are not (further apart).

You're All Set

Review your choices below and begin exploring your subscription.



Spacer components are the easiest way to create irregular spacing in Figma.

Responsive spacing

You may want to adjust spacing on different screen sizes. Consider decreasing the vertical space between elements on smaller screen sizes to reduce page height and make scrolling less taxing.

The figure displays two side-by-side mobile news application interfaces. The left interface, labeled "Vertical Spacing (4)", features larger vertical gaps between its sections. It includes a navigation bar with links like "Summit", "Updates", "What Is NATO?", and "What NATO Promised Ukraine". Below this is a teal header section with the number "4". Further down are news cards for "Erdogan's U-Turn on Sweden" and "Will Turkey Join the E.U. Now?". A timestamp at the bottom indicates "Current time in: Kyiv July 12, 6:54 p.m. Moscow July 12, 6:54 p.m. Washington July 12, 11:54 a.m.". The right interface, labeled "Vertical Spacing (2.5)", shows a more compact design with smaller vertical gaps. It has a similar navigation bar and a teal header section with the number "2.5". The news cards for "Erdogan's U-Turn on Sweden" and "Will Turkey Join the E.U. Now?" are also present. Both interfaces include a "LIVE" button and a timestamp at the bottom indicating "Current time in: Kyiv July 12, 6:54 p.m. Moscow July 12, 6:54 p.m.".

We recommend limiting horizontal groupings of elements at smaller breakpoints because of the limited screen space available.

Specifications

Spacing scale

Token	.5	1 (Base)	1.5	2	2.5	3	4	5	6	7	8	9
-------	----	----------	-----	---	-----	---	---	---	---	---	---	---

Pixel	4px	8px	12px	16px	20px	24px	32px	40px	48px	56px	64px	72px
REM	.25 rem	.5 rem	.75 rem	1 rem	1.25 rem	1.5 rem	2 rem	2.5 rem	3 rem	3.5 rem	4 rem	4.5 rem

What's your fortune? 🎡 Getting to use TPL! 😊

[See in Figma ↗](#)

[See in Storybook ↗](#)

Typography

Overview

TPL offers a range of The New York Times' most essential typography tokens for our digital products.

Summer Politics Quiz	20 Uptick Seen in Red Knots on Jersey Shore <i>Uptick Seen in Red Knots on Jersey Shore</i>	12 The aim of The New York Times is to "give the news, all the news, in concise and attractive form ... to give the news impartially, without fear or favor" ... —Adolph S. Ochs, Aug. 18, 1896	16 Secttuna
r Summer Politics Quiz	22 Uptick Seen in Red Knots on Jersey Shore <i>Uptick Seen in Red Knots on Jersey Shore</i>	14 The aim of The New York Times is to "give the news, all the news, in concise and attractive form ... to give the news impartially, without fear or favor" ... —Adolph S. Ochs, Aug. 18, 1896	18 Sillisbc
ur Summer Politics Quiz	28 Uptick Seen in Red Knots on Jersey Shore <i>Uptick Seen in Red Knots on Jersey Shore</i>	16 The aim of The New York Times is to "give the news, all the news, in concise and attractive form ... to give the news impartially, without fear or favor" ... —Adolph S. Ochs, Aug. 18, 1896	20 Sisivena
, Hawaii	32 Uptick Seen in Red Knots on Jersey Shore <i>Uptick Seen in Red Knots on Jersey Shore</i>	18 The aim of The New York Times is to "give the news, all the news, in concise and attractive form ... to give the news impartially, without fear or favor" ... —Adolph S. Ochs, Aug. 18, 1896	
hu, Hawaii	36 Uptick Seen in Red Knots on Jersey Shore <i>Uptick Seen in Red Knots on Jersey Shore</i>	20 The aim of The New York Times is to "give the news, all the news, in concise and attractive form ... to give the news impartially, without fear or favor" ... —Adolph S. Ochs, Aug. 18, 1896	
Oahu, Hawaii	40 Uptick Seen in Red Knots on Jersey Shore <i>Uptick Seen in Red Knots on Jersey Shore</i>		
isit in Oahu, Hawaii	44 Uptick Seen in Red Knots on Jersey Shore <i>Uptick Seen in Red Knots on Jersey Shore</i>		
Visit in Oahu, Hawaii	48 Uptick Seen in Red Knots on Jersey Shore <i>Uptick Seen in Red Knots on Jersey Shore</i>		

1. [@Text](#) - For product copy
2. [@Title](#) - For headings & visual weight
3. [@Title Karnak](#) - For representing sub-brands & newsletters
4. [@Label](#) - For categorization
5. [@Headline](#) - For any story headline, agnostic of tone
6. [@Headline News](#) - For News headlines only
7. [@Headline Feature](#) - For Feature headlines only
8. [@Headline Opinion](#) - For Opinion headlines only
9. [@Body](#) - For story content only

Headlines ([@Headline](#), [@Headline News](#), [@Headline Opinion](#), [@Headline Feature](#)) and [@Body](#) are only used for our journalism. Strong signaling for our journalism is core to our brand identity, so these tokens should never be used elsewhere.

All font sizes fall on an incremental scale from 10 to 72.

Text Styles

React / CSS-in-JS

Typography Token

Compose

[See in Figma ↗](#)

[See in Storybook ↗](#)

Guidance

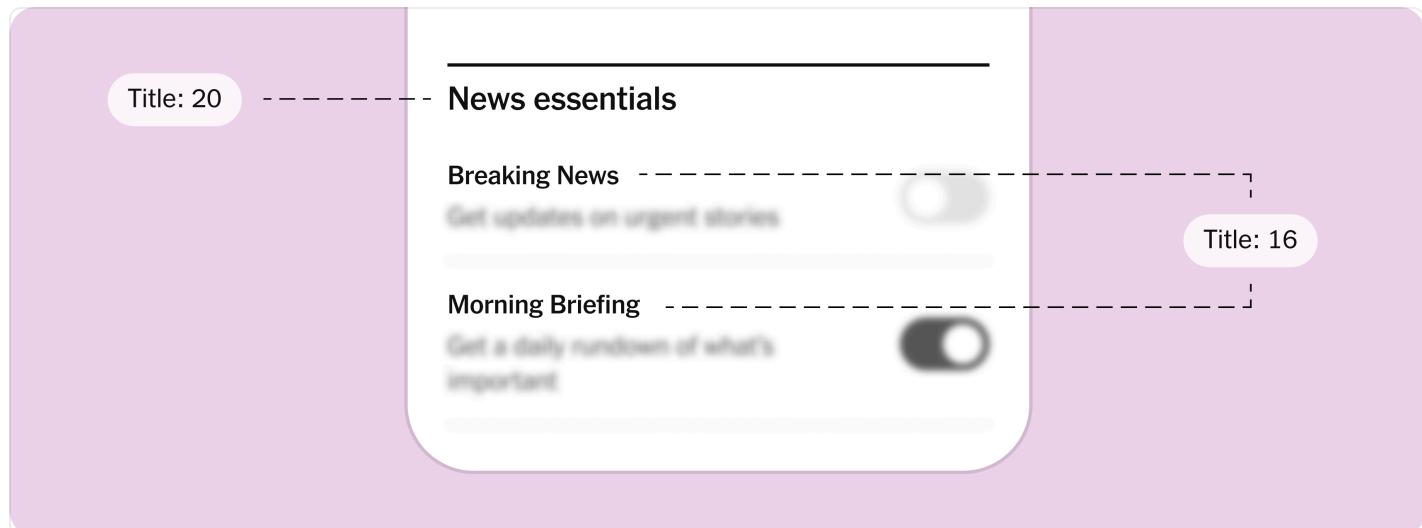
For product copy

@Text is appropriate for short to long passages of non-journalism copy. @Text can be paired with @Title to form basic hierarchies. For inline links, we suggest keeping the text as Content.Primary, along with an underline.



For headings & visual weight

@Title offers extra visual weight and larger sizes. It's appropriate for short pieces of copy to help readers navigate and understand the content of a screen. Most often @Title is paired with @Text to provide more information and hierarchy. @Title may be used for headings, legends, and emphasis copy.



For categorization

@Label adds extra context to other pieces of text and comes in two weights: @**L** Regular and @**L** Emphasis. It is always used alongside other copy for short text strings and is never used as a heading.

Trump, Canada's Fighting Wildfires and more. **LIVE**

1 MIN READ

NEWSLETTER

Read The Interpreter

These books will help you understand the significance of the Supreme Court's ruling on

Label: Standard

A Visibly Angry Putin Emphasizes the Failure of the Wagner Revolt

Vladimir Putin's remarks came after Yevgeny Prigozhin, the founder of the Wagner mercenary force, claimed he was not seeking to oust the Russian president.

Label: Emphasis

Use @Label carefully as text in all-caps can be less readable.

For story headlines

@Headline can be used to communicate the headline of any story no matter which desk or section it came from. Because it can be used for any type of story, @Headline is ideal for a package of different kinds of stories.

In State Legislatures, Old Rivalries Are Reheated by National Politics

The Crypto World Is on Edge After a String of Hacks

After Million-Mile Journey, James Webb Telescope Reaches Destination

Headline (16)

Use @Headline News only to communicate the headline of a News story; it may additionally be italic.

-- The Supreme Court upheld an unusual Pennsylvania law on corporate lawsuits.

3 MIN READ

-- The First Amendment puts limits on laws that ban online threats, the justices ruled in a separate case.

4 MIN READ

News: Headline (20)

Supreme Court Rejects Theory That Would Have Transformed American Elections

The 6-3 majority dismissed the “independent state legislature” theory, which would have given state lawmakers

News: Headline (32, Italic)

Use [@Headline Opinion](#) only to communicate the headline of an Opinion story.

A Handy Guide to the Republican Definition of a Crime

June 27, 2023

Opinion: Headline (36)



@Headline Opinion uses the font Cheltenham Condensed Text Bold in our web code library. Due to font limitations, our Figma library currently uses Cheltenham Condensed Bold instead. [Contact us](#) with any questions.

Use [@Headline Feature](#) only to communicate the headline of a Feature story.

The Terror of Threes in the Heavens and on Earth

Physicists have long explored how phenomena in groups of three can sow chaos. A new three-body problem, they warn, could lead to thermonuclear war.

7 MIN READ

Feature: Headline (32)

For story copy

@Body is only used for communicating the contents of a story and is appropriate for medium to long passages of text.

chased after three shoplifters who had taken off with several packages of laundry soap. Before the police arrived, he sat at a back-room desk to do something important: Capture the culprits' faces.

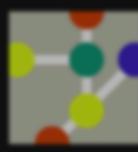
Body (18)

On an aging desktop computer, he pulled up security camera footage, pausing to zoom in and save a photo of each thief. He then logged in to a facial recognition program. Facewatch, which his store

For representing sub-brands & newsletters

Use @Title Karnak only when referencing sub-brands (The Athletic, Audio, Cooking, Games, and Wirecutter) and Newsletters within the News product. It's appropriate for short pieces of copy, typically referencing the name of the sub-brand.

Recommended Newsletters



WEEKLY

Science Times

Get stories that capture the wonders of nature, the cosmos and the human body.

[See the latest](#)

Audio

Hear reporter Adam Liptak on how abortion ended up back in front of the Supreme Court.



... [audio progress bar] ...

0:37

[Listen to the full episode](#)

Title Karnak (18)



This is an emerging pattern. Expect changes to naming and availability of font sizes and weights. Please send feedback to the TPL team.

To apply color

The default color for type is **Content Primary**. Consider using **Content Primary (Dim)** for long form content that may benefit from slightly less contrast in order to reduce eye strain, such as story content.

News essentials

Breaking News

Get updates on urgent stories



Morning Briefing

Get a daily rundown of what's important



Simon Mackenzie, a security officer at the discount retailer QD Stores outside London, was short of breath. He had just chased after three shoplifters who had taken off with several packages of laundry soap. Before the police arrived, he sat at a back-room desk to do something important: Capture the culprits' faces.

Content: Primary

Content: Primary (Dim)

Use **Content Secondary** to reinforce typographic hierarchy. Use **Content Secondary (Dim)** to reduce contrast on secondary text for an additional level of subtlety.

Content: Secondary

Barred From Grocery Stores by Facial Recognition

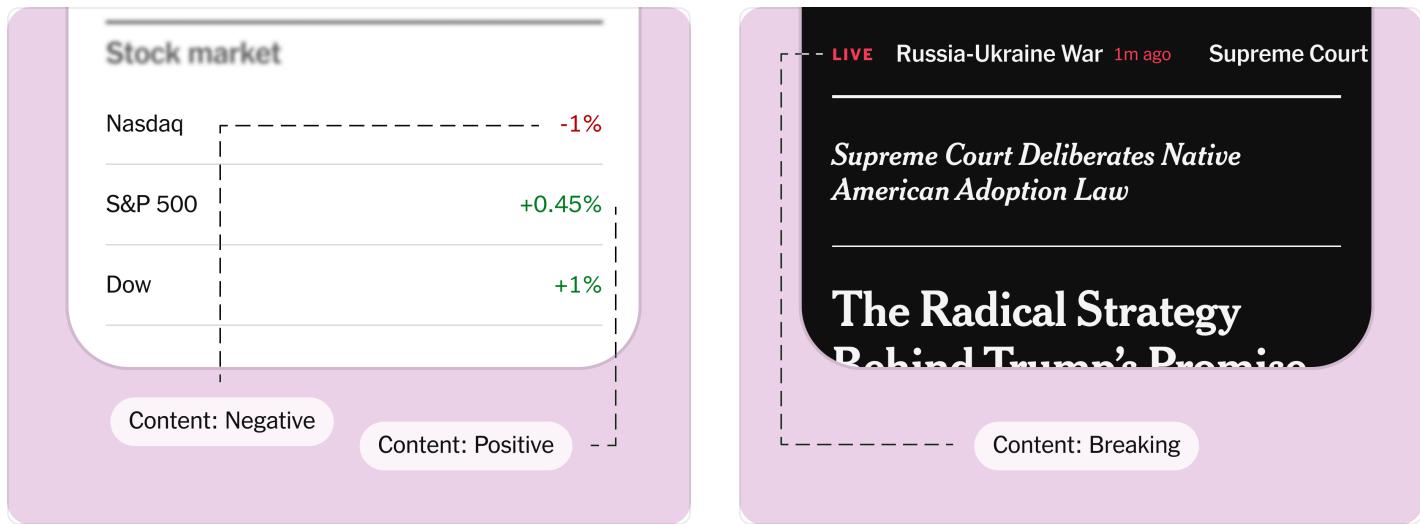
Vladimir Putin's remarks came after Yevgeny Prigozhin, the founder of the Wagner mercenary force, claimed he was not seeking to oust the Russian president.

4 MIN READ

Shein Flew Influencers to China to

Content: Secondary (Dim)

Use **Positive** to signify success or "up" and use **Negative** to signify error or "down". Use **Breaking** to signify breaking journalism.



- ⓘ Never use color alone to communicate important information.

Accessibility

Use color tokens to guarantee sufficient color contrast for type on top of **Background Primary** and **Background Secondary**.

Adequate color contrast, or the perceived visual difference in the lightness/darkness of a foreground and background, is crucial for readability. All type at The New York Times up to 18px must meet a contrast ratio of 4.5:1 and type 18px and larger must meet a contrast ratio of 3:1 (per [WCAG 2.1 AA](#) standards).

Aa Aa Aa Aa
Aa Aa Aa Aa Aa
Aa Aa Aa Aa
Aa Aa Aa Aa Aa

Background: Primary and Background: Secondary

Aa Aa Aa Aa
Aa Aa Aa Aa

Background: Tertiary and Background: Highlight

- ⓘ Only **Content Primary**, **Content Primary (Dim)**, **Content Secondary**, and **Content Secondary (Dim)** meet contrast standards on **Background Tertiary** and **Background Highlight**. No other content colors should be used on these background colors.

Specifications

Typography tokens

Category	Use(s)	i Fs / Lh / Ls	Fonts
Text	For product copy, used for short-long copy	12 / 1.3 / 0 14 / 1.3 / 0 16 / 1.3 / 0 18 / 1.3 / 0 20 / 1.2 / 0	Franklin Medium
Title	For added visual weight, used for headings, legends, emphasis	12 / 1.3 / 0 14 / 1.3 / 0 16 / 1.3 / 0 18 / 1.3 / 0 20 / 1.2 / 0 22 / 1.2 / 0 24 / 1.2 / 0 28 / 1.2 / 0 32 / 1.2 / 0 48 / 1.1 / 0 56 / 1 / 0	Franklin Semibold
Title Karnak	For representing sub-brands & newsletters	16 / 1.15 / 0 18 / 1.15 / 0	Karnak Bold
Label	For adding helpful context to other pieces of text, used only for short copy. Available in @↳ Regular and @↳ Emphasis weights.	-	-
↳ Regular	Weight of Label for general use	11 / 1.25 / .10	Franklin Caps Medium
↳ Emphasis	Weight of Label for extra attention	11 / 1.25 / .10	Franklin Caps Extra Bold
Headline	For communicating the headline of a story, appropriate for all story types	16 / 1.2 / .01 18 / 1.2 / .01 24 / 1.2 / 0 28 / 1.2 / 0	Cheltenham Medium (16) Cheltenham Book (18, 24, 28)
Headline News	For headlines of News stories only	16 / 1.2 / .01 18 / 1.2 / .01 20 / 1.2 / .01 22 / 1.15 / 0 28 / 1.15 / 0 32 / 1.15 / 0 36 / 1.15 / 0 40 / 1.1 / 0 48 / 1.1 / 0	Cheltenham Bold
Headline Feature	For headlines of Feature stories only	24 / 1.15 / 0 28 / 1.15 / 0 32 / 1.15 / 0 48 / 1.1 / 0 56 / 1 / 0	Cheltenham Light (24) Cheltenham Extra Light (28, 32, 48, 56)

Headline Opinion	For headlines of Opinion stories only	18 / 1.15 / .0025 20 / 1.15 / .0025 22 / 1.15 / .0025 24 / 1.15 / .0025 36 / 1.15 / .0025 56 / 1 / 0	Cheltenham Condensed Text Bold (18, 20, 22, 24, 36) Cheltenham Condensed Bold (56)
Body	For communicating the content of a story only	14 / 1.39 / 0 16 / 1.39 / 0 18 / 1.39 / 0	Imperial Book

You made it this far? Well done champ ☺

[See in Figma ↗](#)

[See in Storybook ↗](#)

Components

Components are standard, reusable UI elements with Accessibility, Brand alignment, technical quality, and design quality built in.

We will be releasing more components and patterns on an ongoing basis. [We welcome your feedback](#) on what is currently available and what you'd like to see next.

Our Resources

[TPL Figma Library](#)

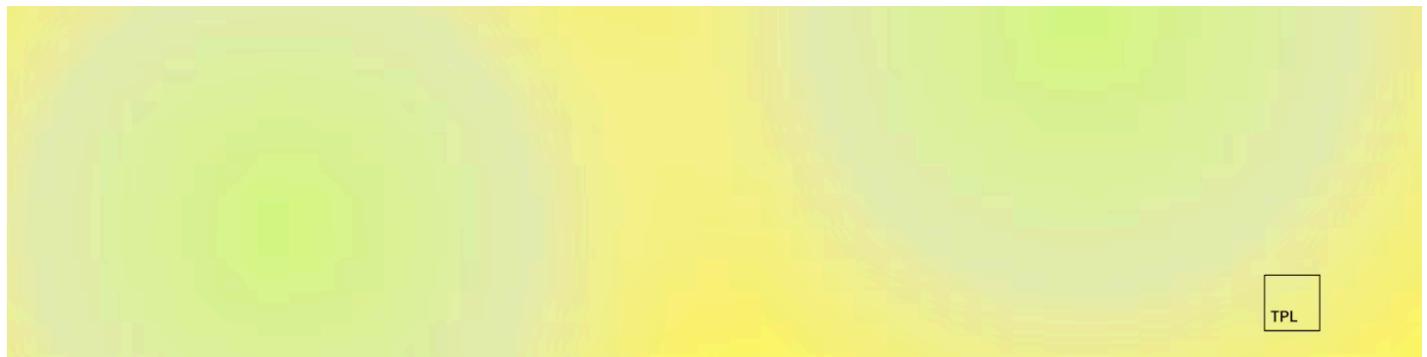
[TPL Storybook](#)

[iOS-TPL](#)

[TPL GitHub](#)

[Our Figma Team](#)

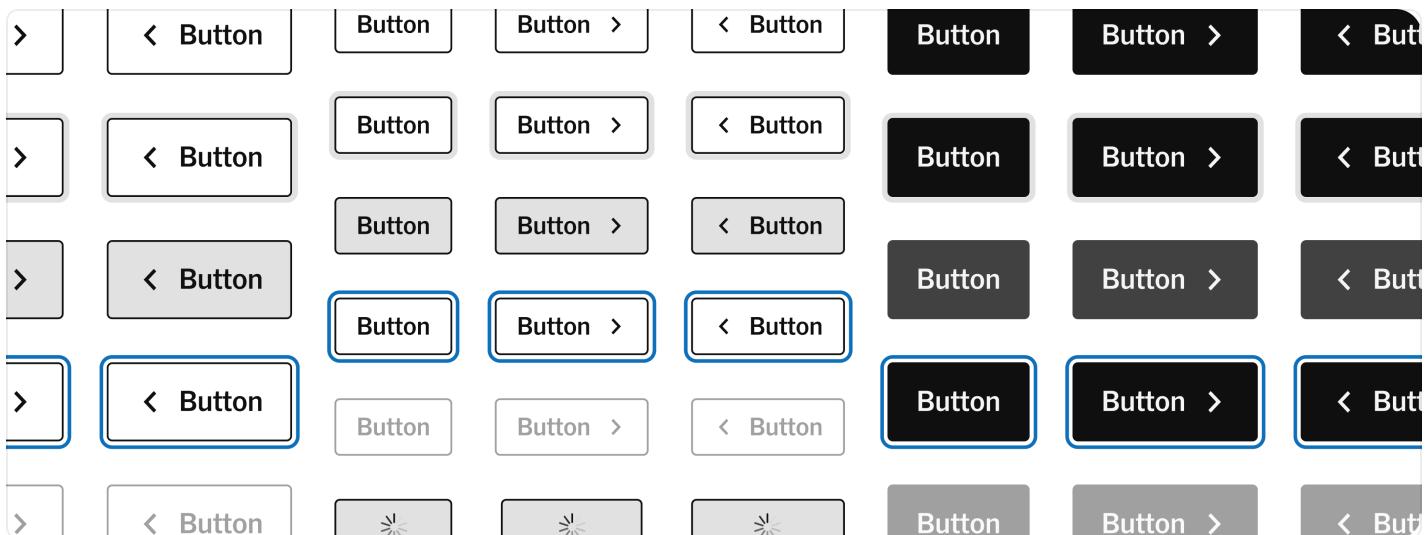
[Design Systems Jira](#)



Buttons

Overview

Buttons allow users to take actions, giving them agency and control over their experience with a tap or click.



Figma

Web

iOS

Android

Components

React / CSS-in-JS

SwiftUI ButtonStyle

No

Components

React / CSS-in-JS

SwiftUI ButtonStyle

No

Components

React / CSS-in-JS

SwiftUI ButtonStyle
(excluding
extraCompact height)

No

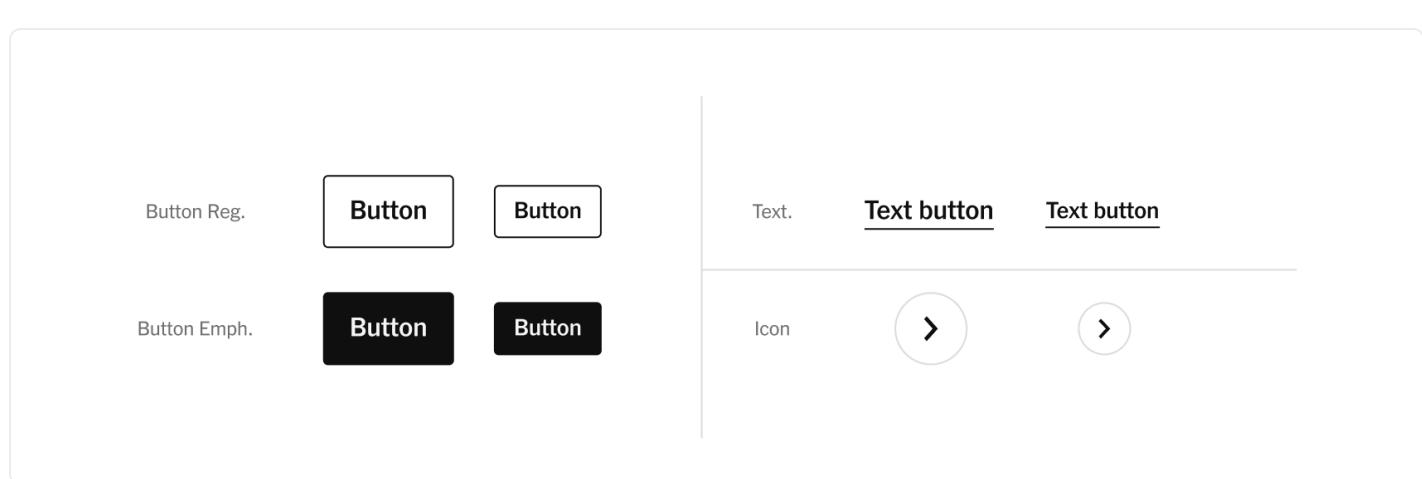
[See in Figma ↗](#)

[See in Storybook ↗](#)

Guidance

Button types

TPL Foundations offers three types of buttons: **Button**, **Text Button**, and **Icon Button**.

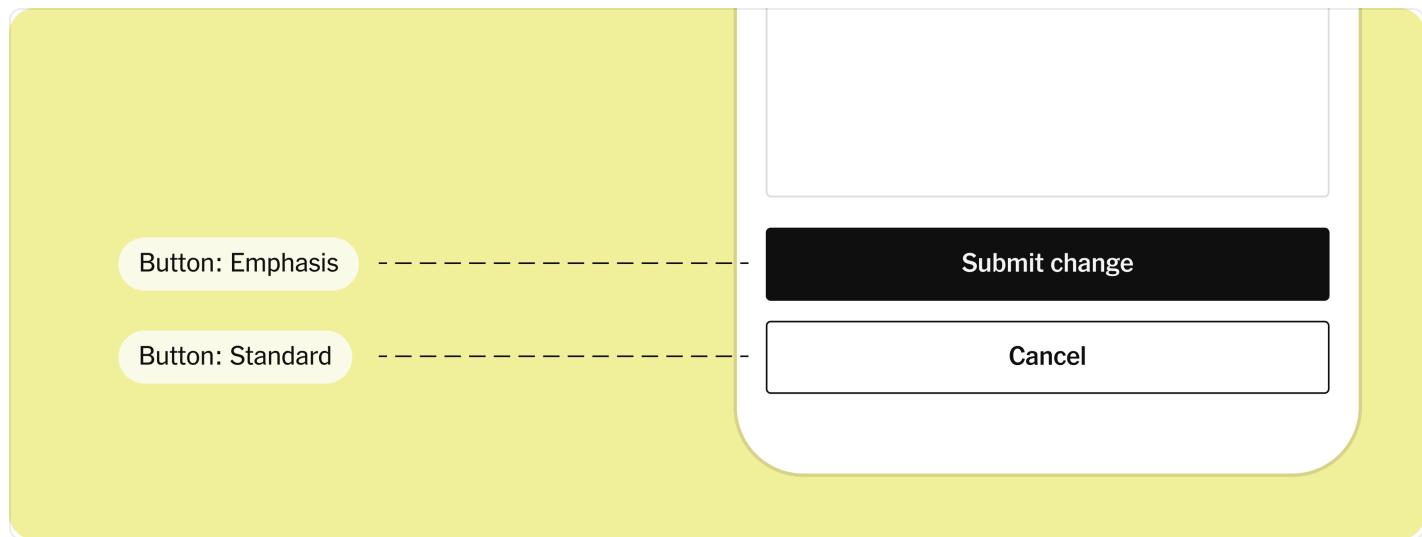


Text Button and Icon Button components include additional sizes not shown here.

These button components are designed for all user-facing surfaces at the Times, including both editorial and non-editorial use cases.

For most actions

Our default ● **Button** in two variants (Standard and Emphasis) should be useful for most actions.



We recommend using **Button: Standard** for all single button experiences, and reserving **Button: Emphasis** for actions that require additional prominence.

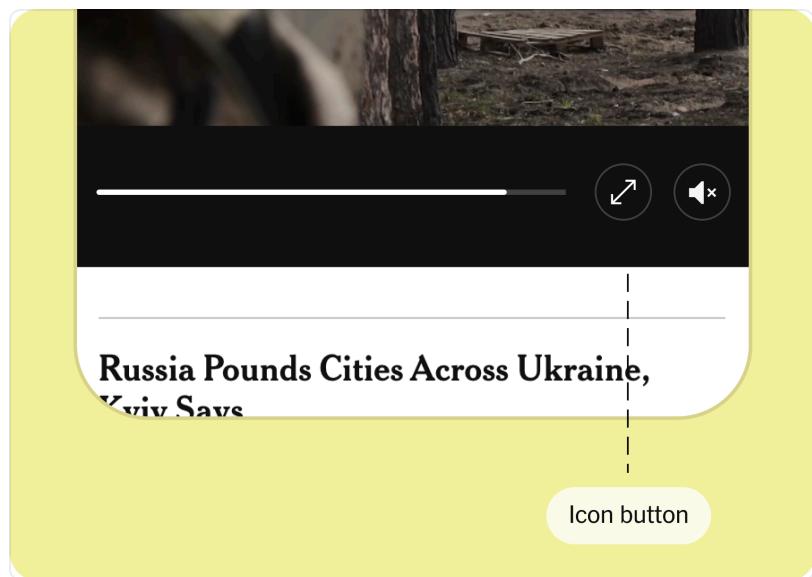
For lower prominence

- **Text Button** is appropriate when you need a user to take an action, without distracting from the content. This is the least visually prominent button.



For quick actions

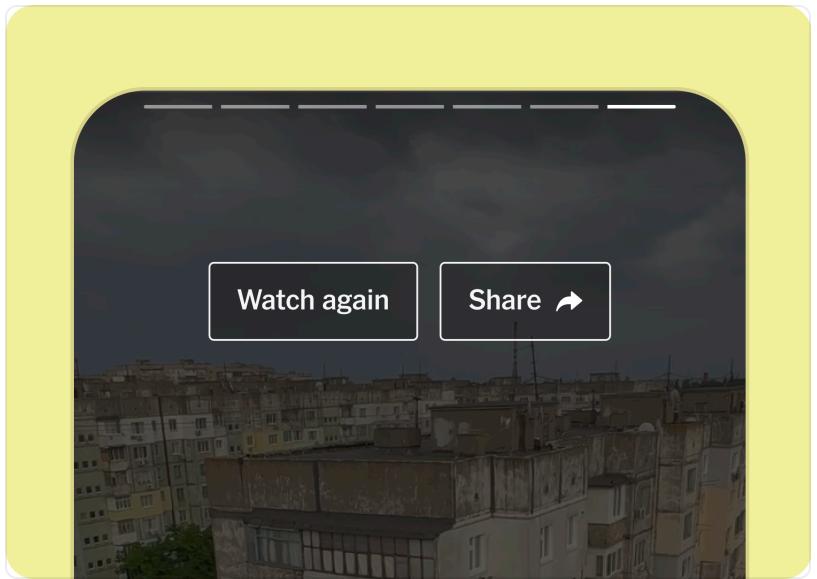
- **Icon Button** is useful for quick actions in compact spaces. Since Icon Buttons lack text labels, they should only contain universally understood and accessible icons and an accessibility label for screen readers.



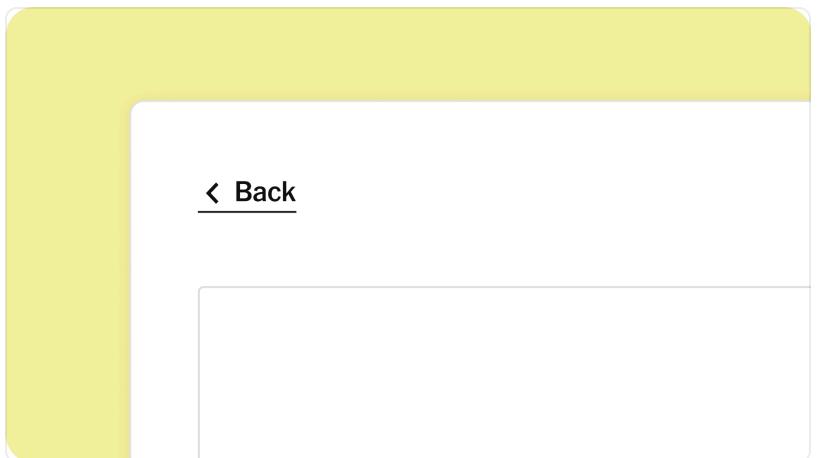
Writing for buttons

Text inside a button is used to describe the action the button performs. Button content is always in sentence case with no punctuation.

Buttons can contain icons in addition to their label.
Buttons are a language-led component, so **trailing icons** are the default setting.



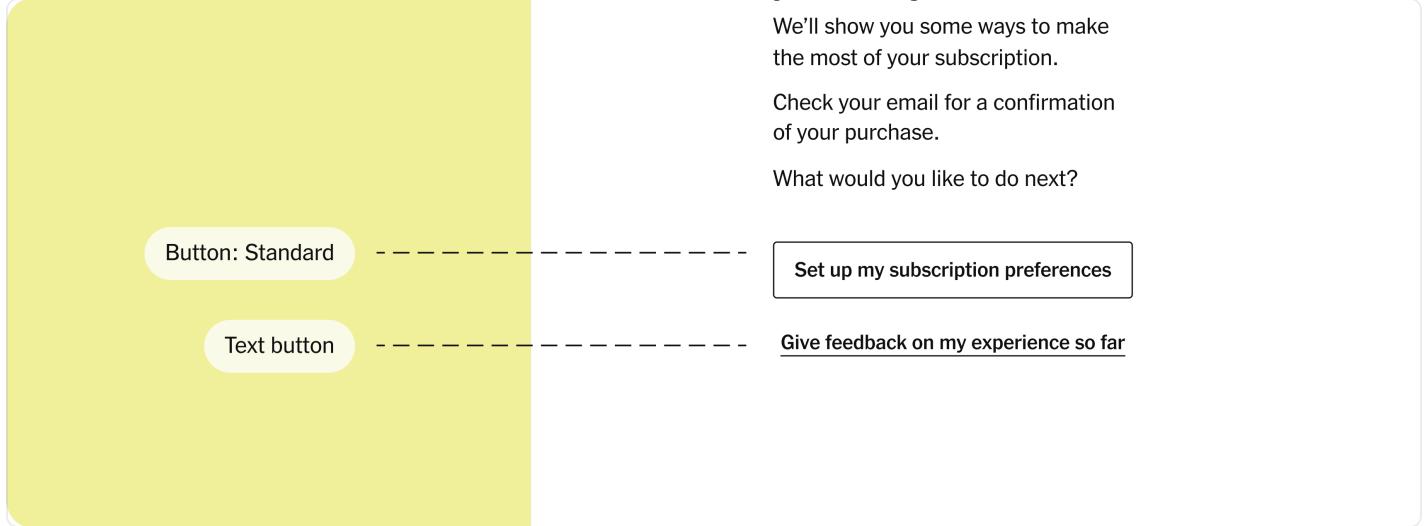
Leading icons can be used for navigational buttons, or other instances where it makes more semantic sense to lead with an icon.



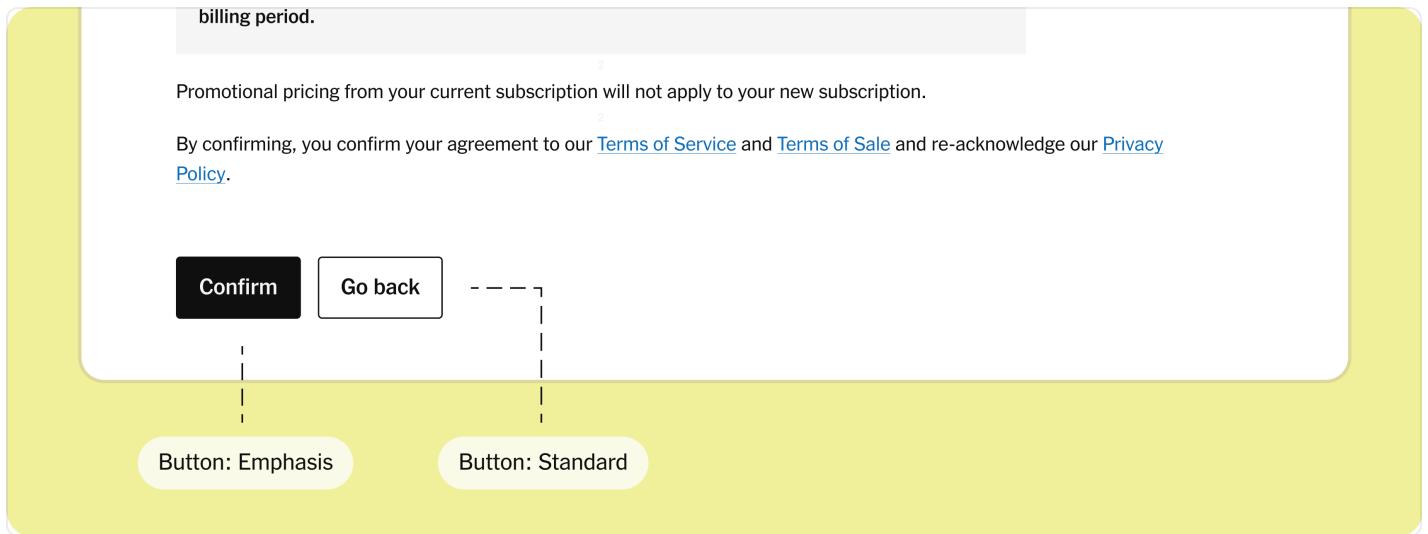
Button hierarchy

All button components have an established hierarchy. For a flow with primary and secondary actions, only use two different buttons in one screen.

Button Standard is the default primary CTA — pair it with **Text Button** for a secondary action.



For scenarios that require a more visually prominent CTA, you can use **Button Emphasis** with an optional **Button Standard** for the secondary action.

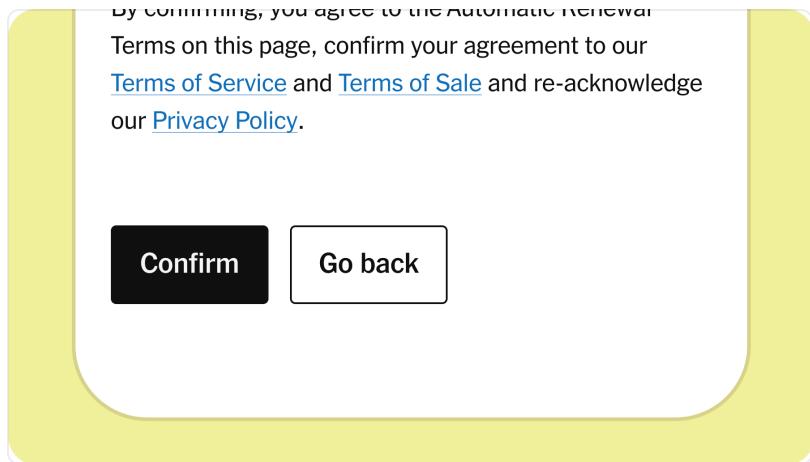


Aligning button groups

We recommend a 1.5 [spacing token](#) spacing between buttons when stacked horizontally or vertically. When aligning buttons, they should be within the same column as the surrounding related content.

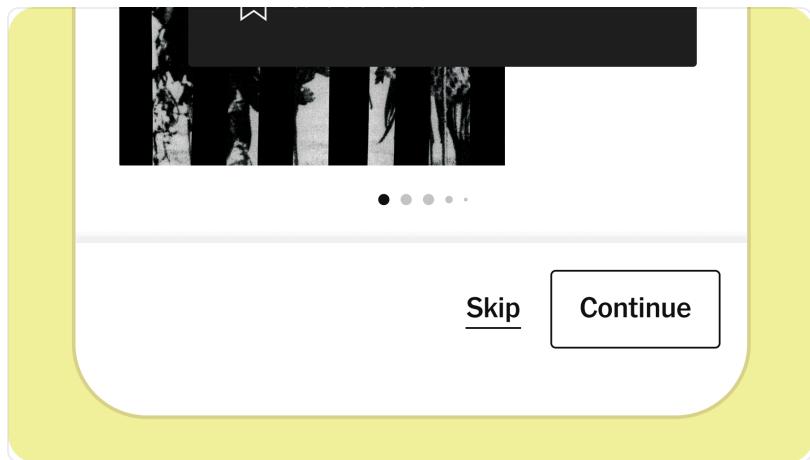
We recommend aligning buttons to the left edge of the column.

In these cases the primary CTA should be on the left.



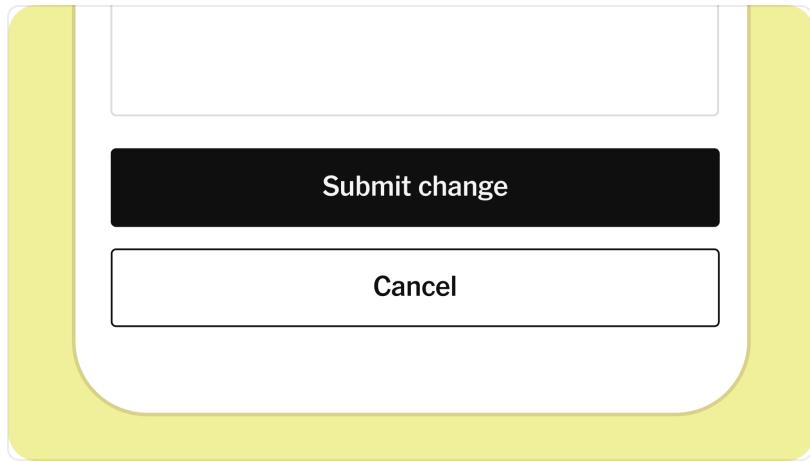
You can also align the button to the right to reinforce progressive direction when moving a user through a flow.

In this case the primary CTA should be on the right.



There may be some cases where a flex-width button group is appropriate.

In this case, the primary CTA should be above the secondary action.



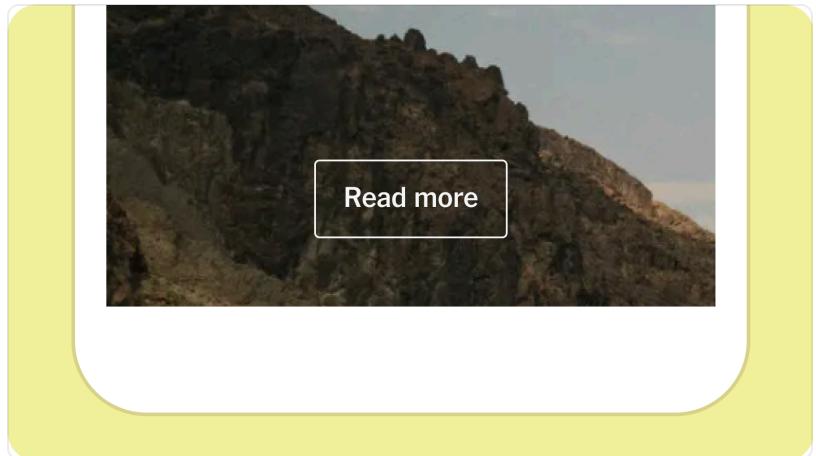
Accessibility

Icon buttons should always be labelled with an accessible aria label. This label should describe what the button *does* rather than what the icon *is*. (ie “Menu” rather than “Hamburger”).

Button keyboard navigation

Key	Action
Tab	Button is focused (non-disabled)
Space, Enter	Button is activated (non-disabled)

If you opt for a custom/non-TPL background behind the **Standard Button Standard**, make sure to [test](#) to ensure a 4.5:1 contrast ratio between them.



Behavior

Buttons use **states** to communicate their interactivity.

Default	Hover	Pressed	Focused	Disabled	Processing
<u>Text button</u>					

Button states have built-in interaction animations that are standard across all buttons and interactive elements.

You made it! Much wow, such amaze 🐾

Button

Text Button

Icon Button

TPL

Button

Overview

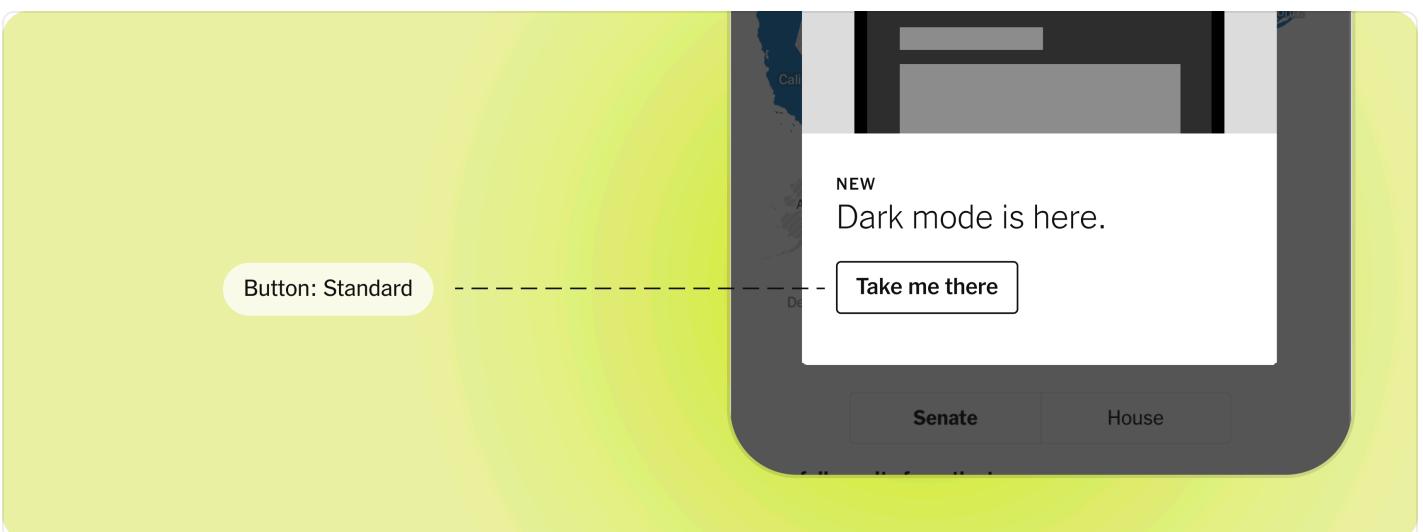
Button is the default button for both editorial and non-editorial surfaces.

Figma	Web	iOS	Android
<input checked="" type="checkbox"/> Components	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> SwiftUI ButtonStyle	<input checked="" type="checkbox"/> No

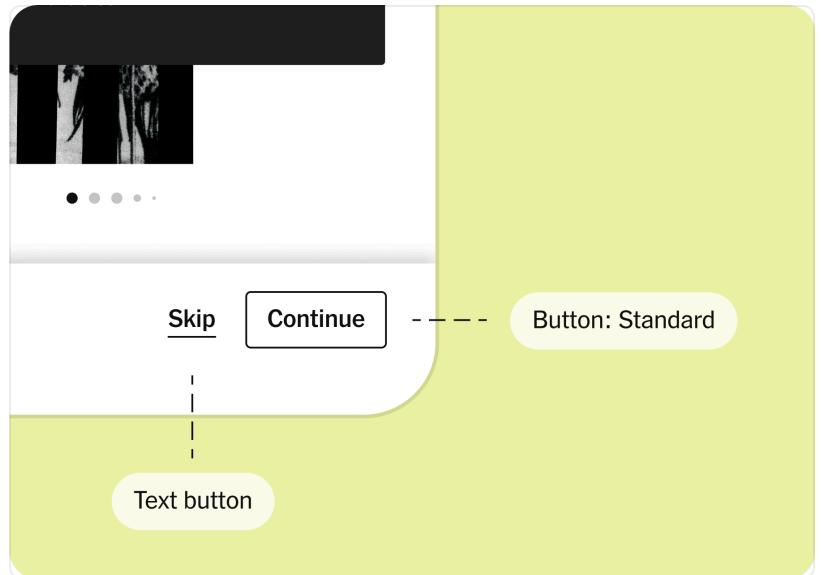
[See in Figma ↗](#)[See in Storybook ↗](#)

Standard

The default button, “Standard”, is a box button with a stroke outline.



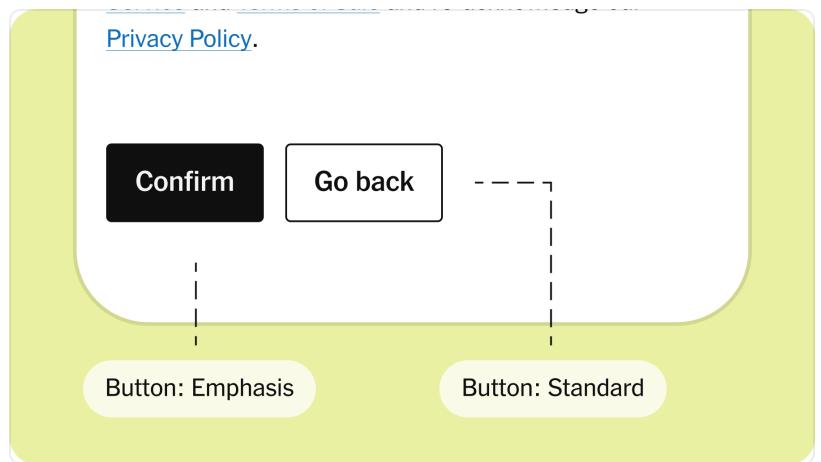
This button can be used as a standalone primary CTA, or as the most prominent CTA when paired with Text button.



Emphasis

Emphasis buttons are filled box buttons. They are used for a more prominent call to action, and are ideal for experiences where the CTA requires more visual weight. They are always the primary action in an experience.

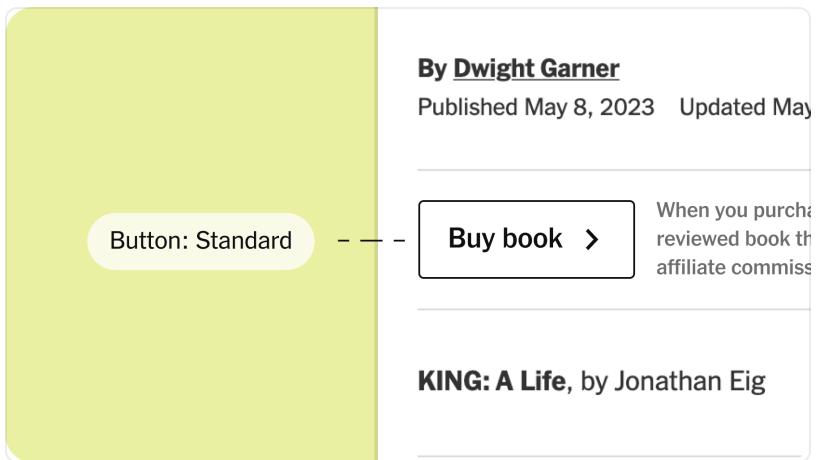
Emphasis buttons can be paired with Standard button for a hierarchy with more visual emphasis than Standard + Text.



Content

As [mentioned](#) in the general guidance, writing for Button should be sentence case with no punctuation.

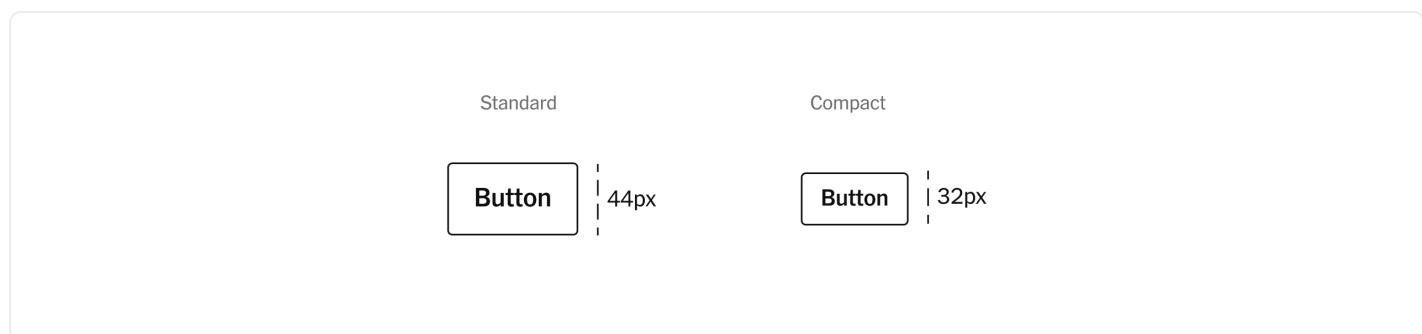
Trailing icons (language first, icons after) are the default for Buttons with an icon. Leading icons can be used for navigational purposes.



Specifications

Height

Like all TPL buttons, Button comes in two heights: **standard** and **compact**. They differ in component height and font size. The height of the standard button is set to meet accessibility tap target requirements.



The compact button height is intended for surfaces with less available space. Its 32px height still meets accessibility guidelines.

Width

Button has two width variants: **content-width** and **flex-width**.

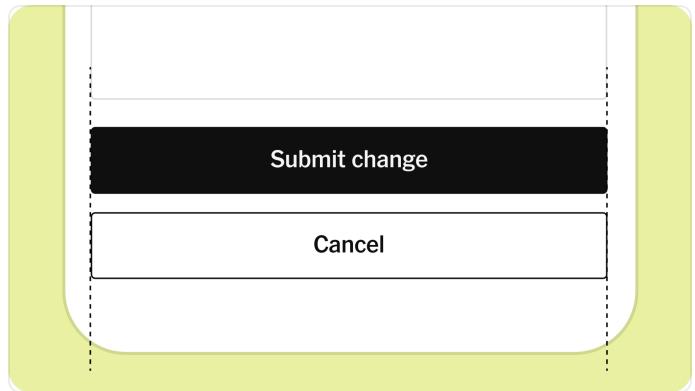
Content

The default content-width is determined by the length of content and the left and right padding.

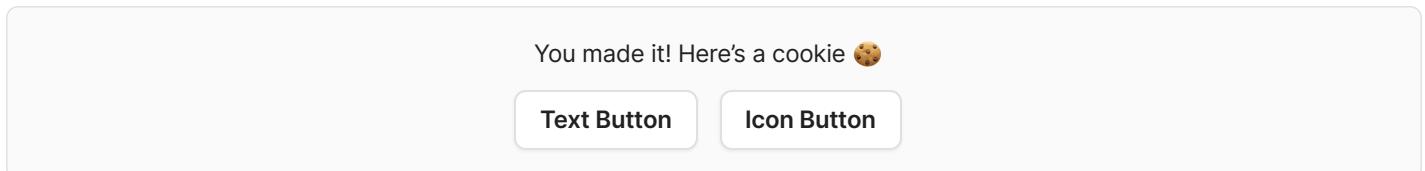


Flex

Flex-width buttons can be used when you would like the button width to take up the entire width of the screen or a custom container. This is ideal for smaller screen sizes, to ensure an ample tap target size for users.



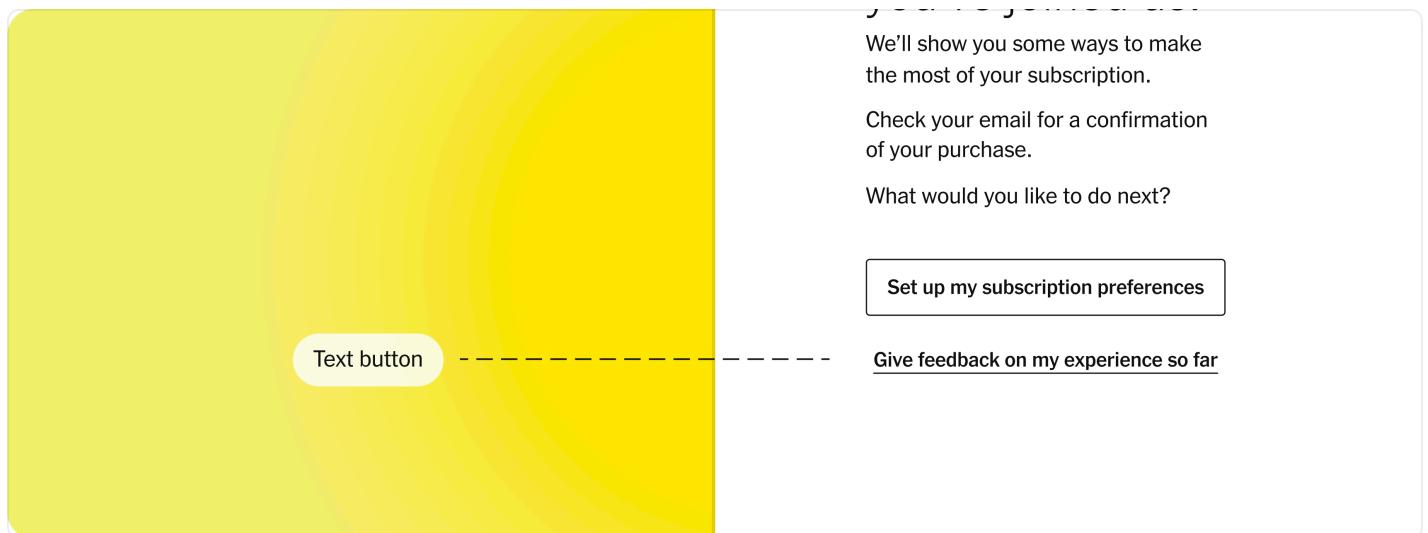
We encourage using content-width buttons whenever possible. However, there may be exceptions where flex-width buttons with custom-set widths may be more appropriate.



Text Button

Overview

Text buttons are the least prominent call to action. They are used to avoid distracting readers from important content.



Figma

Web

iOS

Android

Components

React / CSS-in-JS

SwiftUI ButtonStyle

No

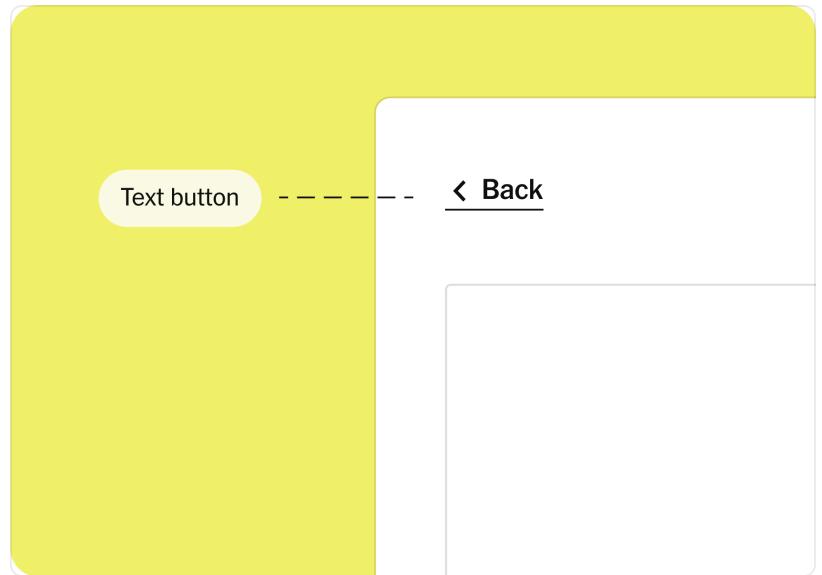
[See in Figma ↗](#)

[See in Storybook ↗](#)

Content

As [mentioned](#) in the general guidance, writing for Text Button should be sentence case with no punctuation.

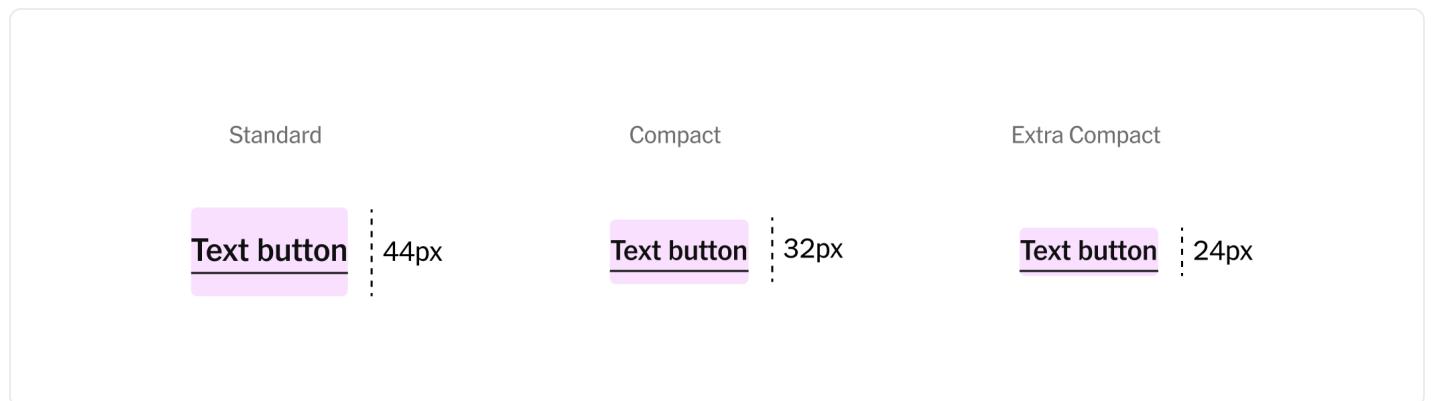
Trailing icons (language first, icons after) are the default for Text Buttons with an icon. Leading icons can be used for navigational purposes.



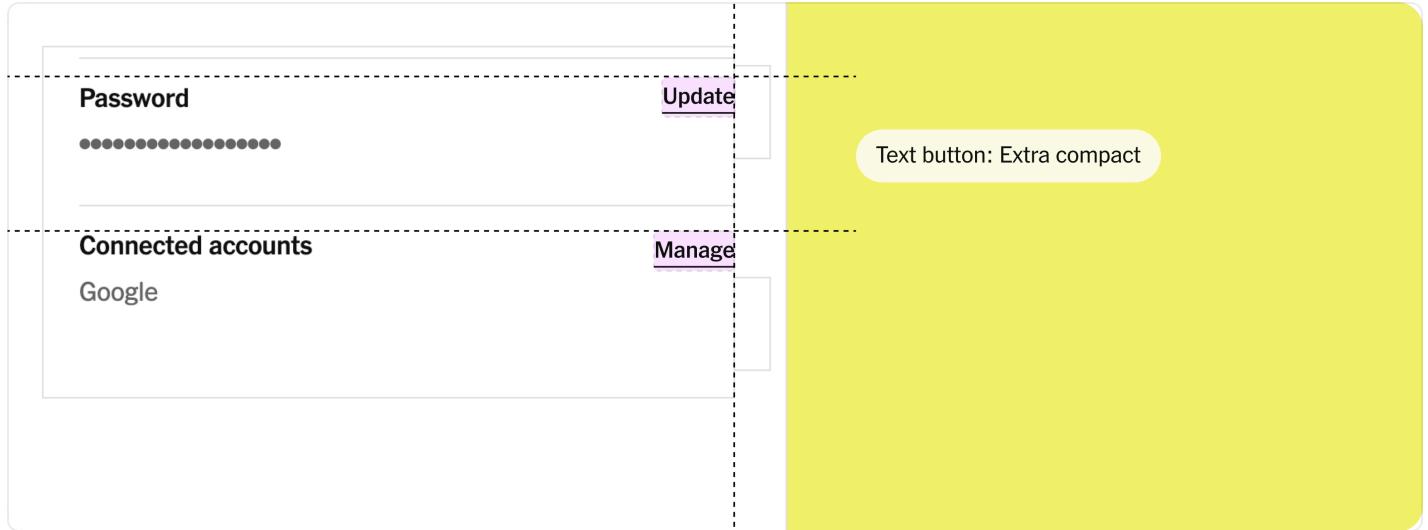
Specifications

Height

Text button comes in three sizes: **Standard**, **Compact** and **Extra Compact**.



Extra Compact is intended to accommodate tighter inline use cases. We strongly recommend extra compact text buttons as accessible alternatives to standalone inline links.



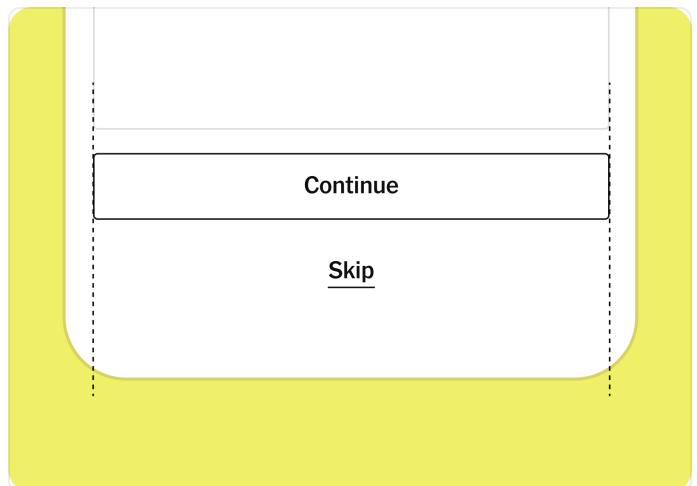
Width

Same as with Button, Text Button has two width variants: **Content-width** and **Flex-width**.

The default **Content-width** is determined by the length of content and padding.



Flex-width buttons can be used when you would like the button width to take up the entire width of the screen or a custom container. This is ideal for smaller screen sizes, to ensure an ample tap target size for readers.



We recommend keeping text buttons at content-width whenever possible. Since text buttons have no visible bounding box, wider text buttons don't communicate the tap target as clearly.

That's all they wrote 

Button

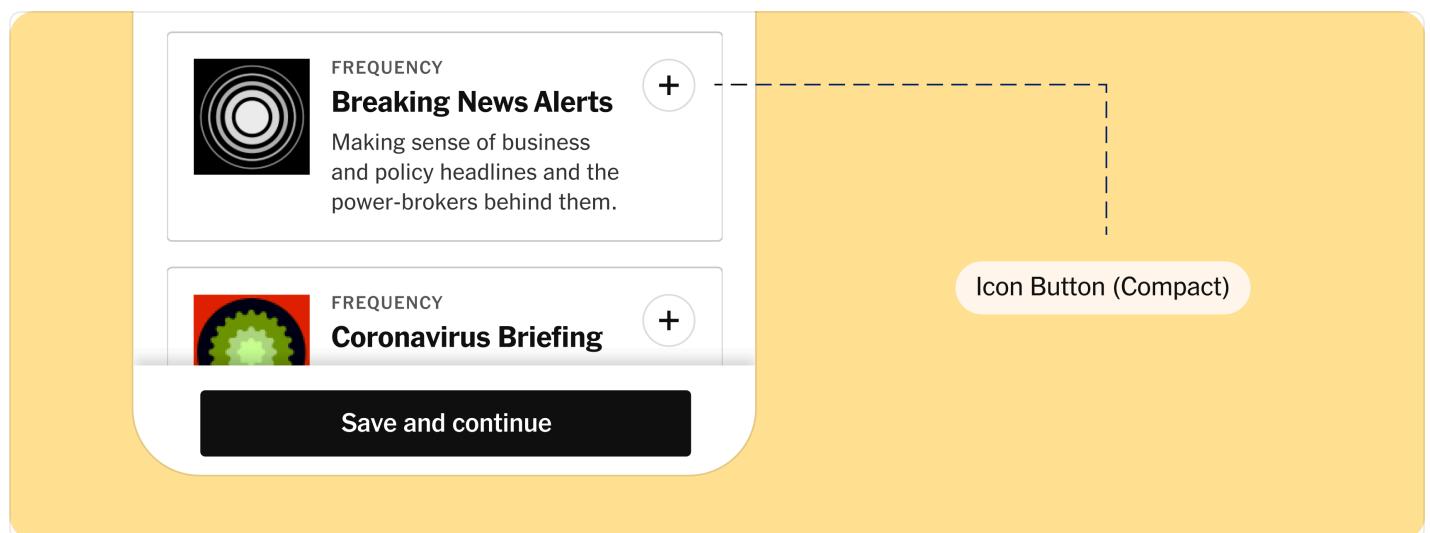
Icon Button

TPL

Icon Button

Overview

Icon buttons are used for toggles and quick actions take quick action with a tap or click, such as in tool belts and media controls. Icon buttons can be paired together for related actions.



Figma

Web

iOS

Android

Components

React / CSS-in-JS

SwiftUI ButtonStyle
(excluding
extraCompact height)

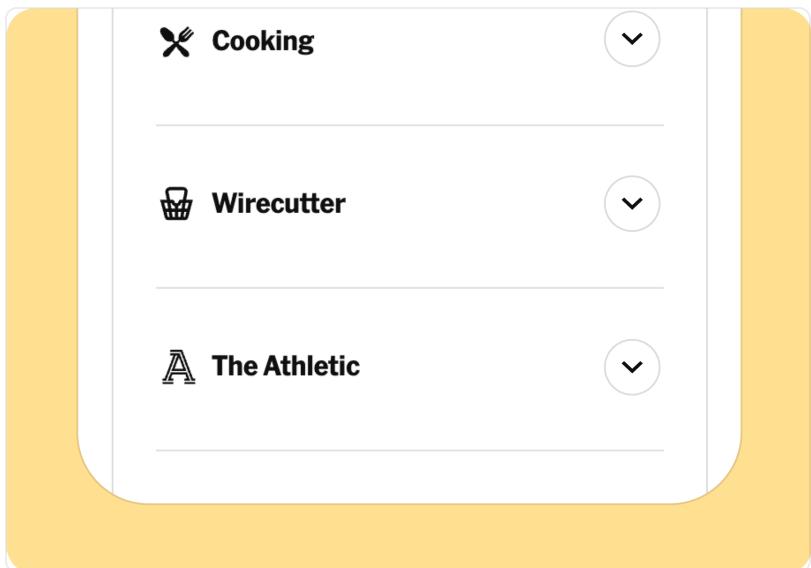
No

[See in Figma ↗](#)

[See in Storybook ↗](#)

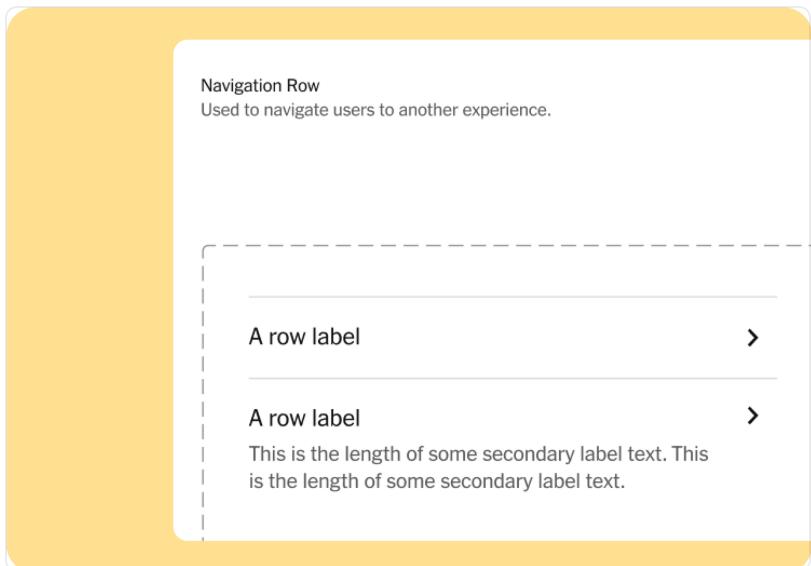
Standard

The default button, “Standard”, is a button with a tertiary stroke outline.



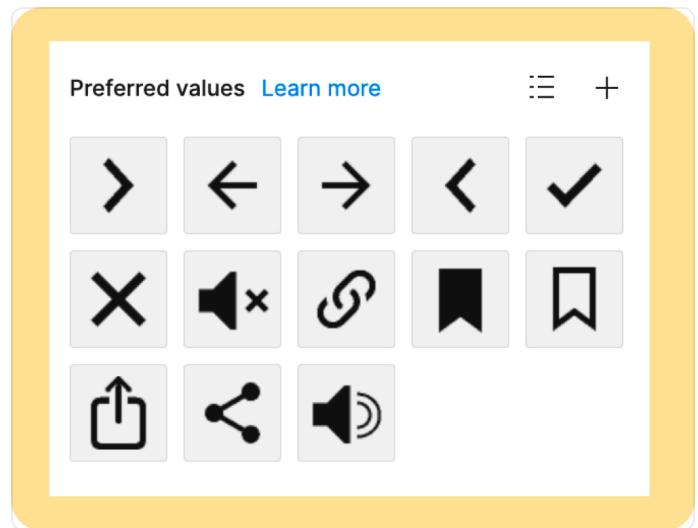
Light

Light Icon Buttons do not have a stroke on default. These buttons are ideal for experience that are visually dense, or that have multiple interactive elements. TPL's Row component uses Light Icon Button to indicate navigation.



Content

Although all [TPL icons](#) can be used in Icon button, we recommend using a subset of icons that have been vetted for this use case. These are maintained as “preferred values” in our [Figma library](#).

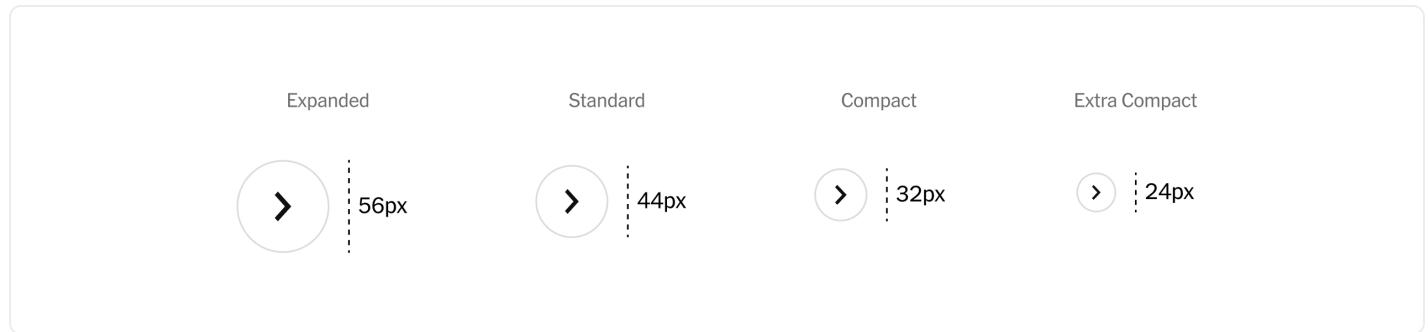


 We recommend using **Button** or **Text Button** with an icon whenever possible. Icon buttons should be used sparingly, and only in situations where the icon is universally understood.

Specifications

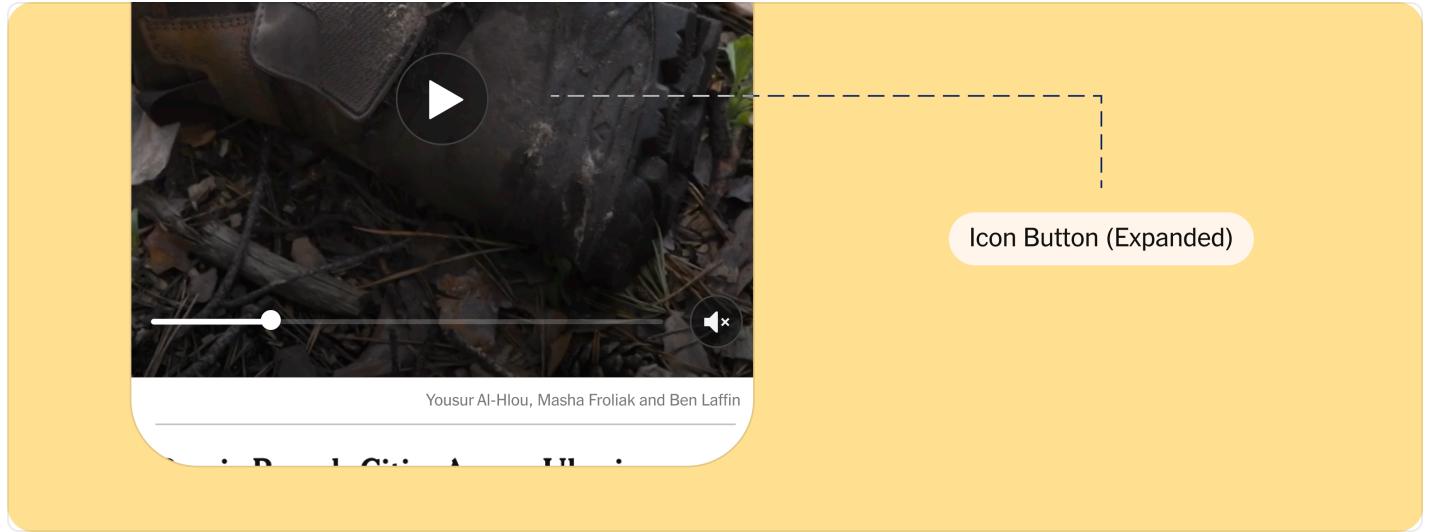
Height

Icon button comes in four sizes: **Expanded**, **Standard**, **Compact** and **Extra Compact**. They correspond directly with the four available icon sizes.



Standard and compact sizes match directly with **Button** and **Text Button**.

The **expanded** size can be used for larger CTAs and mobile-specific actions.



The **extra compact** size is for tighter quick actions, such as navigation on a burst.

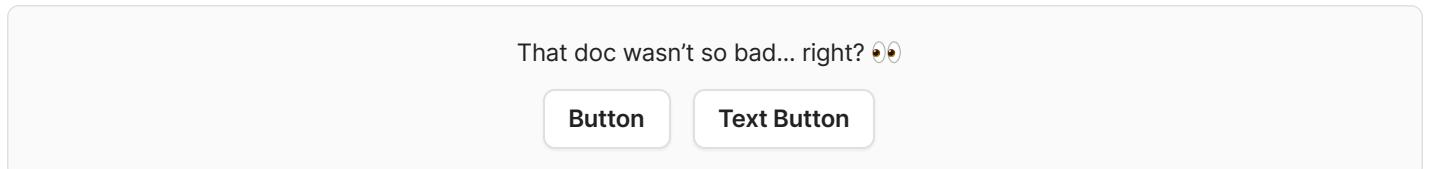
new paperback releases: Annie Proulx's history of the world's wetlands, Tess Gunty's award-winning novel of a young woman in crisis, David Maraniss's biography of Jim Thorpe and more.

Here are six paperbacks we recommend →



Width

Icon button widths are always content-width. Icon buttons should be a perfect circle.

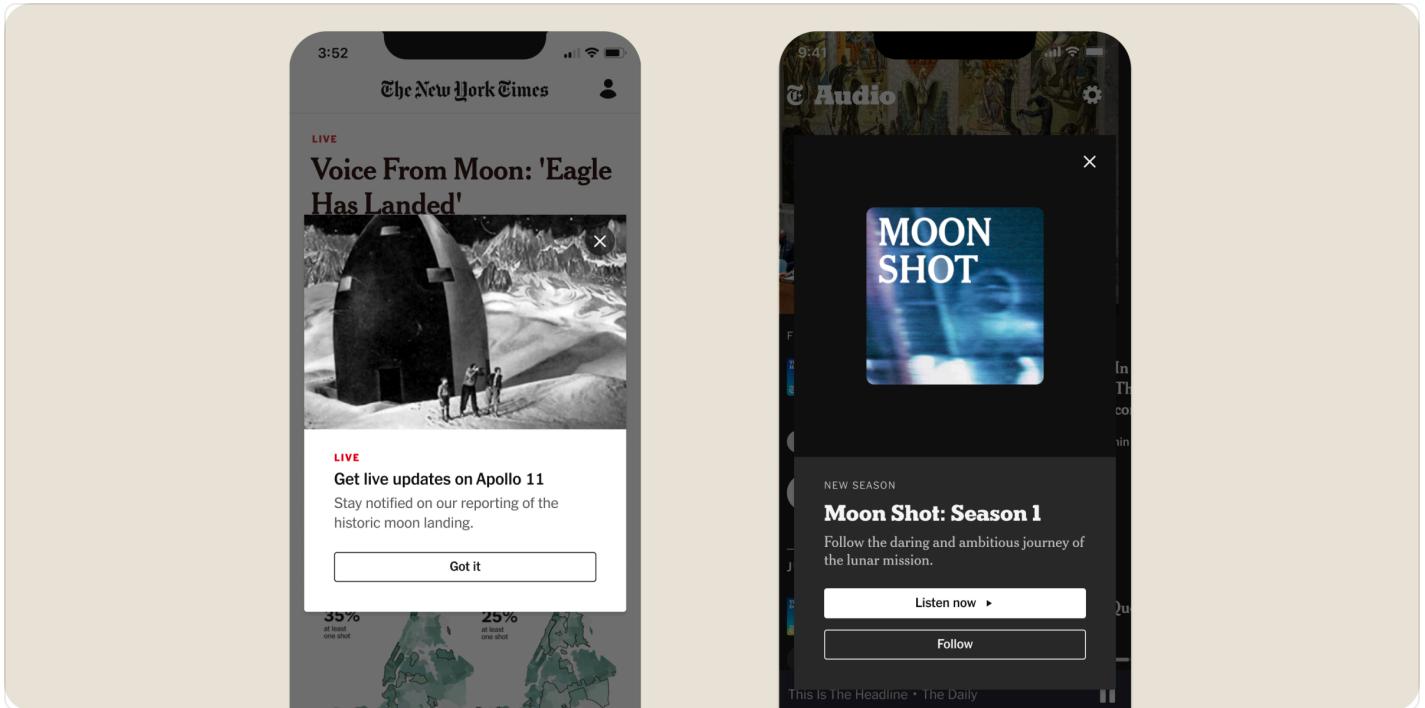




Dialog

Overview

Dialog is a messaging component which appears as a modal covering the entire screen. It is often used to promote features or direct users to something new.



Figma

iOS

Web

Android

Components

SwiftUI

React / CSS-in-JS

No

[See in Figma ↗](#)

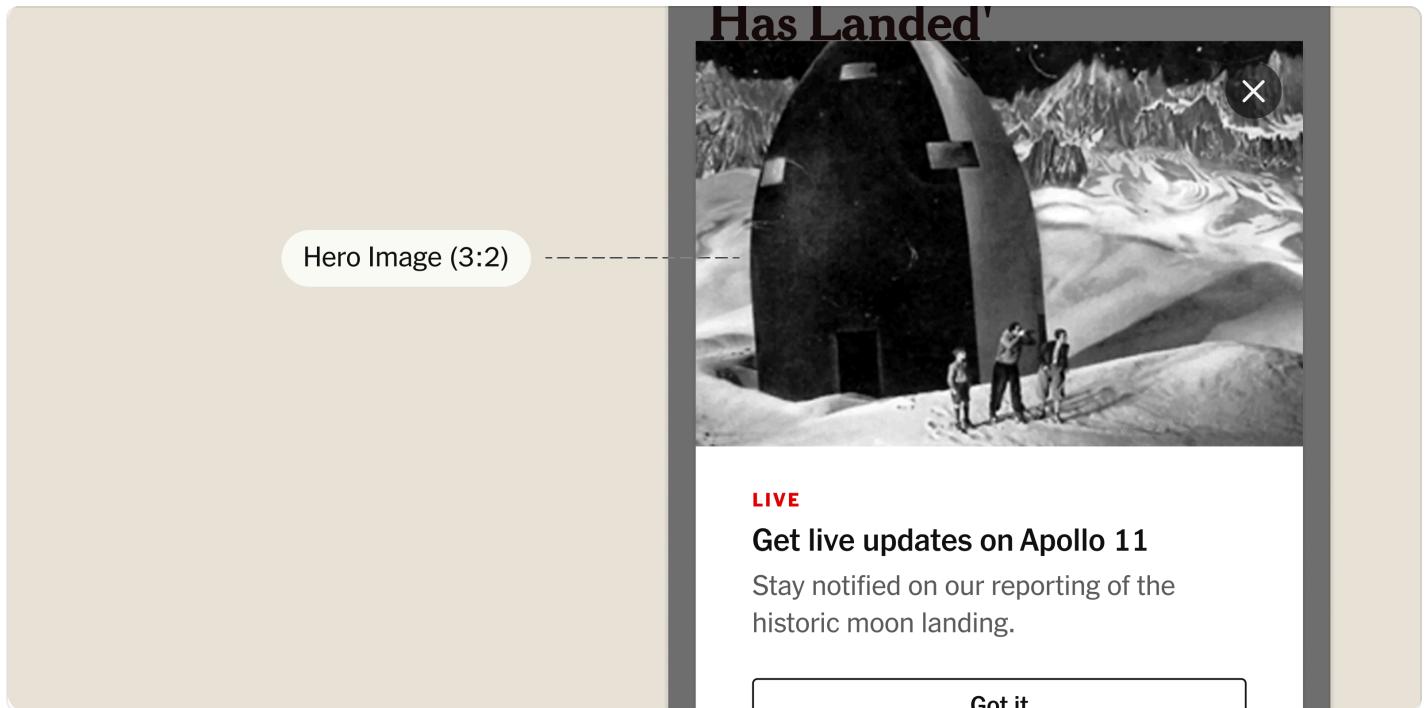
[See in SwiftUI ↗](#)

[See in Storybook ↗](#)

Guidance

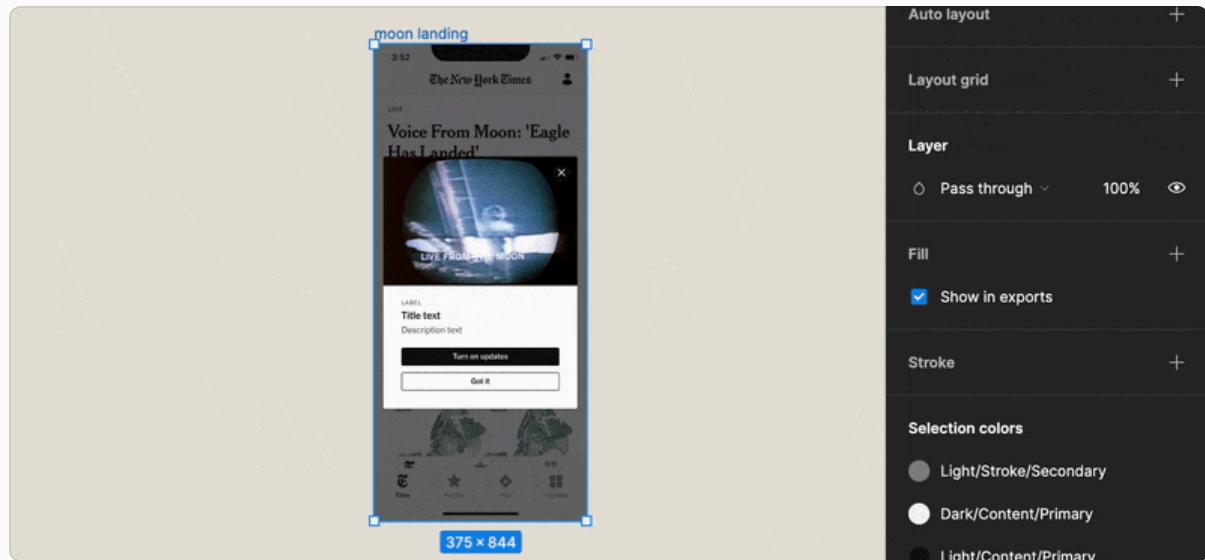
Hero image

By default we offer a Hero slot (`hero`) that can accommodate graphics with any aspect ratio. We recommend using either static PNG images or [Lottie files](#) for motion graphics.

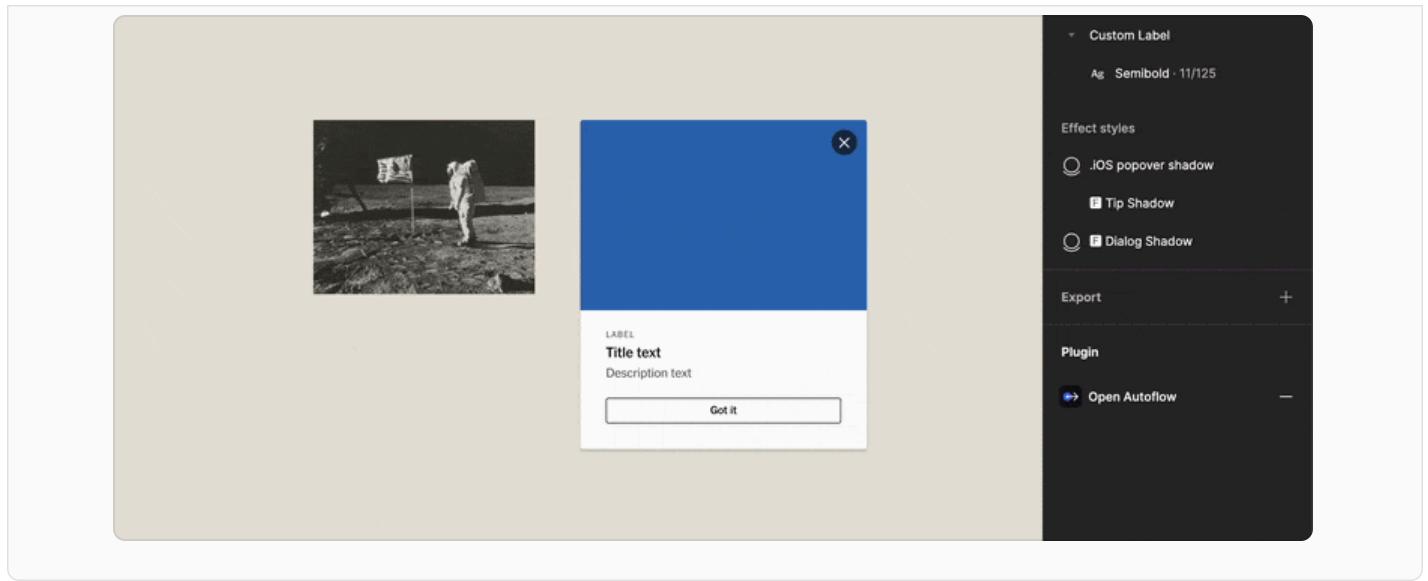


Our Figma component offers locked **1:1** and **3:2** image ratios to help designers mock up graphics for those preset ratios.

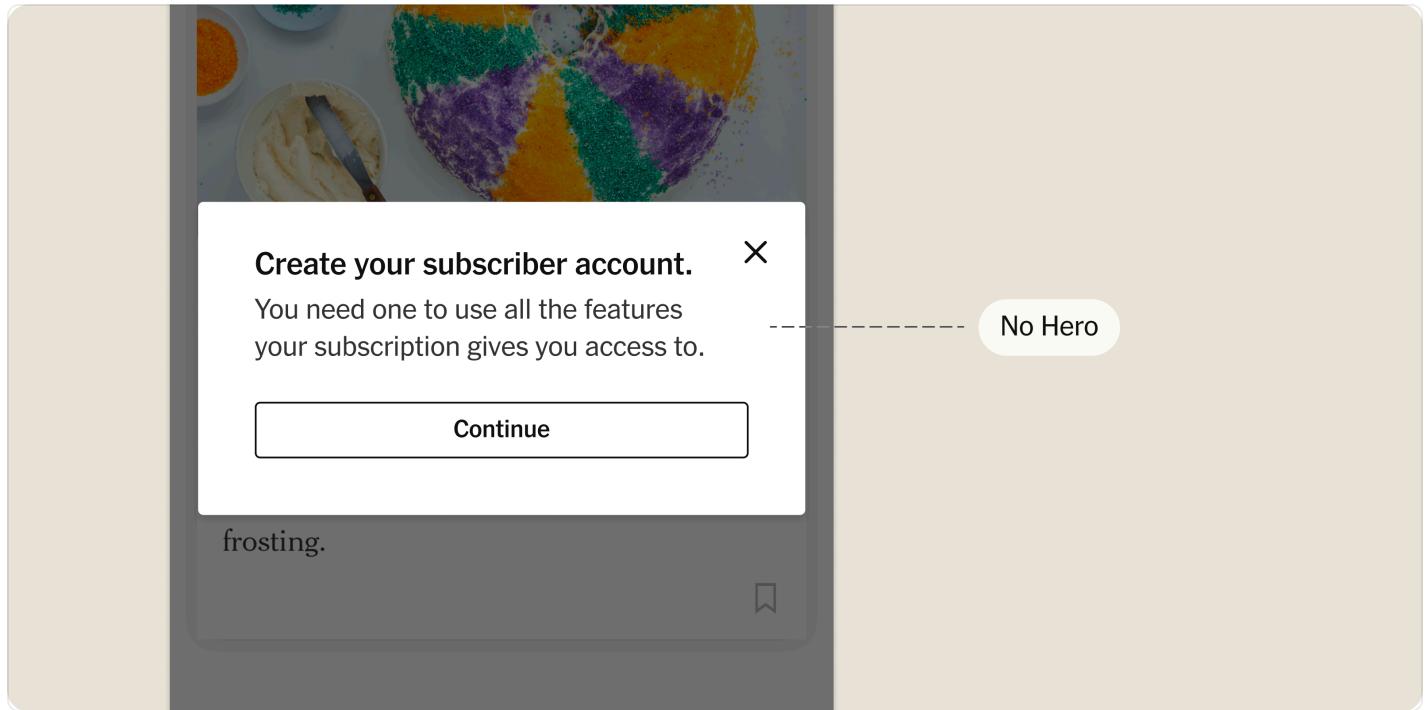
"Custom" aspect ratio lets you adjust the hero's height by changing the height of the component manually.



Paste an Image fill on the Hero frame to replace our placeholder hero with your own.

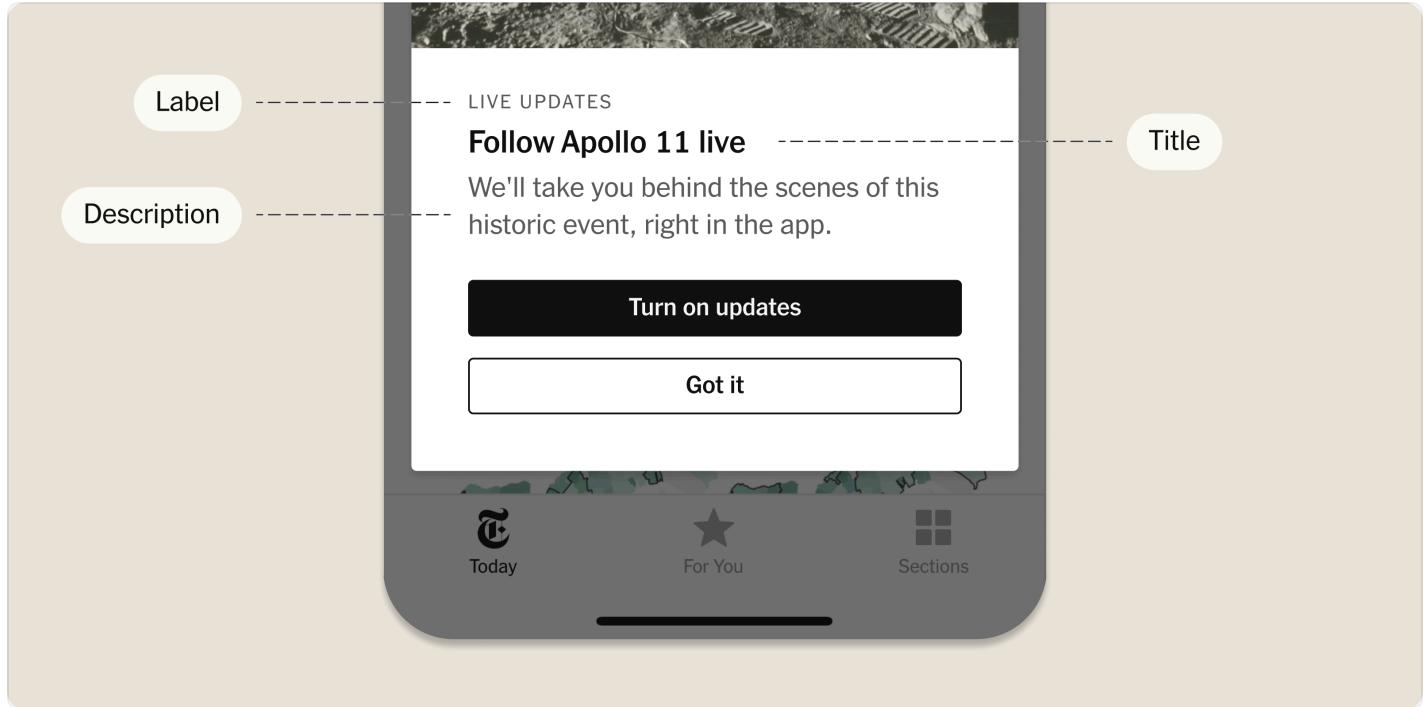


Hero is optional and can be disabled for a more content-lead message.



Content

Dialog offers required `@Title` text (`title`), optional `@Label` text (`label`), and optional `Description` text (`description`).

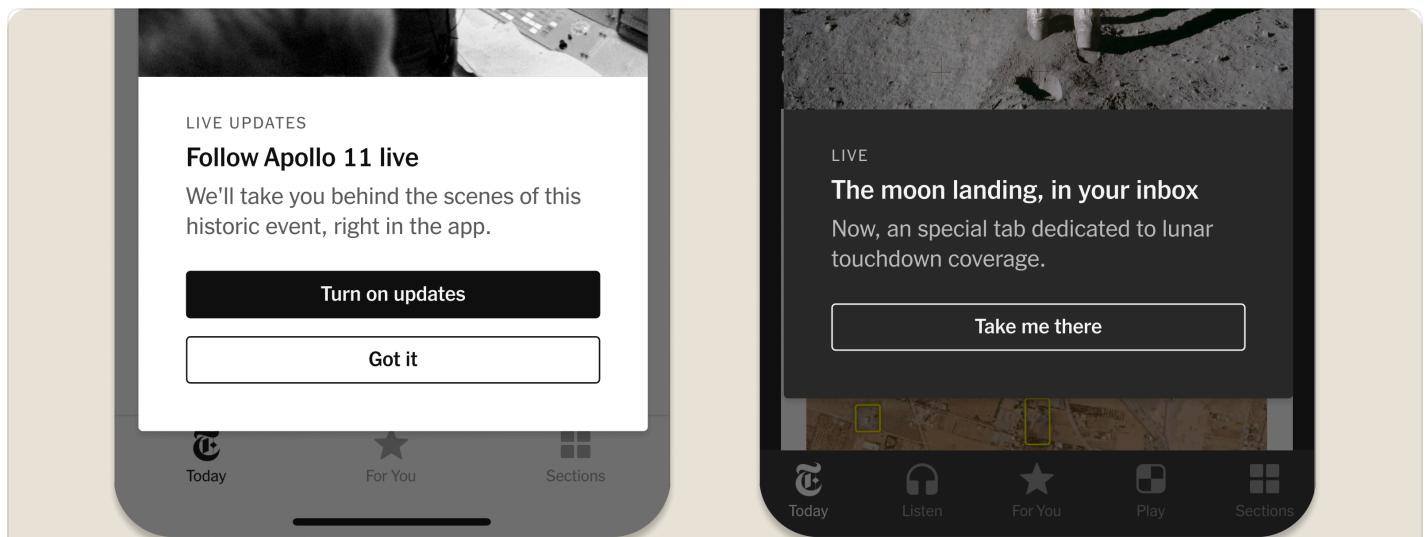


Label can only be used alongside a Title, and should be limited to one or two words.

Try to keep the length of Title under 80 characters. Description can be used for longer, more detailed messages, but should still maintain an [efficient and personable tone](#).

Actions

Dialog requires an action button at the bottom of the content. By default, it uses a single [Standard Button](#). Two buttons of decreasing hierarchy, [Emphasis Button](#) and [Standard Button](#), can also be used.



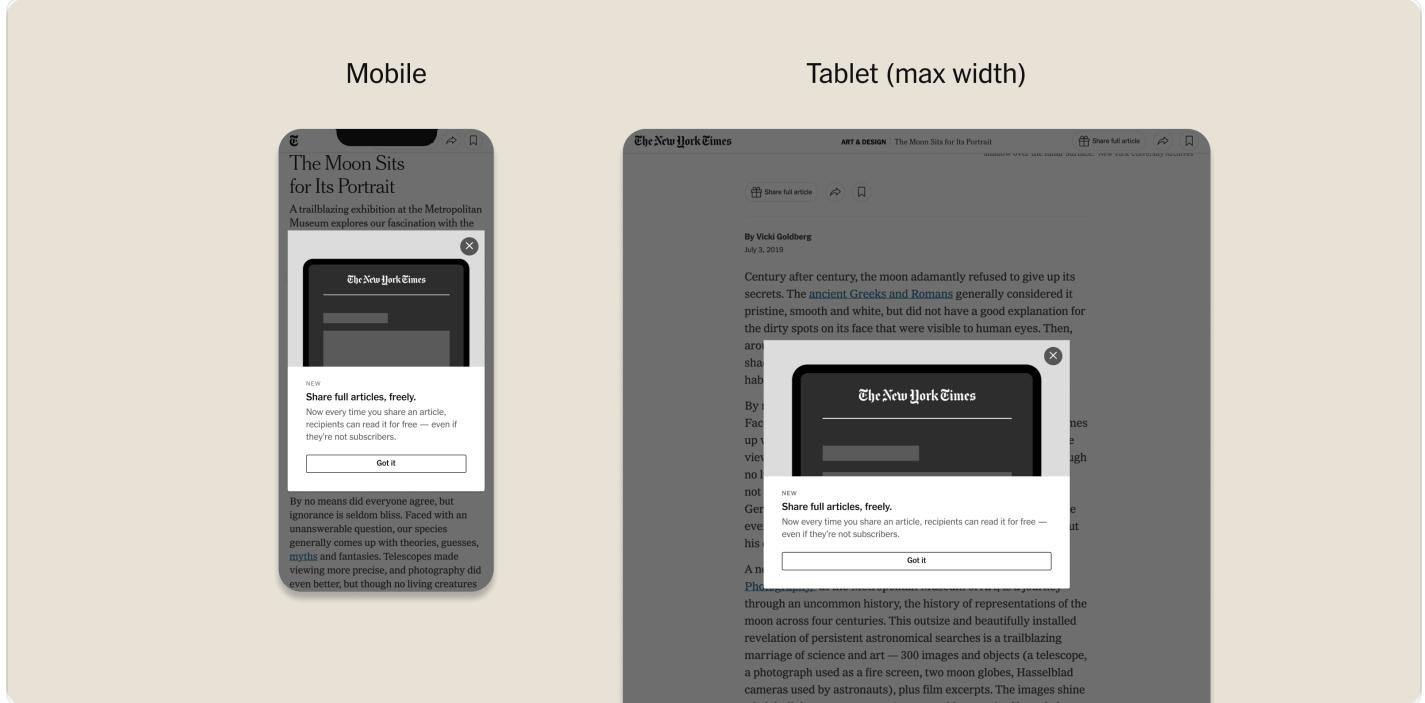
Standard and Emphasis Buttons

Standard Button

Specifications

Layout

Dialogs fill the screen horizontally. To account for larger devices like iPad, Dialog has a max width of 536px.



The Figma component includes a **16px margin** on each side by default: designers can set the width of the Dialog to the screen size.

Dialogs are horizontally and vertically centered aligned by default, which can be changed if necessary.

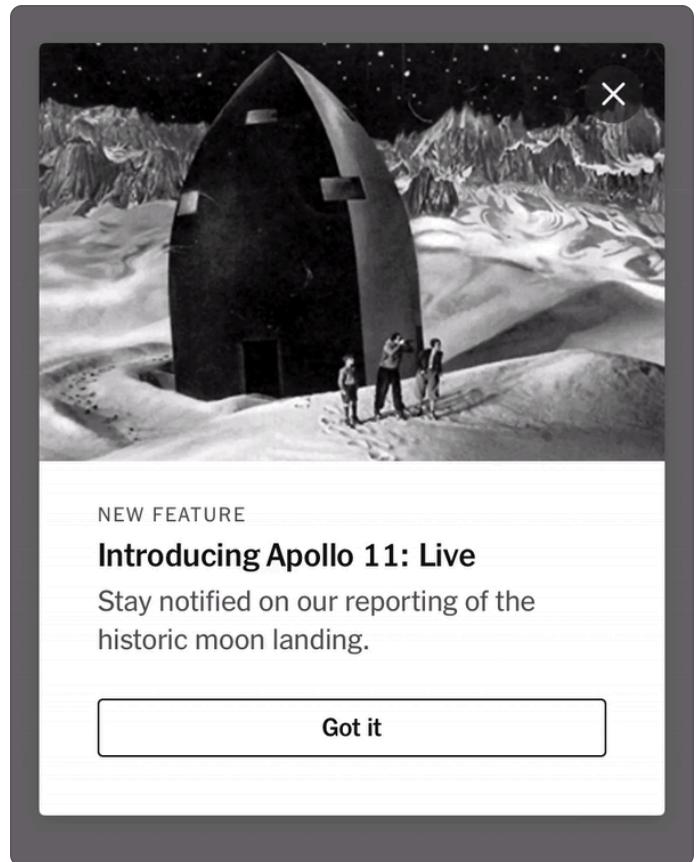
You can change the space in between label, title, description and the space between content and actions.

Behavior

Dialogs can be dismissed in three ways:

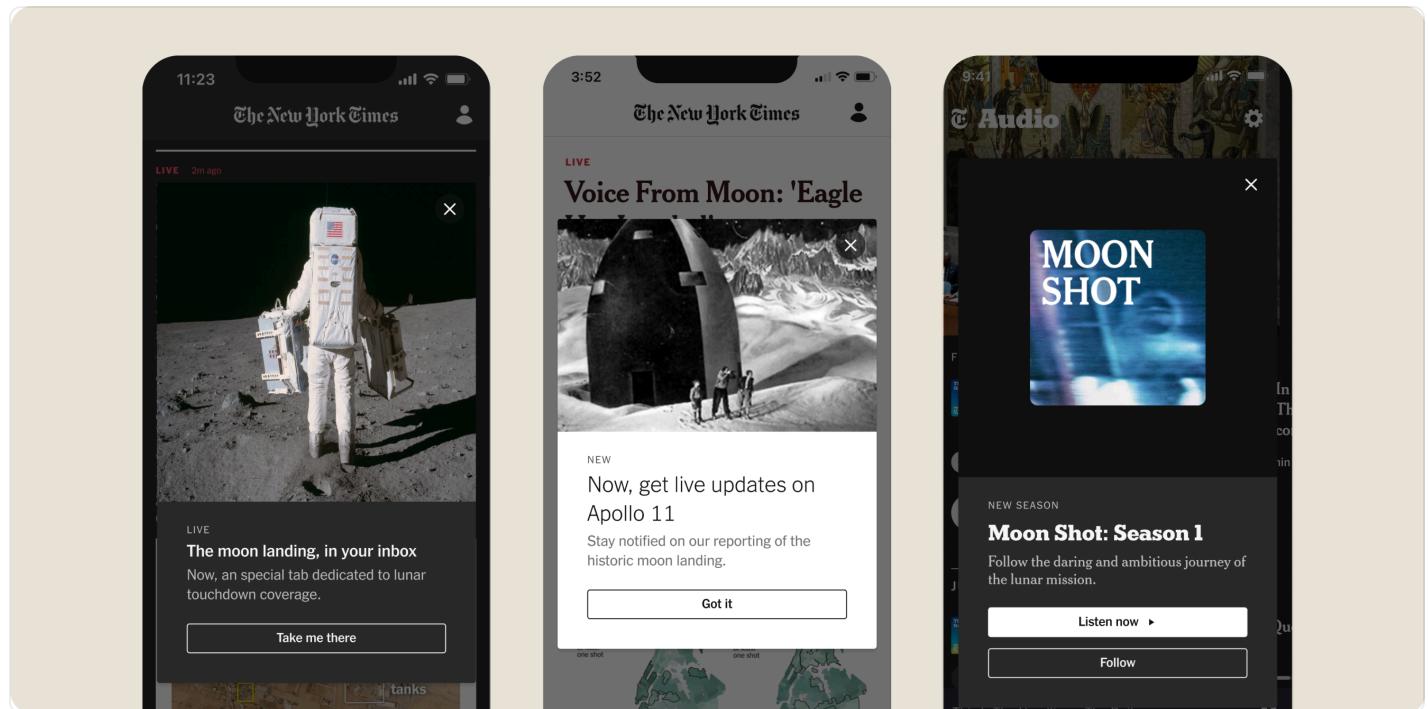
1. The close Icon button in the top right
2. The Action button, which is set to "Got it" on default.
3. Tapping anywhere on the screen outside of the Dialog container.

When dismissed, Dialogs will briefly fade out.



Typography

While Dialog has default type settings, you *can* implement your own type. Simply adjust the type style to suit your use case. We recommend using [TPL type tokens](#).



You can also change the text alignment to be left, right, or center.

In use

- **Cooking:** [Instacart Onboarding](#)
- **All NYT iOS apps:** [App Tracking Transparency](#)



Big thanks to Jamie Liao and the AppX team for contributing the starter kit Product Messaging Component to this work.

Thank you to pilot team members Alvin Andino, Lavi Tang, Rimar Villasenor, and Grace LaRosa from Audio, to Véronique Brossier, Paige Collins, Mimi Fang from User Space, and to John Andrews, Mark Keefe, Paroma De, Olivia Cheng and Kyle Oye from App Platforms for collaborating, testing and supporting TPL's work on this component.

Additional thanks to Zev Eisenberg for code review and multiple consults; and Goran Svorcan for contributing to ios-games' CustomModal.

Special shout out to Cam Pulsford for his comprehensive support and guidance through TPL's first published iOS view.

If you're using Dialog, let us know. We'd love to chat about your experience with the component and feature your collaboration here.

Have a question or suggestion? Want to chat more about Dialog? We're [happy to help](#).

Forms

Overview

Forms encompass at least one input as well as a method to submit. For ease of data collection and better accessibility, almost all form experiences should be wrapped in a container.

The image displays two mobile application screens side-by-side, illustrating the use of form containers. The left screen shows a newsletter sign-up form with an email input field containing 'name@example.com' and a black button labeled 'Subscribe to newsletter'. The right screen shows a login form with an email input field containing 'dietrich.lubs@gmail.com', a password input field with a 'Show' link, a 'Log in' button, and a 'Forgot your password?' link. Both screens are set against a light green background.

[See in Figma ↗](#) [See in Storybook ↗](#)

Guidance

To contain a form experience

Generally, all instances of inputs should be wrapped in a container except for when data is not sent to the backend. This does not change the visual appearance of the form, but it utilizes native technology to make data collection simpler and to properly communicate with assistive technologies.



When to use a container

- Account creation
- Login



When not to use a container

- Filter
- Sort

- Comments

- Changing the view of a screen

Your updates have been saved ✓.

[See in Figma ↗](#)

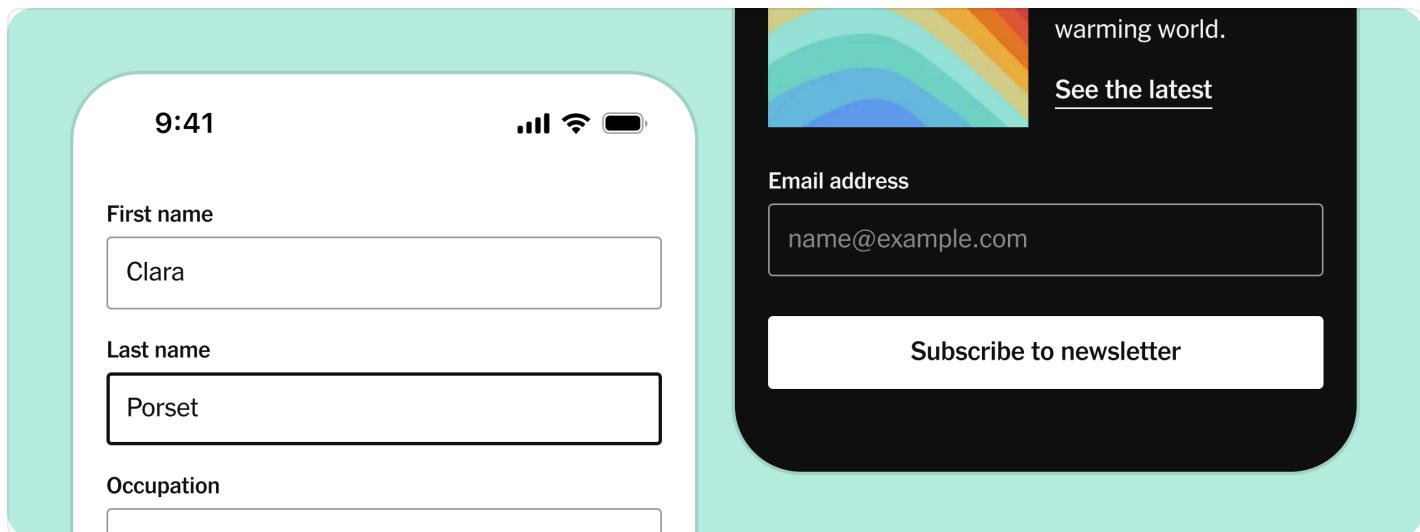
[See in Storybook ↗](#)

Text input

Overview

Text input collects plain text from readers. It can also collect text that follows a certain pattern, such as an email address.

Text input is also the foundation in which every other input is built upon, making the visual style and basic functionality of all TPL inputs cohesive.



Figma

Web

iOS

Android

Components

React / CSS-in-JS

No

No

[See in Figma ↗](#)

[See in Storybook ↗](#)

Guidance

For labels

All inputs require a label to state what information should be entered. This label is visually displayed by default, though it can be hidden only if it is replaced by other text that visually acts as a label.

Labels should be succinct and written in sentence case (Source: [Product Tone](#)).

The image displays two side-by-side screenshots of mobile devices. The left screenshot shows a form with three text input fields labeled 'First name', 'Last name', and 'Occupation'. The right screenshot shows a form with a single text input field labeled 'Enter your email to receive updates from DealBook Summit.' and a button labeled 'Get updates'.

9:41

First name

Last name

Occupation

Ing-wen of Taiwan.

Enter your email to receive updates from DealBook Summit.

name@example.com

Get updates

For required inputs

Use the `required` property to specify if an input must be filled. Text input automatically indicates required fields by adding the text "(required)" to the label, which can optionally be removed.

The image shows a mobile device screenshot with a single text input field labeled 'First name (required)'.

9:41

First name (required)

For errors

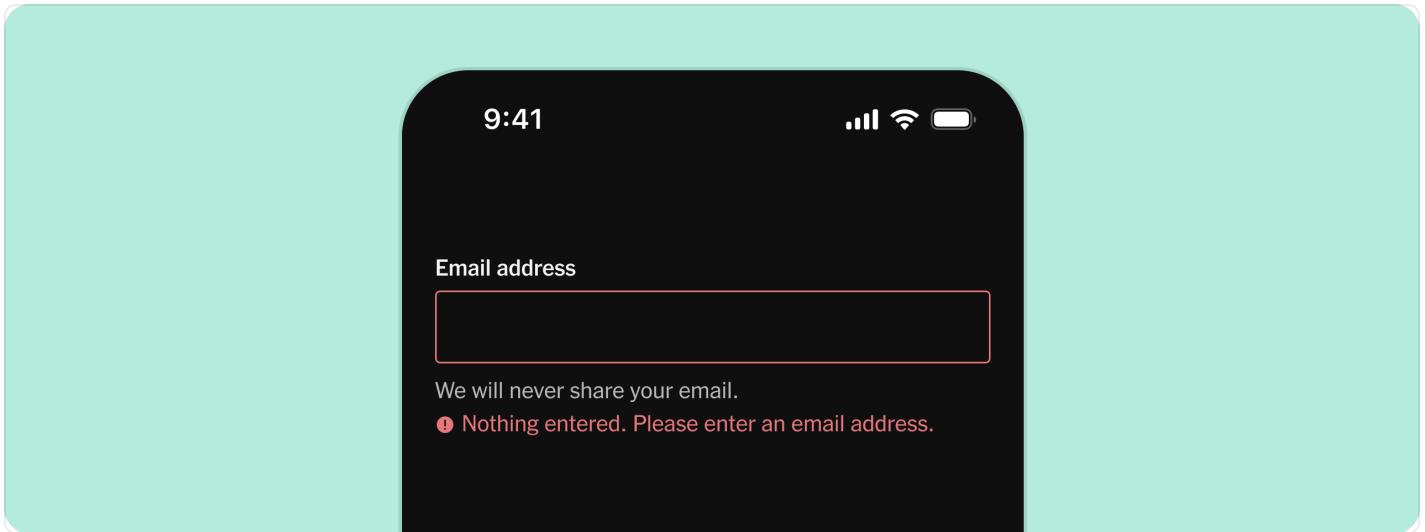
Inputs often need to be validated for various types of errors. Make sure to account for the following errors:

Missing value	When nothing is entered and the input is required (e.g. Leaving password blank when creating an account)
Type mismatch	When something is entered that does not conform to the specified input type (e.g. Entering a name when the input expects an email)
Pattern mismatch	When something is entered that does not match a specified pattern (e.g. Entering a non-NYT email when the pattern requires "@nytimes.com")
Too long	When an entry exceeds the maximum character length (e.g. Exceeding a 60 character limit for a name input)

Too short	When an entry does not exceed the minimum character length (e.g. Not meeting 20 characters for a personal bio input)
Bad input	When something is entered that the input can't convert, like a text when number is expected (e.g. Entering a character inside a zip code)

Inputs respect each platform's native error validation wherever possible. You can additionally specify your own error messages and validation, which is especially useful when an error can only be found by talking to a server (e.g. creating an NYT account with an existing email).

Default error messages are always provided with native error validation, though you can optionally write your own copy. You must always include your own copy when specifying your own error validation. Prioritize offering the reader a way to resolve the issue or accomplish their goal (Source: [Product Tone](#)). Sometimes it also makes sense to tell users what the problem is—if so, present the problem first (Source: [Inclusive UX Writing](#)).

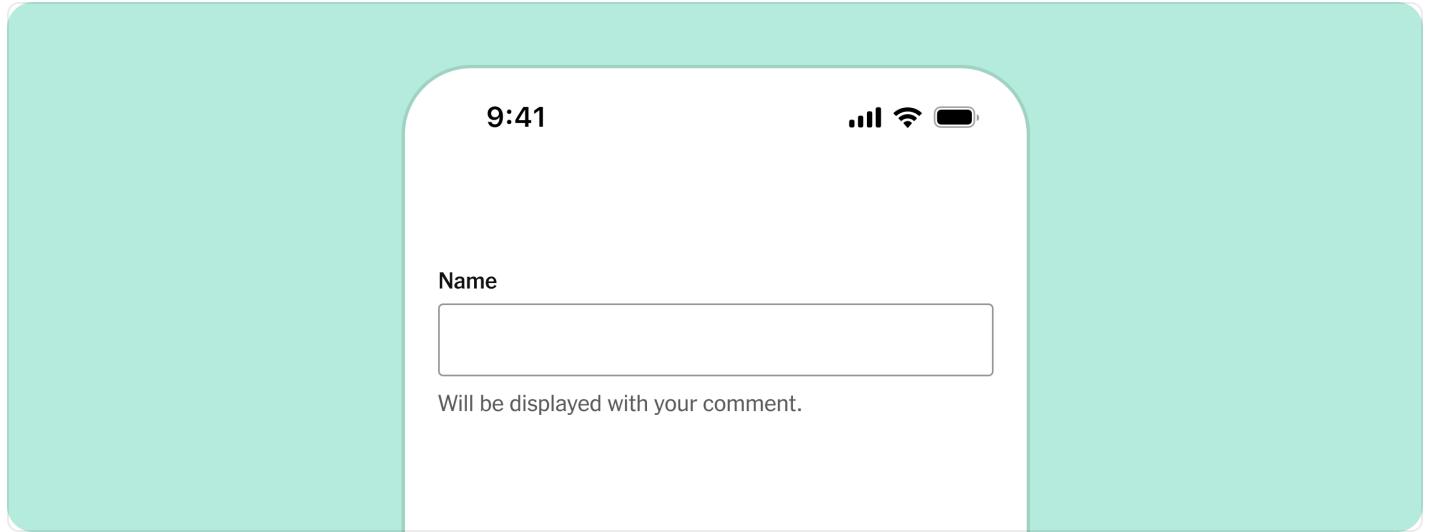


For added context

Consider adding helper text when a reader may need more support to correctly fill an input. Common examples include:

- When an input must meet certain criteria (e.g. minimum length)
- Explaining how the input will be used and reassuring the reader that their information will be respected properly

Use caution with directional language (left, down, etc.). Page layouts might change across devices, and directional language isn't useful for people using screen readers. (Source: [Inclusive UX Writing](#)).



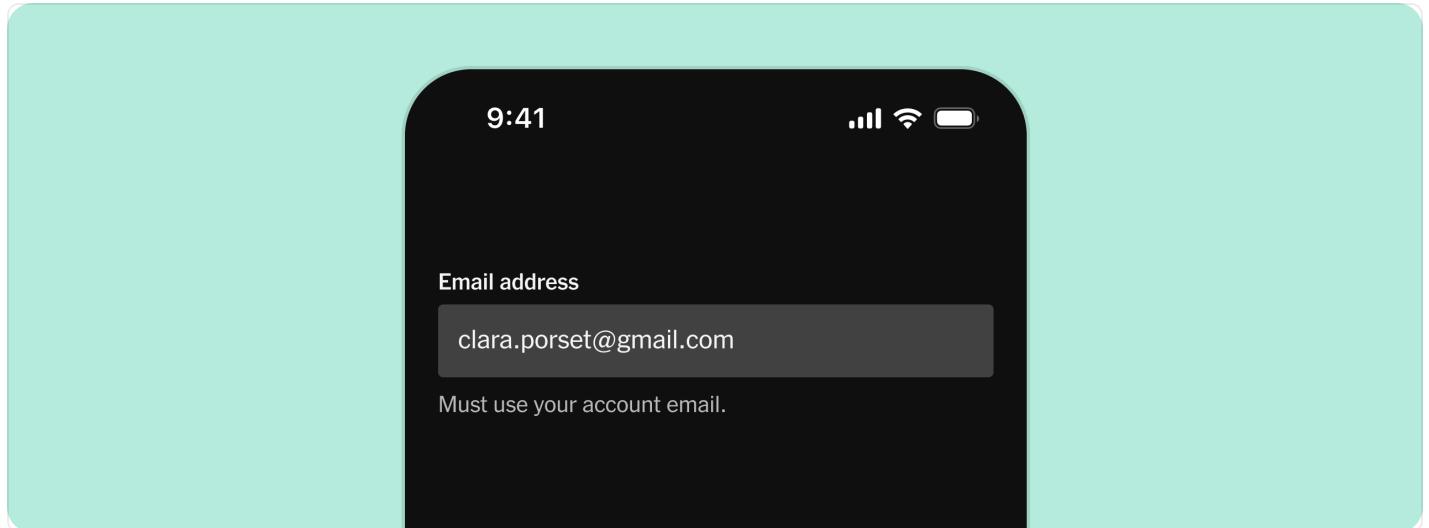
For placeholder text

Do not use placeholder text as a label or helper text. Placeholder text should act as a progressive enhancement, aiding a reader only when helpful. Placeholder text fails color contrast standards, disappears on click, and is not always read aloud by screen readers.

To lock inputs

Use a locked input in the rare case an input must remain present but not able to be interacted with.

- i TPL's locked input state differs from standard disabled inputs by still allowing text to be selected and recognized by assistive technologies.



Accessibility

Keyboard navigation color

When a reader navigates The New York Times using the tab key, inputs are focused in `Content/Accent` to aid recognition.

When the reader starts typing or clicks into the input, the input is focused in `Stroke/Primary`.

The image shows two side-by-side screenshots of a mobile device displaying a form. Both screenshots are timestamped at 9:41 and show signal, Wi-Fi, and battery icons at the top.

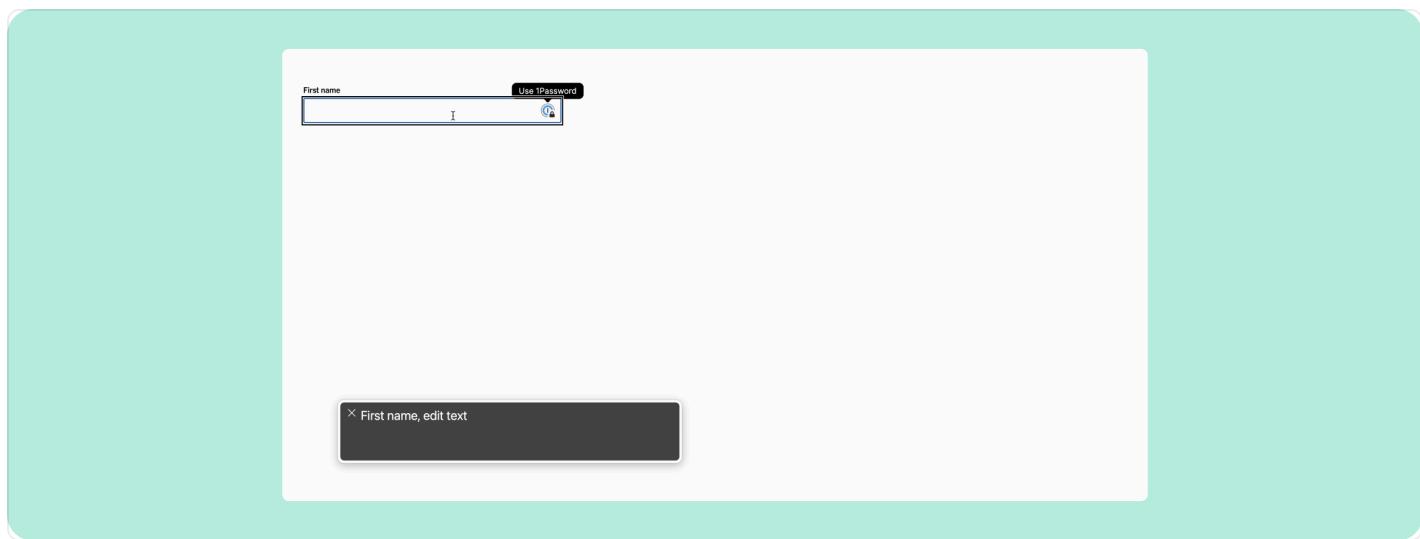
Left Screenshot (Content/Accent state): The first input field, labeled "First name", has a blue border around its entire bounding box, indicating it is the currently focused element.

Right Screenshot (Stroke/Primary state): The same "First name" field now has a black border around its stroke, and the text "Clara" is visible inside, indicating the input is active.

Below the first input are two more fields: "Last name" and "Occupation", each with a standard gray border.

Labels required

Every input requires a label even if the label is visually removed. Labels are always read aloud by screen readers.



♪♪ The rest is still unwritten, yeah, oh, oh. ♪♪

[See in Figma ↗](#)

[See in Storybook ↗](#)

Appendix

Error messages on Chrome, Safari, Firefox

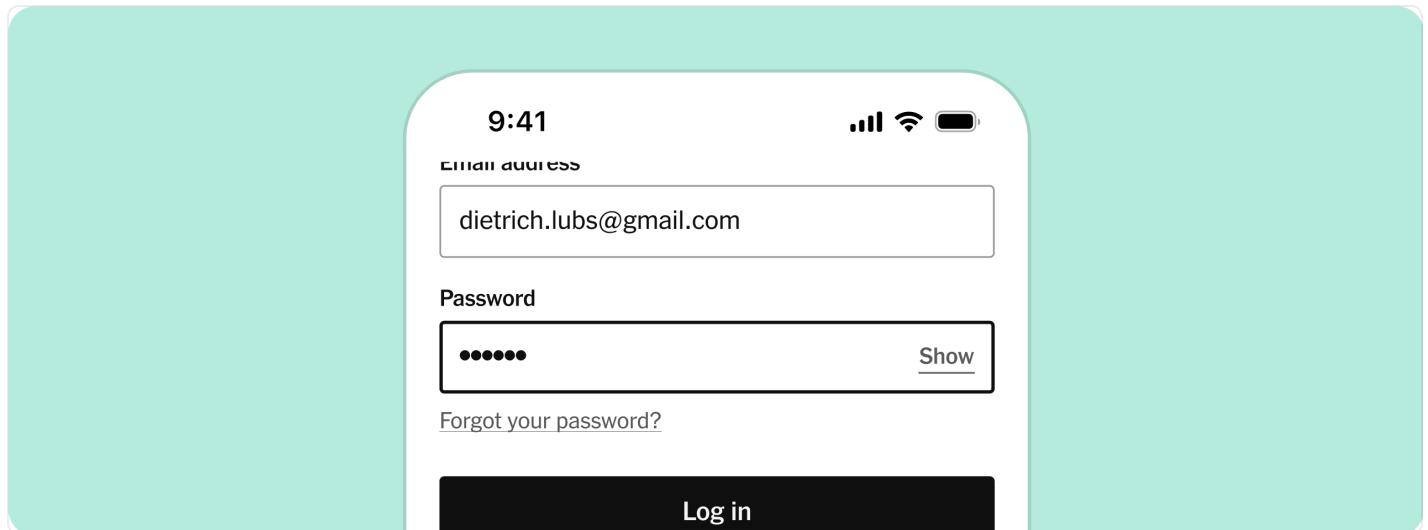
Client error type	Chrome message	Safari message	Firefox message	How to recreate
<code>required</code>	Please fill out this field	Fill out this field	Please fill out this field	Enter nothing
<code>minlength</code>	Please lengthen this text to [] characters or more (you are currently using [] character).	Use at least [] characters	Please use at least [] characters (you are currently using [] characters).	Enter a string too short
<code>maxlength</code>	-	-	-	Not possible to enter
<code>min</code>	Value must be greater than or equal to [].	Value must be greater than or equal to []	Please select a value that is no less than [].	Enter a number too small
<code>max</code>	Value must be less than or equal to [].	Value must be less than or equal to []	Please select a value that is no more than [].	Enter a number too big
<code>pattern</code>	Please match the requested format.	Match the requested format.	Please match the requested format.	
<code>type="email"</code>	Please include an '@' in the email address. '[]' is missing an '@'.	Enter an email address	Please enter an email address.	Enter a
	Please enter a part following '@'. '[]' is incomplete.	Enter an email address	Please enter an email address.	Enter a@
	'.' is used at a wrong position in 'a.'	Enter an email address	Please enter an email address.	Enter a@a.
<code>type="number"</code>	-	Enter a number	Please enter a number.	a
	Please enter a number.	Enter a number	Please enter a number.	.
<code>type="url"</code>	Please enter a URL.	Enter a url	Please enter a URL.	
<code>type="search"</code>	-	-	-	Never shows an error
<code>type="tel"</code>	-	-	-	Never shows an error
<code>type="text"</code>	-	-	-	Never shows an error
<code>type="datetime-local"</code>	-	-	-	Not possible to enter
<code>type="month"</code>				
<code>type="time"</code>				
<code>type="week"</code>				

TPL

Password input

Overview

Password input securely collects a password with the ability to show or hide characters. Password input is built on top of [Text Input](#), sharing the same visual style and basic functionality.



Figma

Web

iOS

Android

Components

React / CSS-in-JS

No

No

[See in Figma ↗](#)

[See in Storybook ↗](#)

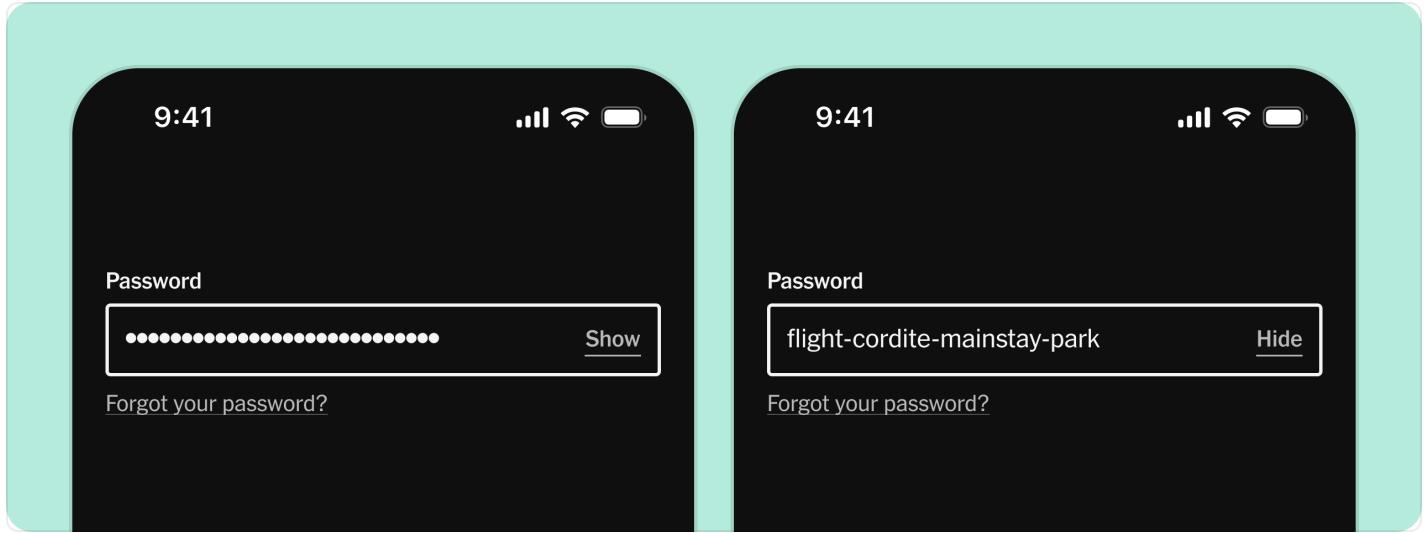
Guidance



Please refer to [Text Input](#) for more guidance.

To reveal and hide password

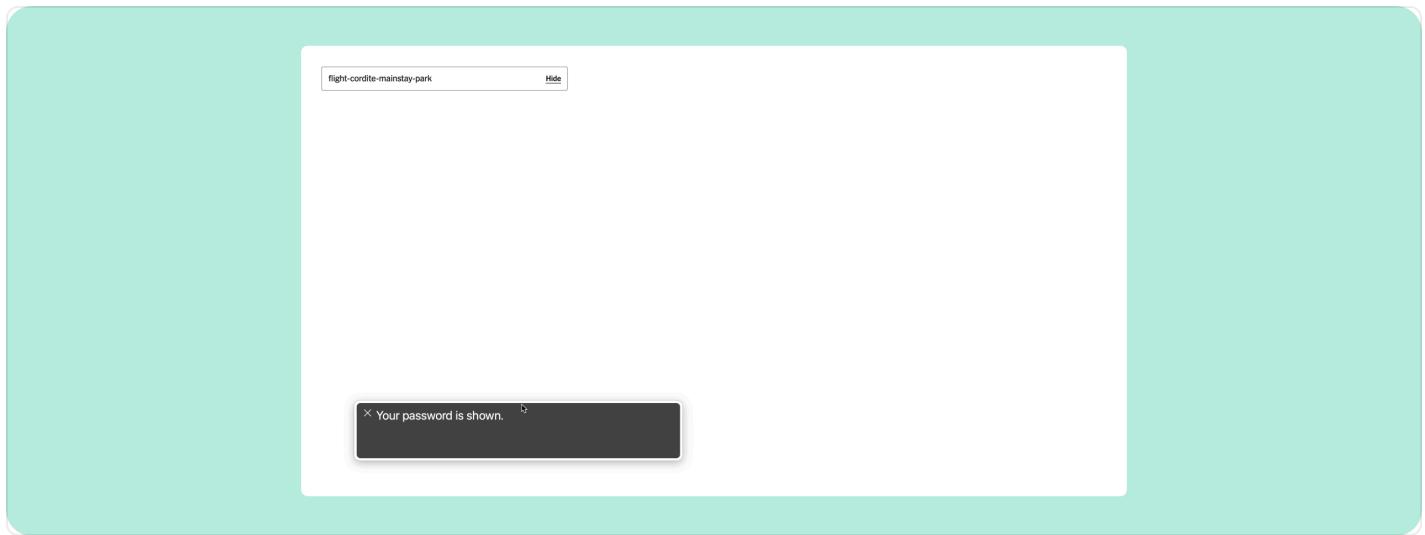
Password input includes a button that allows a reader to reveal or hide the characters they've typed. Characters are hidden by default.



Accessibility

Password visibility announcement

Toggling the Show/Hide button is accompanied by the screen-reader-only announcement “Password is visible” or “Password is hidden”. This is a helpful security feature for visually impaired readers who need to be aware if their password is currently visible to onlookers.



Caput draconis ■.

[See in Figma ↗](#)

[See in Storybook ↗](#)

Link

Overview

Link turns a piece of text into a navigational element that brings readers to a new place. Link is commonly used within a sentence with an underline beneath it. It can also be used without an underline when grouped together with other links, as in a navigation bar or footer.

The screenshot shows a mobile device displaying the New York Times website. The top status bar indicates the time is 9:41 and shows signal, Wi-Fi, and battery icons. The main content area has a light gray background and displays the heading "Inflation F.A.Q." Below this, a box contains the text: "What is inflation? Inflation is a loss of purchasing power over time, meaning your dollar will not go as far tomorrow as it did today. It is typically expressed as the annual change in prices for everyday goods and services such as food, furniture, apparel, transportation and toys." To the right of the content area is a sidebar with a white background and rounded corners. It contains copyright information ("© 2023 The New York Times Company"), links to NYTCO, Contact Us, Accessibility, Work with us, Advertise, T Brand Studio, Your Ad Choices, Privacy Policy, Terms of Service, Terms of Sale, Site Map, Help, and Subscriptions.

Figma

Web

iOS

Android

Example

React / CSS-in-JS

No

No

[See in Figma ↗](#)

[See in Storybook ↗](#)

Guidance

For links surrounded by text

By default, Link includes an underline to make it stand out from the surrounding text.

Opt in

To opt out of updates and offers sent directly by The Athletic, you must separately [contact The Athletic](#).

Privacy

The New York Times takes your privacy seriously. [Visit our privacy policy](#) to learn more.

For link-heavy experiences

Link can be styled without an underline by using the `Subtle` variant. To reduce visual clutter, consider using `Subtle` when grouping many links together, such as in nav menus, footers or other link-heavy layouts. Only remove the underline when you're certain the reader can infer from context that the text is interactive.

NEWS

Home Page	Wildlife Tracker
World	Obituaries
Coronavirus	The Upshot
U.S.	International
Politics	Canada
New York	Español
Business	中文网
Tech	Today's Paper

[Link \(Subtle\)](#)

For linking to outside-NYT content

It's typically most accessible to open links in the same tab. However, per [Newsroom standards](#) all non-NYT links should open in a new tab.

Also consider opening in a new tab when:

- Leaving the page causes readers to lose their progress, such as on a form or multi-step flow.
- The link opens instructions that helps readers complete the task they're working on, such as a pop-up window.

To apply color

Link inherits the color of surrounding text by default and accepts any [Content color token](#). Err toward Primary or Secondary link colors for distraction-free monochromatic experiences.

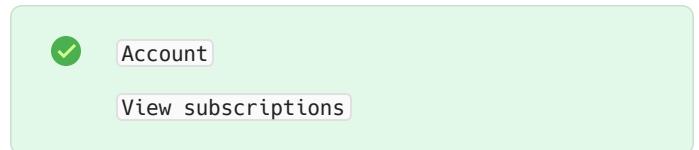
Accessibility

Describe the destination

The text used within Link must clearly and concisely describe the reader's destination. This text is crucial for assistive technologies as it becomes the [@accessible name](#).



These are poor [@accessible names](#) because they do not offer enough information to the reader.



These are good [@accessible names](#) because they describe a destination.

It's cumbersome for link text to be overly long. The headline of a story might make a great link, for example, but including the author, date, and summary is likely too wordy.

New tab announcement

Opening links in a new tab is especially jarring for readers who use a screen reader. Link provides the visually hidden announcement "Opens in new tab" after the name of the link.



It's dangerous to go alone! Take this ✂.

[See in Figma ↗](#)

[See in Storybook ↗](#)

Link Box

Overview

Link Box is a container that combines a collection of elements into one link. Link Box handles several important accessibility and SEO considerations and should always be utilized over wrapping several elements in a plain link tag.

Figma	Web	iOS	Android
Not needed	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> No

[See in Storybook ↗](#)

Guidance

To choose what the link is

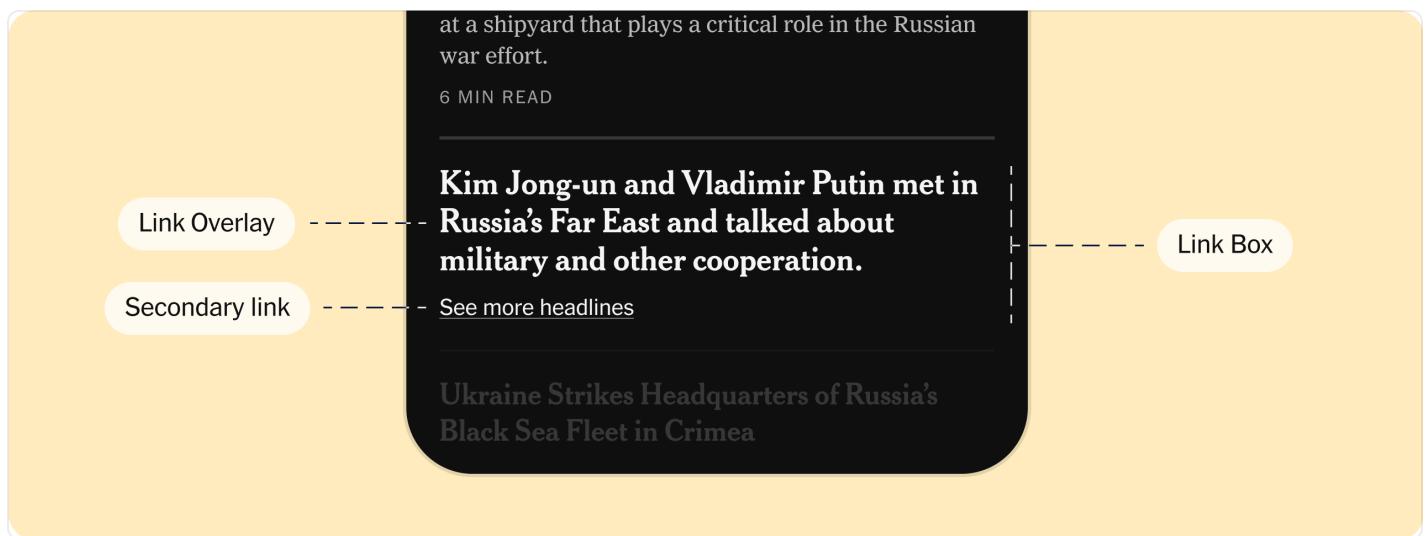
Link Box offers the ability to choose which element will act as the link for the entire container, called the overlay link. The overlay link should typically be the main piece of text inside the group, such as a heading. Make sure the link text accurately describes where the reader will navigate to.



In this example the Headline, Summary, and Reading time are the Link Box and the Headline acts as the Overlay.

To add other links

Link Box can contain additional links that do not act as the overlay link. Simply add them using the [Link](#) component.



To use your own custom component as the overlay link

[Advanced] Link Box can accept a custom component to act as the Overlay; this can be helpful for adding tracking links—add your link as a component to the `component` prop.

```
<LinkBox.Overlay href="https://nytimes.com" className={basicLinkClass} Component={LinkWithMockTracking}>  
  Link text goes here  
</LinkBox.Overlay>
```

"I'm not a box," she said "I don't have sides. This is it. One side fits all. This is it."

[See in Storybook ↗](#)



Menu

Overview

Apple's native [Menu](#) is the preferred way to display actions in context within a screen. TPL offers no customizations to Apple Menu. We recommend attaching Menu to a system icon or icon button.

A screenshot of an iPhone displaying a context menu. The screen shows the time as 9:41 and various status icons at the top. A circular '+' button is in the top right. The main content area shows a list with two items: 'Remove' and 'Reorder'. In the top right corner of this list, there is a small '...' button. The entire interface has a light gray background with rounded corners on the list items.

Figma	iOS	Web	Android
<input checked="" type="checkbox"/> Components	<input checked="" type="checkbox"/> Apple Components	<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> No

Figma

iOS

Web

Android

Components

Apple Components

No

No

[See in Figma ↗](#)

[See in SwiftUI docs ↗](#)

Guidance

Use Apple's native Menu in SwiftUI or UIKit

TPL suggests Apple's native Menu in [SwiftUI/UIKit](#). Apple Menu handles presentation, positioning, and accessibility in a reliable and easy to use component. It is impossible to customize background color, corner radius, and font.

TPL does not offer guidance for using icons at the moment, but please Slack [@Matt Argomaniz](#) if you're interested.



Ordering actions

When choosing the order in which actions are displayed in a Menu, start with those that allow a reader to take action(s) on the things in front of them—**customize and react**.

Then display actions that support a reader taking future actions—**store**.

Lastly, display actions that support actions beyond the reader—**connect**.

Examples:

1. Customize and react

- Listen to article
- Delete

2. Store

- Save
- Add to queue

3. Connect

- Share
- Gift

Destructive actions

Consider using the red destructive role for actions that deletes user data or performs an irreversible operation (Source: [Apple HIG](#)).



Accessibility

Consider using Accessibility Custom Actions

For Voiceover and assistive technology users, Menu actions are sometimes more convenient to reach through Accessibility Custom Actions. This appends actions to a group, typically eliminating the need for a reader to open up a Menu. Accessibility Custom Actions is only available in UIKit.

To learn more:

- [WWDC19 Tutorial](#)
- [UIKit Documentation](#)
- [Mobile A11y blog post by Rob Whitaker](#)

[See in Figma ↗](#)

[See in SwiftUI docs ↗](#)

Thank you to Véronique Brossier, Mimi Fang, Victoria Pandeirada, Jamie Liao and Chris Ballard.

Special shout out to Cam Pulsford for his comprehensive support and guidance through TPL's first iOS component releases.

Have a question or suggestion? We're [happy to help](#).

Radio

Overview

Radios allow readers to make a single selection from a group of options.

Give this article
You have 7 gift articles left this month.
Anyone can read them.

Share with anyone
 Share with New York Times subscribers

Upgrade to Home Delivery
 Change to News
 Change to Games
 Change to Cooking

Radios must always be in a group of at least two options; we recommend a maximum of six. In cases where more than six options are required, consider using a different input, like a[Select List](#).

Figma

Web

iOS

Android

Components

React / CSS-in-JS

No

No

[See in Figma ↗](#)

[See in Storybook ↗](#)

Guidance

Content

The content of [@Radio Labels](#) and [@Group Labels](#) should always be in sentence case. We recommend keeping these labels brief, ideally under five words, for ease of understanding. If more text is needed to describe Radio or Group, we recommend using a secondary label or surrounding text elements to explain it in more detail.

Replace your current subscription with another subscription.

- Upgrade to Home Delivery, including weekend or weekday newspaper delivery, plus unlimited digital access with a free All Access subscription.
- Change to News, including access to reporting, investigation and analysis, online and in The New York Times app.
- Change to Games, including access to Spelling Bee, Wordle, over 10,000 crosswords and more, online and in The New York Times Crossword app.
- Change to Cooking, including access to thousands of recipes, plus advice and inspiration, online and in The New York Times Cooking app.
- Change to Wirecutter, including access to product reviews and recommendations for smarter shopping.

Change subscription

Replace your current subscription with another

- Upgrade to Home Delivery
Includes weekend or weekday newspaper delivery, plus unlimited digital access with a free All Access subscription
- Change to News
Includes access to reporting, investigation and analysis, online and in The New York Times app
- Change to Games
Includes access to Spelling Bee, Wordle, over 10,000 crosswords and more, online and in The New York Times Crossword app
- Change to Cooking
Includes access to thousands of recipes, plus advice and inspiration, online and in The New York Times Cooking app
- Change to Wirecutter
Includes access to product reviews and recommendations for smarter shopping

X ✓

-  A group label must be defined for readers who use a screen reader, even if it is visually hidden.

Order

The way you order radios should support reader needs and can influence decision-making. It's important to consider what the most appropriate order of options is for your use case. Examples include alphabetical, cost low-to-high, priority high-to-low, or likeliness to be selected (preferred options by readers).

Confirmation buttons

Including a button to confirm a selection allows the reader to select an option without that selection taking effect immediately. This is a common pattern in [Forms](#), where multiple inputs can share the same confirmation button.

\$5/month billed every 4 weeks

\$40/year billed once annually

Subscribe now

Radio buttons can exist without a confirmation button in cases where the change should happen instantly and is clearly noticeable to the user.

Appearance

Automatic

Use your device setting to determine appearance.
The app will change modes when your device
setting is changed.

Dark

Ignore your device setting and always render in
dark mode.

Light

Ignore your device setting and always render in
light mode.

Appearance

Automatic

Use your device setting to determine appearance.
The app will change modes when your device
setting is changed.

Dark

Ignore your device setting and always render in
dark mode.

Light

Ignore your device setting and always render in
light mode.

Default selection

Not setting a default requires the reader to intentionally select an option before continuing. This requires [disabling](#) the button that allows the reader to progress until a radio button is selected.

Subscribe to New York Times Cooking

- \$5/month billed every 4 weeks
- \$40/year billed once annually

Subscribe now

Disabled Button

Subscribe to New York Times Cooking

- \$5/month billed every 4 weeks
- \$40/year billed once annually

Subscribe now

Active Button

Once a reader selects a radio button, the list cannot be reset to have no option selected. A “none of the above” option can be helpful in cases where readers may want to proceed without selecting one of the other options.

Step 1 of 4

Please tell us why you'd like to cancel your subscription.

- I don't use my subscription enough
- My subscription is too expensive
- I want a break from the news
- My rate is increasing
- I have concerns about the coverage
- None of the above

Continue to cancel

Setting a default value can bias a decision, seem pushy, or alienate readers who don't fit your assumptions. Only use a default selection if you have data to back it up.^[1] Always choose a default value based on what is most useful to the reader.

With these considerations in mind, providing a default value can make it easier for a reader to navigate through an experience smoothly. For example, if a reader has selected an option previously in their experience, you can preselect that option for them to make progressing easier.

Specifications

Height

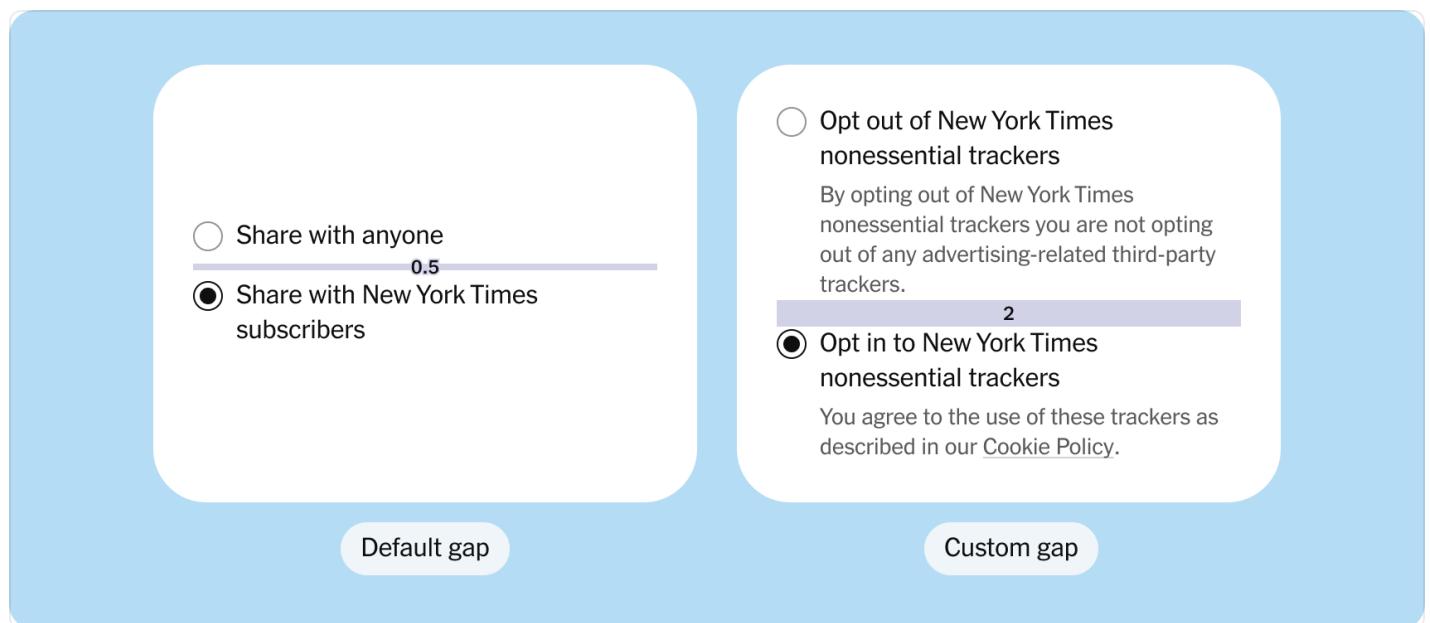
The height of radio buttons are set to a 32px minimum. This matches the height of our [compact buttons](#) and exceeds the [WCAG 2.2 tap target](#) requirement.

Width

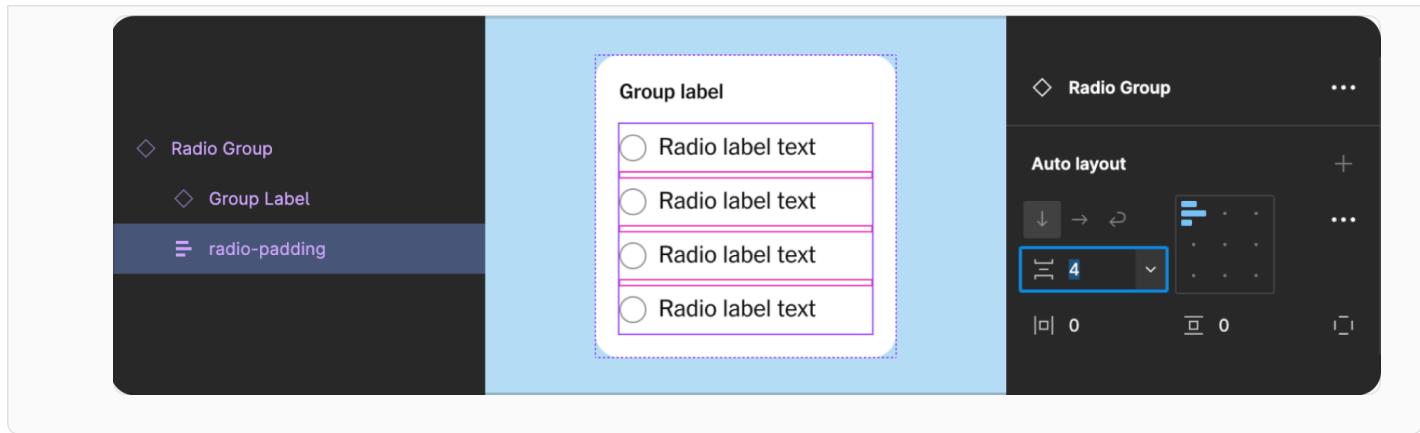
Radio buttons are **flex-width** to properly align radios within a group, especially when Radio labels or descriptions wrap to two or more lines.

Gap spacing

We recommend a .5 gap (4px) between radio buttons as a default value built into our Radio Group component. This value is customizable, particularly for cases where [@Radio Descriptions](#) affect the visual density of a [@Radio Group](#).

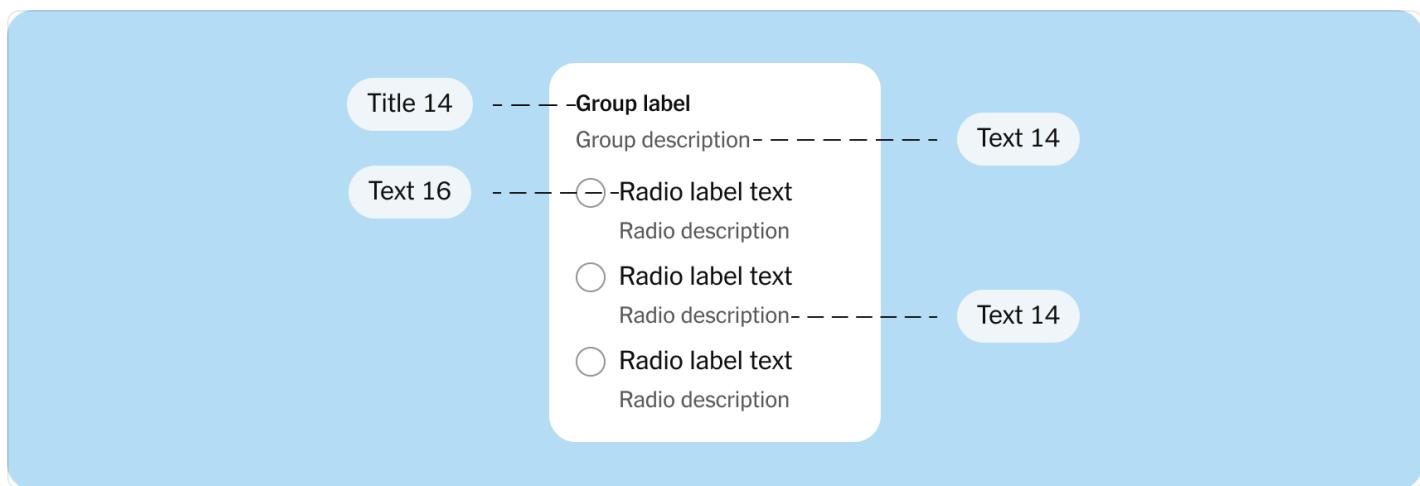


You can override spacing in Figma by clicking into the auto-layout frames and adjusting the gap manually. Be sure to use variables (tokens) from our [Spacing](#) scale.



Typography

@Radio Labels and @Radio Descriptions use the @Text 16 typography token by default. @Group Labels use @Title 14 and @Group Descriptions use @Text 14.



Customization

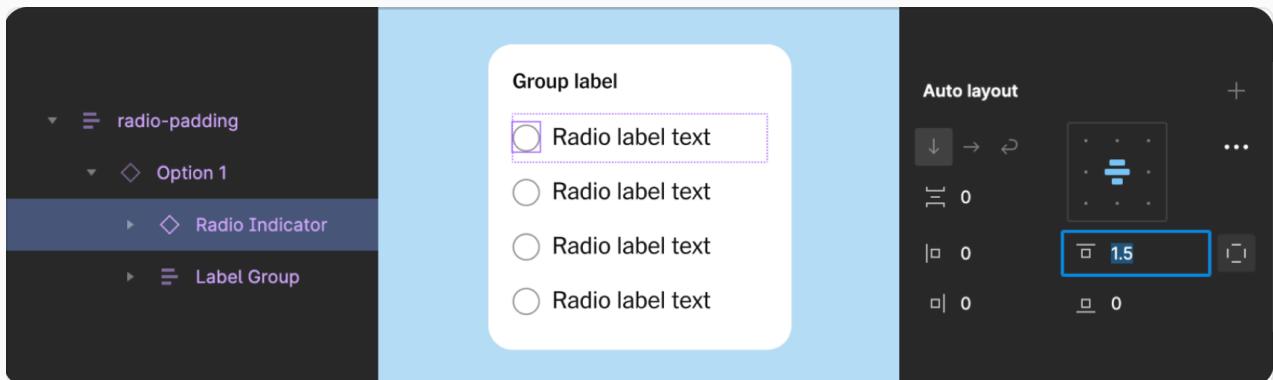
! Customizing Radio risks not passing accessibility requirements and reducing cohesion with The New York Times product experience.

We recognize that some cases may require customizations beyond what is currently provided. [Reach out](#) to our team for guidance customizing TPL components.

If you choose to customize the font size of the Radio, the Radio Indicator is positioned based on the height of the label text, including line height. If you choose a text style with a different height, you will need to customize the top padding of the Radio Indicator to maintain alignment, which can be calculated with this formula: `RadioTopPadding = (LabelHeight - RadioHeight) / 2`.



You can adjust the Radio Indicator alignment in Figma by clicking into the auto-layout frames and adjusting the top-padding manually. Be sure to still use increments from our [Spacing scale](#).



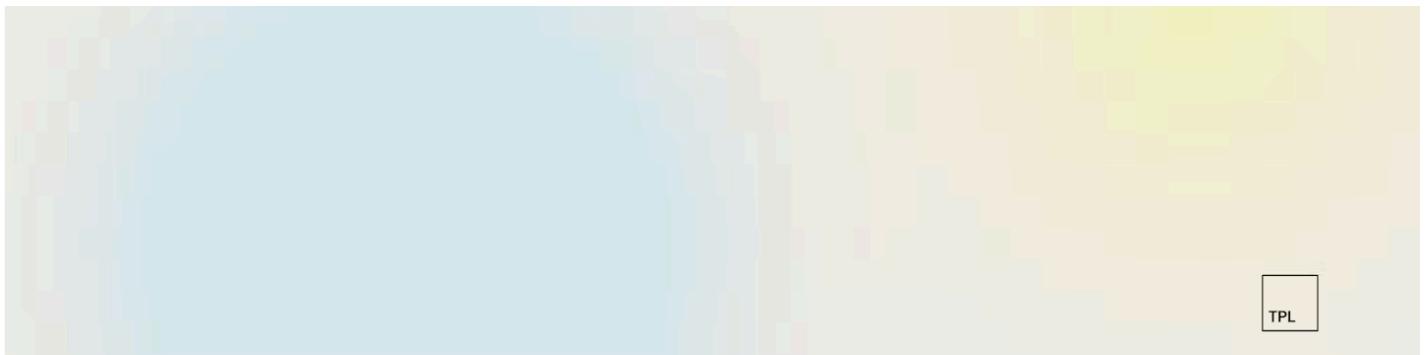
Glossary

Term	Description
Group Description	Optional description text that can be enabled to further describe a @Radio Group .
Group Label	The primary text at the top of a @Radio Group that briefly describes the group.
Radio	A combination of a @Radio Indicator and a @Radio Label . It can optionally include description text.
Radio Description	Optional description text that can be enabled to further describe a @Radio .
Radio Group	A group of @Radios with a @Group Label , which can be optionally hidden.
Radio Indicator	The circular element in a @Radio that indicates whether it has been selected or not.
Radio Label	The primary text in a @Radio that briefly describes the option.

The world cannot be found in our documentation. It's out there.

[See in Figma ↗](#)

[See in Storybook ↗](#)



Row

Overview

Rows are used to clearly display information in single-column lists. They can optionally include Buttons, Secondary Text, Icons, and Media.

The Veggie
Weekly | [See Latest](#)

The Morning
Daily | [See Latest](#)

Open Thread
Weekly | [See Latest](#)

Find newsletters
Our journalists provide commentary and analysis on topics from Politics and Climate Change

Connect

Account

Your profile

Account number
XXXXXXXXXXXXXXXXXXXX

Email address
olivia.rodrigo@nytimes.com

Update email

Password

Update

Figma

Web

iOS

Android

Components

React / CSS-in-JS

No

No

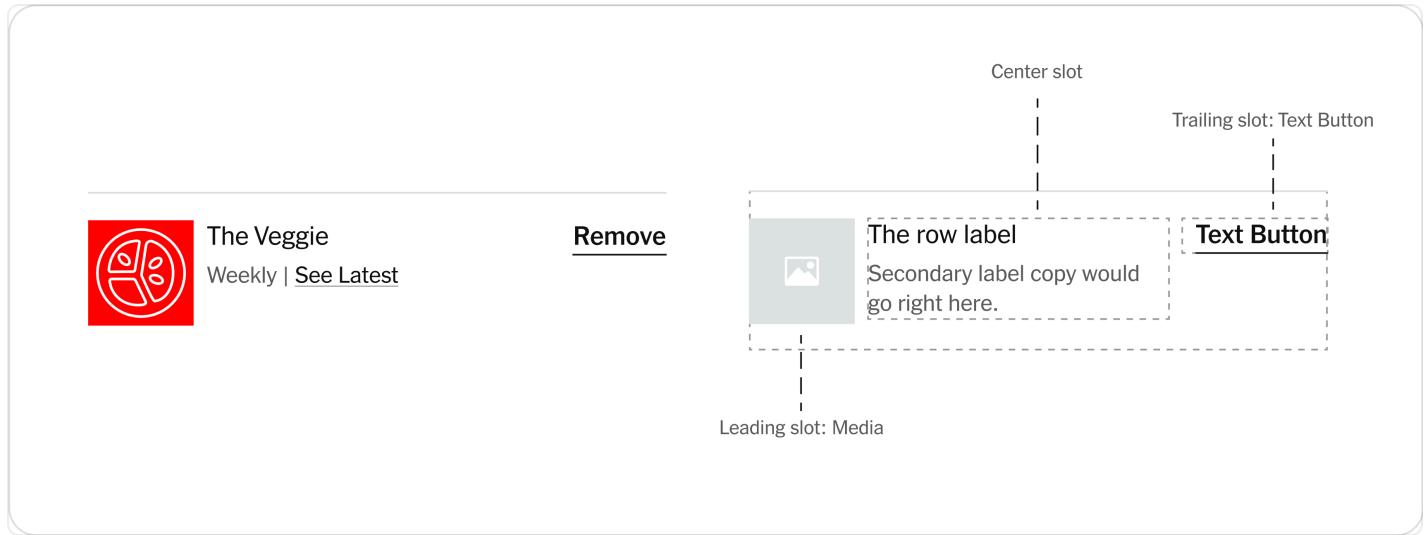
[See in Figma ↗](#)

[See in Storybook ↗](#)

Anatomy

Row components are composed of three slots: leading, center, and trailing. Depending on the use case, leading and trailing slots can be optional.

The **center slot** contains the main text content. The **leading slot** can be used to insert relevant media, and **trailing slot** determines the functionality of the row.



Rows should be scannable, clear, and consistent with other rows in the same list.

Guidance

Types

For displaying information

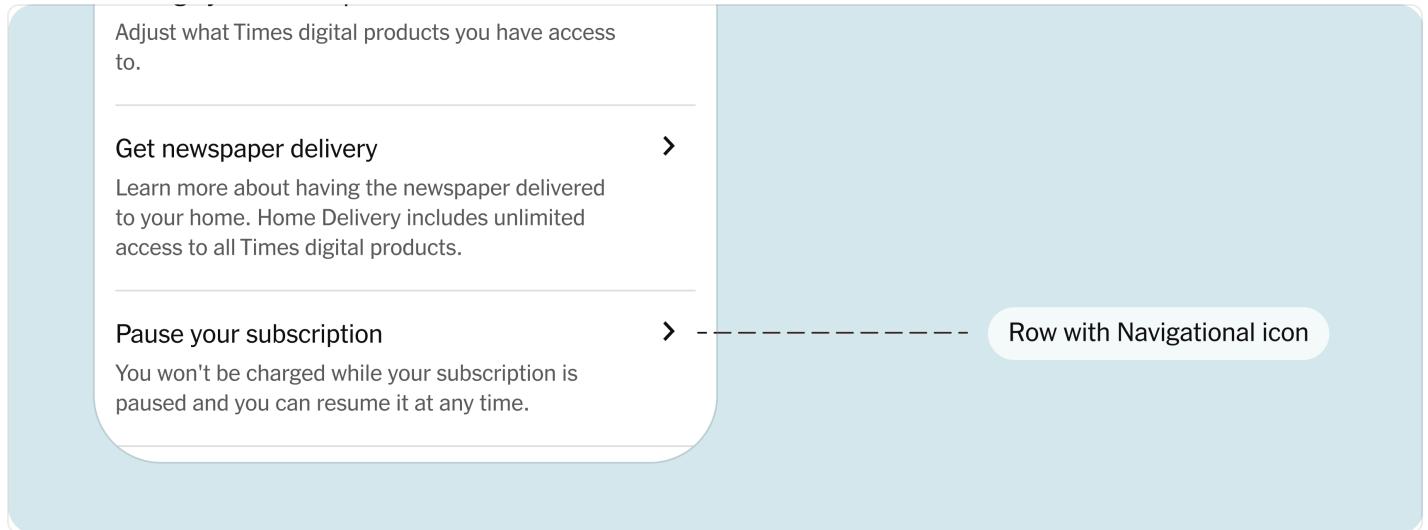
Rows that are not made interactive through buttons or navigational indicators are useful for presenting static information. This information can be generated by the user, or by The Times on behalf of the user.

A screenshot of a mobile application interface showing a list of payment methods. The list includes:

- Mastercard ***3
- Email address: olivia.rodrigo@nytimes.com
- Subscription rate: \${Price} \${Promotion} every {Billing cycle} through {Month ##, #####}

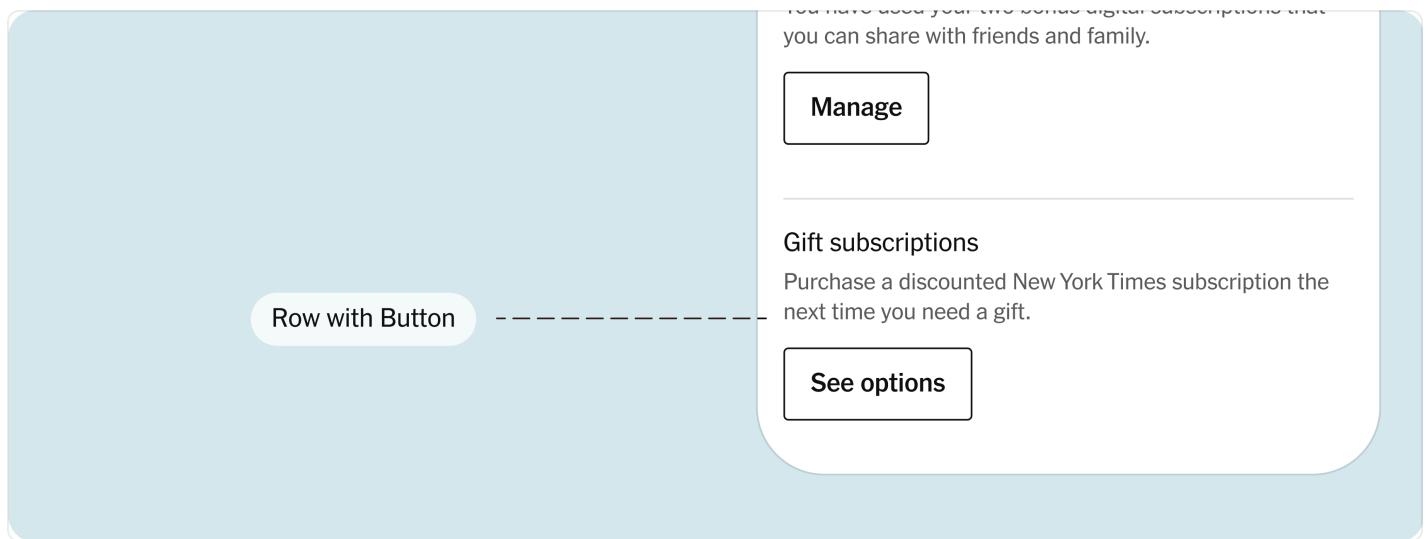
For navigating to a new experience

Rows can be used to navigate users to another page. Use a caret in the trailing slot to reinforce the progressive action.



For taking an action

By using a **Button** or **Text Button** in the trailing slot, rows can be used to prompt the user with an action.



We recommend using **Text Button** in cases where the action directly applies to the content of the row, such as editing or updating the information displayed.

Account number
XXXXXXXXXXXXXXXXXX

Email address [Update](#)
olivia.rodrigo@nytimes.com

Password [Change password](#)
.....

Trailing: Text CTA

We recommend using **Button** in the trailing slot is for initiating a new action, such as navigating to a new flow or signing up for a newsletter.

Share the Times

Bonus subscriptions
You have used your two bonus digital subscriptions that you can share with friends and family. [Manage](#)

Gift subscriptions
Purchase a discounted New York Times subscription for the whole family. [Purchase](#)

Trailing: CTA

Layout

Spacing

Rows provide their own spacing through top and bottom padding (See [height](#)). You shouldn't need to add any additional spacing between rows.

Row List Header

Providing a header for a row list is essential to orient the user to its content. We recommend using consistent header styles within an experience for quicker recognition.

Rule Header

Connected

Google

Disconnect your Google account from The New York Times.

Disconnect

Not connected

Apple

Connect your Apple account to

Connect



We offer **Rule Header** as a [helper template in Figma](#). It is not currently available as a component in TPL. Reach out for guidance on building your own locally.



List Consistency

To reduce visual noise, Rows in a list should not be too diverse from each other. We don't recommend mixing leading content or combining multiple button styles in the same row.

Newsletters



The Morning
Daily | [See Latest](#)



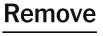
Open Thread
Weekly | [See Latest](#)



The Veggie
Weekly | [See Latest](#)





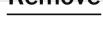




Open Thread
Weekly | [See Latest](#)



The Amplifier
Twice a Week | [See Latest](#)







Get Home Delivery >

Our journalists provide commentary and analysis on topics from Politics and Fashion to Climate Change.





However, the [Button: Weight](#) property can be leveraged to describe inverse actions.

Social accounts



Google

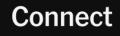
Disconnect your Google account from The New York Times.





Apple

Connect your Apple account to login to The New York Times.



Breakpoints and vertical stacking

Slots in a Row are always stacked horizontally on wider viewports.

We offer the option to vertically stack the center and trailing slots when using a Text Button or Button in the trailing slot on narrower viewports. This is particularly important with Rows that have a lot of text content and/or Media in the leading slot.

Share The New York Times

Bonus subscriptions
You have used your two bonus digital subscriptions that you can share with friends and family.

Gift subscriptions
Purchase a discounted New York Times subscription the next time you need a gift.

Manage

See options

Default behavior at large breakpoint

Bonus subscriptions
You have used your two bonus digital subscriptions that you can share with friends and family.

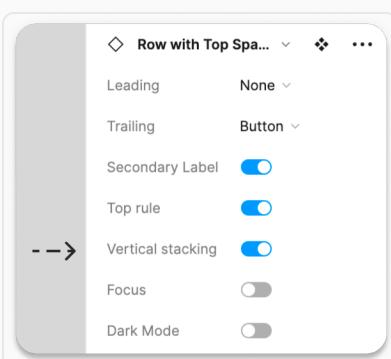
Manage

Gift subscriptions
Purchase a discounted New York Times subscription the next time you need a gift.

See options

Vertical stacking with small breakpoint

For consistency, vertical stacking should be applied to all rows in a list.



"Vertical stacking" is a toggle property in the Figma component.

Top rule

Rows are separated with a top **Horizontal rule** by default. This can be optionally removed for all the rows in a list.

Today's Paper

Stay focused on the day's articles — scroll through section by section, whether you're offline or online

Replica Edition

Read the print edition online, in its traditional layout

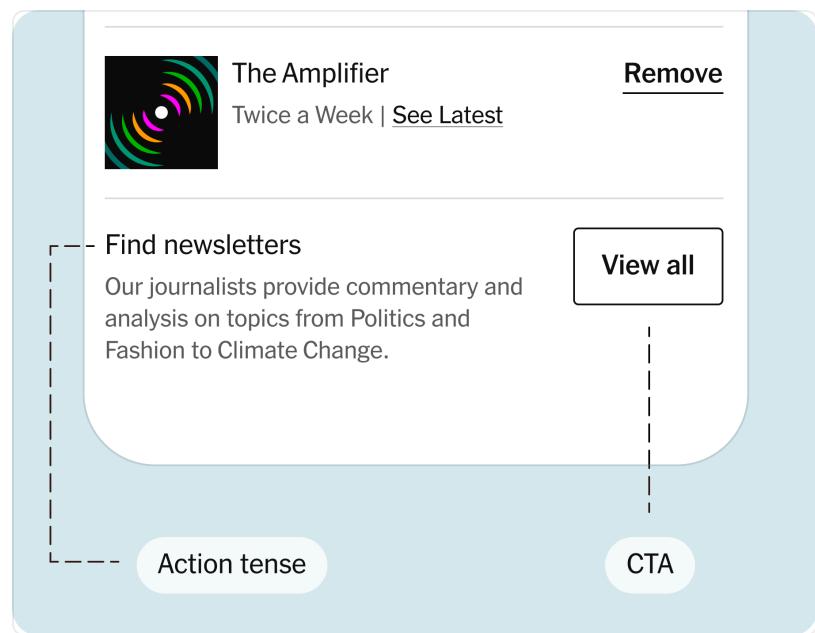
Times Store

15% off a selection of items for Times enthusiasts. Use 15%Sub at checkout.

Content

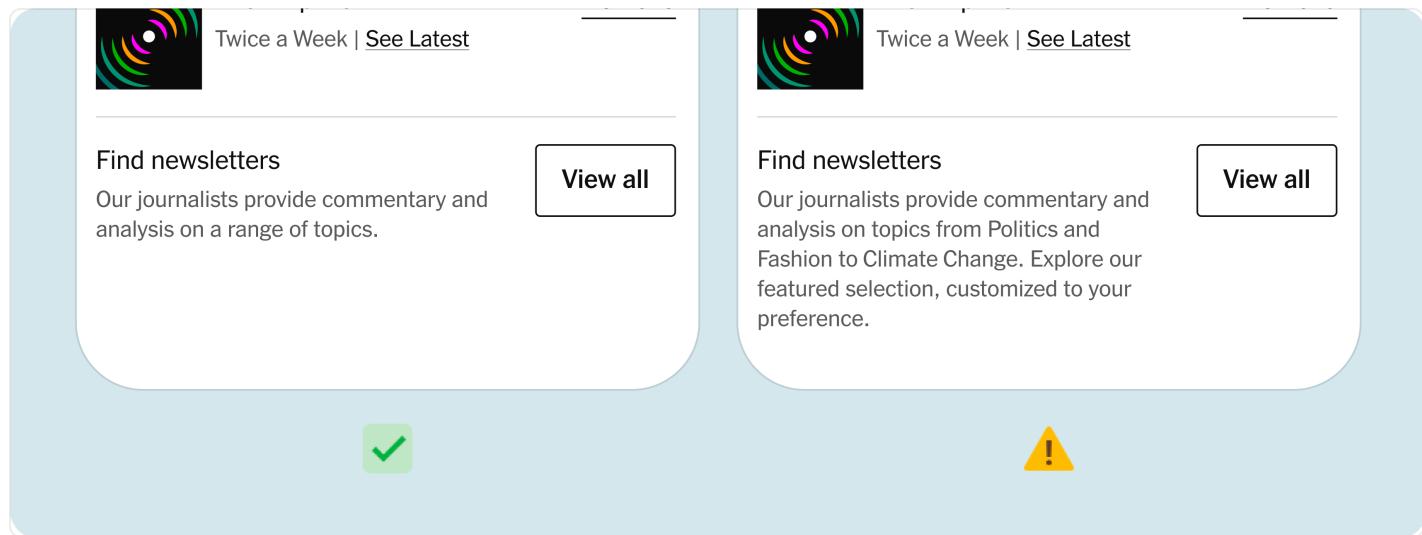
Label

Row labels should be concise and clear. We recommend keeping them to a **maximum of one line** on the smallest viewport. For action rows, we encourage using present tense for the header to reinforce the action.



Secondary Label

Secondary Labels can be used to provide supporting context or relevant details. We recommend keeping secondary Row Label text to a *maximum of 3 lines* at the smallest viewport.

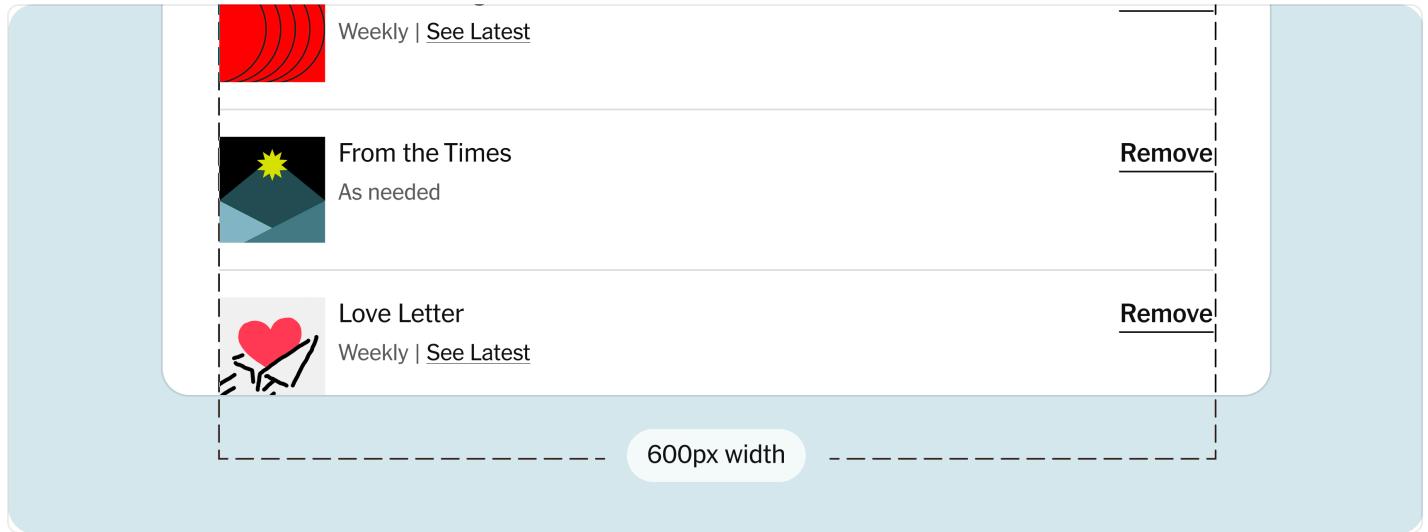


If using a Secondary Row Label in an Action Row, consider using [vertical stacking](#) to allow ample room for copy.

Specifications

Width

Row width is determined by the width of the container its' inside of. All rows in a list should maintain the same width.



We offer “Row Group” in Figma as a helper template grouping a **Rule Header** and multiple **Rows**.

Row Group Template

Your profile

Account number
XXXXXXXXXXXXXX

Email address
olivia.rodrigo@nytimes.com

Update

Password

Update email

Frame

X 40 Y -2151

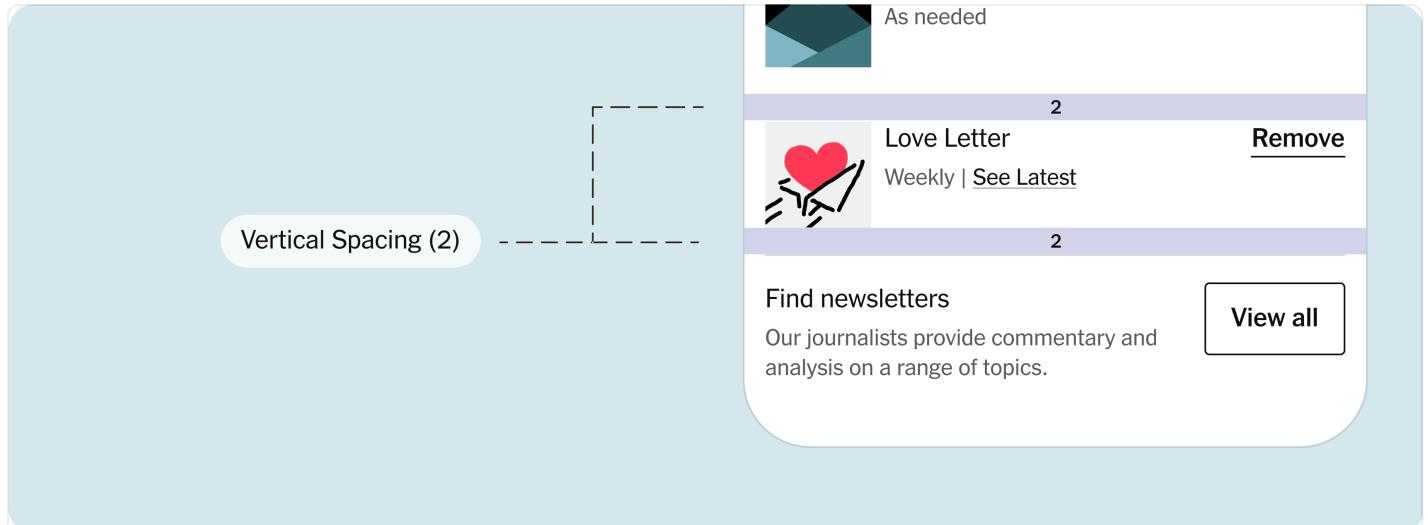
W 350 H 361

Auto layout

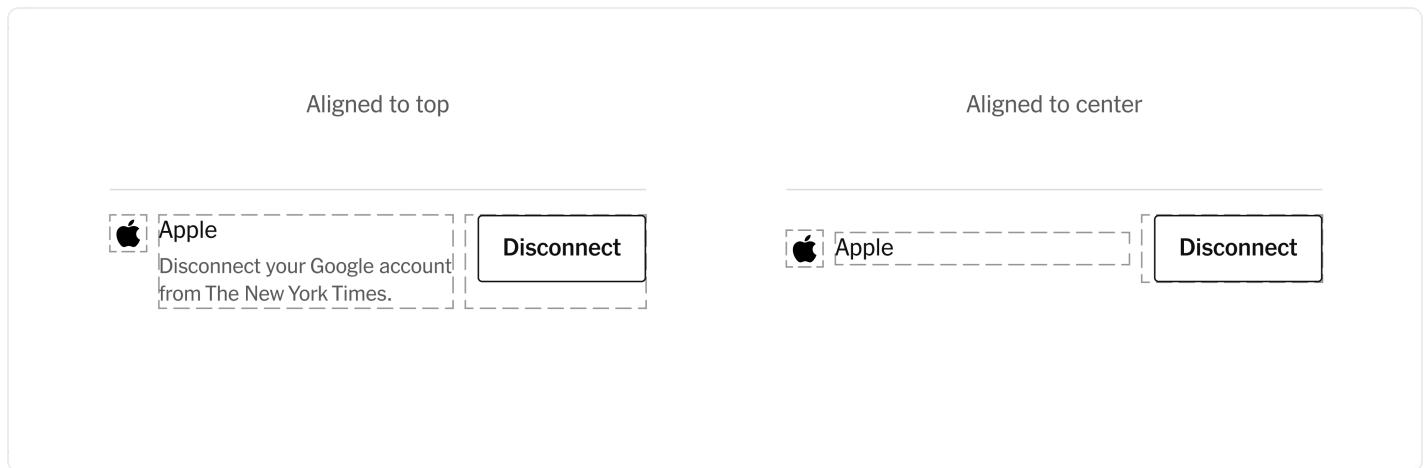
Use this container to establish the width of the row list. Make sure each nested component's width is set to “Fill container.”

Height

Row height is determined by its content and built-in spacing. Rows maintain consistent padding between the top rule (if using) and the content, and bottom padding.



When a row has a secondary label, all of the content aligns to top. Without a secondary label, content is vertically centered in the container.



Behavior

Navigational row interaction

Rows used for navigation leverage Link Box [link] to make the entire row a directional link. Navigational rows have a unique hover and pressed state triggered by interacting with any part of the row.

Manage subscription

Change your subscription

Adjust what Times digital products you have access to.



Get newspaper delivery

Learn more about having the newspaper delivered to your home. Home Delivery includes unlimited access to all Times digital products.



Button row interaction

Rows that use a **Button** or **Text Button** in the trailing slot respect [the states from that component](#). Unlike with navigational rows, *only* the button is interactive.

The image displays two identical card components side-by-side, separated by a vertical blue bar. Each card features a red circular icon at the top left, followed by a white rectangular button with a black border containing the text "Sign up +". Below this button, there is a black square icon with a colorful, concentric spiral pattern. To the right of the icon, the text "The Amplifier" is displayed in bold, followed by "Twice a week | See Latest" in a smaller font. At the bottom of each card is another white rectangular button with a black border and the same "Sign up +" text. A light blue circle highlights the area around the bottom button of the right card, with the text "On tap" positioned to its right, indicating that this specific button is the target of a tap action.

Focus states

In general, rows respect the default focus behavior of the elements that compose them.

Get newspaper delivery

Learn more about having the newspaper delivered to your home. Home Delivery includes unlimited access to all Times digital products.

Sign up

Because navigational rows use [Link Box](#), the entire row becomes one focusable object.

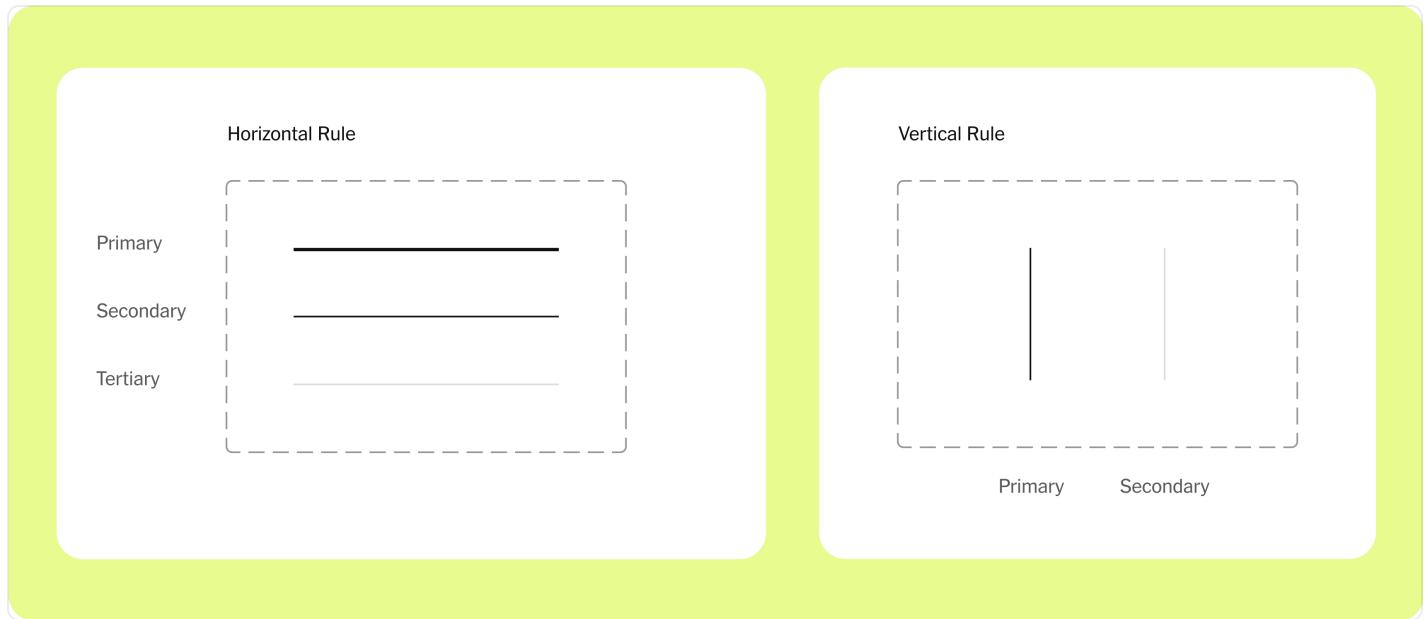
Change your subscription >

Adjust what Times digital products you have access to.

Rule

Overview

Rules are lines used to both separate and anchor content. They define the boundary between elements and help readers way-finding while scrolling.



We offer three types of Horizontal Rules: Primary, Secondary, and Tertiary. We offer two types of Vertical Rules: Primary and Secondary.

Figma

Web

iOS

Android

Components

React / CSS-in-JS

SwiftUI

Compose

[See in Figma ↗](#)

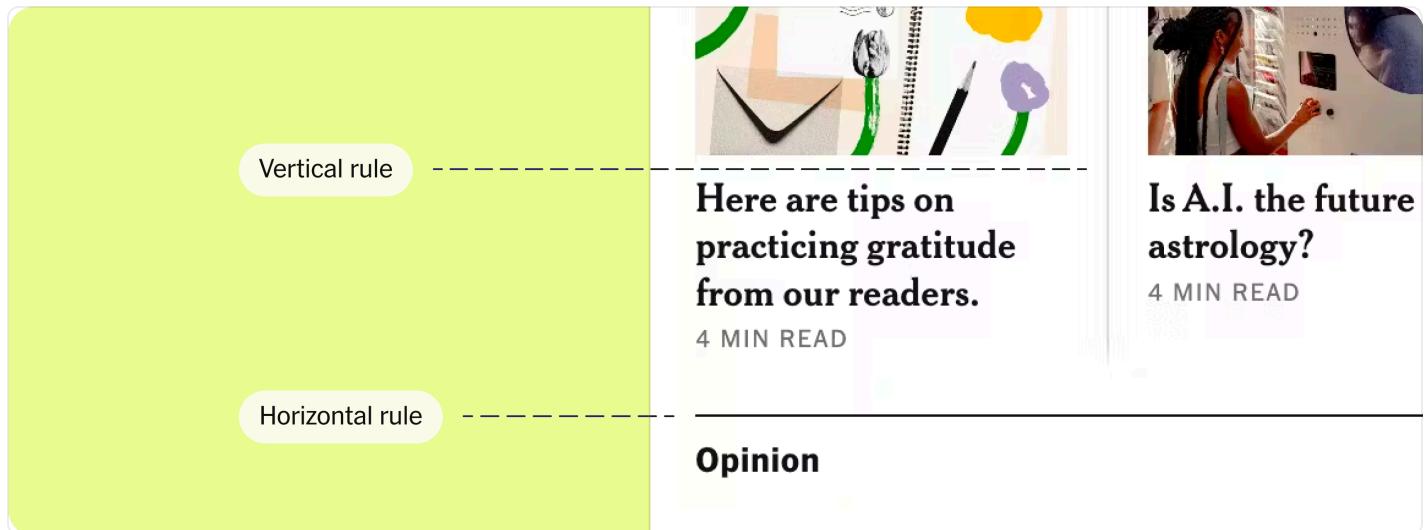
[See in Storybook ↗](#)

Guidance

Separating elements

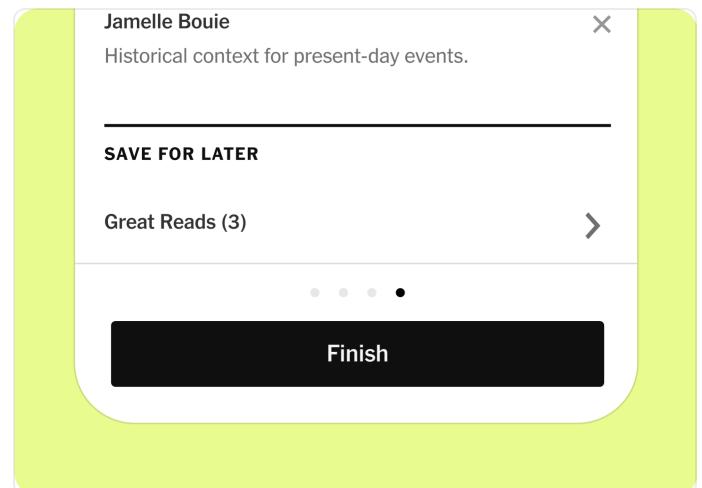
Rule is used to separate elements on a page. We recommend using even spacing tokens above and below the rule when using Rule this way. Both Horizontal and Vertical Rule can be used this way.

Rules with lower visual prominence are best suited for separating elements because they are less distracting from their surrounding elements.



Anchoring attention

We recommend using Primary and Secondary Horizontal Rules to anchor attention on the page when readers are scrolling, helping them way-find through a page more easily.



Specifications

Component	Type	Weight	Color	Use(s)
▼ Horizontal Rule 3	Primary	2px	Stroke Primary	Anchoring attention with the highest visual prominence.

	Secondary	1px	Stroke Primary	Anchoring attention or separating elements.
	Tertiary	1px	Stroke Tertiary	Separating elements with the lowest visual prominence.
▼ Vertical Rule 2	Primary	1px	Stroke Primary	Separating elements with visual prominence.

You made it! 🎉 get into ittttt 🎉

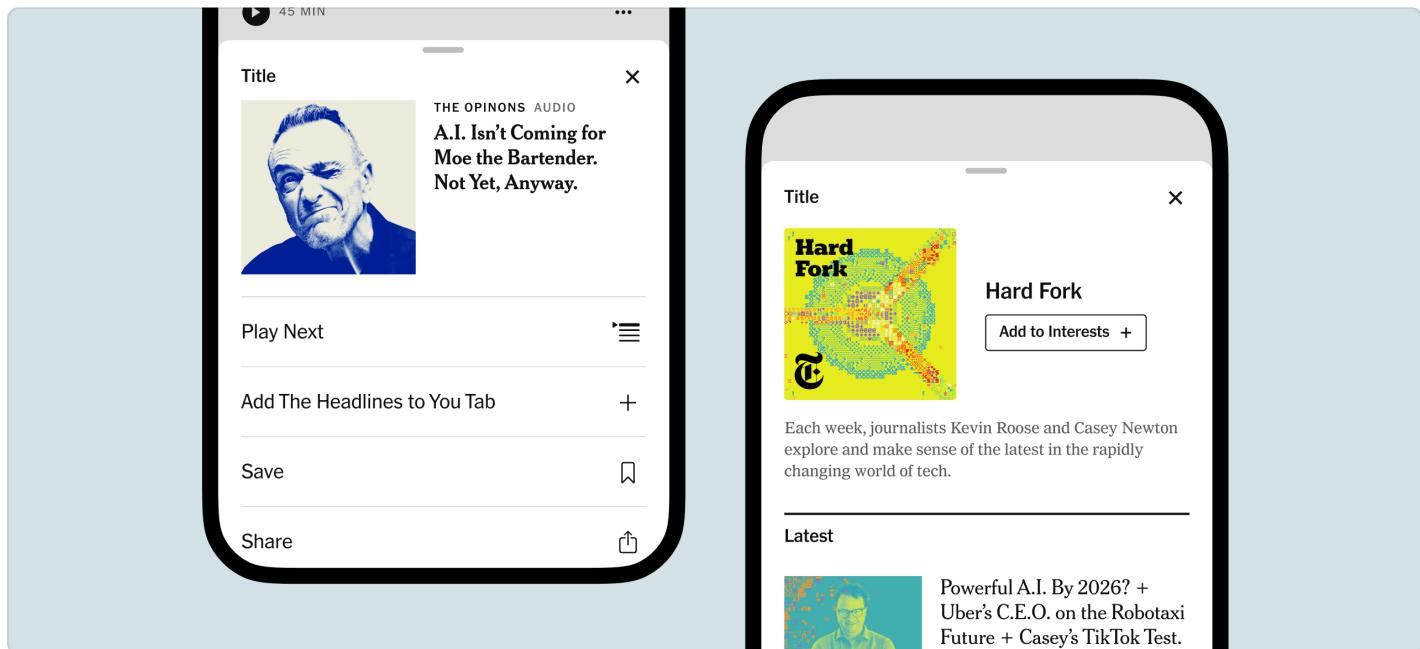
[See in Figma ↗](#)

[See in Storybook ↗](#)

Sheet

Overview

We use Apple's native [Sheet](#) component to help readers perform tasks without leaving the context of an experience.



Figma

Web

iOS

Android

Components

No

Example Code

No

[See in Figma ↗](#)

[See in Github ↗](#)

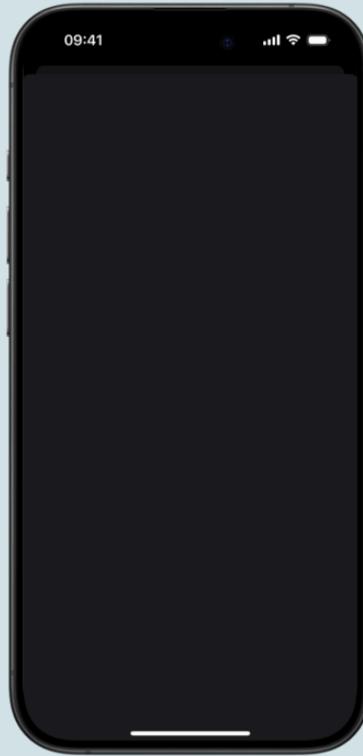
Guidance

Controlling height

Apple Sheet offers two standard [detent properties](#) to control height: `.medium` and `.large`.



.medium



.large

The `.medium` detent covers around half of the screen, maintaining more context with the previous view. The `.large` detent covers the full height of the view and creates a minimizing effect on it with a scrim behind.



Detents can also be set as percentages and fixed point values. However, we recommend using `.medium` and `.large` whenever possible for consistency across global implementations.

Supporting multiple heights

Sheets can have multiple heights by adding multiple detent values in an array.

```
.presentationDetents([.medium, .large])
```

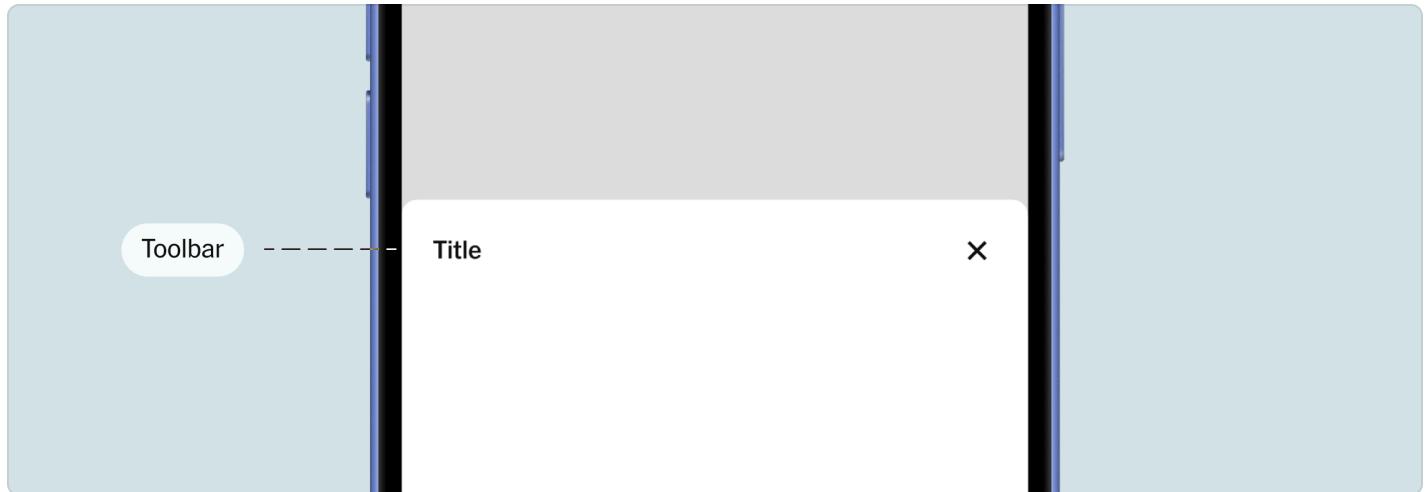
When multiple detents are set, the Sheet will appear at the first listed height and the reader can move between heights by swiping.



This also makes the grabber visible by default. Grabber visibility can be controlled manually with the `prefersGrabberVisible` property.

Toolbar content and actions

We recommend using Apple's [Toolbar](#) to position TPL Icon buttons and/or Title text in three positions: topBarLeading, Principal (Center), and topBarTrailing.



Toolbar Title text does not wrap. We recommend using as few words as possible, ideally 1-2 words, to ensure it appears properly across device sizes and [Dynamic Type](#) settings.

Toolbar can be removed or replaced.



You can replace Toolbar in Figma with [instance swapping](#).

Toolbar uses Icon Buttons in the Leading and/or Trailing position as Actions. We recommend reserving one action for dismissal, represented by the X icon. Sheet can also be dismissed by swiping down or tapping outside of the Sheet.



Icon buttons can be removed.

Adding custom content

Sheet is highly flexible and can include any view within it.

We recommend using [scroll views](#) for Sheet content, even when the content fits within the initial view by default. This ensures that content is always viewable across device sizes and Dynamic Type settings.

A screenshot of the Figma prototyping panel. On the left, the file structure shows a "Sheet demo for docs" project with various components like "Sheet Helper", "Local Sheet Content", and "iPhone 16 - Sheet". The main canvas displays an iPhone 16 - Sheet component with a yellow scroll view. A dashed line indicates the scroll view's boundaries. The prototyping panel on the right shows the "Design" tab selected, with a preview of the sheet and various prototyping tools at the bottom.

Navigating between Sheets

We are currently investigating how to best navigate between multiple Sheets in the best way for the user experience and reduces technical complexity.

If you are interested in this topic or want to contribute in any way, [please reach out!](#)

Specifications

Typography

Sheet uses [@Title](#) 16 by default. As with all typography, the default token can be customized to [any other token in the system](#).

Color

Sheet uses [@Background Primary](#) for Sheet in Light mode and [@Background Secondary](#) in Dark mode. Toolbar uses Apple's default [Chrome Material](#) as a background on scroll.

Corner Radius

Sheet uses Apple's default corner radius.

Holy Sheet 🐝

[See in Figma ↗](#)

[See in Github ↗](#)

Story List

Overview

Story List powers an enormous variety of lockups and journalism across the bundle.

Most Shared

The Morning Ritual That Helps Me Resist the Algorithm

4 MIN READ

5 Exercises to Improve Your Balance and Help You Age Well

12:11

36 Hours in Banff, Alberta ›

Are Runny Egg Yolks Safe to Eat? ›

Books to Improve Your Friendships ›

4 Black Women Who Changed NYC Dining ›

Figma

Web

iOS

Android

Components & Template

React / CSS-in-JS

SwiftUI

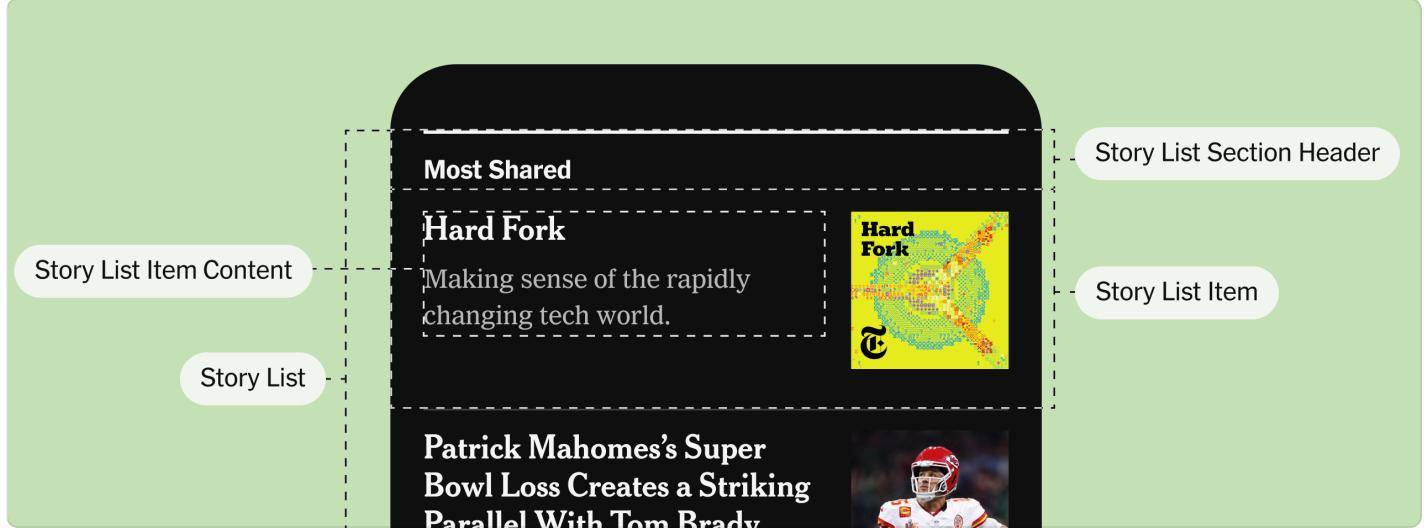
Compose

See in Figma ↗

See in SwiftUI docs ↗

Guidance

What's in a Story List?



Story List combines **Story List Section Headers**, **Rules** (or dividers), and **Story List Item**. **Story List** automatically handles the layout of a package and the spacing between multiple packages.

Story List Item is a single story lock up. It contains media and **Story List Item Content**. In Figma, **Story List Item** also contains dividers for convenience; in code, to apply dividers, you must use **Story List Item** within a **Story List** or manually add a **Rule**.



In Figma, we offer a **Story List** as template. To make your own, stack **Story List Items** and **Story List Section Headers** without any gaps. Remember to toggle off the top rule on the first **Story List Item** and enable "First section" for the first **Story List Section Header**.

Media Position

Story List Item can include media, which can be positioned in 8 ways:

- Top
- Top full bleed
- Trailing
- Leading

- Bottom
- Bottom full bleed
- Poster top
- Poster bottom

In code, you can use anything you like as media.



In Figma, we manually program Media sizes and aspect ratios. We currently offer:

- 16:9 (Landscape)
- 2:3 (Portrait, Landscape)
- 1:1 (Square)

If you'd like to use an unavailable size in Figma, please [let us know](#) or communicate your preference to your engineer.

Actions

`Story List Item` contains both an **actions** and **inline actions** parameter/slot. They are similar, but actions will always be positioned at the bottom of `Story List Item` while inline actions are placed at the end of the text content.



Customization

All of `Story List`'s typography is customizable. You can also customize many more details:

Name	Description	SwiftUI property
Alignment	The horizontal and vertical alignment of internal story list layouts.	<code>alignment</code>
Multiline Text Alignment	The text alignment when text spans more than one line.	<code>multilineTextAlignment</code>
Outer Padding Horizontal	The leading and trailing padding around <code>TPLStoryListItem</code> s and dividers. Expands tap target.	<code>outerPaddingHorizontal</code>

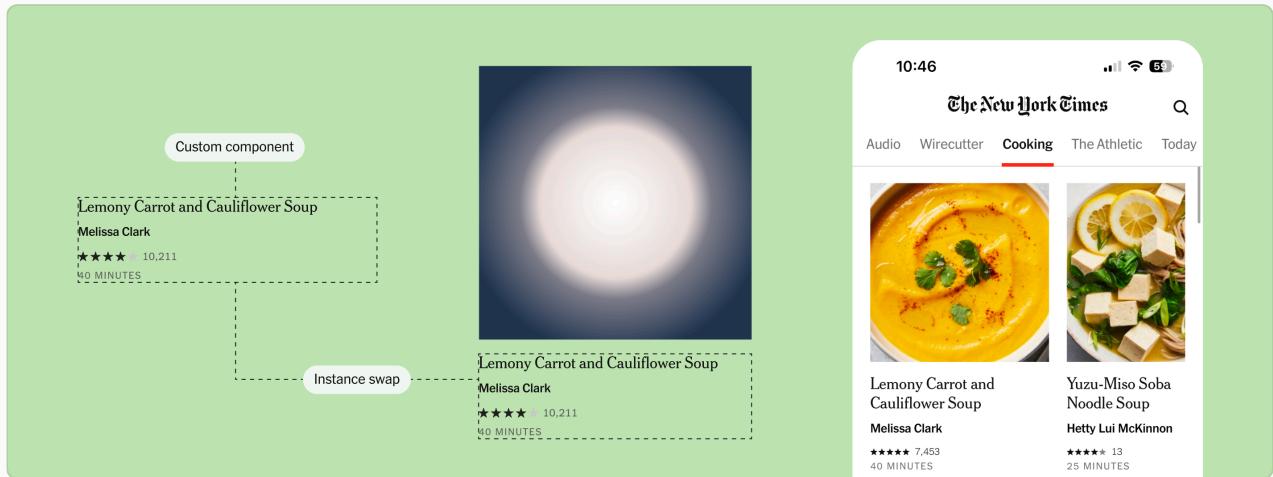
Section divider	The divider (or rule) between sections in <code>TPLStoryList</code> . Setting to <code>nil</code> disables section dividers.	<code>sectionDivider</code>
Section header padding top	The space above section header.	<code>sectionHeaderPaddingTop</code>
Section header padding bottom	The space below section header.	<code>sectionHeaderPaddingBottom</code>
Section spacing	The space in between sections.	<code>sectionSpacing</code>
Item divider	The divider (or rule) between <code>TPLStoryListItem</code> s. Setting to <code>nil</code> disables item dividers.	<code>itemDivider</code>
Item outer padding top	The padding above <code>TPLStoryListItem</code> . Expands tap target.	<code>itemOuterPaddingTop</code>
Item outer padding bottom	The padding below <code>TPLStoryListItem</code> . Expands tap target.	<code>itemOuterPaddingBottom</code>
Item content media spacing	The inner space between media and text content in <code>TPLStoryListItem</code> .	<code>itemContentMediaSpacing</code>
Item actions spacing	The space between actions and text content in <code>TPLStoryListItem</code> .	<code>itemActionsSpacing</code>
Item content inner spacing	The space between every text element in <code>TPLStoryListItemContent</code> .	<code>itemContentInnerSpacing</code>
Item top label padding bottom	The space below the top label in <code>TPLStoryListItemContent</code> .	<code>itemTopLabelPaddingBottom</code>

Content and actions

You can often build your own views and place them inside `Story List` because `Story List` is largely built with views (rather than strings, for example).



In Figma, you can mimic this by replacing `Story List Item Content`, `actions`, and `inline actions` with [instance swapping](#).



Accessibility

All text combined for VoiceOver users

`Story List` allows VoiceOver users to move between stories with fewer swipes because all the text in a `Story List Item` is combined and read together.

[See in Figma ↗](#)

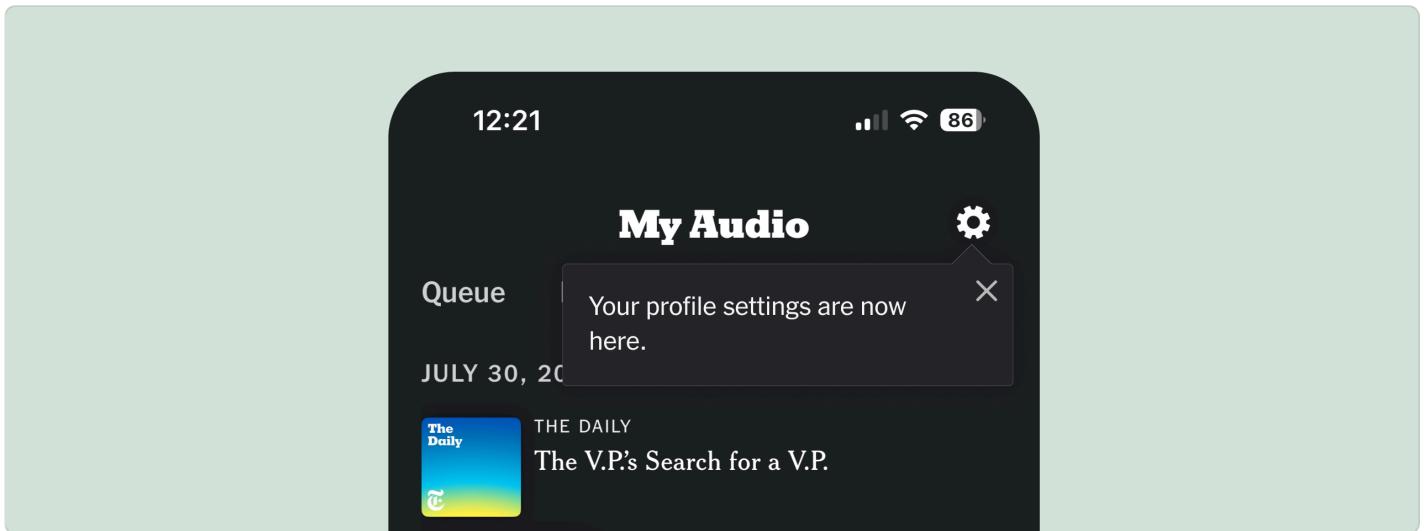
[See in SwiftUI docs ↗](#)



Tip

Overview

TPL Tip displays a contextual, dismissible message that brings attention to a feature.



Figma

Web

iOS

Android

Components

No

SOON In Code Review

No

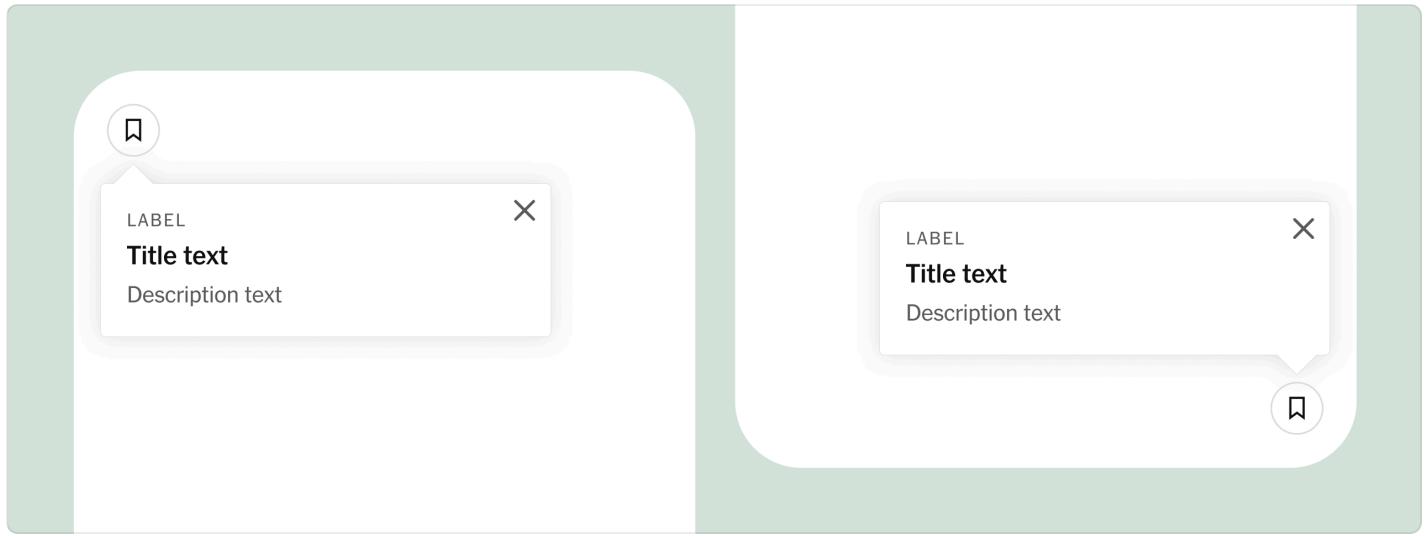
[See in Figma ↗](#)

[See in SwiftUI docs ↗](#)

Guidance

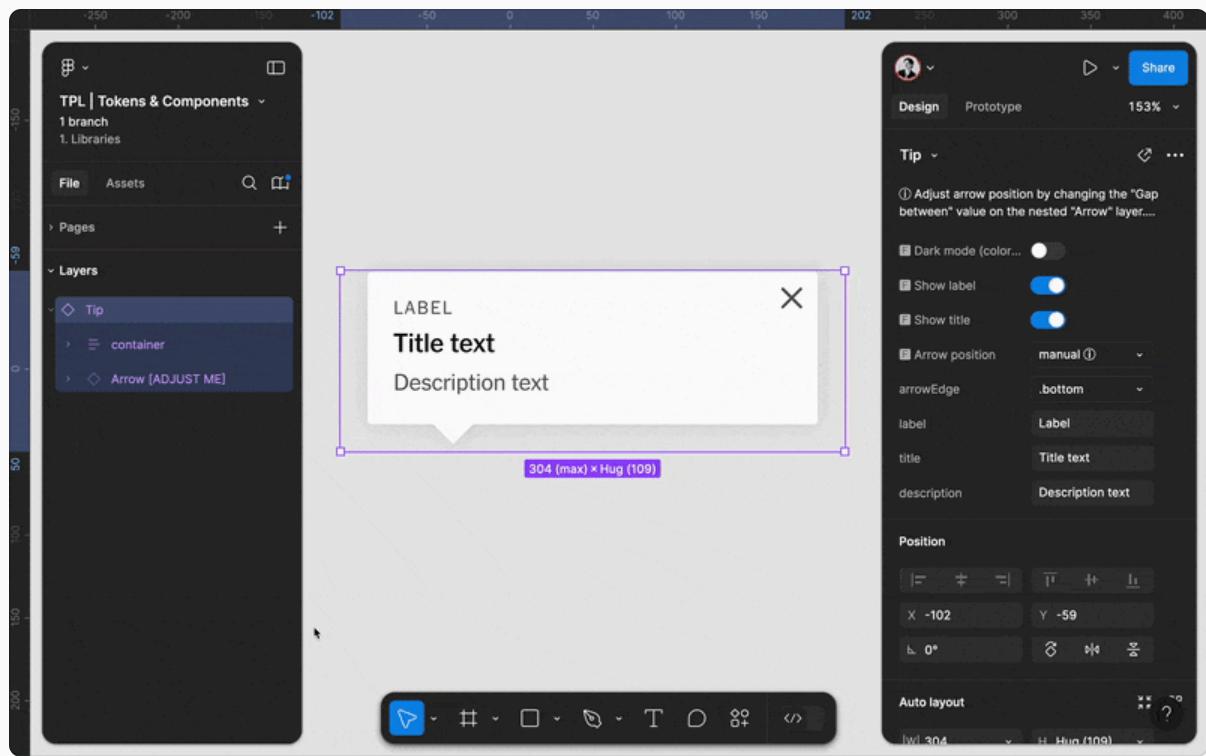
Positioning

TPL Tip automatically handles positioning to ensure it never appears off screen. You can override part of this behavior by specifying whether the tip should always appear above or below the attached element.



In Figma, we offer `center` and `manual` options for horizontal Arrow alignment.

Select the Arrow layer and adjust the Space between value to set Arrow position manually.



Content

Tip offers three content configurations by default:



Label

`Label/Standard`

`Content/Secondary`

Title

`Label/Standard`

`Content/Primary`

Description

`Text/16`

`Content/Primary`

Title

`Title/16`

`Content/Primary`

Description

`Label/Standard`

`Content/Secondary`

Description

`Text/14`

`Content/Secondary`

Customization

All of Tip's typography is customizable. You can also customize many more details:

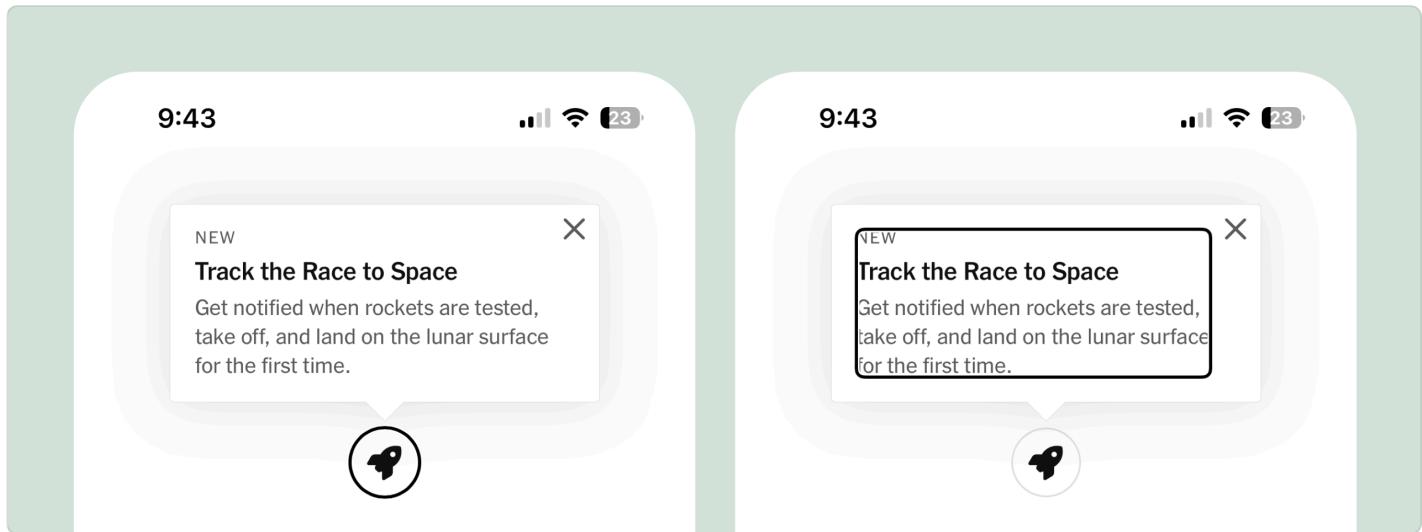
Name	Description	SwiftUI property
Background	The background color of the tip.	<code>background</code>
Color scheme	The background color of the tip.	<code>colorScheme</code>
Corner radius	The corner radius on all corners of the the tip.	<code>cornerRadius</code>
Shadow color	The color of the shadow behind the tip. Recommended to use alongside <code>shadowRadius</code> , <code>shadowX</code> , <code>shadowY</code> .	<code>shadowColor</code>
Shadow radius	The radius of the shadow behind the tip.	<code>shadowRadius</code>

Shadow x	The horizontal offset of the shadow behind the tip.	<code>shadowX</code>
Shadow y	The vertical offset of the shadow behind the tip.	<code>shadowY</code>

Accessibility

Reading order

Assistive technologies navigate to Tip directly after the element it is attached to. For VoiceOver users, a visually hidden “Tip” is announced before reading the content.



[See in Figma ↗](#)

[See in SwiftUI docs ↗](#)

Big thanks to our Pilot partners Bon Champion, Daman Chatha, Lavi Tang Rimar Villasenor and Alvin Andino from **Audio**, Jayne Lee, Jun Chou and Tulio Jarocki from **Cooking**, Véronique Brossier and Lareen Melo from **AppX**, Mimi Fang, and Victoria Pandeirada from **User Space** and Zev Eisenberg from **Games**.

Special shout out to Cam Pulsford for his comprehensive iOS support and guidance.

If you're using Tip, [let us know!](#) We'd love to chat about your experience with the component and feature your collaboration here.

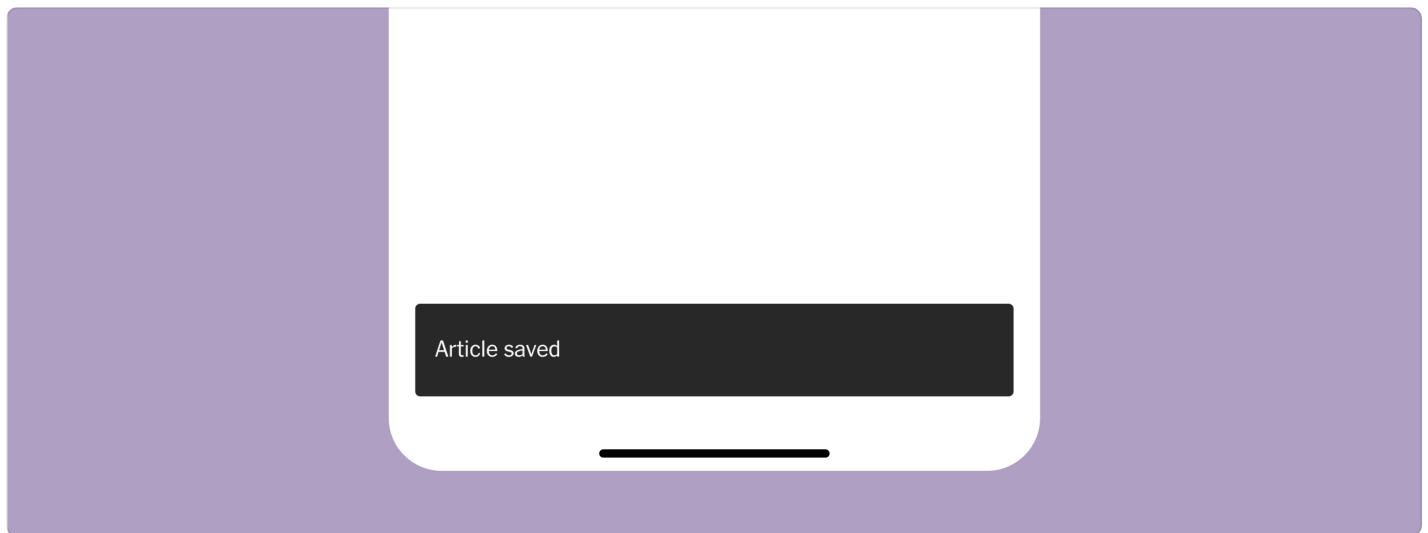
Have a question or suggestion? Want to chat more about Tip? We're [happy to help](#).



Toast

Overview

Toast is a messaging component that announces a change of status. It is often used to acknowledge or confirm an action.



Figma

iOS

Web

Android

Components

Components

No

No

[See in Figma ↗](#)

[See in SwiftUI docs ↗](#)

Guidance

Positioning

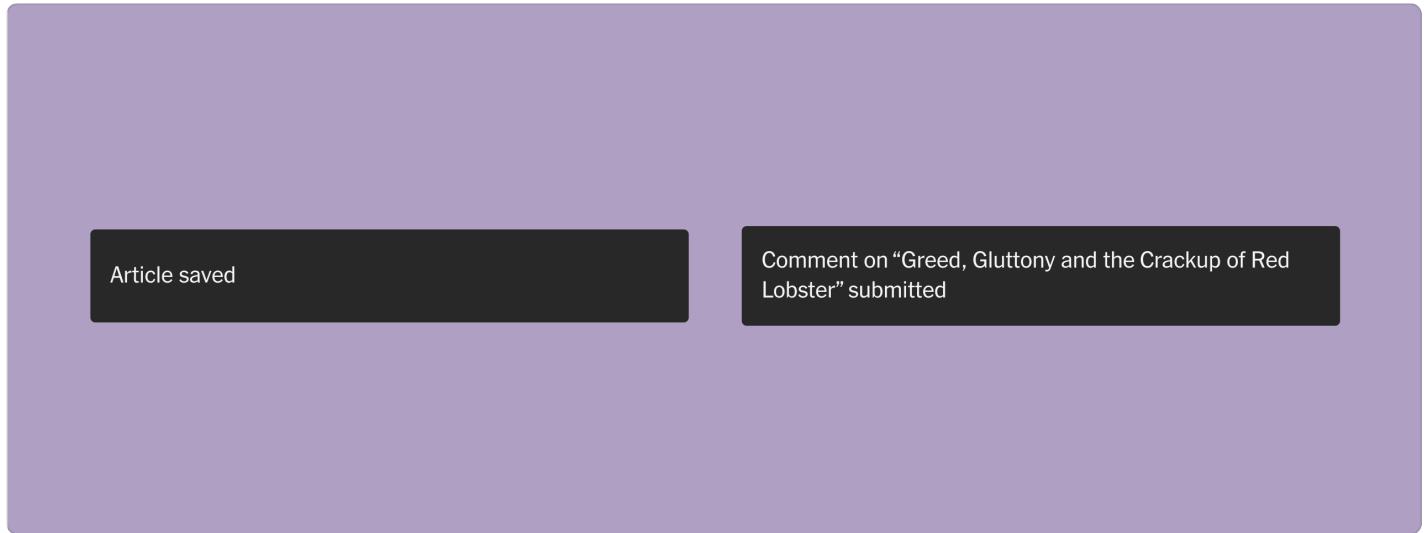
Toast should be positioned in a location on the screen—typically the bottom—where it can be seen immediately after its corresponding action is taken. If your app has a bottom bar (e.g. charm bracelet, tab bar), consider placing the toast above it.



TPL Toast does not currently handle animation. [Contact us](#) with any suggestions.

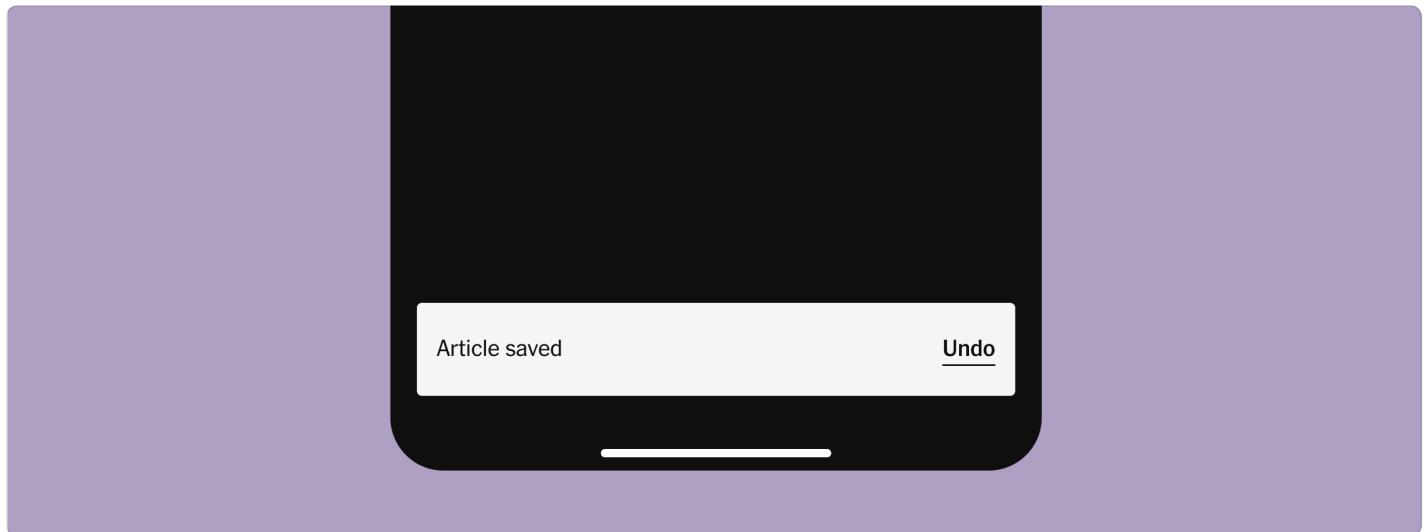
Content

The Toast message should describe the outcome of the action taken. Try to keep it short and to the point. This message also acts as the announcement for screen reader users.



Action

Toast can display an optional action. Try to limit the text to one or two words. Ensure the action can also be performed elsewhere in the app. Actions cannot be announced by VoiceOver.



Accessibility

VoiceOver announcement

The message of the toast is announced automatically by VoiceOver without needing to be focused. “Status” is prepended to the announcement. Toast can also be read by VoiceOver if a reader drags their pointer to it.



[See in Figma ↗](#)

[See in SwiftUI docs ↗](#)

Have a question or suggestion? We're [happy to help](#).

Resources

Times Product Language offers services outside of Foundations as well as multiple methods of support. You can also always reach out if you need more help.

Quick Resources

[TPL Figma Library](#)

[TPL Storybook](#)

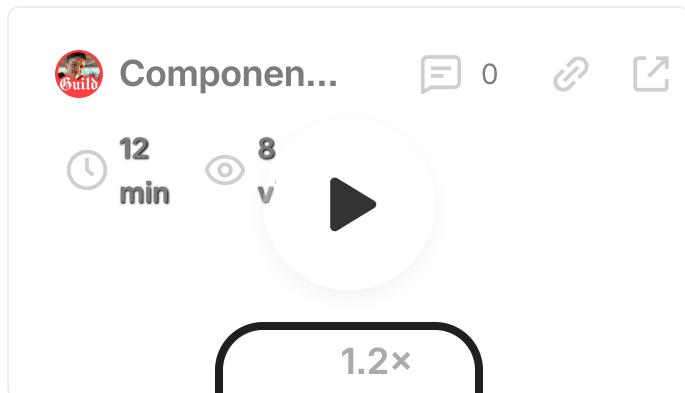
[TPL GitHub](#)

[Our Figma Team](#)

[Design Systems Jira](#)

Get Started

A collection of walkthroughs and tutorials to get familiarized with everything TPL offers.



Components Part 1: What Are They?

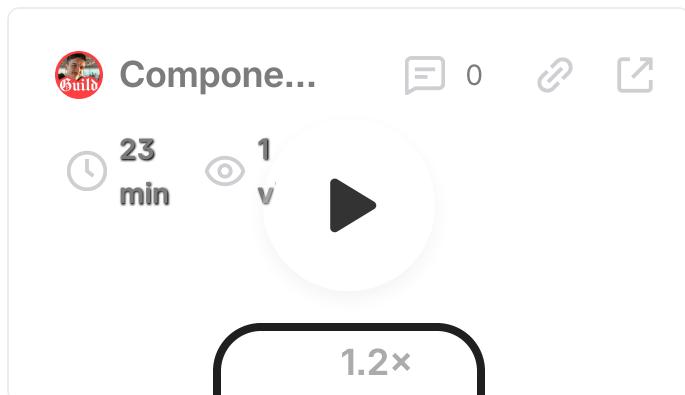
12 min 8 v

1.2x

This block displays a thumbnail for a video titled "Components Part 1: What Are They?". The thumbnail shows a person speaking. It includes a play button, a timestamp of 12 minutes, and a view count of 8. Below the thumbnail is a playback control bar set to 1.2x speed.

Components Part 1: What Are They?

An introduction to components in Figma, in Code, and in design systems, explaining what goes into making components and what makes them impactful for teams.



Components Part 2: Designing with Components

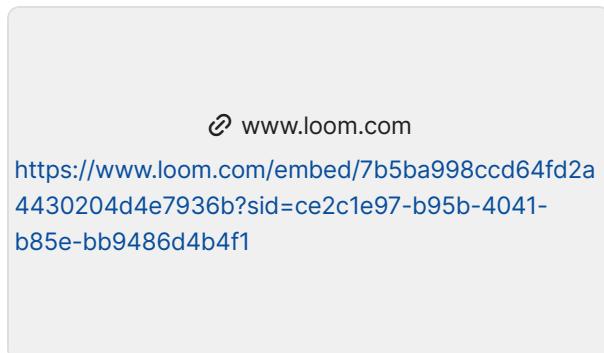
23 min 1 v

1.2x

This block displays a thumbnail for a video titled "Components Part 2: Designing with Components". The thumbnail shows a person speaking. It includes a play button, a timestamp of 23 minutes, and a view count of 1. Below the thumbnail is a playback control bar set to 1.2x speed.

Components Part 2: Designing with Components

A recommended process for working with components in iOS and Android, explaining and demonstrating how to use TPL, customize it, and go beyond it.



www.loom.com

<https://www.loom.com/embed/7b5ba998ccd64fd2a4430204d4e7936b?sid=ce2c1e97-b95b-4041-b85e-bb9486d4b4f1>

This block displays a thumbnail for a video titled "Using TPL Colors as Variables in Figma". The thumbnail shows a Loom logo and a URL. Below the thumbnail is a link to the video's URL.

Using TPL Colors as Variables in Figma

By leveraging Figma's new variable feature, we can easily switch between light and dark modes, enhancing design efficiency and hand-off to engineering.

9/6/2024

A screenshot of a video player interface. At the top left is a circular profile picture of a man with a beard and orange hair. To his right, the text "TPL Roads..." is partially visible. In the top center are three icons: a speech bubble with "0", a link icon, and a share icon. Below these are two circular progress indicators: one showing "4 min" and another showing "11 v". To the right of these is a large black play button. Underneath the play button is a curved progress bar with the text "1.2x" in the center.

An Introduction to TPL for Product Designers

Learn about TPL and the team that maintains it.

10/17/2024

A screenshot of a video player interface. At the top left is a circular profile picture of a man with a beard and orange hair. To his right, the text "TPL Roads..." is partially visible. In the top center are three icons: a speech bubble with "0", a link icon, and a share icon. Below these are two circular progress indicators: one showing "8 min" and another showing "5 vi". To the right of these is a large black play button. Underneath the play button is a curved progress bar with the text "1.2x" in the center.

TPL Resources Walkthrough

Tour the Figma library, documentation, and code.

10/17/2024

Contact Us

Is any of this confusing? Are you wondering what this means for you and your team? Do you want to shower us with love and praise? No matter the topic, please [reach out to us on Slack!](#) Or, if you're ready, dive into our resources.



On Slack

TPL channel: [#times-product-language](#)

Hyphen support channel: [#hyphen](#)

Piano support channel: [#piano](#)

Slack groups:

@design-systems-team, @design-systems-eng, @design-systems-design



Our Resources

[TPL Figma Library](#)

[TPL Storybook](#)

[TPL GitHub](#)

[Our Figma Team](#)

[Design Systems Jira](#)

Feature Availability

A regularly updated guide to which TPL features are available on which platforms.

Web	iOS	Android		
Feature	Figma	Web	iOS	Android
Apple Menu	<input checked="" type="checkbox"/> Components	<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> Apple Components	<input checked="" type="checkbox"/> No
Button	<input checked="" type="checkbox"/> Components	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> SwiftUI ButtonStyle	<input checked="" type="checkbox"/> No
Color	<input checked="" type="checkbox"/> Color Styles <input checked="" type="checkbox"/> Color Variables	<input checked="" type="checkbox"/> JS / CSS-in-JS	<input checked="" type="checkbox"/> SwiftUI Extensions <input checked="" type="checkbox"/> UIKit Color Extension	<input checked="" type="checkbox"/> Kotlin <input checked="" type="checkbox"/> XML Resource
Dialog	<input checked="" type="checkbox"/> Components	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> SwiftUI	<input checked="" type="checkbox"/> No
Icon Button	<input checked="" type="checkbox"/> Components	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> SwiftUI ButtonStyle (excluding extraCompact height)	<input checked="" type="checkbox"/> No
Icons	<input checked="" type="checkbox"/> Components	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> No
Link	<input checked="" type="checkbox"/> Example	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> No
Link Box	Not needed	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> No
Password input	<input checked="" type="checkbox"/> Components	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> No
Radio	<input checked="" type="checkbox"/> Components	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> No
Row	<input checked="" type="checkbox"/> Components	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> No
Rule	<input checked="" type="checkbox"/> Components	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> SwiftUI	<input checked="" type="checkbox"/> Compose
Sheet	<input checked="" type="checkbox"/> Components	<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> Example Code	<input checked="" type="checkbox"/> No
Spacing	<input checked="" type="checkbox"/> Components <input checked="" type="checkbox"/> Variables	<input checked="" type="checkbox"/> JS / CSS-in-JS	<input checked="" type="checkbox"/> CGFloat Extension	<input checked="" type="checkbox"/> Kotlin <input checked="" type="checkbox"/> XML Resource
Story List	<input checked="" type="checkbox"/> Components & Template	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> SwiftUI	<input checked="" type="checkbox"/> Compose
Text Button	<input checked="" type="checkbox"/> Components	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> SwiftUI ButtonStyle	<input checked="" type="checkbox"/> No
Text input	<input checked="" type="checkbox"/> Components	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> No
Tip	<input checked="" type="checkbox"/> Components	<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> In Code Review	<input checked="" type="checkbox"/> No
Typography	<input checked="" type="checkbox"/> Text Styles	<input checked="" type="checkbox"/> React / CSS-in-JS	<input checked="" type="checkbox"/> TypographyToken Extension (supports SwiftUI & UIKit)	<input checked="" type="checkbox"/> Compose

Mobile Resources

iOS Dynamic Type Calculator Table

Font Size (pt)	Relative To	Dynamic Type Size	Scale Factor	Scaled Font Size
11	Body	AX 5	311.76%	34.29
16	Headline	AX 5	311.76%	49.88
14	Body	AX 5	311.76%	43.65

Dynamic Type Size Names

Dynamic Type Size Name

X Small

Small

Medium

Large (default)

X Large

XX Large

XXX Large

AX 1

AX 2

AX 3

AX 4

AX 5

iOS, iPadOS Dynamic Type Sizes

Style	X Small	Small	Medium	Large (def)	X Large	XX Large	XXX Large	AX 1	AX 2	AX 3	A:
Large Title	31	32	33	34	36	38	40	44	48	52	
Title 1	25	26	27	28	30	32	34	38	43	48	
Title 2	19	20	21	22	24	26	28	34	39	44	
Title 3	17	18	19	20	22	24	26	31	37	43	
Headline	14	15	16	17	19	21	23	28	33	40	
Body	14	15	16	17	19	21	23	28	33	40	
Callout	13	14	15	16	18	20	22	26	32	38	

Subhead	12	13	14	15	17	19	21	25	30	36
Footnote	12	12	12	13	15	17	19	23	27	33
Caption 1	11	11	11	12	14	16	18	22	26	32
Caption 2	11	11	11	11	13	15	17	20	24	29

Source: Apple HIG → Typography → [iOS, iPadOS Dynamic Type sizes](#) and [iOS, iPadOS larger accessibility type sizes](#)

SwiftUI & Jetpack Compose Comparison Table

iOS (SwiftUI component)	Android (Composable)	Summary
Disclosure group	-	Accordion
Picker	-	Afford ability to choose one item among a group; can be displayed as a wheel, menu, or navigation link
Toolbar	Bottom app bar	Bar at bottom of screen that displays tools; not used for navigation
-	Snackbar	Brief notification that appears at the bottom of the screen
Action sheet (a.k.a. confirmation dialog)	-	Buttons that appear at the bottom of the screen to confirm or cancel an action
-	Carousel	Collection of items that can move in/out of view
Label	-	Combo of an icon and text, meant to introduce content
-	Chip	Compact element akin to button with four variations: assist, filter, input, suggestion
ScrollView	-	Container for scrollable content within a screen
List	-	Container that presents rows of data arranged in a single column
Grid	Lists and grids	Container view that arranges content. On Android, lists and grids can handle lazy loading. Note: Lists on iOS are not the same as lists on Android
Toggle	Switch	Control to choose between on and off states
Slider	Slider	Control to select from a defined range of values
-	Icon	Displays a glyph
Pull down button	-	Displays a menu of options + scrim upon tap (e.g. "More" button)
Context menu	Menu	Displays a menu of options upon long press
Pop up button	-	Displays a menu of options upon tap
Menu	DropdownMenu	Displays a series of actions upon tap
Image	Image	Displays an image

Navigation stack	-	Displays full screen breadcrumb pattern; same experience as navigation split view on mobile
Tab view	Bottom navigation	Displays interactive elements to switch between child views. Typically acts as navigation. On Android, may also provide access to other key actions such as a floating action button. On iOS, can take the form of "swipeys" with dots signifying each view.
-	Badge	Displays small number or text inside a background; used for displaying notification counts for example
Chart	-	Framework for creating a variety of chart styles
Web view	-	Framework for displaying web content
Page tab view	-	Full screen "swipeys"
-	Floating action button	High-emphasis button typically fixed in the corner of a screen.
Divider	Divider	Line to separate content
Group box	Card	Logical grouping of content inside a stylized view.
Navigation bar	Top app bar	On Android, provides access to key tasks and information. Generally hosts a title, core action items, and certain navigation items.
-	Scaffold	On Android, Scaffold assembles key composables together to form the structure of screen (e.g. <code>topBar</code> , <code>bottomBar</code> , <code>floatingActionButton</code>). On iOS, there is no specific equivalent. VStack is a suitable alternative.
Popover	-	On larger screens, displays a tooltip fixed to an element; On smaller screens, displays a full screen sheet with the element labelled
Navigation split view	-	On larger screens, displays navigation 2-3 side columns on larger screens. On smaller screens, displays full screen breadcrumb pattern
Alert	Alert dialog	Presents a disruptive message requiring interaction
Sheet	Bottom sheets	Presents a modal from the bottom of the screen
Date picker	Date picker	Selects either a single date or range of dates
Sheet (?)	Bottom drawer	Sheet that appears from the bottom intended for navigation
Gauge	-	Show a value within a range; can be displayed as a line or a semi-circle
Progress view	Progress indicators	Shows the progress toward completion of a task
Sidebar	Navigation drawer	Slide-in menu to display navigation
Checkbox	Checkbox	Toggle that displays a checkbox and label

Button	Button	Triggers an action. Various visual stylings provided by default. On iOS, there are many Button variations to perform specific actions (e.g. share, copy, paste, edit).
-	Modal drawer	
-	Scrollable tabs	
-	Time picker	
TipKit	Tooltip	
Radio group	Radio button	
Search	Search	
Segmented control	Segmented button	
Table	-	

Migration Support

We are offering migration support for teams that are currently leveraging Hyphen and/or Piano. Please [let us know](#) if the above documents do not provide the support you are looking for.

Migrating from Hyphen

Introduction

This page contains information for Hyphen users looking to migrate to TPL. Relative to Hyphen, TPL has an expanded audience. Thus, we've made changes that make TPL relevant to not just the Growth mission, but editorial product creators as well. TPL 1.0 also lacks some of Hyphen's features, such as [List Item](#). Future releases of TPL will aim for feature parity with Hyphen where possible.

Button

Summary

- Default appearance is now outlined, not filled
- Added separate **TextButton** component with a new “extra compact” size
- Added separate **IconButton** component with new “extended” and “extra compact” sizes
- Added a new `weight` property to control visual prominence
- Button states are slightly different
- Added subtle animations
- Various [React component changes](#)
- Added an optional accessibility enhancement to the “processing” state ([details](#))

Button Figma Component Changes

Hyphen	TPL	Notes
Button / Filled Light	<code>Button</code> <code>Weight: Emphasis</code> <code>Dark Mode: False</code>	Renamed and variants included as properties instead of as separate components.
Button / Filled Dark	<code>Button</code> <code>Weight: Emphasis</code> <code>Dark Mode: True</code>	Renamed and variants included as properties instead of as separate components.
Button / Outline Light	<code>Button</code> <code>Weight: Standard</code> <code>Dark Mode: False</code>	Renamed and variants included as properties instead of as separate components.

Button / Outline Dark	<code>Button</code> Weight: Standard Dark Mode: True	Renamed and variants included as properties instead of as separate components.
Button / Text Light	<code>Text Button</code> Dark Mode: False	Renamed and variants included as properties instead of as separate components.
Button / Text Dark	<code>Text Button</code> Dark Mode: False	Renamed and variants included as properties instead of as separate components.
Button / Text Light, Text Dark Height: Default	<code>Text Button</code> Height: Standard	Variant name changed from Default to Standard.
Button / Text Light, Text Dark Height: Min. tap target	<code>Text Button</code> Height: Compact, Extra Compact	Min tap target has changed. Compact (32px) and Extra Compact (24px) both meet our minimum tap target.
Icon: True, False	<code>Icon: Leading, Trailing, None</code>	Icons can now be Leading or Trailing.
Height: Default, Compact	<code>Height: Standard, Compact</code>	Compact variants are 32px tall in TPL, rather than 36px tall.
Width: Content, Flex, Max	Width: Content, Flex	Flex in TPL is used instead of Hyphen's Max.

Button React Component Changes

Summary	Hyphen	TPL
The new <code>weight</code> prop controls the TPL Button's visual prominence.	<code>variant</code> prop (filled, outline, text)	<code>weight</code> prop (standard, emphasis)
TPL Button has a new, less prominent default appearance (<code>weight="standard"</code>).	Default <code>variant</code> is 'filled' (has a black background)	Default <code>weight</code> is 'standard' (has a black outline and transparent background)
Hyphen Button's <code>variant="filled"</code> is equivalent to TPL Button's <code>weight="emphasis"</code> .	<code><Button variant="filled"></code>	<code><Button weight="emphasis"></code>
Hyphen Button's <code>variant="outline"</code> is equivalent to TPL Button's <code>weight="standard"</code> .	<code><Button variant="outline"></code>	<code><Button weight="standard"></code>
TextButton is now a separate component.	<code><Button variant="text"></code>	<code><TextButton></code>
The new <code>colorVariant</code> prop now handles Light / Dark Mode appearance separately.	<code><Button variant="*Dark"></code>	<code><Button colorVariant="alwaysDark"></code>
We've combined multiple system props into a single <code>sx</code> prop.	<code><Button m={[1, 2]} maxWidth="unset"></code>	<code><Button sx={{ m: [1, 2], maxWidth: 'unset' }}></code>
The <code>processing</code> prop now optionally accepts a string, letting TPL Button announce to screen readers what the "processing" state means.	<code><Button processing></code>	<code><Button processing="Signing up"></code>

See Also

- [TPL Button \(pre-release\) Storybook](#)

- [nytimes/news#5379](#)

Color

Except for the tokens named below, all other tokens have the same values and semantic names between TPL and Hyphen.

Hyphen Color Changes

Hyphen	TPL	Notes
Content Placeholder	Content Secondary (Dim)	
Stroke Secondary	Stroke Secondary	color change: • Light: #7F7F7F → #979797 • Dark: no change

Icon

Summary

- Renamed “UI” category to “Standard”
- Renamed “Social” category to “Third-Party”
- Moved “Super-T” icon to “Brand” category
- Renamed some icons
- Added *many* icons
- Added an additional, extra small size (12)

The tables below list each category of icons in both Hyphen and TPL Foundations.

UI Icons → Standard Icons

Hyphen	TPL Foundations
Alert	Alert
-	Arrow up
-	Arrow right
-	Arrow down
-	Arrow left
-	Calendar
Checkmark	Checkmark
Chevron_Down	Caret down
Chevron_Right	Caret right
Chevron_Up	Caret up
-	Caret left

Close	Close
-	Create
Dot	Dot
-	Folder
Help	Help
-	History
-	Home
Info	Info
-	Link
Menu	Menu
-	Muted
Person	Person
-	Play
-	Play circle
Plus	Plus
-	Refresh
-	Save
-	Save filled
-	Search
-	Settings
-	Share (Android)
-	Share (Apple)
-	Star
Success	Checkmark circle
-	Unmuted

Social Icons → Third-Party Icons

Hyphen	TPL Foundations	Notes
Apple	Apple	
Facebook	Facebook	
Google	Google	
NYT—T	<i>Brand Icons → Super-T</i>	Renamed and moved to Brand Icons category

Brand Icons

Hyphen	TPL Foundations	Notes
Games	Games	
Home Delivery	Newspaper	
Cooking	Cooking	
Wirecutter	Wirecutter	
The Athletic	The Athletic	
Social Icons → NYT—T	Super-T	Renamed and moved from Social Icons category

List Item (Pending)

This component has not been added as of TPL 1.0 launch on 7/11/2023. Migration support will be added once this component is available.

Radio (Pending)

This component has not been added as of TPL 1.0 launch on 7/11/2023. Migration support will be added once this component is available.

Rule

The Hyphen Rule component has the same variants as the TPL Rule component. The only differences are caused by changes in our Stroke Secondary value, [documented above](#).

Spacing

TPL's spacing tokens are identical to those in Hyphen.

States

TPL 1.0 does not have formal specifications for element states in the same way that Hyphen does, though TPL's components such as Button reflect our current design thinking.

The `@nyt/tpl` Root component also provides default focus styles to all child elements.

Text Link (Pending)

This component has not been added as of TPL 1.0 launch on 7/11/2023. Migration support will be added once this component is available.

Typography

Summary

- Revised naming conventions to reference font size
- Added more granular size options across groupings (e.g. Title, Text, etc.)
- Deprecated Bold styles of Franklin in favor of Semibold
- Deprecated Specialty font tokens within TPL because they are not all-purpose and conflict with Headline styles used in editorial contexts.
- Deprecated Interactive font tokens in favor of [Text Button](#) and [Inline Link](#) (coming with TPL 1.2).

The tables below list each category of Typography tokens in both Hyphen and TPL Foundations.

Core Typography Changes

Hyphen	TPL	Notes
Core / Headline 1	Title / 28	Hyphen's Headline 1 uses 26px, a size we do not offer. You may opt for Title 28 or Title 24.
Core / Headline 2	Title / 22	
Core / Title 1	Title / 18	TPL Title uses Semibold instead of Bold.
Core / Title 2	Title / 16	TPL Title uses Semibold instead of Bold.
Core / Title 3	Title / 14	TPL Title uses Semibold instead of Bold.
Core / Subtitle 1	Title / 18	
Core / Subtitle 2	Title / 16	
Core / Subtitle 3	Title / 14	
Core / Body 1	Text / 18	
Core / Body 2	Text / 16	
Core / Body 3	Text / 14	

Specialty Typography Changes

Hyphen	TPL	Notes
Specialty / Legal 1	Text / 12	Legal text uses the generic Text token.
Specialty / Display 1—L	Title / 56	Use Title tokens (Semibold Franklin) instead of Cheltenham for this use case.
Specialty / Display 1—M	Title / 48	Use Title tokens (Semibold Franklin) instead of Cheltenham for this use case.
Specialty / Display 1—S	Title / 32	Use Title tokens (Semibold Franklin) instead of Cheltenham for this use case.
Specialty / Display 2—L	Title / 48	Use Title tokens (Semibold Franklin) instead of Cheltenham for this use case.

Specialty / Display 2 - M	Title / 48	Use Title tokens (Semibold Franklin) instead of Cheltenham for this use case.
Specialty / Display 2 - S	Title / 32	Use Title tokens (Semibold Franklin) instead of Cheltenham for this use case.
Specialty / Display 3	Title / 28	Use Title tokens (Semibold Franklin) instead of Cheltenham for this use case.
Specialty / Subscription 1	Title / 32	Use Title tokens (Semibold Franklin) instead of Cheltenham for this use case.

Overall / Subscriptions 2 **Title / 32** [View Title Tokens / Considerations](#)

Interactive Typography Changes

Hyphen	TPL	Notes
Interactive / Headline 1	Text button or Inline Link (coming in TPL 1.2)	We are not offering interactive (underlined) text styles.
Interactive / Headline 2	Text button or Inline Link (coming in TPL 1.2)	We are not offering interactive (underlined) text styles.
Interactive / Title 1	Text button or Inline Link (coming in TPL 1.2)	We are not offering interactive (underlined) text styles.
Interactive / Title 2	Text button or Inline Link (coming in TPL 1.2)	We are not offering interactive (underlined) text styles.
Interactive / Title 3	Text button or Inline Link (coming in TPL 1.2)	We are not offering interactive (underlined) text styles.
Interactive / Subtitle 1	Text button or Inline Link (coming in TPL 1.2)	We are not offering interactive (underlined) text styles.
Interactive / Subtitle 2	Text button or Inline Link (coming in TPL 1.2)	We are not offering interactive (underlined) text styles.
Interactive / Subtitle 3	Text button or Inline Link (coming in TPL 1.2)	We are not offering interactive (underlined) text styles.
Interactive / Body 1	Text button or Inline Link (coming in TPL 1.2)	We are not offering interactive (underlined) text styles.
Interactive / Body 2	Text button or Inline Link (coming in TPL 1.2)	We are not offering interactive (underlined) text styles.
Interactive / Body 3	Text button or Inline Link (coming in TPL 1.2)	We are not offering interactive (underlined) text styles.

Migrating from Piano

Introduction

This page contains information for Piano users looking to migrate to TPL.

Overall, TPL 1.0 includes equivalents to many of Piano's design tokens and components, but is ultimately a different system. Thus, something from Piano may exist in TPL, but with a different name and some modifications. Some items have been removed as well.

Migration Policy

TPL 1.0 is a new design system with new guidance and implementing libraries. This means:

- [Deprecated](#) Piano tokens (e.g. colors) have been outright removed in TPL
- If a Piano token or component *does* have a direct equivalent in TPL, it may have a different name or programming interface

Color

Summary

Relative to Piano, TPL makes the following color token changes:

- Moved Signal Highlight to the Background palette (i.e. [Background Highlight](#))
- Moved all other Signal colors to the Content palette ([details](#))
- Combined Stroke Tertiary and Quaternary into Stroke Tertiary ([details](#))
- Removed Piano [deprecated](#) colors ([example](#))
- Renamed several colors ([example](#))
- Changed a few color values ([example](#))

Content Palette Changes

Piano	TPL	Notes
Content Primary	Content Primary	
Content Inverse Primary (Deprecated)	Content Primary with "User Inverse" color behavior	Use Content Primary with the "User Inverse" color behavior instead.

Content Secondary	Content Primary (Dim)	<p>Color value changed:</p> <ul style="list-style-type: none"> Dark: Gray 20 → Gray 15 <p>Regarding name collisions:</p> <p>TPL also has a color named Content Secondary, but it is instead equivalent to Piano's Content Tertiary.</p> <p>Thus, TPL Root defines color content secondary, which overrides Piano's color content secondary.</p>
Content Tertiary	Content Secondary	
Content Quaternary	Content Secondary (Dim)	
Content Quintary	Content Placeholder	
Signal Accent	Content Accent	<p>Moved to Content palette.</p> <p>Color values changed:</p> <ul style="list-style-type: none"> Light: Blue 50 → Accent Blue 55 Dark: Blue 35 / #5A9AF2 → New Blue 35 / #6BA1DD
Signal Positive	Content Positive	Moved to Content palette.
Signal Negative	Content Negative	Moved to Content palette.
Signal Editorial	Content Accent (Dim)	Moved to Content palette.
Signal Breaking	Content Breaking	Not used in nytimes/ios-newsreader-fusion

Signal → Content Palette Changes

Piano	TPL	Notes
Signal Accent	Content Accent	<p>Moved to Content palette.</p> <p>Color values changed:</p> <ul style="list-style-type: none"> Light: Blue 50 → Accent Blue 55 Dark: Blue 35 / #5A9AF2 → New Blue 35 / #6BA1DD
Signal Positive	Content Positive	Moved to Content palette.
Signal Negative	Content Negative	Moved to Content palette.
Signal Developing	-	
Signal Editorial	Content Accent (Dim)	Moved to Content palette.
Signal Breaking	Content Breaking	Not used in nytimes/ios-newsreader-fusion
Signal Highlight	Background Highlight	Moved to Background palette.

Background Palette Changes

Piano	TPL	Notes
-------	-----	-------

Background Primary	Background Primary	
Background Inverse Primary (Deprecated)	Background Primary with "User Inverse" color behavior	
Background Secondary	Background Secondary	
Background Inverse Secondary (Deprecated)	Background Secondary with "User Inverse" color behavior	
Background Tertiary	Background Tertiary	Color value changed: <ul style="list-style-type: none">Dark: Gray 85 → Gray 80
Background Quaternary	–	
Background Elevated	–	
Background Overlay (Deprecated)	Background Scrim	We previously deprecated Piano's Background Overlay color. This non-deprecated TPL color replaces it.
Signal Highlight	Background Highlight	Moved to Background palette.

Stroke Palette Changes

Piano	TPL	Notes
Stroke Primary	Stroke Primary	
Stroke Inverse Primary (Deprecated)	Stroke Primary with "User Inverse" color behavior	
Stroke Secondary	Stroke Secondary	Color values changed: <ul style="list-style-type: none">Light: Gray 50 → Gray 55Dark: Gray 50 → Gray 45
Stroke Tertiary	Stroke Tertiary	We've combined Stroke Tertiary and Quaternary into a single color token named Stroke Tertiary. Color values changed: <ul style="list-style-type: none">Light: Gray 25 → Gray 15Dark: Gray 70 → Gray 80
Stroke Quaternary	Stroke Tertiary	We've combined Stroke Tertiary and Quaternary into a single color token named Stroke Tertiary. The color values for Piano Stroke Quaternary and TPL Foundations 1.0 Stroke Tertiary are the same: <ul style="list-style-type: none">Light: Gray 15 (Piano · TPL)Dark: Gray 80 (Piano · TPL)

Other Color Changes

Piano	TPL	Notes
Overlay Black	-	In most cases, use TPL Background Scrim instead. Note: Piano Overlay Black is defined as pure black without transparency (source). By comparison, TPL Background Scrim is a semi-transparent Gray 100 (source).
Static White (Deprecated)	-	
Static Gray 100 (Deprecated)	-	

Appendix A: Sources

- [Tokens & Components | Piano](#)
- [TPL | Tokens & Components](#)
- [💀 DEPRECATED: Color & Type Extended | TPL Foundations \[Alpha\]](#)
- Piano Deprecated Colors: [colors-deprecated.json](#) · [SUBX-2833](#)
- [NYT Color Spectrums \[Beta\]](#) (source of named color values, e.g. [Accent Blue 35](#))
- [Work Zone | TPL Foundations](#) (source of the “New Blue” used for the new [Content Accent](#) colors, which differ from [Accent Blue 55](#) and [Accent Blue 35](#))

Appendix B: All Color Changes

Piano	TPL	Notes
Content Primary	Content Primary	
Content Inverse Primary (Deprecated)	Content Primary with “User Inverse” color behavior	Use Content Primary with the “User Inverse” color behavior instead.
Content Secondary	Content Primary (Dim)	Color value changed: <ul style="list-style-type: none">• Dark: Gray 20 → Gray 15 Regarding name collisions: TPL also has a color named Content Secondary , but it is instead equivalent to Piano’s Content Tertiary . Thus, TPL Root defines color content secondary , which overrides Piano’s color content secondary .
Content Tertiary	Content Secondary	
Content Quaternary	Content Secondary (Dim)	
Content Quintary	Content Placeholder	
Stroke Primary	Stroke Primary	
Stroke Inverse Primary (Deprecated)	Stroke Primary with “User Inverse” color behavior	

Stroke Secondary	Stroke Secondary	<p>Color values changed:</p> <ul style="list-style-type: none"> Light: Gray 50 → Gray 55 Dark: Gray 50 → Gray 45
Stroke Tertiary	Stroke Tertiary	<p>We've combined Stroke Tertiary and Quaternary into a single color token named Stroke Tertiary.</p> <p>Color values changed:</p> <ul style="list-style-type: none"> Light: Gray 25 → Gray 15 Dark: Gray 70 → Gray 80
Stroke Quaternary	Stroke Tertiary	<p>We've combined Stroke Tertiary and Quaternary into a single color token named Stroke Tertiary.</p> <p>The color values for Piano Stroke Quaternary and TPL Foundations 1.0 Stroke Tertiary are the same:</p> <ul style="list-style-type: none"> Light: Gray 15 (Piano · TPL) Dark: Gray 80 (Piano · TPL)
Background Primary	Background Primary	
Background Inverse Primary (Deprecated)	Background Primary with "User Inverse" color behavior	
Background Secondary	Background Secondary	
Background Inverse Secondary (Deprecated)	Background Secondary with "User Inverse" color behavior	
Background Tertiary	Background Tertiary	<p>Color value changed:</p> <ul style="list-style-type: none"> Dark: Gray 85 → Gray 80
Background Quaternary	–	
Background Elevated	–	
Background Overlay (Deprecated)	Background Scrim	We previously deprecated Piano's Background Overlay color. This non-deprecated TPL color replaces it.
Signal Accent	Content Accent	<p>Moved to Content palette.</p> <p>Color values changed:</p> <ul style="list-style-type: none"> Light: Blue 50 → Accent Blue 55 Dark: Blue 35 / #5A9AF2 → New Blue 35 / #6BA1DD
Signal Positive	Content Positive	Moved to Content palette.
Signal Negative	Content Negative	Moved to Content palette.
Signal Developing	–	
Signal Editorial	Content Accent (Dim)	Moved to Content palette.
Signal Breaking	Content Breaking	Not used in nytimes/ios-newsreader-fusion

Signal Highlight	Background Highlight	Moved to Background palette.
Overlay Black	-	<p>In most cases, use TPL Background Scrim instead.</p> <p>Note: Piano Overlay Black is defined as pure black without transparency (source). By comparison, TPL Background Scrim is a semi-transparent Gray 100 (source).</p>

Appendix C: Piano Colors Whose TPL Equivalent Has a Different Color Value

Piano	TPL	Notes
Content Secondary	Content Primary (Dim)	<p>Color value changed:</p> <ul style="list-style-type: none"> Dark: Gray 20 → Gray 15 <p>Regarding name collisions: TPL also has a color named Content Secondary, but it is instead equivalent to Piano's Content Tertiary. Thus, TPL Root defines color content secondary, which overrides Piano's color content secondary.</p>
Stroke Secondary	Stroke Secondary	<p>Color values changed:</p> <ul style="list-style-type: none"> Light: Gray 50 → Gray 55 Dark: Gray 50 → Gray 45
Stroke Tertiary	Stroke Tertiary	<p>We've combined Stroke Tertiary and Quaternary into a single color token named Stroke Tertiary.</p> <p>Color values changed:</p> <ul style="list-style-type: none"> Light: Gray 25 → Gray 15 Dark: Gray 70 → Gray 80
Background Tertiary	Background Tertiary	<p>Color value changed:</p> <ul style="list-style-type: none"> Dark: Gray 85 → Gray 80
Signal Accent	Content Accent	<p>Moved to Content palette.</p> <p>Color values changed:</p> <ul style="list-style-type: none"> Light: Blue 50 → Accent Blue 55 Dark: Blue 35 / #5A9AF2 → New Blue 35 / #6BA1DD

Appendix D: Piano Colors With No TPL Equivalent

Piano	TPL	Notes
Background Quaternary	-	
Background Elevated	-	
Signal Developing	-	

Overlay Black	-	In most cases, use TPL Background Scrim instead.
		Note: Piano Overlay Black is defined as pure black without transparency (source). By comparison, TPL Background Scrim is a semi-transparent Gray 100 (source).
Static White (Deprecated)	-	

Static Gray 100	-	
		(Deprecated)

Typography

Summary

- Implement all text tokens in Figma as text styles
- Changed some line heights
- Removed several supported font sizes
- Removed default responsive sizes
- Some components with customizable font weight are now separate React components (e.g. StandardHeading)
- Some type tokens have direct equivalents:
 - Paragraph → Body
 - **Note:** TPL Body lacks built-in inline link styles
 - StandardHeading 400 → Headline
 - StandardHeading 700 → NewsHeadline
 - LightHeading → FeatureHeadline
 - OpinionHeading → OpinionHeadline
- These Figma and React components do *not* have direct equivalents:
 - Generic Text
 - Blockquote
 - Byline
- Added these type tokens:
 - Text (not compatible)
 - Title
 - Label
- Added generic Typography React component ([more migration guidance](#))

The following tables use a “slash” notation to denote Font Size / Line Height Ratio.

Example: **20 / 1.5** means, “20 point font size on 1.5 (150%) line height”.

Paragraph → Body Sizes

Piano	TPL
-	14 / 1.39
-	16 / 1.39
18* / 1.4	18 / 1.39
20* / 1.4	20 / 1.5

* These font sizes were only available on certain screen sizes. By comparison, TPL's Body text token supports multiple font sizes and responsive sizing.

Note: Body lacks built-in inline link styles.

StandardHeading 400 → Headline Sizes

Piano	TPL
14/1.25	-
16 / 1.2	16 / 1.2
18 / 1.2	18 / 1.2
20/1.15	-
22/1.15	-
24 / 1.15	24 / 1.15
28 / 1.15	28 / 1.15
32/1.15	-
36/1.15	-
40/1.1	-
48/1.1	-
56/1	-
64/1	-
72/1	-

StandardHeading 700 → NewsHeadline Sizes

Piano	TPL
14/1.25	-
16 / 1.2	16 / 1.2
18 / 1.2	18 / 1.2
20 / 1.15	20 / 1.15
22 / 1.15	22 / 1.15
24/1.15	-

28 / 1.15	28 / 1.15
32 / 1.15	32 / 1.15
36 / 1.15	36 / 1.15
40 / 1.1	40 / 1.1
48 / 1.1	48 / 1.1
56 / 1	-
64 / 1	-
72 / 1	-

LightHeading → FeatureHeadline Sizes

Piano	TPL
22 / 1.15	-
24 / 1.15	24 / 1.15
28 / 1.15	28 / 1.15
32 / 1.15	32 / 1.15
36 / 1.15	-
40 / 1.1	-
48 / 1.1	48 / 1.1
56 / 1	56 / 1

OpinionHeading Chelt Cond → OpinionHeadline Sizes

Piano	TPL
-	Chelt Text Cond 18 / 1.15
-	Chelt Text Cond 20 / 1.15
Chelt Cond 22 / 1.2	Chelt Text Cond 22 / 1.15
Chelt Cond 24 / 1.2	Chelt Text Cond 24 / 1.15
Chelt Cond 28 / 1.2	-
Chelt Cond 32 / 1.2	-
Chelt Cond 36 / 1.15	Chelt Text Cond 36 / 1.15
Chelt Cond 40 / 1.15	-
Chelt Cond 48 / 1.1	-
Chelt Cond 56 / 1.1	Chelt Cond 56 / 1.1

OpinionHeading Chelt Text Cond → OpinionHeadline Sizes

Piano	TPL
-	Chelt Text Cond 18 / 1.15
-	Chelt Text Cond 20 / 1.15
Chelt Text Cond 22 / 1.15	Chelt Text Cond 22 / 1.15
Chelt Text Cond 24 / 1.15	Chelt Text Cond 24 / 1.15
Chelt Text Cond 28 / 1.15	-
Chelt Text Cond 32 / 1.15	-
Chelt Text Cond 36 / 1.15	Chelt Text Cond 36 / 1.15
Chelt Text Cond 40 / 1.15	-
Chelt Text Cond 48 / 1.1	-
Chelt Cond 56 / 1.1	Chelt Cond 56 / 1.1

Generic Text

Piano offers a [generic Text component](#) which supports *all* possible typefaces, font sizes, line height and letter spacing values. It was also the main way to access Franklin-based type tokens.

TPL 1.0 does not offer a direct equivalent. Instead, please see:

- The [Text](#), [Title](#) and [Label](#) text tokens for Franklin-based typography
- The [Typography component](#) for advanced programmatic text token usage (note: more limited than Piano)

Blockquote

TPL 1.0 does not offer a Blockquote component.

It's possible to construct a similar layout using TPL spacing, color and type tokens. For example:

- **Color:** [Content Secondary](#)
- **Typography:** [Body/16 or Body/18](#)
- **Vertical Rule:** Secondary
- **Spacing** [between rule and text]: 2

Byline

TPL 1.0 does not offer a Byline component.

Please consider using these TPL tokens to construct a similar layout:

- **Color:** [Content Primary Dim](#)
- **Typography:** [Title/14](#)

Note: TPL 1.0 does not currently provide inline link styles.

Figure & FigCaption

TPL 1.0 does not offer a Figure or FigCaption component.

For FigCaption typography, please consider using these TPL tokens to construct a similar layout:

Caption Text

- **Color:** [Content Secondary Dim](#)
- **Typography:** [Body/14 or Body/16](#)

Credit Text

- **Color:** [Content Secondary Dim](#)
- **Typography:** [Text/12 or Text/14](#)

LinkBlock

This component has not been added as of TPL 1.0 launch on 7/11/2023. Migration support will be added once this component is available.

Rule (Pending)

Rule is available in TPL but migration support has not been added as of TPL 1.0 launch on 7/11/2023. Instead, please reach out to us on Slack or in Office Hours.

Spacing

TPL's spacing tokens are identical to Piano's.

Text Link

This component has not been added as of TPL 1.0 launch on 7/11/2023. Migration support will be added once this component is available.

Work Zone

View of Appendix B: All Color Changes

Piano	TPL	Notes
Content Primary	Content Primary	
Content Inverse Primary (Deprecated)	Content Primary with "User Inverse" color behavior	Use Content Primary with the "User Inverse" color behavior instead.

Content Secondary	Content Primary (Dim)	<p>Color value changed:</p> <ul style="list-style-type: none"> Dark: Gray 20 → Gray 15 <p>Regarding name collisions:</p> <p>TPL also has a color named Content Secondary, but it is instead equivalent to Piano's Content Tertiary.</p> <p>Thus, TPL Root defines color content secondary, which overrides Piano's color content secondary.</p>
Content Tertiary	Content Secondary	
Content Quaternary	Content Secondary (Dim)	
Content Quintary	Content Placeholder	
Stroke Primary	Stroke Primary	
Stroke Inverse Primary (Deprecated)	Stroke Primary with "User Inverse" color behavior	
Stroke Secondary	Stroke Secondary	<p>Color values changed:</p> <ul style="list-style-type: none"> Light: Gray 50 → Gray 55 Dark: Gray 50 → Gray 45
Stroke Tertiary	Stroke Tertiary	<p>We've combined Stroke Tertiary and Quaternary into a single color token named Stroke Tertiary.</p> <p>Color values changed:</p> <ul style="list-style-type: none"> Light: Gray 25 → Gray 15 Dark: Gray 70 → Gray 80
Stroke Quaternary	Stroke Tertiary	<p>We've combined Stroke Tertiary and Quaternary into a single color token named Stroke Tertiary.</p> <p>The color values for Piano Stroke Quaternary and TPL Foundations 1.0 Stroke Tertiary are the same:</p> <ul style="list-style-type: none"> Light: Gray 15 (Piano · TPL) Dark: Gray 80 (Piano · TPL)
Background Primary	Background Primary	
Background Inverse Primary (Deprecated)	Background Primary with "User Inverse" color behavior	
Background Secondary	Background Secondary	
Background Inverse Secondary (Deprecated)	Background Secondary with "User Inverse" color behavior	
Background Tertiary	Background Tertiary	<p>Color value changed:</p> <ul style="list-style-type: none"> Dark: Gray 85 → Gray 80
Background Quaternary	-	
Background Elevated	-	

Background Overlay (Deprecated)	Background Scrim	We previously deprecated Piano's Background Overlay color. This non-deprecated TPL color replaces it.
Signal Accent	Content Accent	Moved to Content palette. Color values changed: <ul style="list-style-type: none">• Light: Blue 50 → Accent Blue 55• Dark: Blue 35 / #5A9AF2 → New Blue 35 / #6BA1DD
Signal Positive	Content Positive	Moved to Content palette.
Signal Negative	Content Negative	Moved to Content palette.
Signal Developing	–	
Signal Editorial	Content Accent (Dim)	Moved to Content palette.
Signal Breaking	Content Breaking	Not used in nytimes/ios-newsreader-fusion
Signal Highlight	Background Highlight	Moved to Background palette.
Overlay Black	–	In most cases, use TPL Background Scrim instead. Note: Piano Overlay Black is defined as pure black without transparency (source). By comparison, TPL Background Scrim is a semi-transparent Gray 100 (source).

Glossary

accessible name	The name of an element passed to assistive technologies; can be provided through multiple techniques (e.g. label, aria)
assistive technology	Preferences, software, and hardware that adjusts or interprets a digital experience to match the needs of a user
cinemagraph	Short-form video suitable that acts similarly to imagery; typically does not feature sound and lasts less than 5 seconds
color contrast	The difference in brightness between foreground and background colors, evaluated as a ratio (e.g. 4.5:1, 3:1)
hit area	The spatial region in which an element can be interacted with
interactive element	Elements that a reader can interact with by clicking/tapping; for example, buttons, text inputs, checkboxes, and drop-down inputs
keyboard accessible	Usable for people who rely on the keyboard to navigate digital experiences
keyboard navigation	Using the keys on a keyboard to navigate a website or app; for example, using <code>tab</code> to move to the next interactive element or <code>shift + tab</code> to move to the previous interactive element
screen reader	Software that reads aloud digital experiences and offers non-visual methods of navigation
visual chunking	Grouping together elements in close visual proximity
wayfinding	Determining current location and the options for moving forward, backward, or adjacent in an experience



Legacy

We are responsible for a few legacy design systems at The Times. Although they will not be iterated on, we keep these documented for those who have yet to migrate to TPL. You can find both Hyphen and Piano system documentation [here](#).

⚠ Currently, we are maintaining but no longer updating the Hyphen & Piano Design Systems. We recommend using [TPL Foundations](#) for any new projects.

To migrate off either system, [we're here to help](#). Hyphen will be decommissioned in approximately one year, in July 2024.

Hyphen

! We are currently maintaining but no longer updating the Hyphen Design System. We recommend using [TPL Foundations](#) for any new projects.

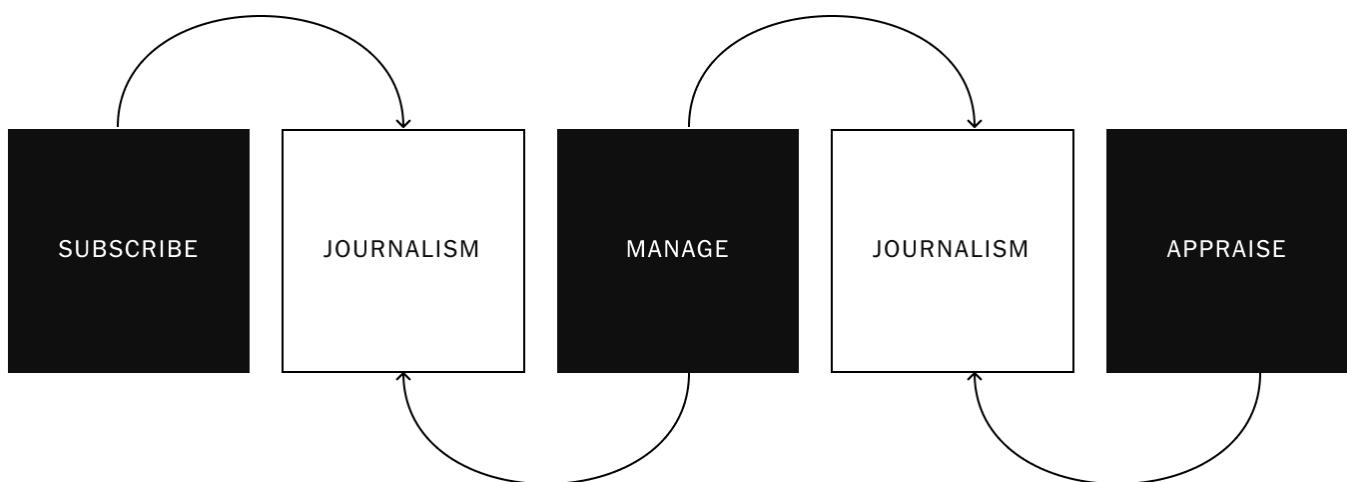
To migrate off the Hyphen system, [we're here to help](#). Hyphen will be decommissioned in approximately one year, in July 2024.

Our system is a collection of design elements that can be assembled together to build any number of experiences. It is a living, breathing system that's flexible, maintainable, stable, scalable – ever-changing with our brand and product needs.

About us

Our focus

Hyphen is one of the multiple design systems used at The Times. It should be used when designing experiences that represent the relationship between our customers and The New York Times as a company.



These experiences act as connections between our journalism, informing customers and allowing them to take action to create their ideal Times experience.

 Our focus is expanding! Interested in where Hyphen is going next? [Find out more](#)

Our goals

By using Hyphen you're helping create consistent, quality experiences while reducing design and engineering time and debt.

- **Deliver coherence and predictability to our products.** All of our guidance is created by auditing our various products and refined to include variations and documentation so they can be used consistently at every touchpoint.
- **Reduce design and engineering time and debt.** By solving the basic building blocks of a user interface, like typography and buttons, designers and engineers can focus on the intricacies and custom UI of their experiences.
- **Raise the quality of our experiences.** All of our tokens and components meet industry quality standards – including states and variations, and meet accessibility standards.

It is not just a pattern library that can be assembled in near-infinite ways – each element has defined standards that help create a shared understanding of an exceptional user experience.

Getting started

Tokens

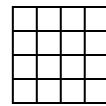
Tokens are foundational elements, like color and typography, used in conjunction to make every product's components and layouts. Tokens are designed to provide broad guidance allowing them to work for current use cases and scale as new ones arise.



TYPOGRAPHY



COLOR



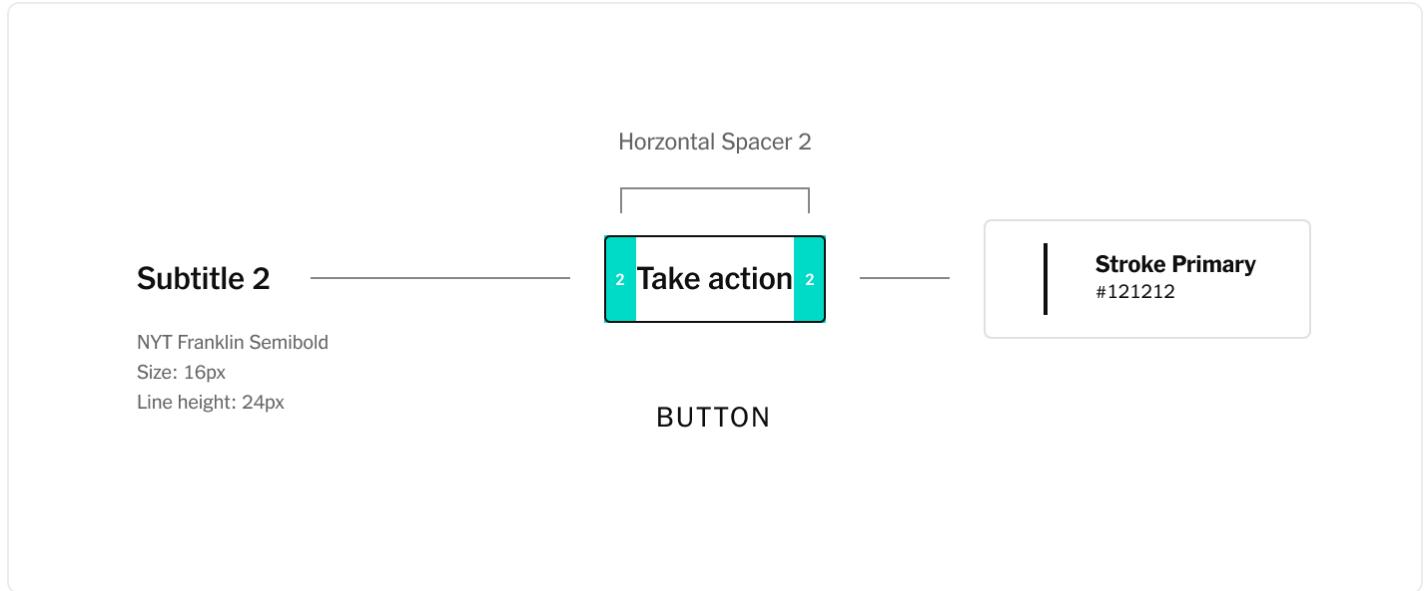
LAYOUT



STATES

Components

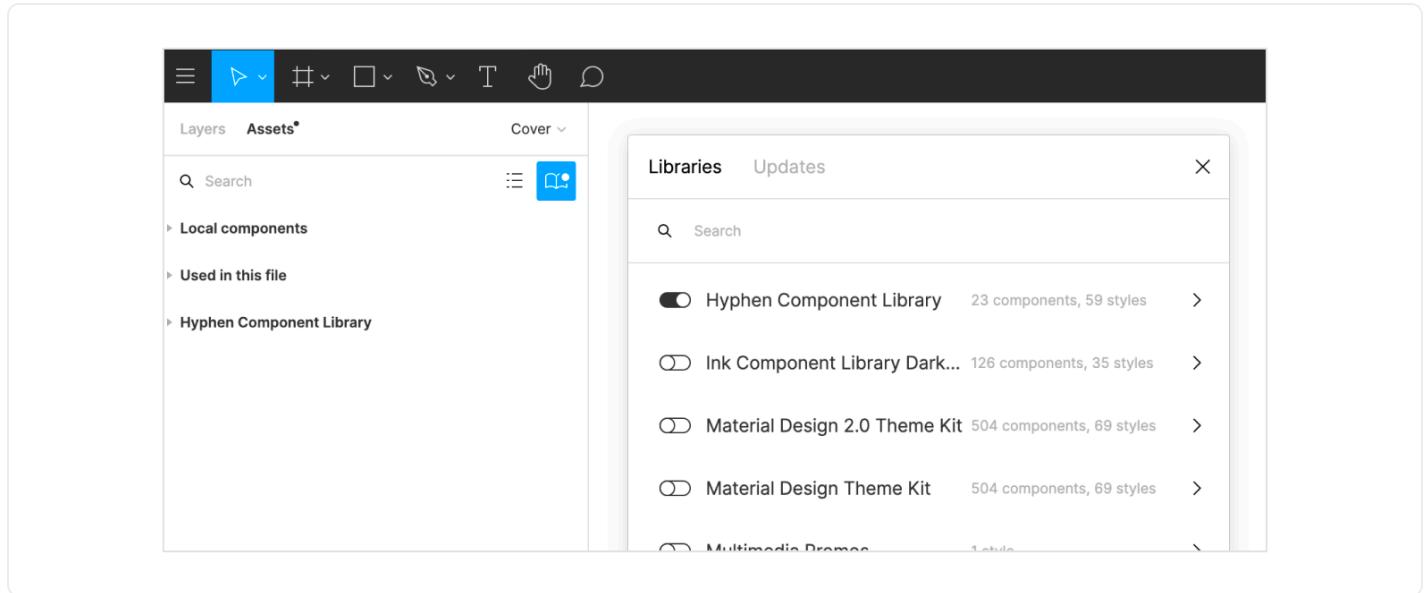
Components are reusable building blocks for creating user interfaces – generic enough to be used across multiple products and platforms. They keep common interaction patterns consistent, reduce engineering time, have built-in solutions for accessibility. They are designed using a combination of our tokens.



Resources

Figma libraries

Hyphen is available for designers in our [Figma library](#) through reusable style and components. Enable the library through the "Assets" tab in the left panel.



Storybook

Hyphen is available as reusable code within the Account product codebase. All tokens and components are available to preview on [Storybook](#).

Support

If you have any questions regarding our resources, you can find us on Slack:



Our Slack channels

Hyphen channel: [#hyphen](#)

Team channel: [#subx-design-systems](#)

Slack group:

[@design-systems-team](#)



Quick links

[Design Systems Figma Docs & Libraries](#)

[Hyphen Storybook](#)

[Design Systems Jira](#)

Button

Buttons allow users to take actions, and make choices, with a single tap. Buttons communicate actions that users can take.

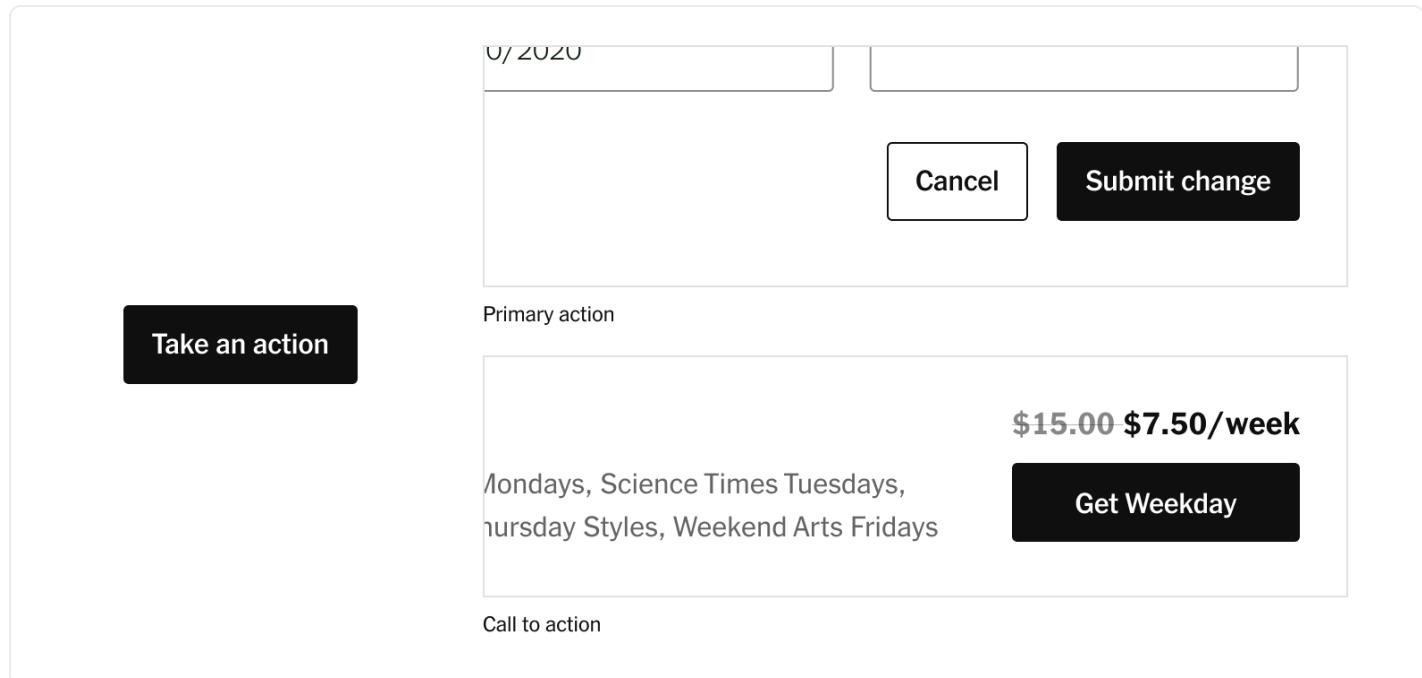
 We are currently maintaining but no longer updating the Hyphen Design System. We recommend using [TPL Buttons](#) for any new projects.

To migrate off the Hyphen system, [we're here to help](#). Hyphen will be decommissioned in approximately one year, in July 2024.

Type

Fill

Fill buttons are used for the most prominent call to action on most screens and flows. They are always the primary action in an experience.



Primary action

Call to action

Take an action

U/2020

Cancel

Submit change

\$15.00 \$7.50/week

Mondays, Science Times Tuesdays,
Thursday Styles, Weekend Arts Fridays

Get Weekday

Outline

Outline buttons are used for less prominent calls to action on most screens and flows. They are best used for the de-emphasized action or actions in an experience, which can be the primary or secondary action.

07/2020

Cancel

Submit change

Secondary action

Take an action

Opt out

Cookie Privacy Act

similar methods to recognize visitors and remember preferences. We also use them to measure ad campaign effectiveness, site traffic. To learn more about these methods, view our cookie policy.

De-emphasized primary action

Text

Text buttons are used for the least prominent calls to action. They are used to avoid distracting the user from important content or for actions that we expect users to rarely interact with.

Cookie Privacy Act

similar methods to recognize visitors and remember preferences. We also use them to measure ad campaign effectiveness, site traffic. To learn more about these methods, view our cookie policy.

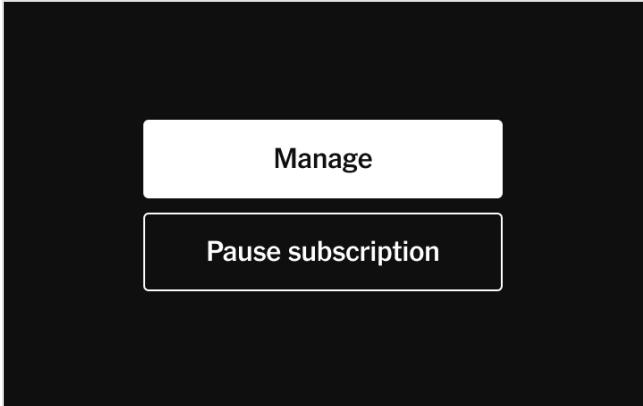
Take an action

Opt out

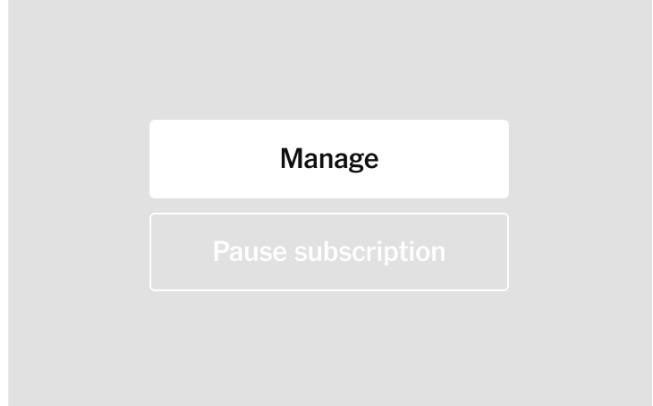
Tertiary action

Inverse

Inverse buttons are offered in Fill and Outline variations to be used on dark backgrounds. Due to the visual appearance, ensure that they are used on backgrounds that meet the contrast accessibility requirements.



✓ Accessible background color



✗ Not accessible background color for Outline Inverse Button

Content

Buttons can either contain text or both text and an icon.

Take an action

TEXT

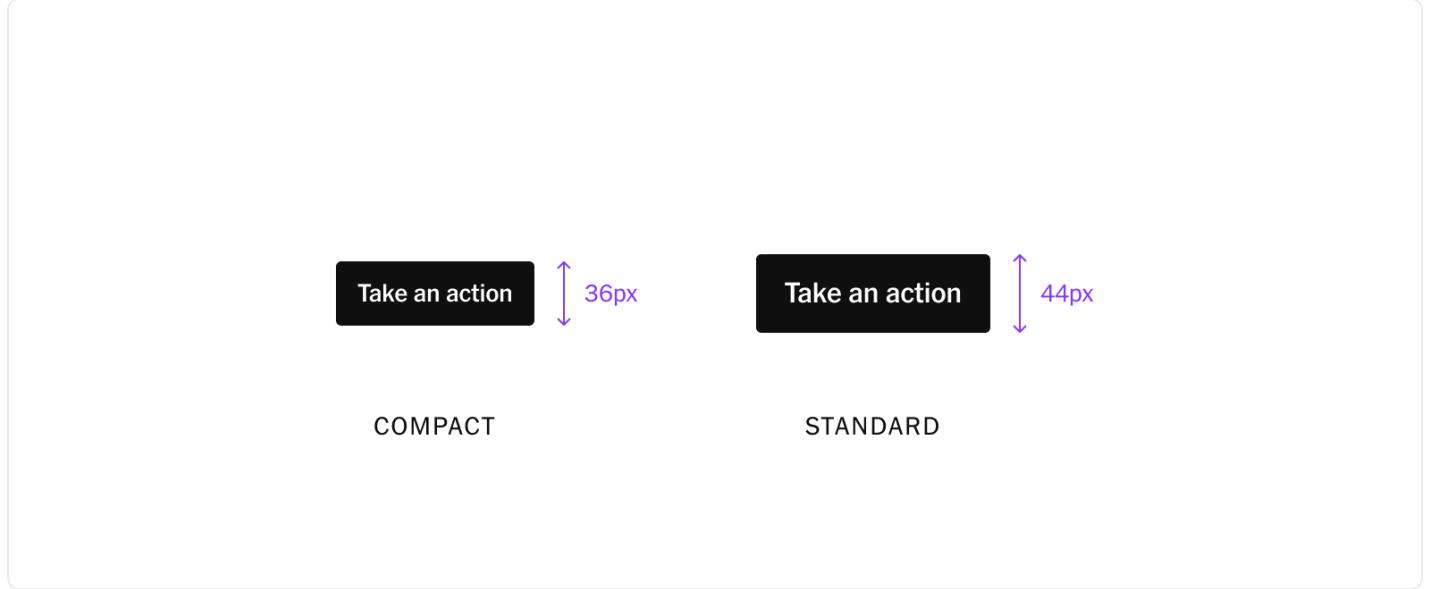
+ Take an action

ICON + TEXT

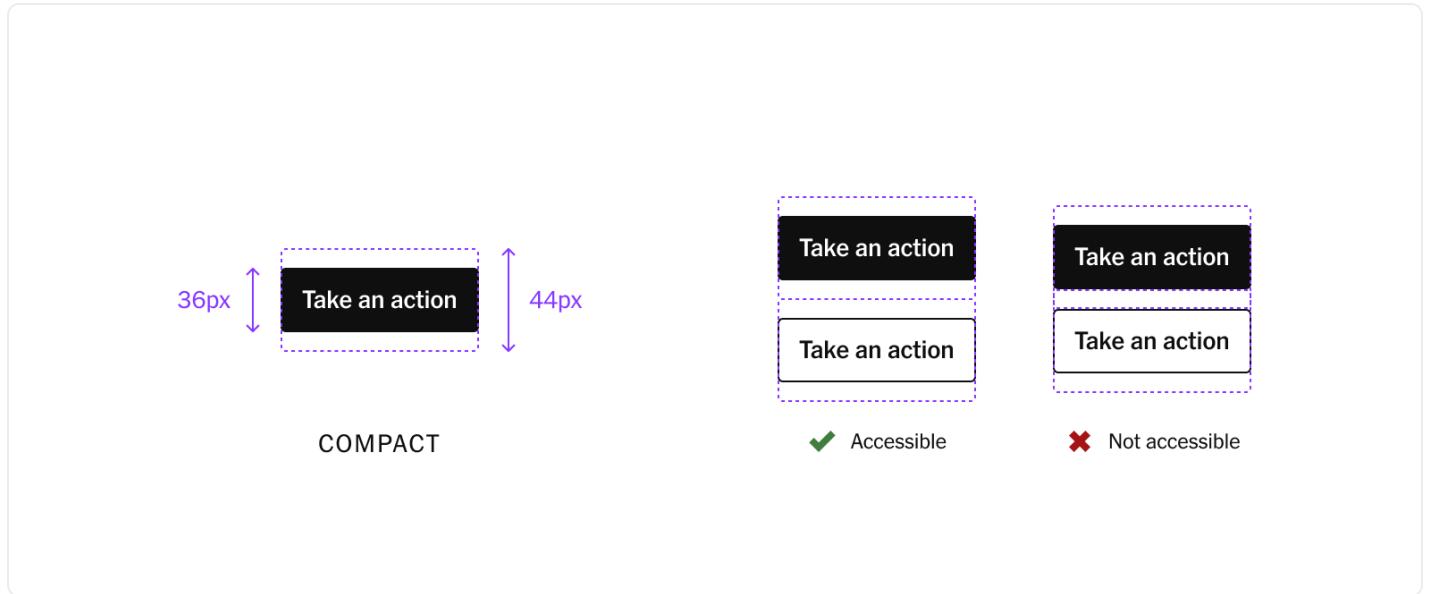
Size

Height

Our buttons come in two heights: standard and compact. They differ in component height. The height of the standard button is set to meet accessibility tap target requirements.

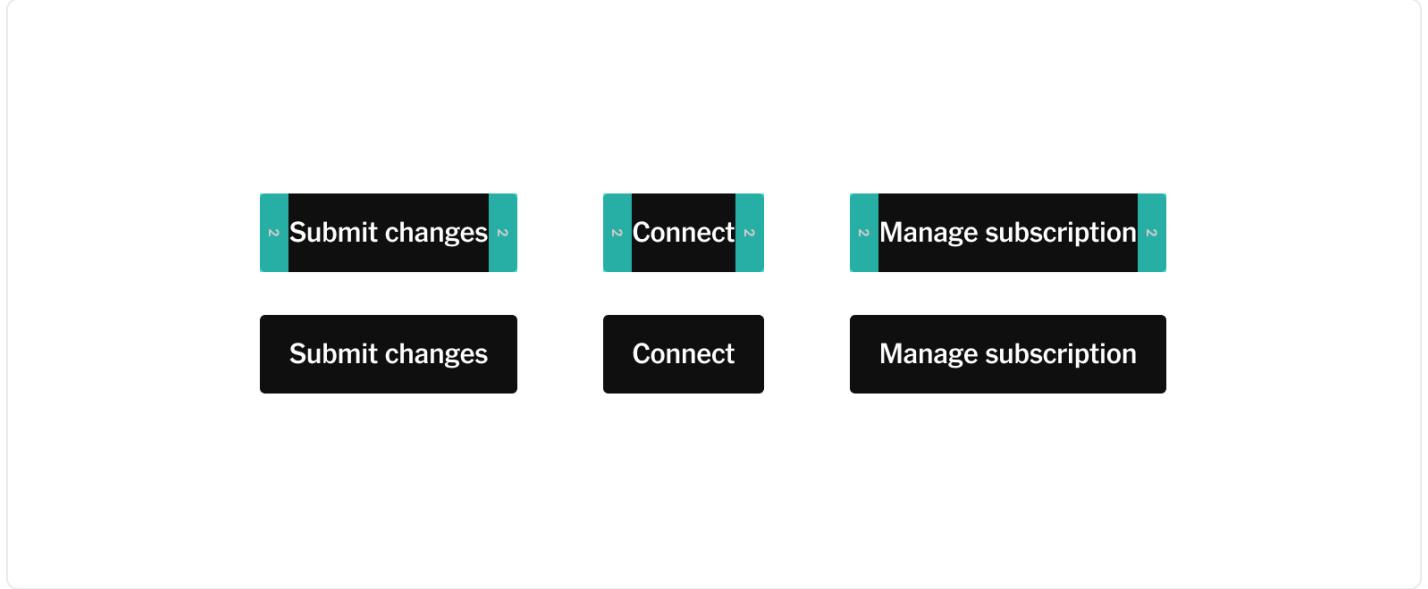


The compact button height is smaller than the tap target size but should provide enough external space to maintain accessibility. We recommend using this size within other components.



Width

The width of our default content-width button is determined by the length of content and the left and right padding.



Maximum-width buttons can be used when center-aligning buttons within a layout. We recommend setting the maximum width as a percentage to allow the button to respond to its container size. Flexible-width buttons can be used when you would like the button width to take up the entire width of the screen or container. This is ideal for layout for smaller screen sizes, to ensure an ample tap target size for users.

Most of the time

Tell us what interests you.
Make sure you see stories on the topics you care about.

[Get started](#)

evanmaeda@gmail.com

Password

Remember me [Forgot your password?](#)

[Log in](#)

Max-width: adheres to the button constraints

Flex-width: adheres to the layout constraints

Layout

Alignment

Our default alignment of Buttons on the web is to align with content. If you are moving a user through a flow, you can also align the button to the right to reinforce progressive direction.

State	ZIP code
<input type="text"/>	<input type="text"/>
Address change date	Phone
The earliest date available is shown.	Used to verify your account.
<input type="text" value="07/20/2020"/>	<input type="text"/>
<div style="text-align: right;">Cancel Next</div>	

Reinforcing progressive direction in a flow

We only recommend center-aligning buttons when content is also center-aligned. Content should only be center-aligned on brief content, as it can be difficult to read when longer.

I know that you're a subscriber, make
the most of The Times.

Tell us what interests you.
We'll make sure you see stories on the topics you care about.

Get started

Center-aligned with content

Horizontal stacking

If the button set is aligned with left-aligned content, horizontally stack the primary button to the left of the secondary button. If the button set is aligned to the right, horizontally stack the primary button to the right of the secondary button to reinforce the progressive direction.

Please check that our system retrieved the correct address.

100 MARTIN LUTHER KING DR.
BROOKLYN, 11207 NY
123-456-2792

[Edit address](#)

Confirm changes

Discard changes

Left aligned button set

Phone

Used to verify your account.

Cancel

Next

Right-aligned button set

Vertical stacking

We stack primary buttons on top of our secondary buttons. We do this because it makes sure that the primary button remains visible if the page scrolls.

BROOKLYN, 11207 NY
123-456-2792

[Edit address](#)

Confirm changes

Discard changes

Left aligned button set at a smaller breakpoint

Phone

Used to verify your account.

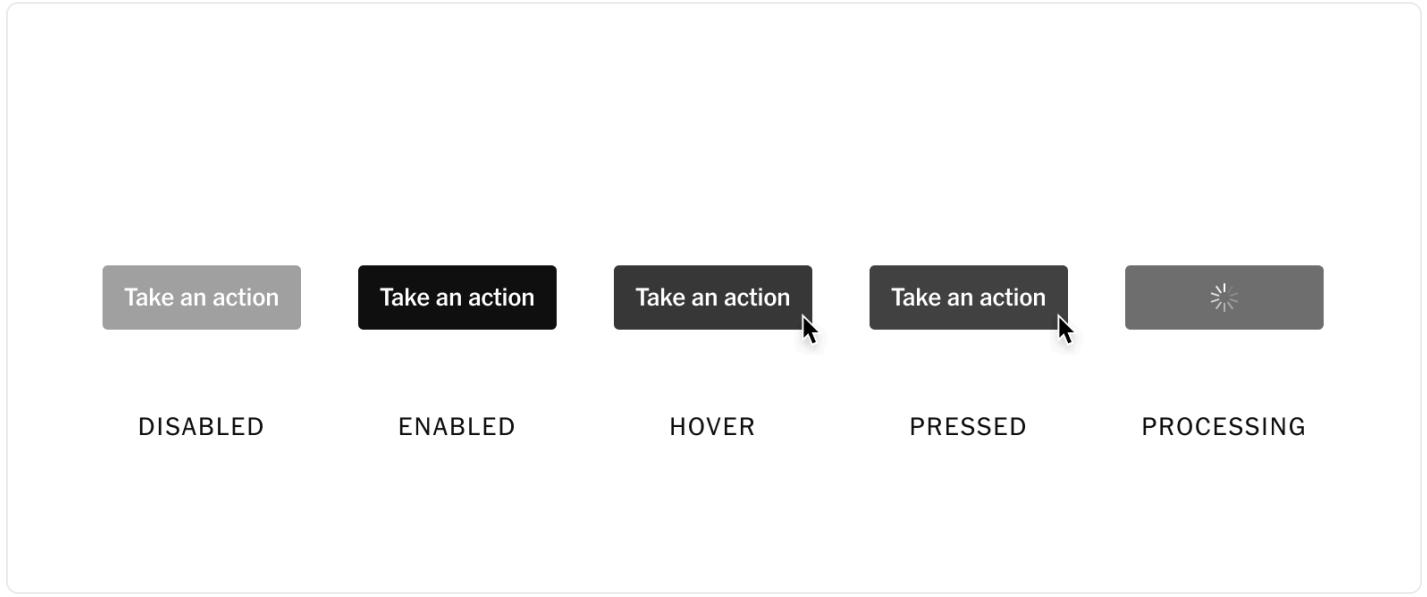
Next

Cancel

Right-aligned button set at a smaller breakpoint

States

Button follows the guidance outlined in the [States](#) token.



Implementation

Design

Button components are available for designers in our [Figma library](#).

A screenshot of the Figma interface showing the LIBRARY, ASSET PANEL, and VARIANTS panels for a button component.

- LIBRARY:** Shows a sidebar with "Cover" and "Libraries" sections. Libraries listed include "Hyphen Component Library" (selected), "Ink Component Library Dark", "Material Design 2.0 Theme", and "Material Design Theme Kit".
- ASSET PANEL:** Shows the "Search" bar and a list of components under "Local components" and "Used in this file". It also shows a list of variants for the "Button" component, each with a "Take an action" label.
- VARIANTS:** A detailed panel for a selected variant. It shows the button's width (131x44), orientation (0°), and state (Enabled). It includes settings for "Clip content", "Filled" style, "Content" width, "Standard" height, and an "Icon" (a small blue circle). The "State" dropdown is set to "Enabled". Other tabs in this panel include "Auto Layout" and "Resizing".

Engineering

Button components are available as reusable code in the Hyphen codebase. All tokens and components are available to preview on [Storybook](#).

Breakpoints + Grid

Grids and breakpoints work together to provide a foundational structure for the layout of our experiences. Combined with Spacing, they enable consistent and intentional interfaces for our users.

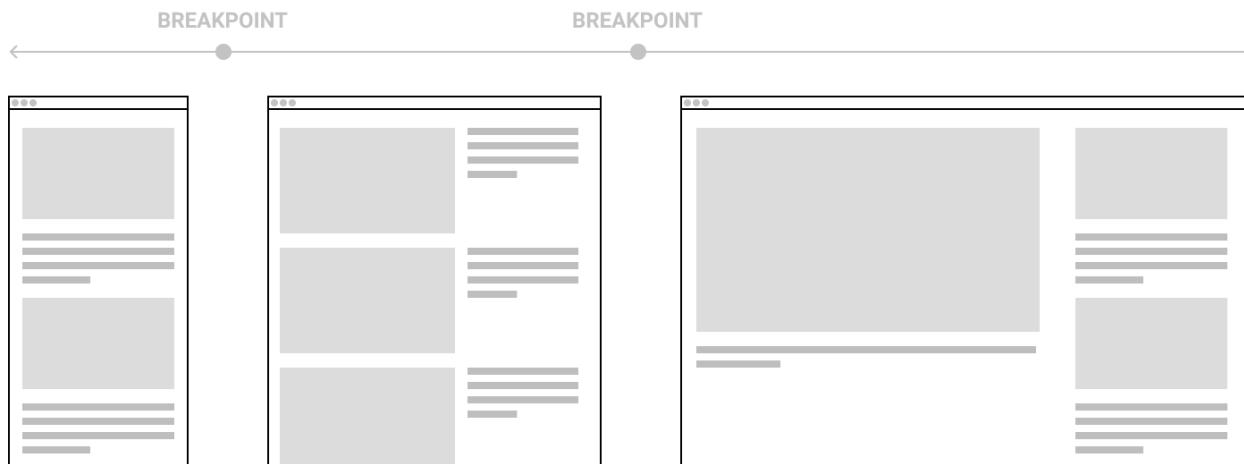


We are currently maintaining but no longer updating the Hyphen Design System.

To migrate off the Hyphen system, [we're here to help](#). Hyphen will be decommissioned in approximately one year, in July 2024.

Breakpoints

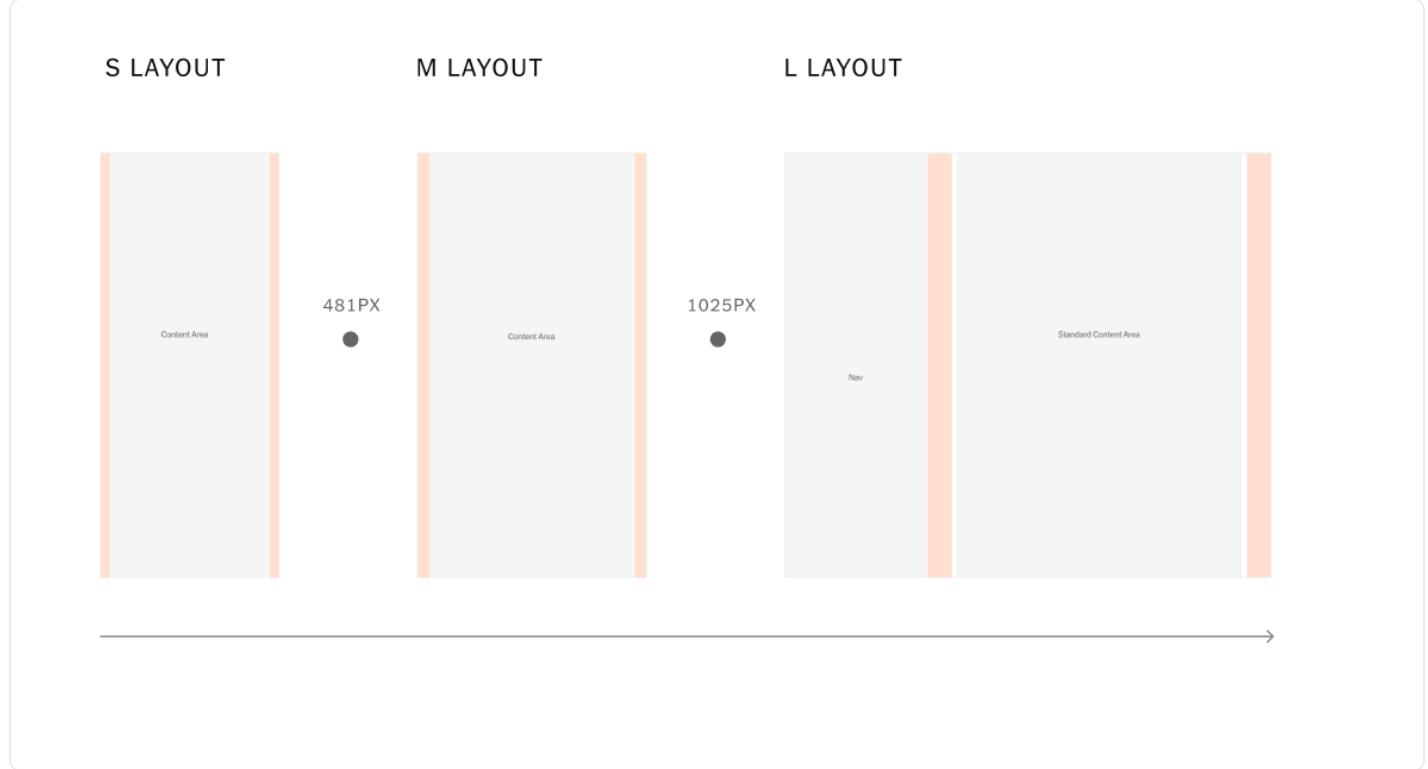
A breakpoint is a specific screen size where layout drastically changes to accommodate the usability needs of the smaller or larger screen size. Between breakpoints, layout remains the same, but the elements adjust as the screen size scales.



Layouts

We offer three layouts, Small, Medium, and Large which are defined by two breakpoints. Designs that serve a user well on a large format device may not be appropriate for a small mobile device. Use these layouts to optimize design for the size of screen you are working with.

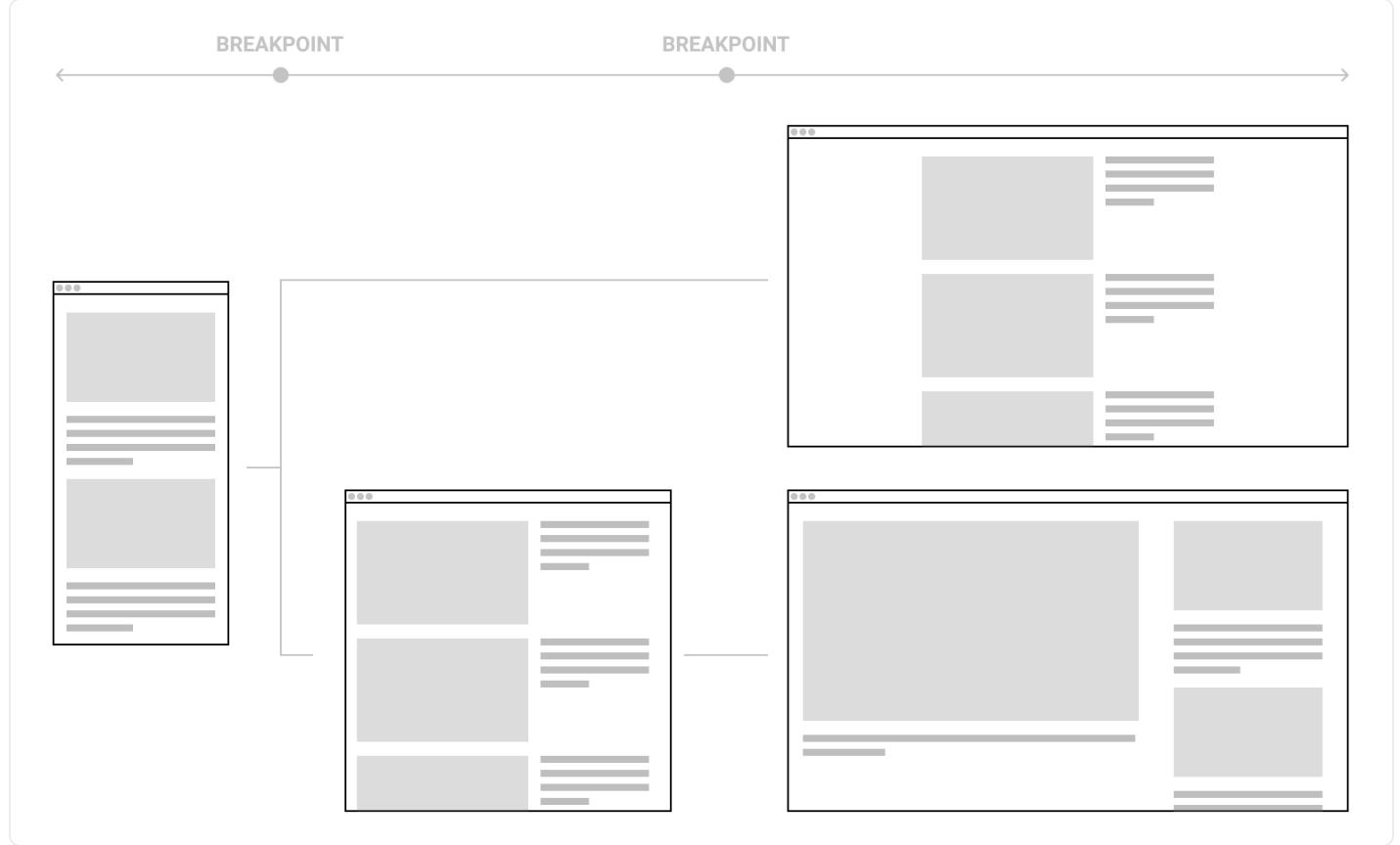
- **481px:** This breakpoint defines the transition from Small to Medium layouts.
- **1025px:** This breakpoint defines the transition from Medium to Large layouts.



Not every experience has a layout that requires change at different screen sizes. Consider breakpoints as optional, and should be used when necessary to ensure that usability and quality needs are met.

Applying breakpoints

Deciding how many breakpoints to apply to an experience is determined experience by experience. Due to the drastic change in size between desktop and mobile devices, we recommend using at least one breakpoint and designing for two layouts.



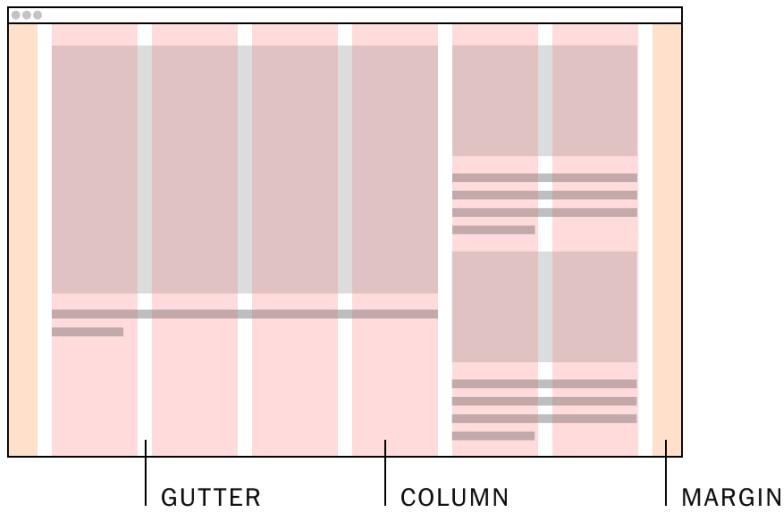
For example, a layout with side navigation doesn't transition easily into smaller screen sizes. We utilize a second breakpoint to accommodate these needs.

Grid

A grid is an underlying structure that guides the top-level spacing on a page. It's used to build consistency in your layout. The grid should align large sections of UI to maintain hierarchy and flow.

Grid breakdown

Grids are made up of margins, gutters, and columns. Some of these elements are responsive and determined at each breakpoint, a specific screen size where layout changes to accommodate the usability needs of the smaller or larger screen size.



Margins

A page margin is the minimum amount of space to the left and right of the page contents. Appropriate margins create room for the eye to focus on and scan content. Our grid uses a page-margin value of 5% across our S, M, and L layouts.

WITH 5% MARGIN

Curabitur eleifend nisi sagittis, mollis est sit amet, consectetur magna. Donec metus felis, dictum nec feugiat ut, consectetur at neque. Etiam vehicula auctor quam. Quisque lobortis dui at placerat lacinia. Nullam at nulla non

Curabitur eleifend nisi sagittis, mollis est sit amet, consectetur magna. Donec metus felis, dictum nec feugiat ut, consectetur at neque. Etiam vehicula auctor quam. Quisque lobortis dui at placerat lacinia. Nullam at nulla non

Curabitur eleifend nisi sagittis, mollis est sit amet, consectetur magna. Donec metus felis, dictum nec feugiat ut, consectetur at neque. Etiam vehicula

NO MARGIN

Curabitur eleifend nisi sagittis, mollis est sit amet, consectetur magna. Donec metus felis, dictum nec feugiat ut, consectetur at neque. Etiam vehicula auctor quam. Quisque lobortis dui at placerat lacinia. Nullam at nulla non

Curabitur eleifend nisi sagittis, mollis est sit amet, consectetur magna. Donec metus felis, dictum nec feugiat ut, consectetur at neque. Etiam vehicula auctor quam. Quisque lobortis dui at placerat lacinia. Nullam at nulla non

Curabitur eleifend nisi sagittis, mollis est sit amet, consectetur magna. Donec metus felis, dictum nec feugiat ut, consectetur at neque. Etiam vehicula

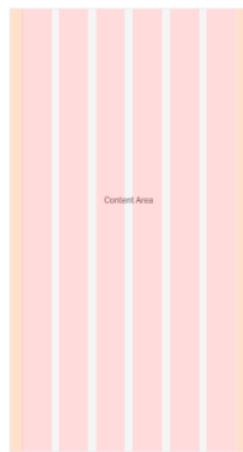
Columns & gutters

Our L & M layouts have 6 fluid columns with 16px gutters. These columns are used for guidance when arranging large content containers, though all containers are not required to align to a column. Fluid columns have no fixed width but stretch to fill the maximum space available to them.

S LAYOUT



M LAYOUT



L LAYOUT



Width

When designing responsively, we start by defining the maximum and minimum width of your experience. By defining these you control the overall limits for the entire experience.

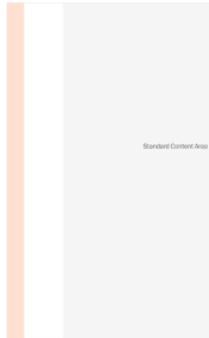
Max width

Max-width grids have a maximum width. When the content area of our grid reaches its max-width it will stop growing. We recommend most experiences using this grid, to ensure the design keeps its intended structure at larger screen sizes.

COMPACT



STANDARD



EXPANDED



We offer three different max-width values but recommend the standard max-width to ensure consistency across experiences.

- **Standard:** 600px (used for default experiences)
- **Expanded:** 700px (used for galleries or multi-column layouts)
- **Compact:** 450px (used for forms)

Min width

Our grid does not have a min-width, instead, when the content area of our container is larger than the space between page margins, the content area shrinks to fit.

Contribution

We have structured our grid and breakpoint system to enable evolution and contribution to meet current and future product needs.

Breakpoints

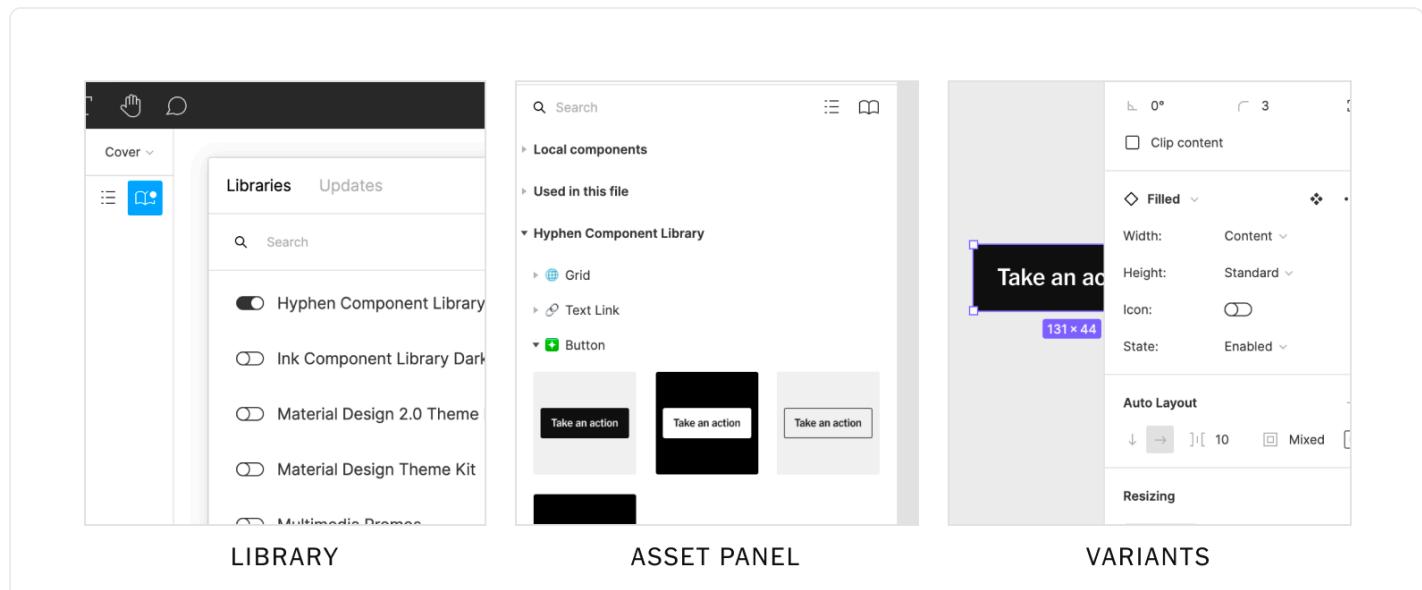
Our breakpoint and grid system was built with the needs of most utilitarian page and form experiences within the Times. For more complex experiences, consider utilizing the [guidance](#) used by Home, Collection, and Story products.

We hope to evolve our guidance to meet current and future product needs, reach out if you would like to add an additional layout or max-width to the system.

Implementation

Design

Grid components are available for designers in our [Figma library](#).



Engineering

Grid and breakpoints are available as reusable code in the Hyphen codebase. All tokens and components are available to preview on [Storybook](#).

Color

Color is used to provide differentiation and distinction between UI elements. Our system prioritizes accessibility for our users.

- ⚠ We are currently maintaining but no longer updating the Hyphen Design System. We recommend using [TPL Color](#) for any new projects.
- To migrate off the Hyphen system, [we're here to help](#). Hyphen will be decommissioned in approximately one year, in July 2024.

Accessibility

Contrast between foreground and background is our main criterion for accessible colors. **Text**, **user interface components**, and **graphical objects** each have their own contrast requirements.

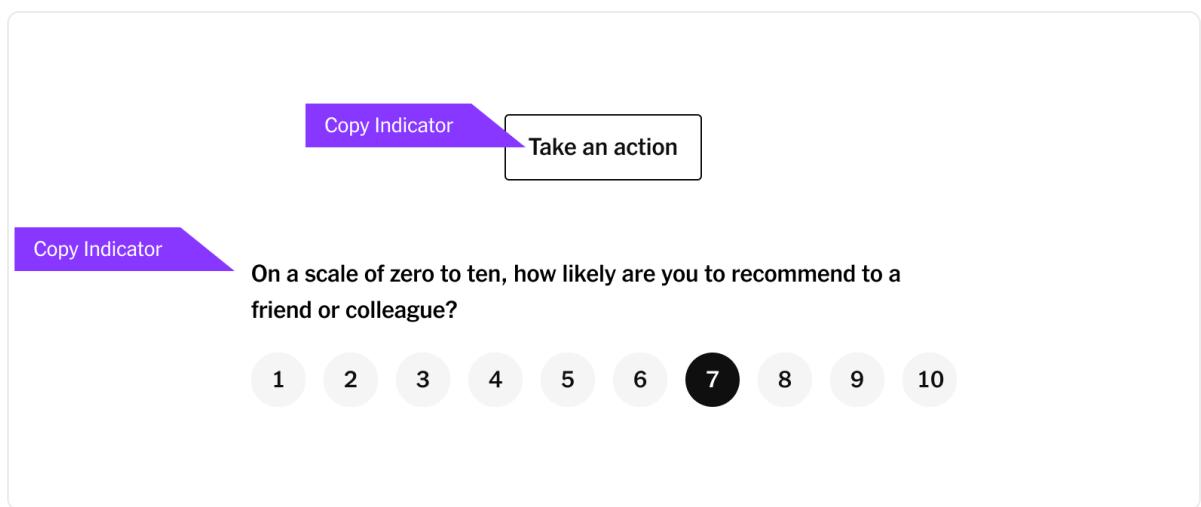
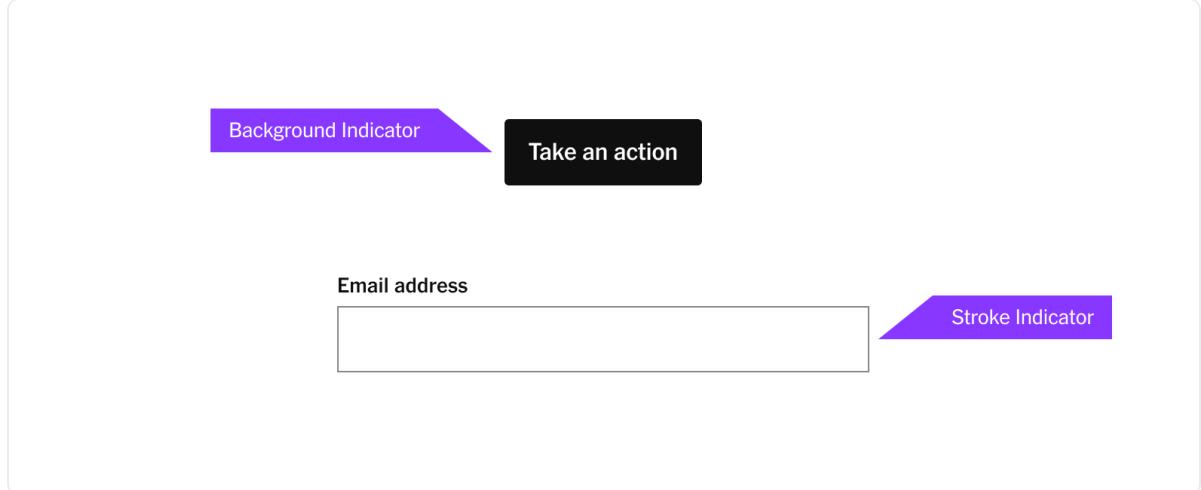
Text

Text must have a contrast ratio of at least 4.5:1 (with rare exceptions). Text that is larger than 18px or 14px when bold must meet a reduced contrast requirement of at least 3:1.

User interface components

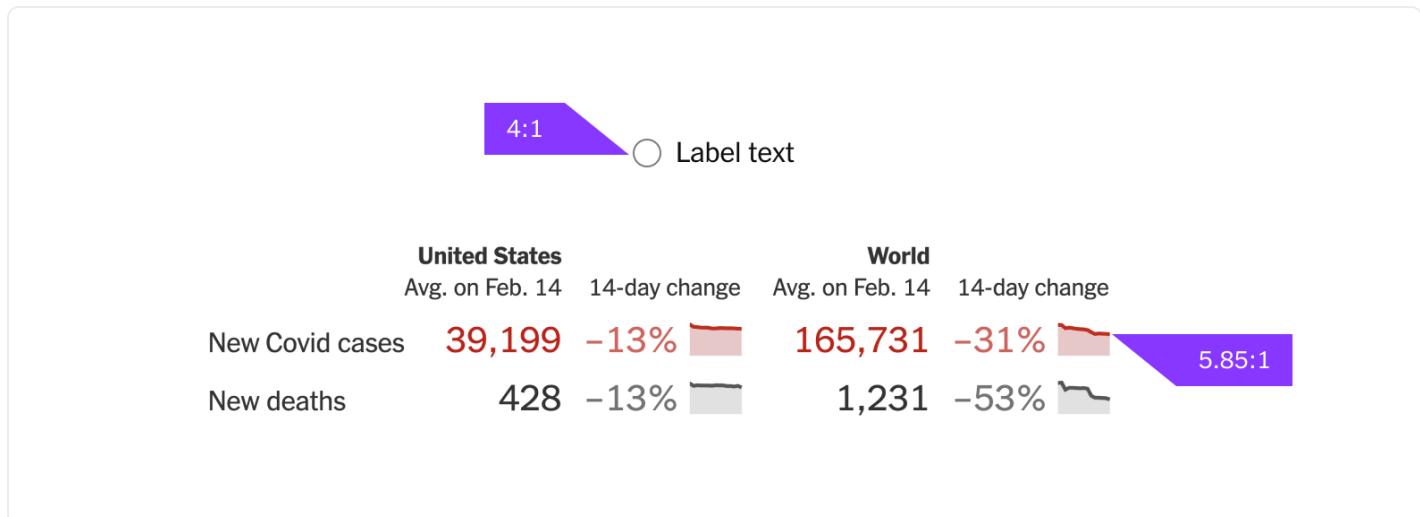
Color contrast helps signify the interactivity and state of user interface components. Generally, we aim for 3:1 contrast for a component's background and/or stroke. Limited exceptions are made for:

1. Components that contain text or icons with 4.5:1 contrast (e.g. buttons)
2. Components that are in a disabled state



Graphical objects

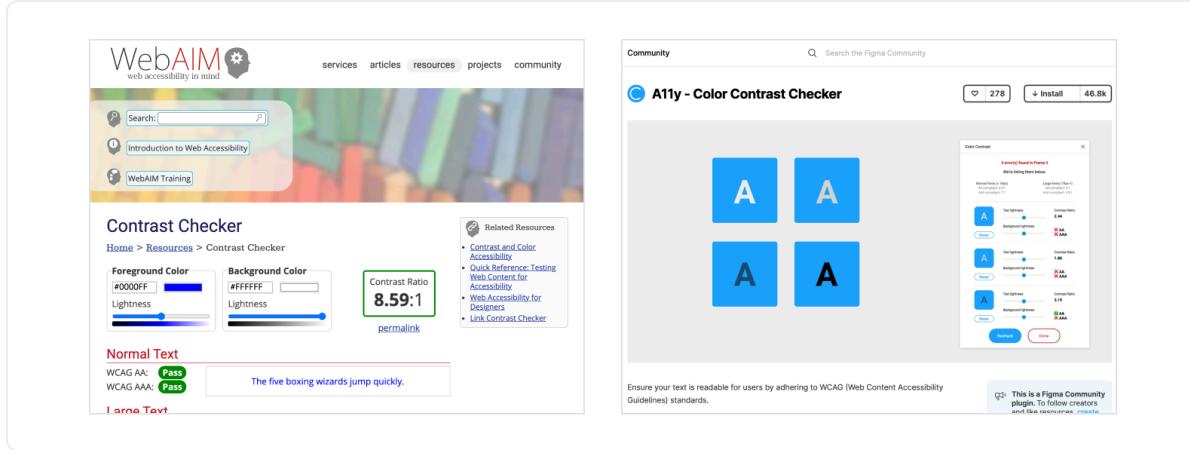
The visual presentation of user interface components and graphical objects must have a contrast ratio of at least 3:1.



Testing color contrast

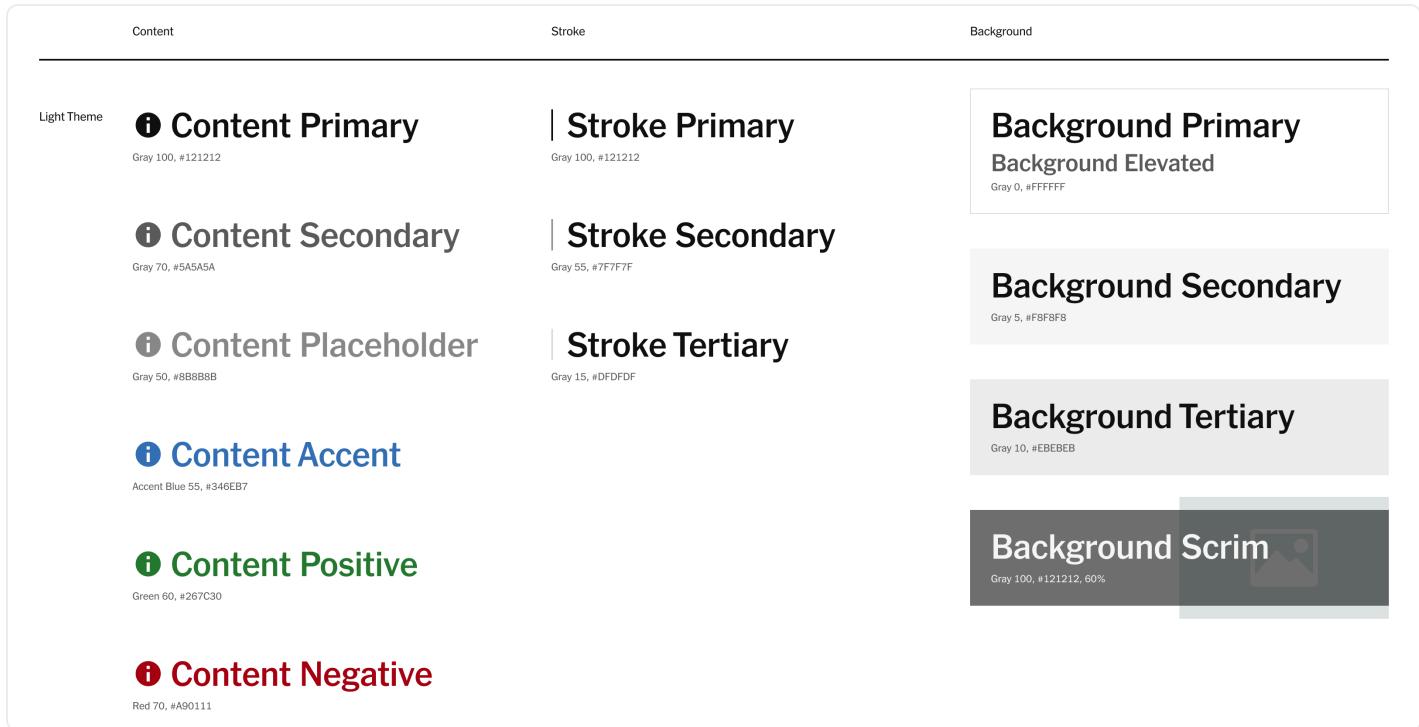
Although our system provides colors that pass contrast requirements, there may be exceptions. Ensure you are using accessible colors using the many resources available online that can be used to ensure that they are being used correctly.

- [WebAIM Contrast Checker](#): A website where you can input specific foreground and background color values to check contrast ratio compliance.
- [A11y - Color Contrast Checker](#): A Figma plug-in that can test colors within your files.
- Storybook's [A11y addon](#) will flag any color contrast violations as well as any other markup issues it can detect.



Palettes

Palettes are used to align common color usages across interfaces. We have rigorously tested our palette colors to ensure accessibility is met.



Content	Stroke	Background
Content Primary Light Theme Gray 100, #121212 Content Secondary Gray 70, #5A5A5A Content Placeholder Gray 50, #888888 Content Accent Accent Blue 55, #346EB7 Content Positive Green 60, #267C30 Content Negative Red 70, #A90111	Stroke Primary Gray 100, #121212 Stroke Secondary Gray 55, #7F7F7F Stroke Tertiary Gray 15, #DFDFDF	Background Primary Background Elevated Gray 0, #FFFFFF Background Secondary Gray 5, #FBFBFB Background Tertiary Gray 10, #E8E8E8 Background Scrim Gray 100, #121212, 60%

Content

The content palette is used when applying color to **text** and **icons**. To ensure accessibility, use these colors with our

background palette colors.

❶ Content Primary

Gray 100, #121212

❶ Content Accent

Accent Blue 55, #346EB7

❶ Content Secondary

Gray 70, #5A5A5A

❶ Content Positive

Green 60, #267C30

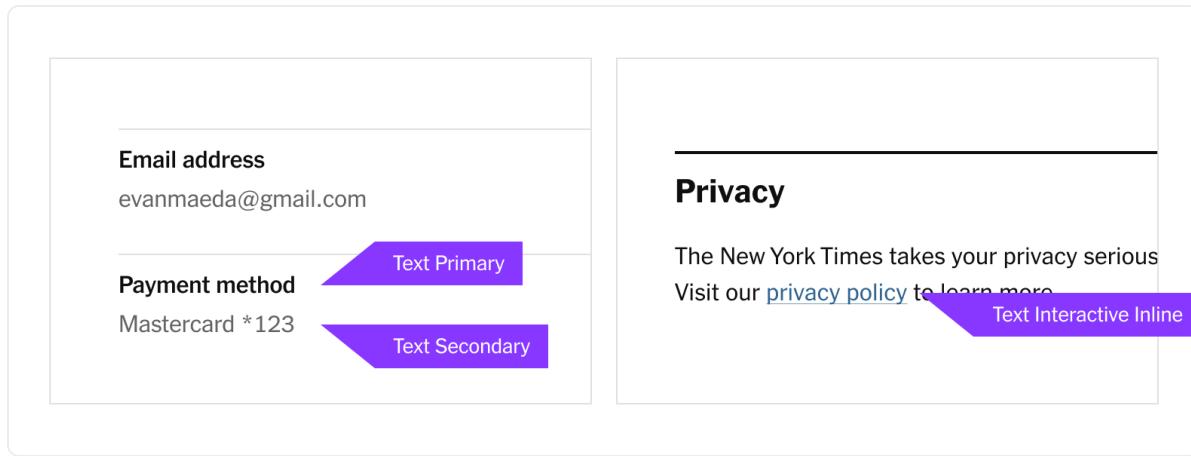
❶ Content Placeholder

Gray 50, #8B8B8B

❶ Content Negative

Red 70, #A90111

Content Primary is our default text style and should be used with Content Secondary or Tertiary to reinforce hierarchy. Text Interactive inline should only be used for inline text links.



Stroke

The stroke palette is used when applying color to strokes.

| Stroke Primary

Gray 100, #121212

| Stroke Secondary

Gray 55, #7F7F7F

Stroke Tertiary

Gray 15, #DFDFDF

Stroke Primary and Stroke Secondary can be used as indicators of an element's interactivity because they meet accessibility requirements. Stroke Tertiary is used for layout purposes only, and should not be used as an indicator of an element's interactivity.

Background

The background palette is used when applying color to the background of a page or a component.

Background Primary

Gray 0, #FFFFFF

Background Tertiary

Gray 10, #E8E8E8

Background Secondary

Gray 5, #F8F8F8

Background Scrim

Gray 100, #121212, 60%

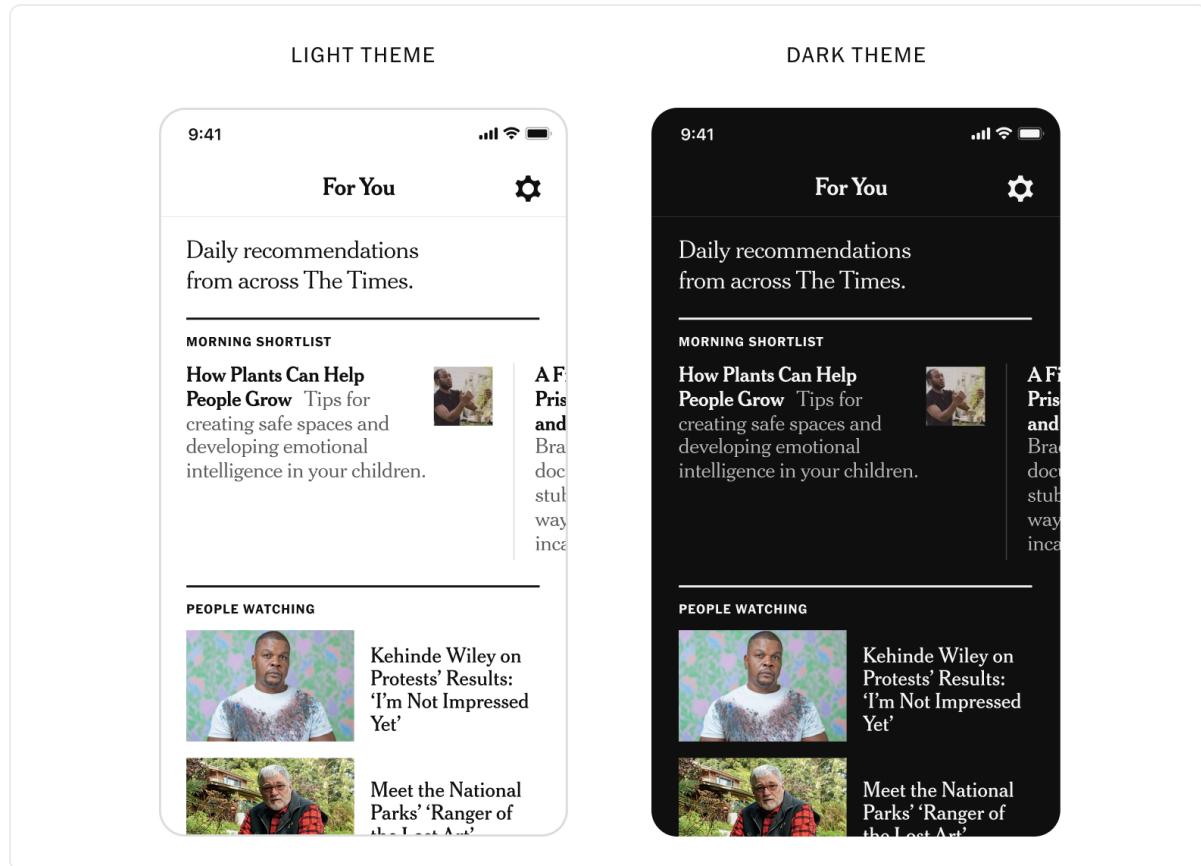
Background Primary is our default UI background color. Background Primary and Secondary should only be used as

supplementary indicators of an element's interactivity, as they do not meet accessibility standards.

Themes

Themes are what allow you to design and build for dark mode.

- **Light theme:** provides **dark content** colors and **light backgrounds**
- **Dark theme:** provides **light content** colors and **dark backgrounds**



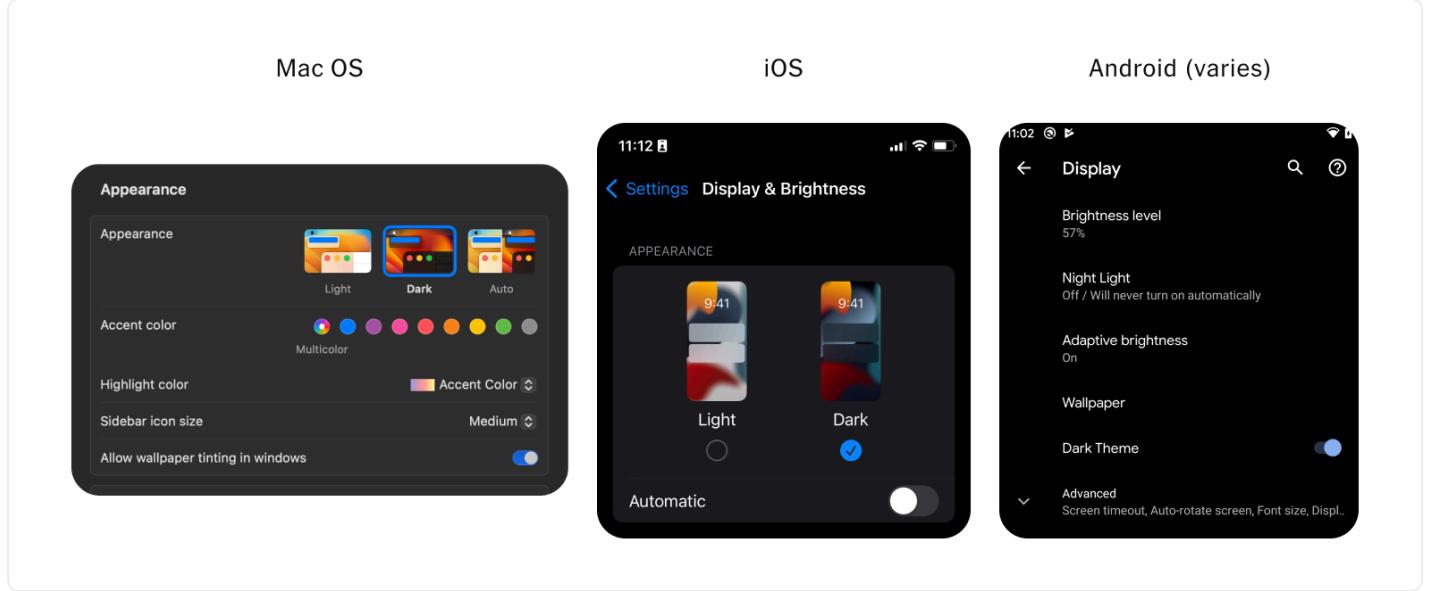
There are two color values associated with every semantic color, one for each theme.

Device and user setting

Our system references device and app settings to determine whether an experience or a container within your experience will appear in the Light or Dark theme.

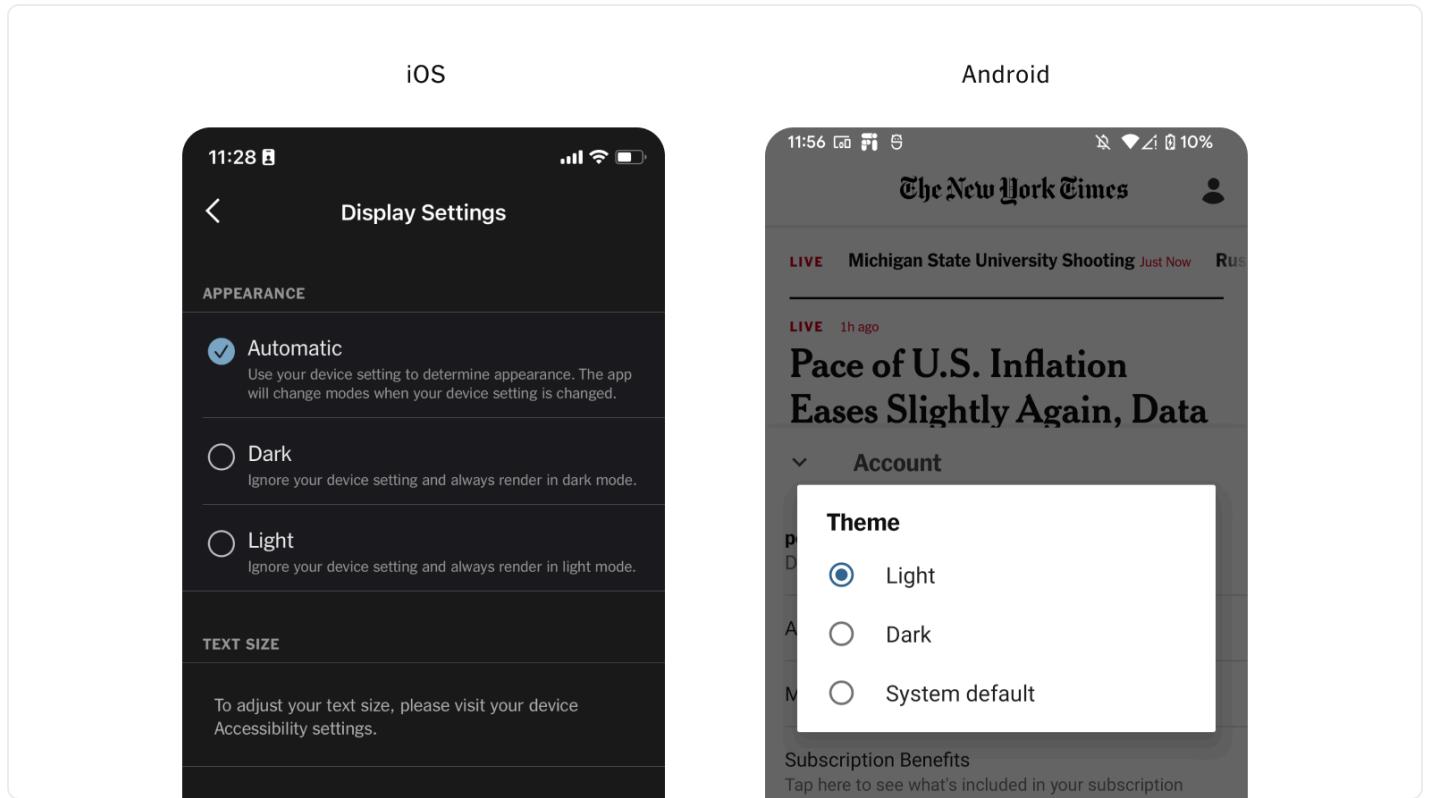
Device setting

Most mobile and desktop devices allow you to set a preference for Light or Dark mode. In the NYTimes iOS and Android apps, our colors respond to a user's device setting by default to determine their theme.



User setting

Our apps also allow the user to set a unique preference in the NYTimes app, allowing users to override their device setting and set a preference for Light or Dark theme manually.



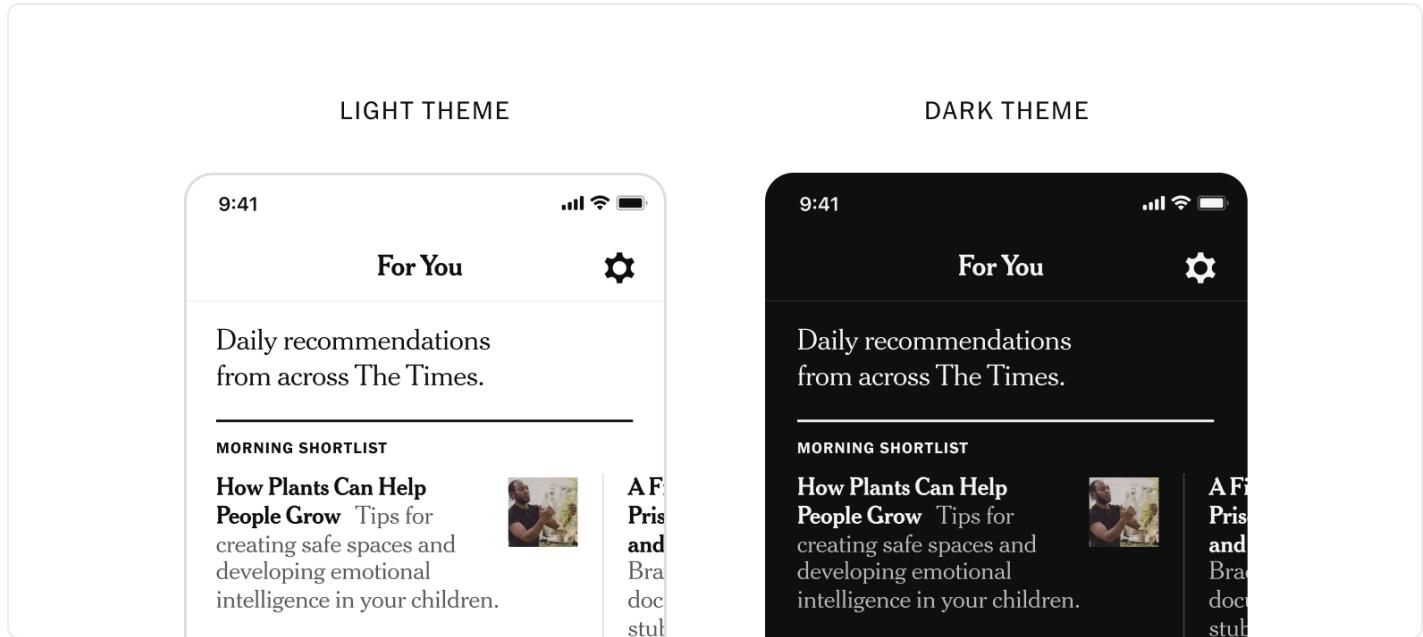
Nytimes.com (Web) does not currently enable dark mode. If you are curious about dark mode for web, please reach out to us on [Slack](#)!

Theme properties

You can control these themes using four properties: Default, Inverse, Always Dark, and Always Light.

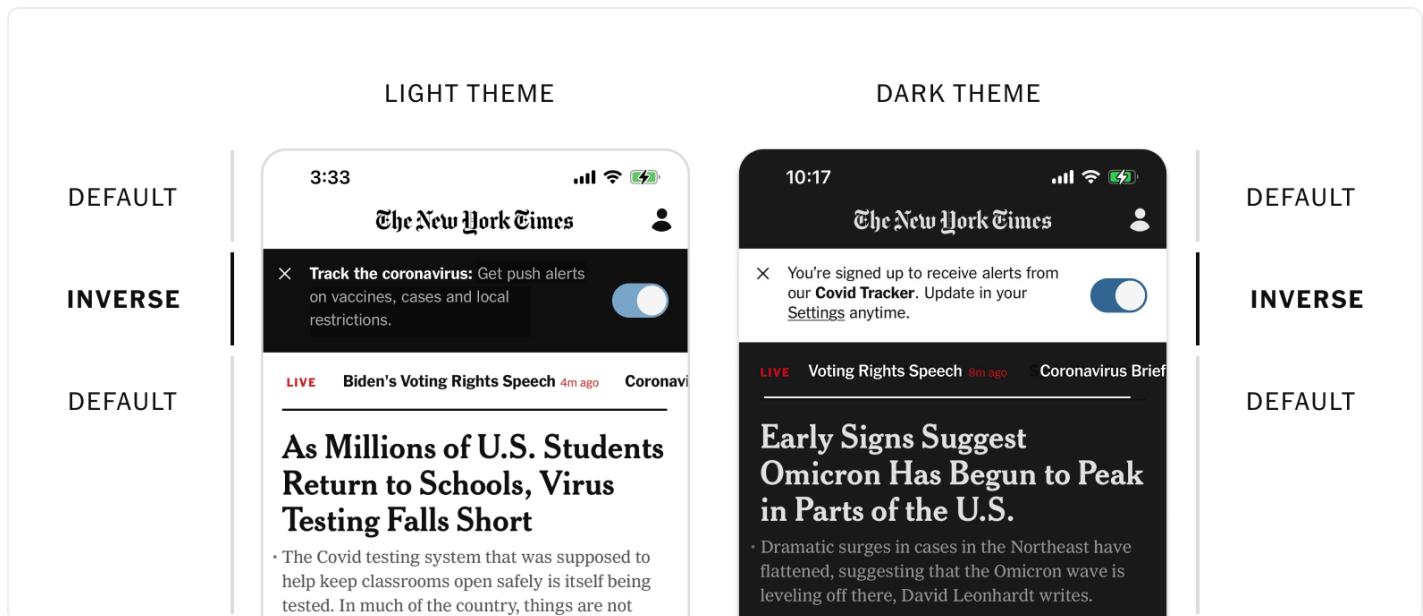
Default

This property respects the user's Device or App setting. Content with this property uses the Light theme in Light Mode and the Dark theme in Dark Mode.



Inverse

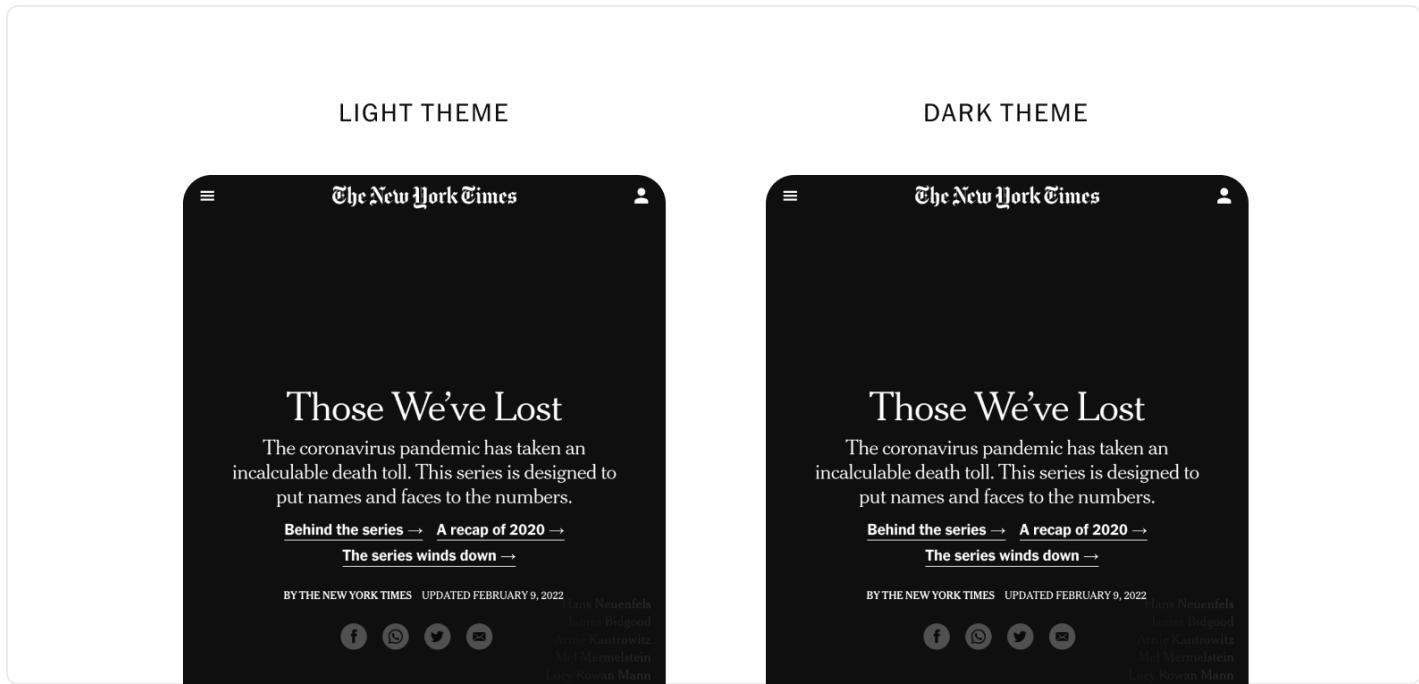
This property inverts the user's Device or App setting. Content with this property uses the Dark theme in Light Mode and the Light theme in Dark Mode.



This should be used for elements that are intentionally offset from the content around it like a Toast or Primary Button. We recommend using it on specific, contained elements of an experience to best uphold the intended offsetting effect.

Always Dark

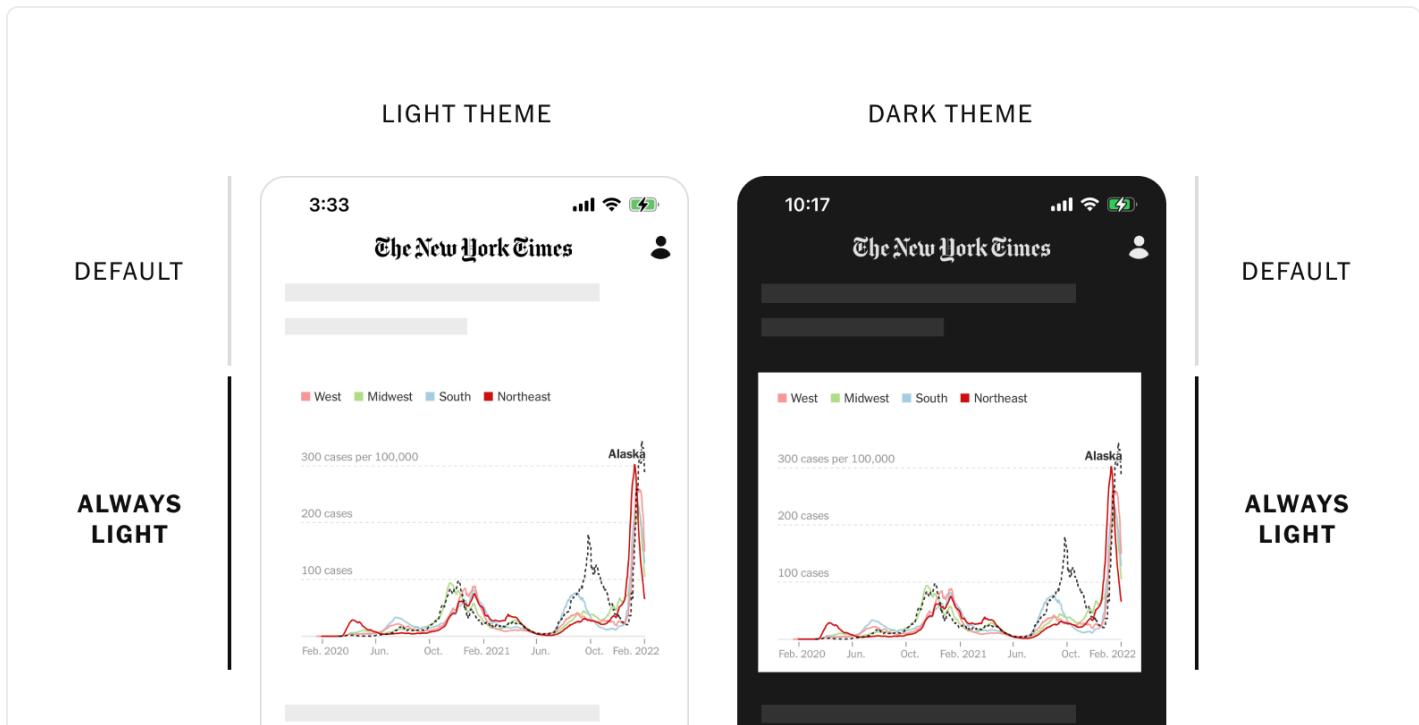
This property ignores the user's Device or App setting. Content with this property uses the Dark theme in both Light and Dark modes.



This property often creates a dramatic or premium effect that would otherwise be lost with the `Invert` property. Use this property with caution; always consider how a user might react to a design decision that does not meet their expectations.

Always Light

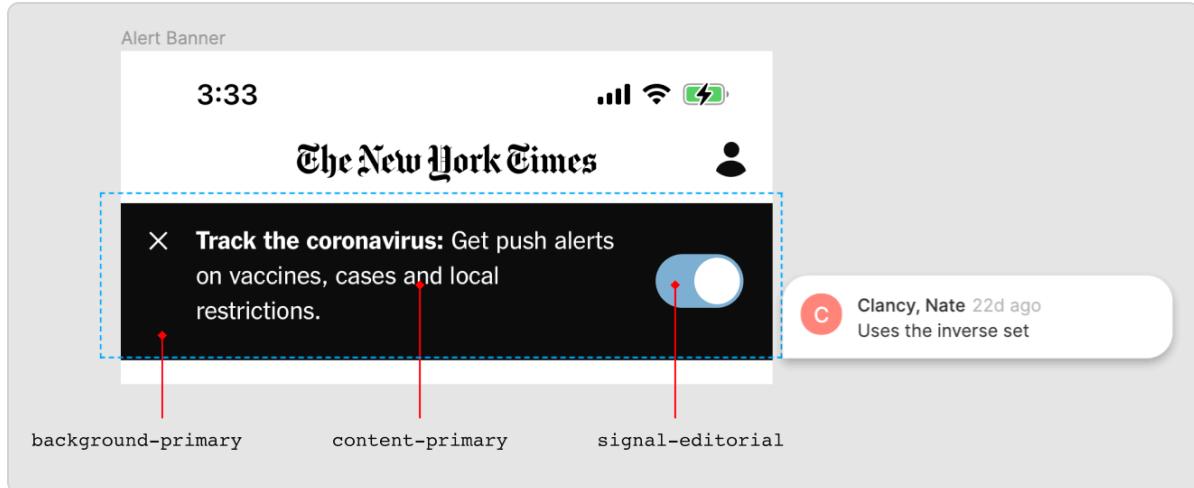
This property ignores the user's Device or App setting. Content with this property uses the Light theme in both Light and Dark modes.



⚠ This property should be used with caution. This is used primarily to ensure elements using custom colors do not break in dark mode. Consider how a user might react to an experience that does not meet their expectation.

Handoff

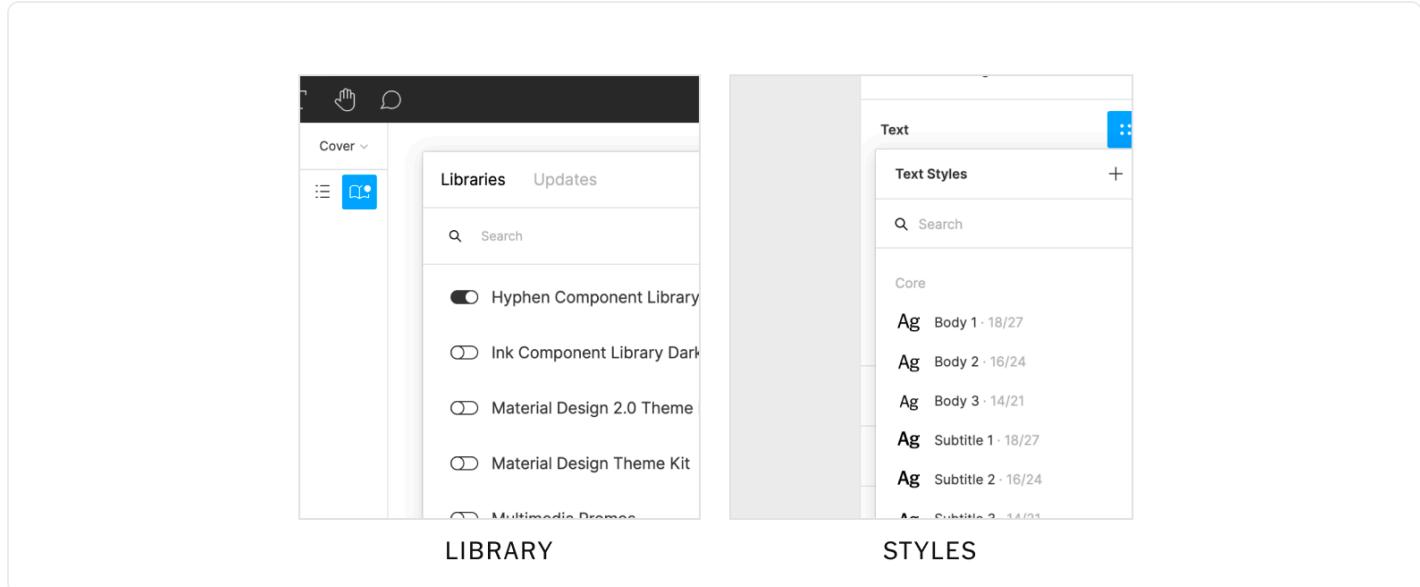
We recommend indicating what elements of your experience are using any property other than the default. This will help your Engineering counterpart to understand what color values to use. You can learn more about the implementation guidelines [here](#).



Implementation

Design

Our color tokens are available in our [Figma library](#).



Engineering

Colors are available as reusable code in the Hyphen codebase and support CSS, CSS-in-JS, and iOS (Android coming soon). More details can be found in our [Storybook](#).

Icon

Iconography uses simplified graphics to communicate content or actions. They should be used to add clarity, reinterpret copy, and make a more accessible product.

 We are currently maintaining but no longer updating the Hyphen Design System. We recommend using [TPL Icons](#) for any new projects.

To migrate off the Hyphen system, [we're here to help](#). Hyphen will be decommissioned in approximately one year, in July 2024.

Usage

We use iconography sparingly within our product experiences. When determining if you should use an icon for a particular use case, consider the following:

When to use

- **To create recall:** An icon that builds on recognition over time, removing the cognitive load to understand the icon/interaction in a single instance, and allowing the user to move more quickly through the experience.
- **To add clarity:** Icons leveraging universal meanings to create a more accessible product by visually reinterpreting text.
- **To set tone:** Icons are used to indicate a pass/fail state of a flow and remind the user to slow down and read their options carefully.

Select an option

Secure transaction

Email address

9252863087

This is not a valid email address

When to avoid

- **As a decorative element:** Icons should not be used as a purely visual element, to fill up space.

Type

We have different types of icons that have different properties.

UI

UI icons are the most common icon type and can be used in most use cases. They do not maintain a specific color and can utilize any color from our icon palette.

[+ Add a report](#)

Keep me signed in

Email address

9252863087

! This is not a valid email address

Social

Social icons are only to be used when referring to the specific brand product they are connected to. Their appearance is set by their specific brand guidelines and is not customizable.

You are connected to Facebook

Dagobah sith wedge binks hutt wedge organa
kamino. Darth skywalker darth darth fett jango.

 Sign in with Apple

Size

Our iconography system provides three distinct sizes: S, M, and L. These sizes were determined to work well with other UI elements such as text, as well as maintain their visual integrity.

SMALL	✓	<input checked="" type="checkbox"/> ↑ 16px
MEDIUM	✓	<input checked="" type="checkbox"/> ↑ 20px
LARGE	✓	<input checked="" type="checkbox"/> ↑ 24px

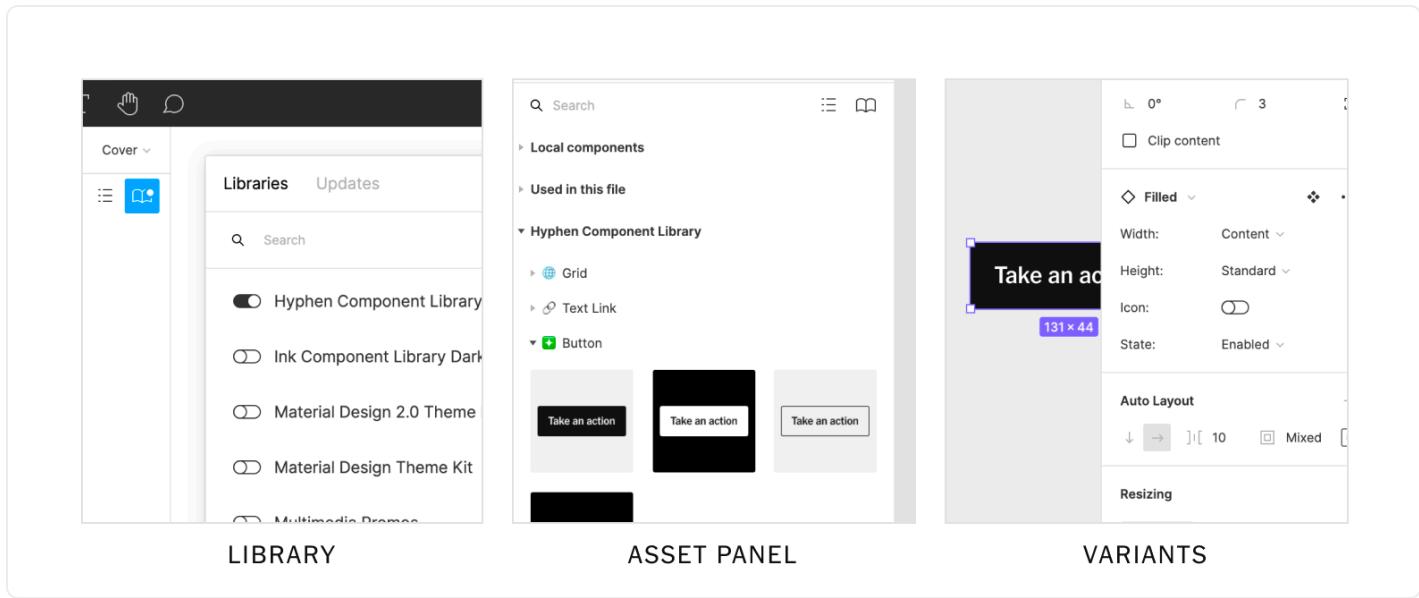
Contribution

In the event that our library doesn't have the icon you are looking for, you can create and contribute that icon for others to use in the future. This process has yet to be determined, reach out to us on slack for the next steps.

Implementation

Design

Icons are available for designers in our [Figma library](#).



Engineering

Icons are available as reusable code in the Hyphen codebase. All tokens and components are available to preview on [Storybook](#).

List Item

A list item presents information cleanly and efficiently as a single-column list of rows.



We are currently maintaining but no longer updating the Hyphen Design System.

To migrate off the Hyphen system, [we're here to help](#). Hyphen will be decommissioned in approximately one year, in July 2024.

Anatomy

List Items are composed of three areas: start, center, and end. Depending on the type of list items, start and end areas can be optional.

The diagram illustrates the anatomy of a List Item through four variations:

- Top Left Variation:** Shows a header icon followed by a "List item header" section. Below it is a horizontal bar divided into three colored segments: purple, light blue, and orange. The purple segment is labeled "START", the light blue segment is labeled "CENTER", and the orange segment is labeled "END". To the right of the bar is a white rectangular button labeled "Action".
- Top Right Variation:** Shows a header icon followed by a "List item header" section. Below it is a horizontal bar divided into two colored segments: light blue and orange. The light blue segment is labeled "CENTER" and the orange segment is labeled "END". To the right of the bar is a white rectangular button labeled "Action".
- Bottom Left Variation:** Shows a header icon followed by a "List item header" section. Below it is a horizontal bar divided into one large light blue "CENTER" segment.
- Bottom Right Variation:** Shows a header icon followed by a "List item header" section. Below it is a horizontal bar divided into two colored segments: light blue and orange. The light blue segment is labeled "CENTER" and the orange segment is labeled "END". To the right of the bar is a white rectangular button labeled "Action".

Start

The start area of the List Item is used for iconography, images, or selection controls. In a singular list, all items should contain the same start type in order to maintain consistency.

- Iconography and images can be used to increase visibility and recognition within a list, helping to move the user through it more quickly.
- Selection controls can be used to allow one or multiple items in the list to be selected.

ICON	IMAGE
<p>List item header This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.</p> <p>List item header This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.</p> <p>List item header This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.</p> <p>Connect</p>	<p>List item header This is the information</p> <p>List item header This is the information</p> <p>List item header This is the information</p> <p>Action</p>

Center

The center area of the List Item is used for text content. It can comprise of multiple hierarchical text styles based on your needs.

CONTRASTED	SUBTLE
<p>List item header This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.</p> <p>List item header This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.</p> <p>List item header This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.</p>	<p>List item header This is the information</p> <p>List item header This is the information</p> <p>List item header This is the information</p>

End

The end area of the List Item is used for actions. These actions can vary depending on the intent of the action.

- Buttons and text links can be used when the action is best communicated as a “call to action”.
- Selection controls can be used when the action is giving the user control over a particular experience.
- A Chevron icon can be used when the action is navigating a user to another experience.

NAVIGATION	CALL TO ACTION	CONTROL
<p>List item header</p> <p>This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.</p>	<p>List item header</p> <p>This is the information</p>	<p>List item header</p> <p>This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.</p>
<p>List item header</p> <p>This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.</p>	<p>List item header</p> <p>This is the information</p>	<p>List item header</p> <p>This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.</p>

Type

To ensure we maintain the right usage intention, we create different list item types to be used in different situations.

Action

The Action List Item is used to prompt the user with an action.

- It has a required end area action, the primary purpose of the list item. To maintain prominence, we use a Button component.
- In order to reinforce the action, we recommend using active-tense content for the header.
- It allows an optional start area.

 **Connect to Facebook**

Link your Apple account and use it to log in to The New York Times.
We will never send messages to your contacts without your permission.

 Connect

 **Connect to Google**

Link your Google account and use it to log in to The New York Times.
We will never send messages to your contacts without your permission.

 Connect

 **Connect to Apple**

Link your Apple account and use it to log in to The New York Times.
We will never send messages to your contacts without your permission.

 Connect

Navigation

The Navigation List Item is used to navigate users to another experience. It has a required end area and allows an optional start area.

- It has a required end area action, a chevron icon, to reinforce its progressive action.
- It allows an optional start area.

Email newsletters

We offer over X emails on a variety of topics. Find the newsletters that interest you, and remove any that don't. >

Interests

We use your interests to customize a daily digest for you. Update your selections to improve your email. >

Saved articles

Save articles to share with others or finish reading when the time is right. >

Information

The Information List Item is used to present information. This information can be generated by the user, or by The Times on behalf of the user.

- If the information can be changed or edited, include an end area action.
- In this case, the action is secondary to the information, differentiating itself from the Action List Item through text link usage and more subtle text hierarchy.
- It allows an optional start area.

ACCOUNT INFORMATION		NEWSLETTERS	
Account number	Y1234567	 Morning Briefing	Remove
Email address	dschrute@dundermifflin.com	 Morning Briefing	Remove
Password	*****	 Morning Briefing	Remove

Size

Width

List item width is determined by the width of the layout container it is in. All list items in a list should maintain the same width for consistency.

List item header This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines. List item header This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines. List item header This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines. List item header This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.	➤	List item header This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines. List item header This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines. List item header This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.	➤
--	---	---	---





Height

List item height is determined by its content. It maintains consistent spacing between the rule and the content, as well as consistent bottom padding.

List item header This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines. The length of this placeholder copy is meant to simulate when content extends to multiple lines. List item header This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines. List item header This is the body description.	➤	List item header First line of information Second line of information  List item header This is the information  List item header First line of information Second line of information Third line of information 	➤
---	---	---	---

Layout

Spacing

List items provide their own spacing through bottom padding. There should not be any spacing between list items.

Header

Providing a header for a list is essential to orient the user to its content.

- The Section component can serve as a list header. For example, Account experiences maintain consistent usage of Sections as headers for lists.
- We recommend using consistent header styles within an experience for quicker recognition.

SECTION COMPONENT

Your title

List item header

This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.

List item header

This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.

List item header

This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.

CUSTOM HEADER

Title header

List item header

This is the information

List item header

This is the information

Title header

List item header

This is the information

List item header

This is the information

Responsive

Depending on its content, a list item can reflow its contents to respond to different screen sizes.

LAYOUT M/L —————→ LAYOUT S

List item header

This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.

Connect

List item header

This is the body description. The length of this placeholder copy is meant to simulate when content extends to multiple lines.

Connect

List item header

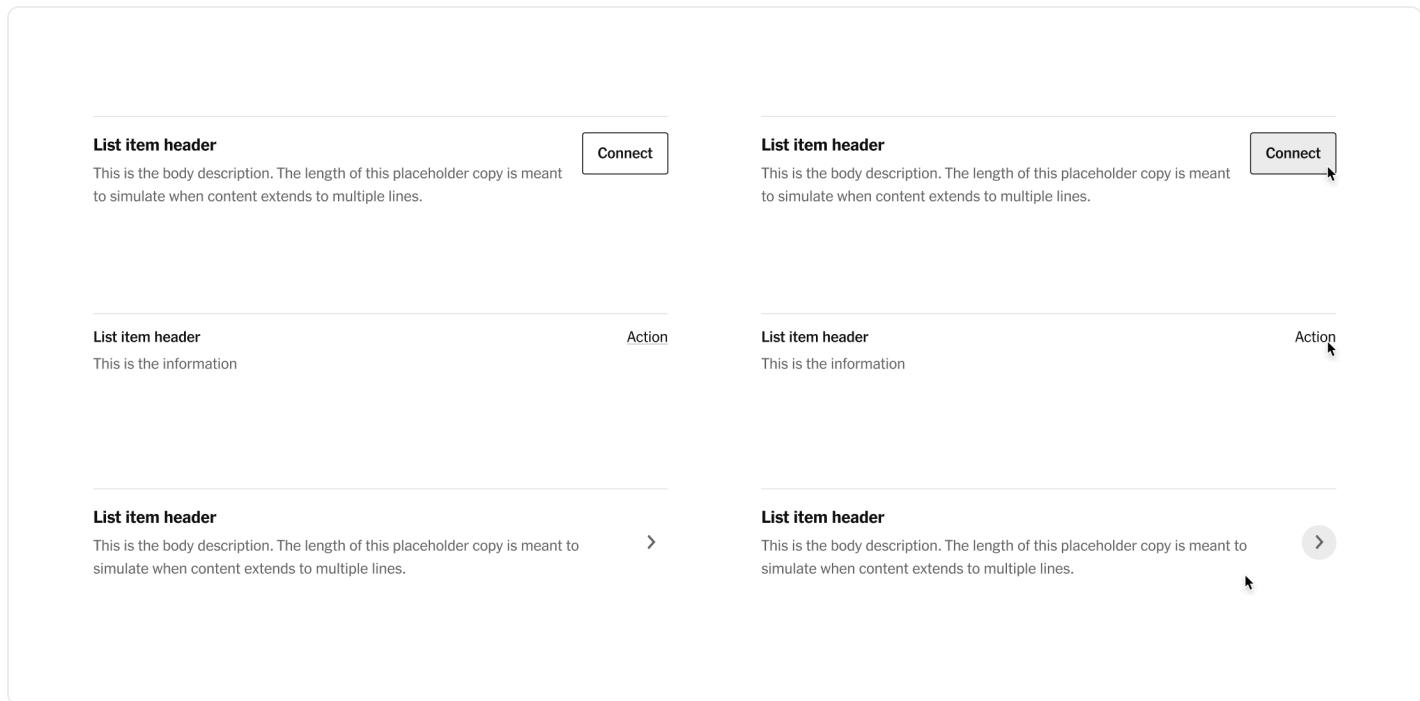
This is the information

List item header

This is the information

States

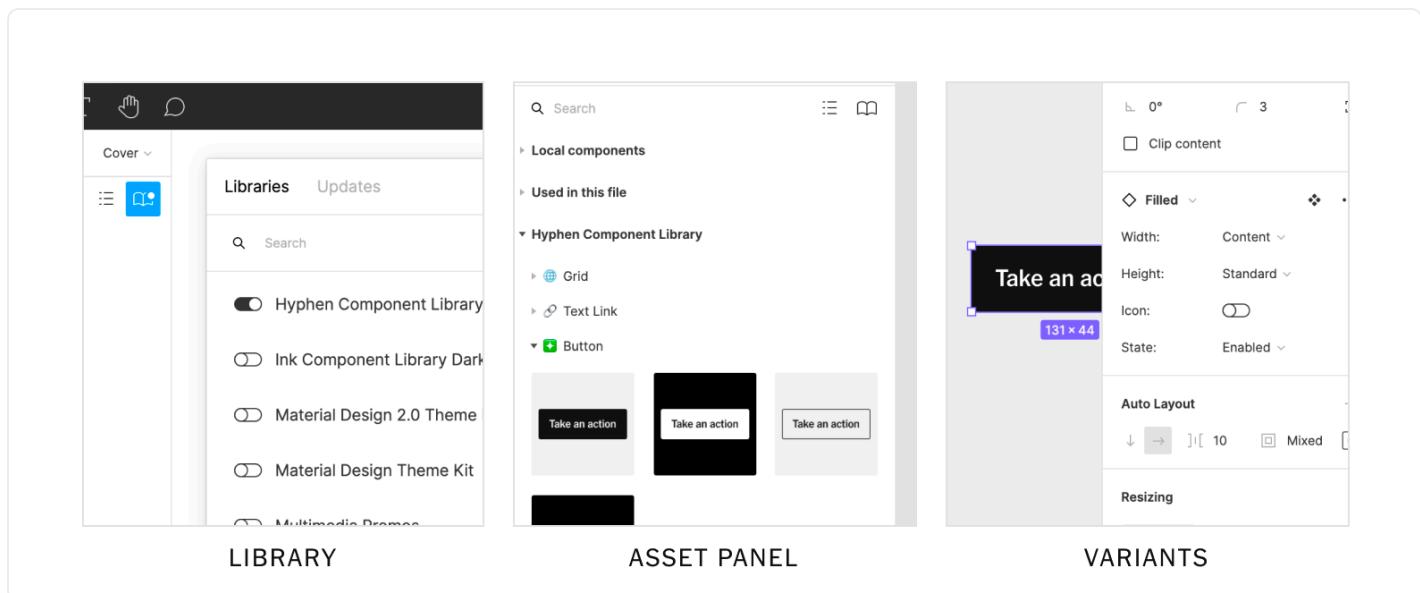
When list items have actions, they need to provide different states. Some list items contain action components such as Buttons that provide state feedback, while others are entirely interactive and must showcase different states depending on the action.



Implementation

Design

Link components are available for designers in our [Figma library](#).



Engineering

List item components are available as reusable code in the Hyphen codebase. All tokens and components are available to preview on [Storybook](#).

Radio

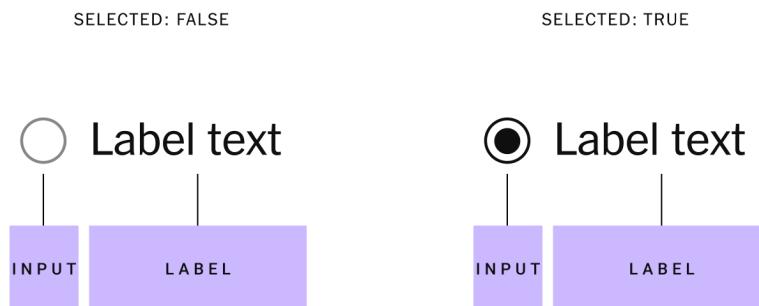
Radio buttons allow users to select a single option from a list of options. Selecting a different option will deselect any other option automatically.

⚠ We are currently maintaining but no longer updating the Hyphen Design System.

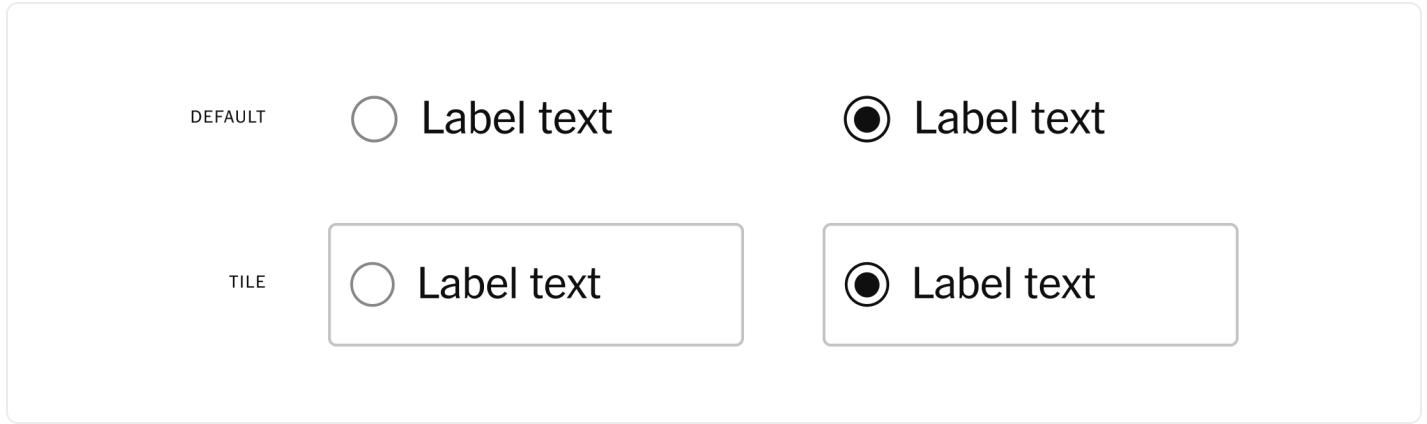
To migrate off the Hyphen system, [we're here to help](#). Hyphen will be decommissioned in approximately one year, in July 2024.

Anatomy

The radio button requires two key elements; the radio input and label. The **radio input** indicates whether the radio button is selected or deselected. The **label** is used to succinctly and clearly describe the option to the user.



Type



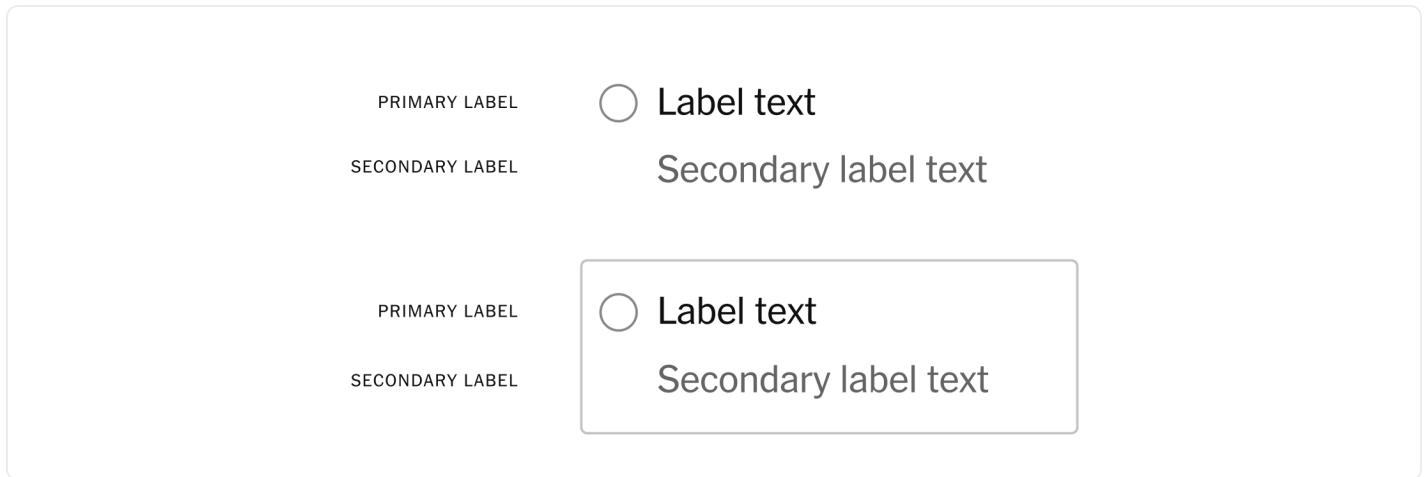
Default

The default radio button is the simplest visual option. It is most useful when there are many options to choose from. It reduces visual complexity in the experience.

Tile

The tile radio button adds a container around the radio button, making the clickable area more obvious to the user. It can help reinforce what information is included in each option when you are using [secondary labels](#) or [slot content](#).

Contents



Primary label

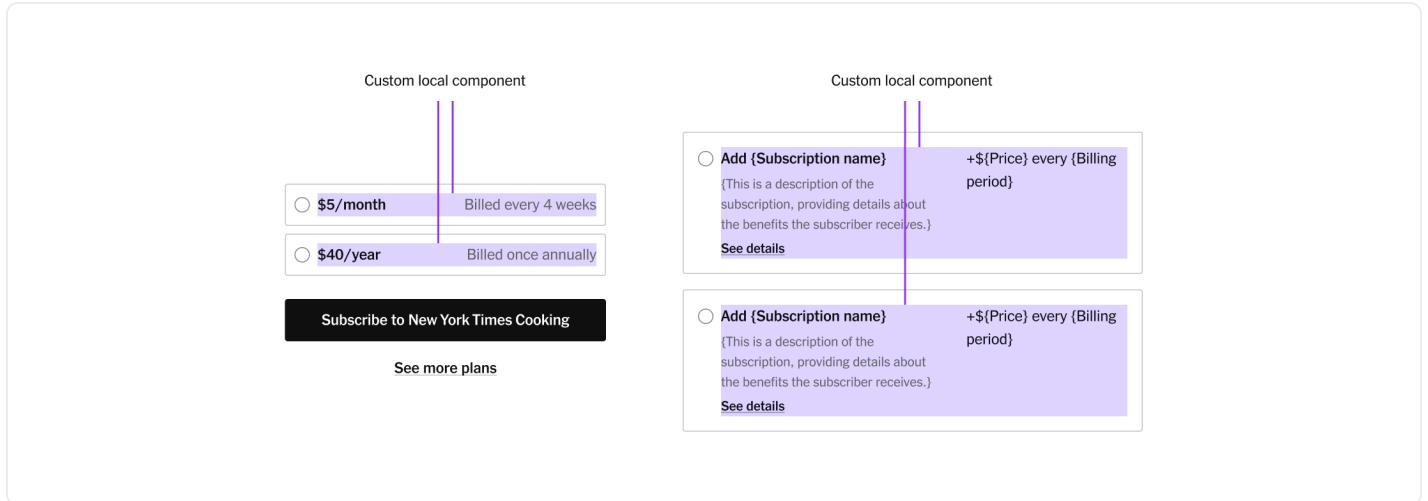
The label should describe each option clearly and succinctly to improve scanability and make selecting an option easy.

Secondary label

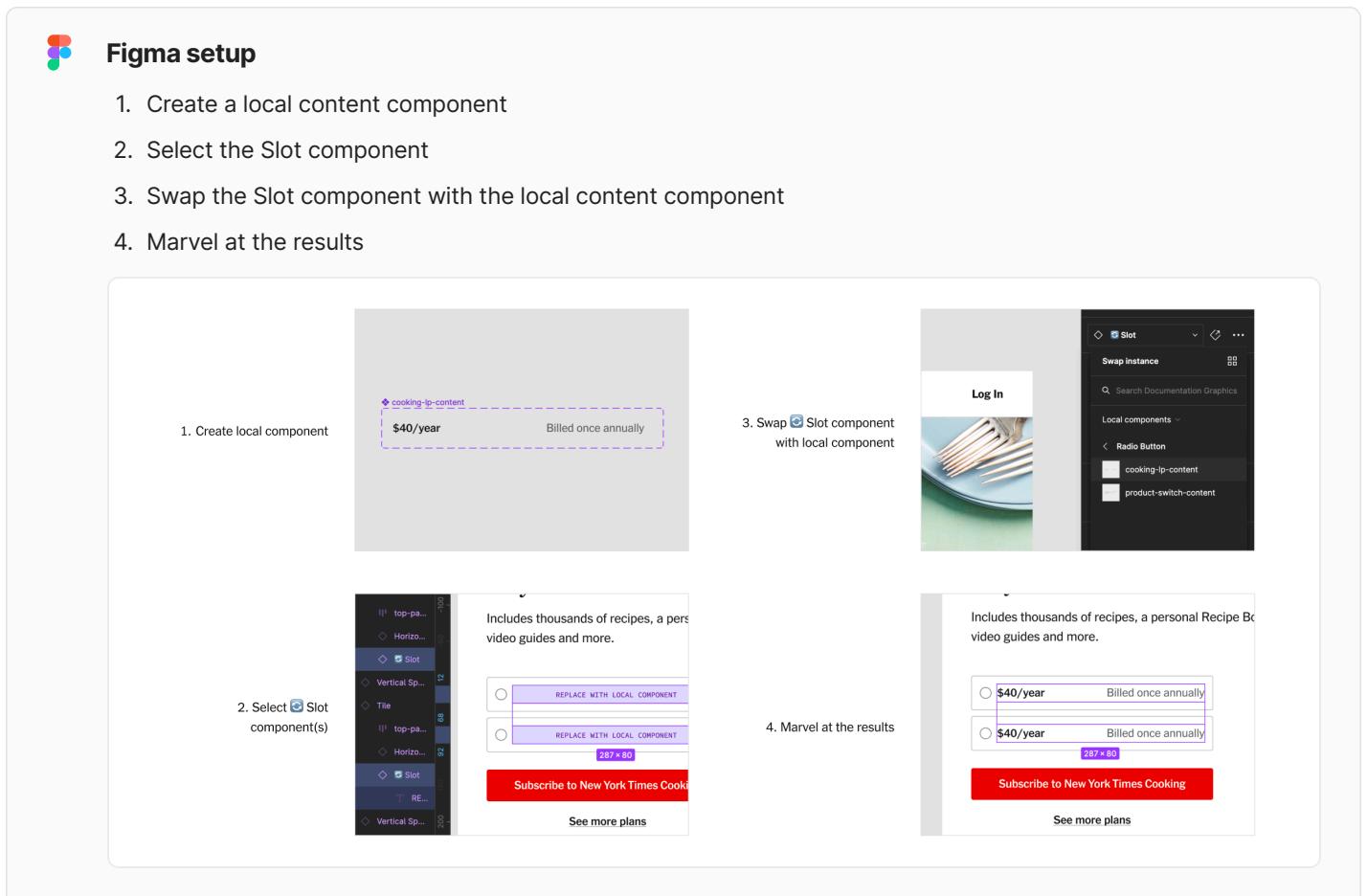
When there is a need to add more detail to an option, instead of writing a long label, you can use a secondary label to expand on the succinct primary label. This label is visually de-emphasized to maintain scanability.

Slot

When there is additional information needed in a radio option beyond the primary and secondary labels, you can use custom content with a slot. This should be used when you've determined that the Primary and Secondary Labels aren't meeting your needs.



The option to create custom content is only available for the Tile variant because its outer boundary allows for more flexible content approaches while still reinforcing what is included in each option.



Content guidance

Use a consistent variant for all radio buttons in a page/experience for consistency and scanability.

\$5/month

\$7/month

\$40/year

\$5/month

\$7/month

\$40/year Billed once annually

✓ Consistent types/variants

✗ Mixed types/variants

Consider what the most appropriate order of options is in your use case, for example: alphabetical, cost low-to-high, priority lot-to-high, or most to least popular. If you aren't sure what order users prefer, that might be a good case to test the best order. Randomizing the order across variants can help remove ordering bias when testing what people select most often.

We recommend using font size 2 for labels. This will ensure the radio input will align properly with the first line of text. If you plan to use another font size you will need to override the radio input spacing, which is not recommended. Reach out in the #hyphen channel if you need support.

Size

Height

The height of the radio button component is set to meet accessibility tap target requirements, rounded to the nearest spacer value, 1.5. If the text wraps or a secondary label is added the height will grow based on the height of that content.

Label text
↓ 12px (1.5)
↓ 12px (1.5)

↑ 48px

Example label text that is
longer than one line

Width

Content-width

Content-width radio buttons respect the width of the label. This ensures that the [hover and focus states](#) properly hug the contents of the option.

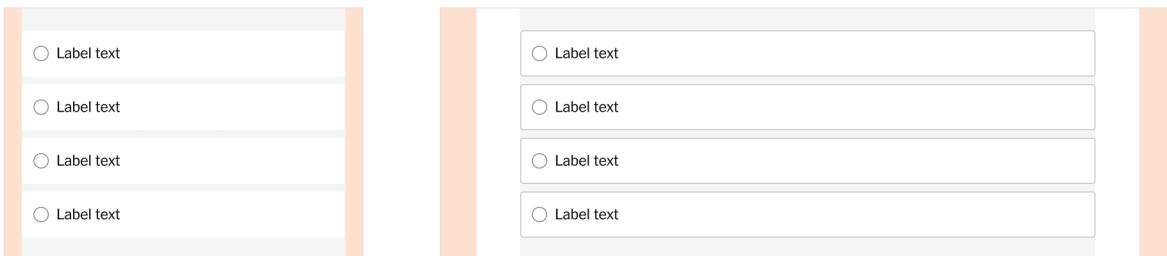
What went wrong on Friday, December 31, 2021?

- Missing paper
- Late paper
- Missing session
- Wet paper
- Damaged paper
- Other

Note: Tile variants do not support the content-width property to maintain visual symmetry.

Fixed-width

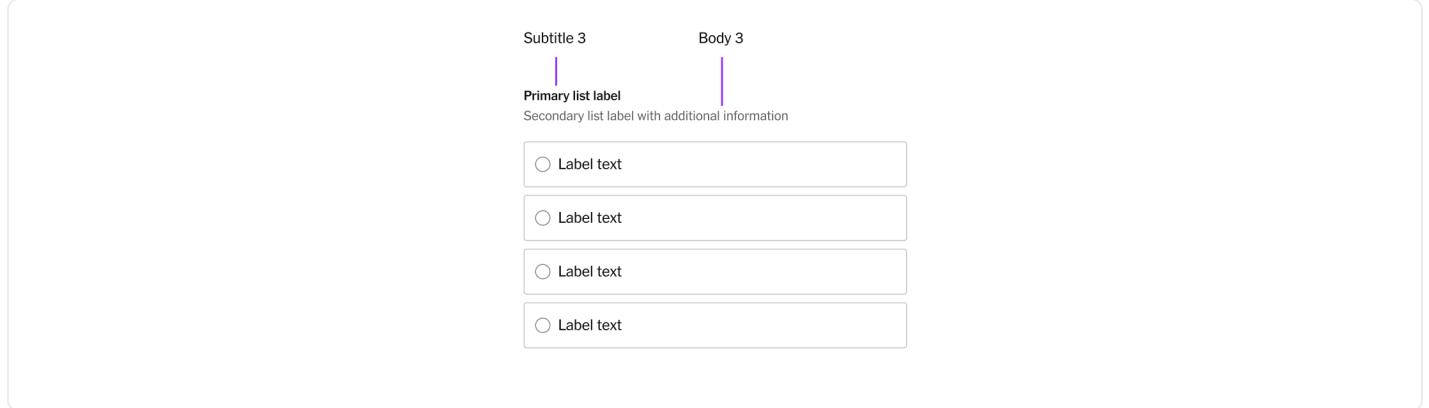
Fixed-width radio buttons allow you to define the width property yourself, allowing their width to be set to fill a container, be a set width, or respect the [Layout Grid](#).



Layout

List

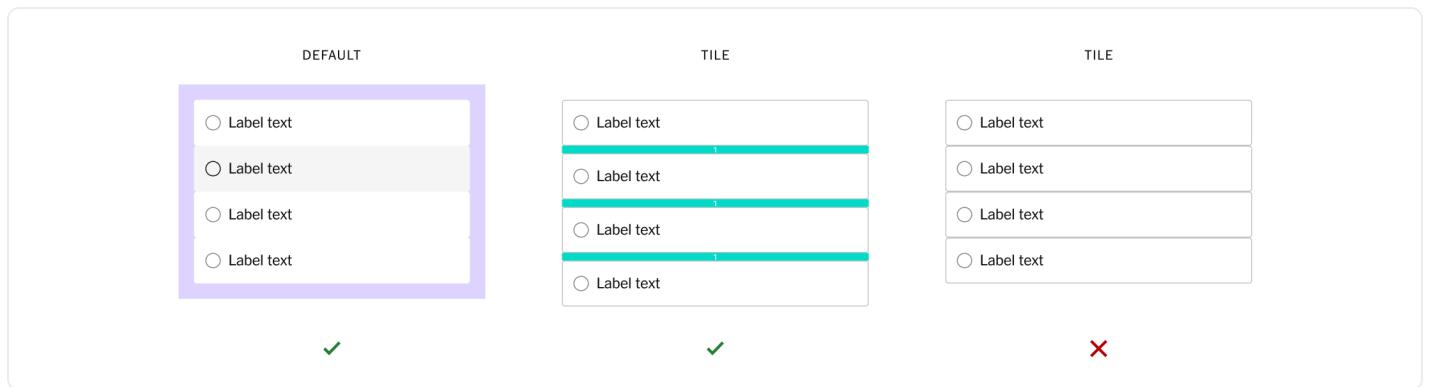
Radio buttons must exist in lists of at least two options for them to function. You can use existing text styles for a Primary and Secondary label if you need to provide context about the list itself. This can be useful when there are multiple radio lists in one experience.



There isn't a recommended maximum number of items. However, long lists of radio buttons can be hard for users to parse and can take up a lot of page real estate. If you would like the same functionality in a more compact form, consider using a dropdown field (coming soon).

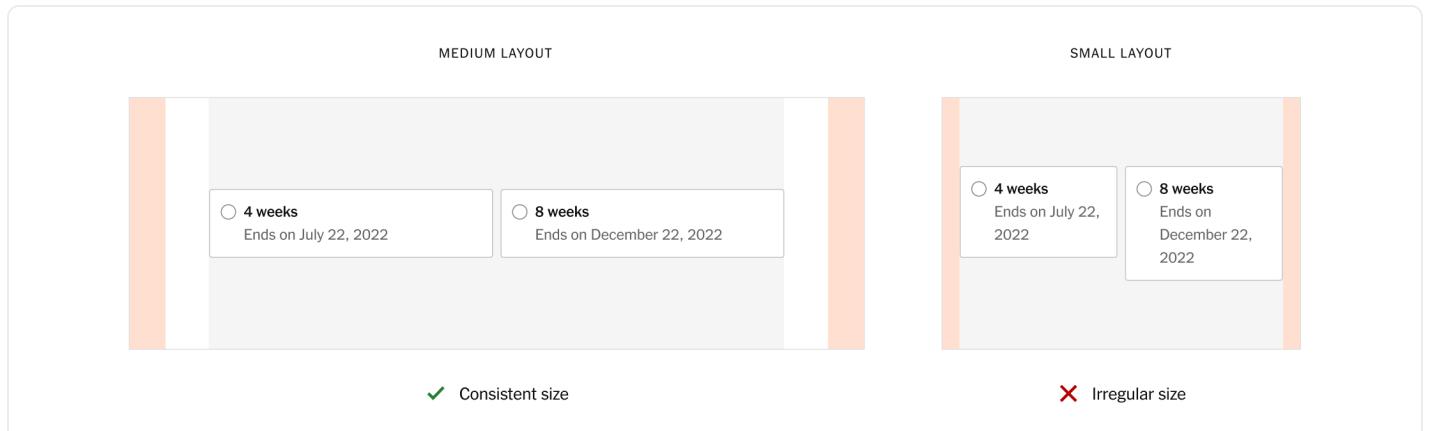
Spacing

Tile radio buttons require some amount of space between them. We recommend using a 1 spacer between tile radio buttons in most cases. There is no such requirement for default radio buttons.



Alignment

Radio buttons can be laid out vertically or horizontally depending on the use case. When possible, arrange the radio button lists vertically for easier reading.



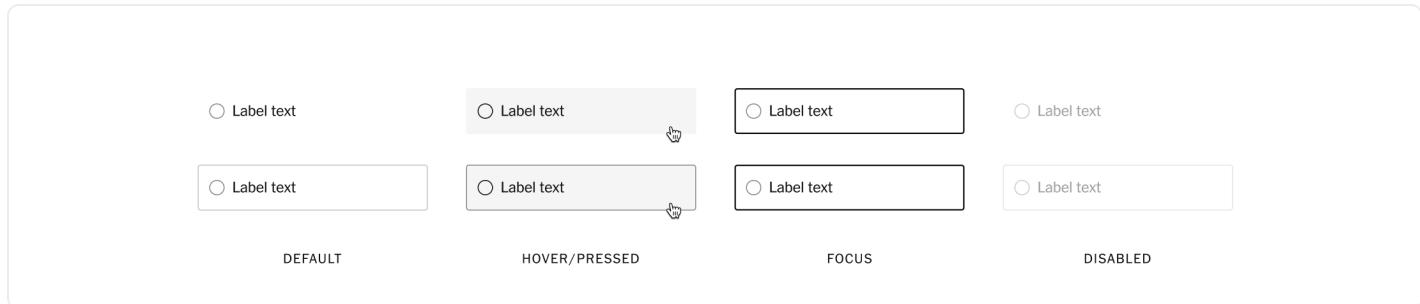
When aligning horizontally you should ensure that the heights of your radio buttons do not become irregular. You may need to set content limits and/or a fixed height on your radio buttons to do so. Use a [breakpoint](#) to switch to a vertical alignment on

smaller screens when a horizontal alignment is no longer appropriate.

Usability

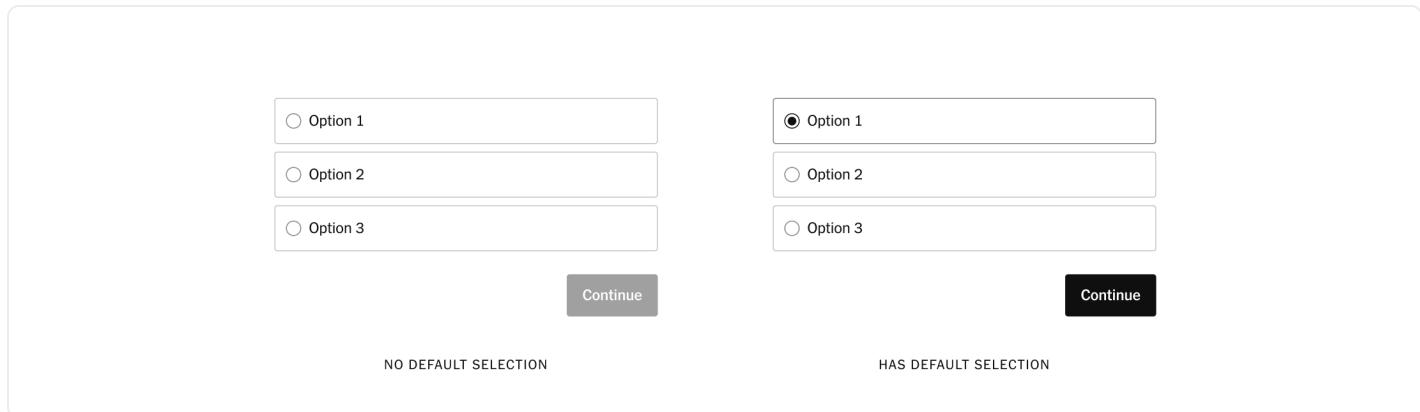
States

Radio buttons can be in a selected or deselected state. Beyond that, they follow the guidance outlined in the [States](#) token.



Default selection

Lists of radio buttons may or may not have one option selected by default when the experience loads. There are important tradeoffs to consider when deciding whether to do so.



No default

Not setting a default requires the user to intentionally select an option before continuing. This requires disabling the option to progress until a radio button is selected. Once a user selects a radio button, the list cannot be reset to have no option selected. Consider a “none of the above” radio option if this becomes an issue.

What went wrong on Friday, December 31, 2021?

Missing paper

Late paper

Missing section

Wet paper

Damaged paper

Other

Report and continue

In the above example, we shouldn't assume what problem a user is facing. If we did, it would alienate those with a different problem. We want to encourage intentional, thoughtful selection. We provide an "Other" option for users who don't want to select any of these reasons but still want to continue.

Has default

Setting a default value can bias a decision, seem pushy, or alienate users who don't fit your assumptions. Only use a default selection if you have data to back it up.^[1] Always choose a default value based on what is most useful to the user. A default selection should never be used as a tool for financial gain. If you have addressed these concerns, providing a default value can make it easier for a user to navigate through an experience smoothly.

What went wrong on Friday, December 31, 2021?

Missing paper

Late paper

Missing section

Wet paper

Damaged paper

Other

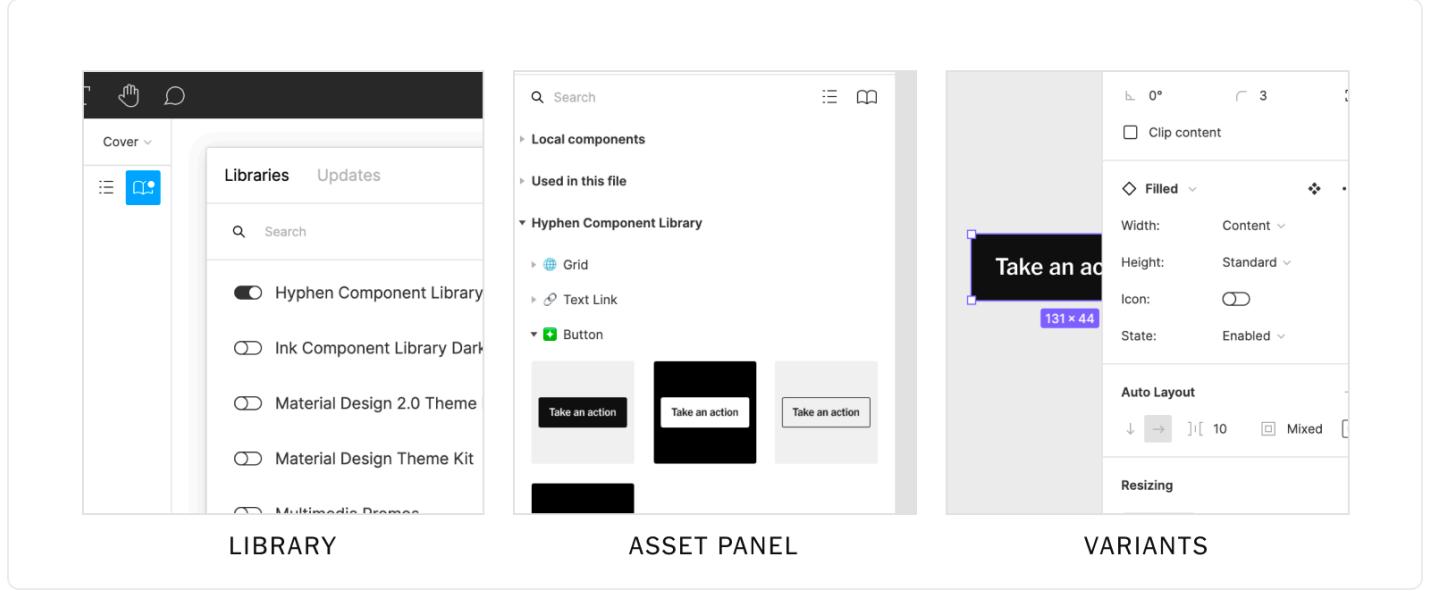
Report and continue

In the above example, testing validated that >80% of users selected "4 weeks" with no default selected. This proves that a strong majority of users prefer that option. Setting a default helps the user take an action quickly.

Implementation

Design

Button components are available for designers in our [Figma library](#).



Engineering

Radio Buttons are available as reusable code in the Hyphen codebase. All tokens and components are available to preview on [Storybook](#).

1. [U.S. Web Design System \(USWDS\) ↪](#)

Rule

Rules reinforce shifts in context and content by dividing or anchoring them. Rules are also known as lines, dividers, and separators.

 We are currently maintaining but no longer updating the Hyphen Design System. We recommend using [TPL Rule](#) for any new projects.

To migrate off the Hyphen system, [we're here to help](#). Hyphen will be decommissioned in approximately one year, in July 2024.

Horizontal Rule

Primary

Primary horizontal rules both divide and anchor content. Their thicker weight and darker color contrast with content, signifying a shift. Primary rules also catch the eye and draw the attention of readers who are scanning a large amount of content quickly.

Privacy

The New York Times takes your privacy seriously. Visit our [privacy page](#) for more information.

California Consumer Privacy Act

We use cookies and similar methods to recognize visitors and remember their preferences. We also use them to measure ad campaign

Section component

When using a primary horizontal rule to anchor, we recommend placing it above and closer to the content it's introducing than the elements above it.

Secondary

Secondary horizontal rules also divide and anchor content, but with less emphasis than the Primary rule. When using it to anchor content, we recommend following the same spatial relationship as the Primary rule.

Digital Subscription

Payment information

Subscription rate
\$7.50 every week

Secondary header

Tertiary

Tertiary horizontal rules only divide content. They have a lower contrast ratio than the other rules, making them the most subtle. Tertiary rules work best alongside spacing.

 You are connected to Facebook
Dagobah sith wedge binks hutt wedge organa kamino. Darth skywalker darth darth fett jango.

 Connect to Google
Dagobah sith wedge binks hutt wedge organa kamino. Darth

List Item component

Size

Height

Height is determined by the weight of the rule, which is defined within each rule type.

Width

We recommend that the rule inherit the width of the layout it is in, or the content it's supporting. There are rare exceptions in editorial contexts where we use a fixed-width format to separate content, but these should be used sparingly.

Vertical Rule

Vertical rules primarily divide content arranged horizontally. They are not used to anchor content.

Primary

Primary vertical rules divide content using high visual contrast and strong emphasis.

✓ **News.** Understand the world around you with original reporting from 1,700 journalists.

✓ **Games.** Unwind with Spelling Bee, Wordle, The Crossword and more.

✓ **Cooking.** Enjoy recipes, advice and inspiration for any occasion.

✓ **Wirecutter.** Explore independent reviews for thousands of products.

✓ **The Athletic.** Discover in-depth, personalized sports journalism.

✓ **News.** Understand the world around you with original reporting from 1,700 journalists.

✗ **Games.** Unwind with Spelling Bee, Wordle, The Crossword and more.

✗ **Cooking.** Enjoy recipes, advice and inspiration for any occasion.

✗ **Wirecutter.** Explore independent reviews for thousands of products.

✗ **The Athletic.** Discover in-depth, personalized sports journalism.

Subscription comparison

Secondary

Secondary vertical rules divide content using a lower level of visual contrast and reduced emphasis.

Outline Framework Pending Deal
President Biden and top aides are likely to leave key parts of his plan to pay for it. That could convince liberals that a deal is within reach.

Starline Reynolds for The New York Times

ax Code; May Cut Paid Leave
ze legislation to pay for new tax usually takes months, but

The Morning: What are the pluses and minuses of a wealth tax?

Eagles, Beavers, Sea Turtles: Why N.Y.C. Is Humming With Wildlife
New York is now "the greenest big city on Earth," one naturalist said. Some creatures have noticed, and are staying for a while.

Adrienne Grunwald for The New York Times

Scientists may have found the secret to being a better runner.

Home page modules

Size

Height

We recommend that vertical rules inherit the height of the container it is inside or the height of the content it is supporting.

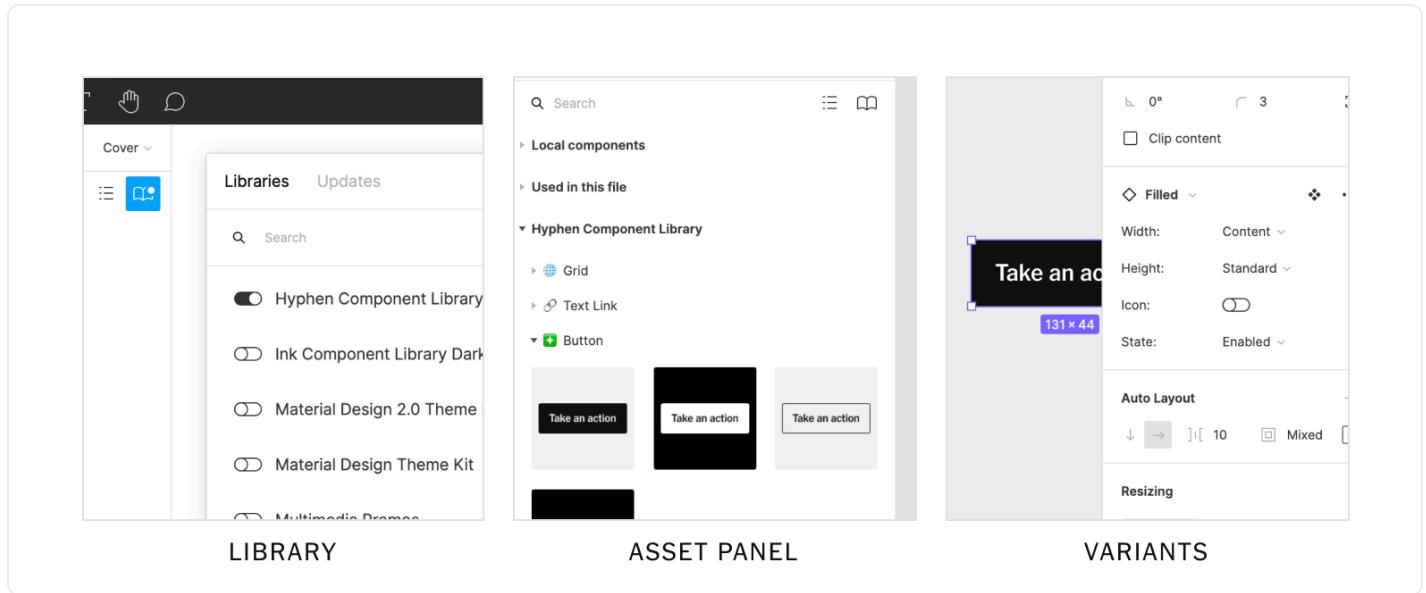
Width

Width is determined by the weight of the rule, which is defined within each rule type.

Implementation

Design

Our rule components are available in our [Figma library](#). You can find further guidance on enabling and using our Figma library [here](#).



Engineering

Rule components are available as reusable code in the Hyphen codebase. All tokens and components are available to preview on [Storybook](#).

Spacing

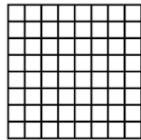
Spacing gives every piece of content its own space on the screen. Combined with Breakpoints and Grids, they enable consistent and intentional interfaces for our users.

⚠ We are currently maintaining but no longer updating the Hyphen Design System. We recommend using [TPL Spacing](#) for any new projects.

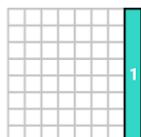
To migrate off the Hyphen system, [we're here to help](#). Hyphen will be decommissioned in approximately one year, in July 2024.

Base grid

We use an 8px base grid to maintain consistency within our spacing system.



Base grid (unit: 8px)



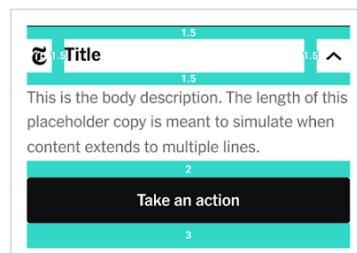
Vertical spacer 1

Height: 1x8=8px
Width: Flexible



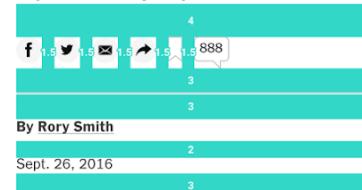
Horizontal spacer 1

Height: Flexible
Width: 1x8=8px



A Roman to the Core, and the Core of Roma

Francesco Totti, a native Roman, has spent his entire career with A.S. Roma: 23 years and counting. He hopes to end his playing days in a Roma jersey.



Token values

Our token values range from .5 - 10 (4px - 80px) The token value 1 represents the 8px base of our grid.

Token	.5	1	1.5	2	2.5	3	4	5	6	7	8	9
-------	----	---	-----	---	-----	---	---	---	---	---	---	---

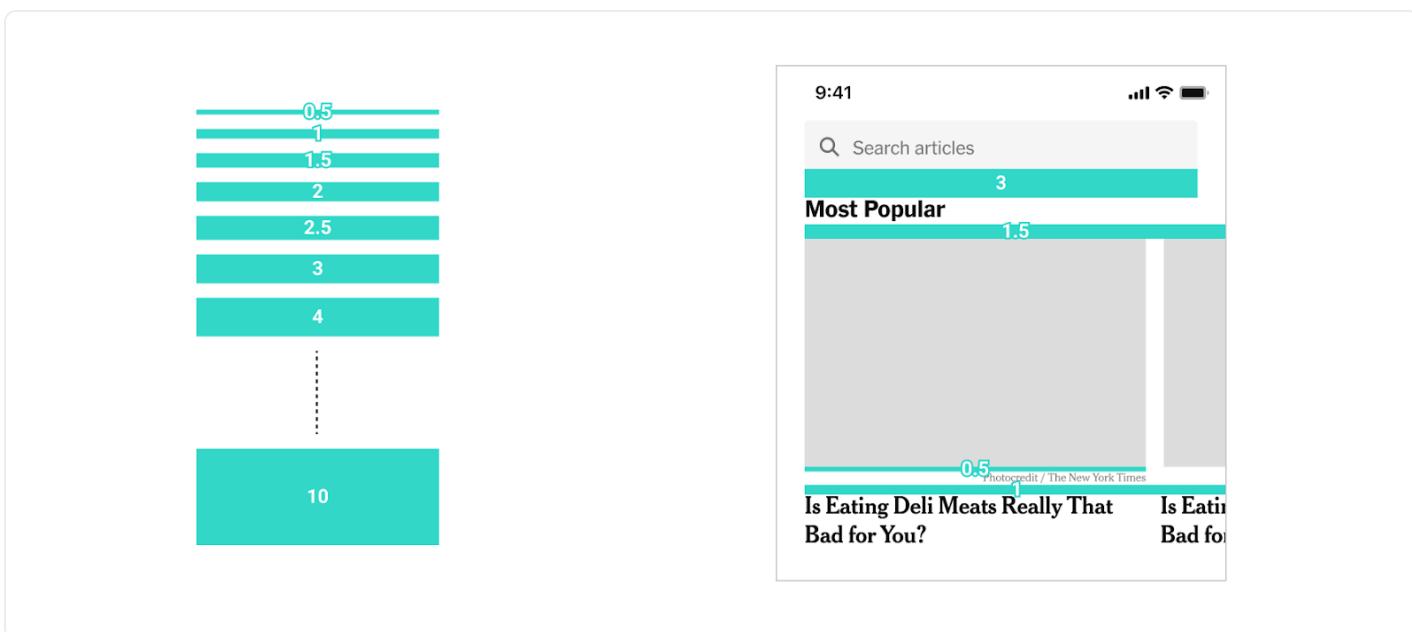
Pixel	4px	8px	12px	16px	20px	24px	32px	40px	48px	56px	64px	72px
REM	.25 rem	.5 rem	.75 rem	1 rem	1.25 rem	1.5 rem	2 rem	2.5 rem	3 rem	3.5 rem	4 rem	4.5 rem
↑ Base												

Component

We represent these spacing token values in Figma as transparent components that come in a set of pre-defined heights & widths. There is a horizontal and vertical spacer component for each token value.

Vertical spacers

Vertical spacers are used to space out elements vertically.

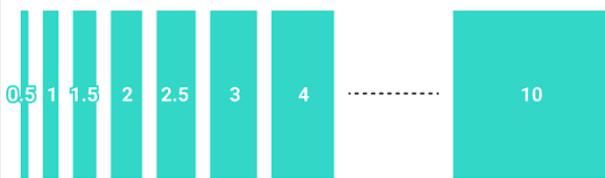


We value flexibility for layout spacing to ensure that individual experience needs can be met. Consider the following when deciding what spacing to use.

- We recommend spacing similar content closer together than dissimilar content.
- We recommend considering changes at breakpoints to honor the difference in screen sizes and user scroll effort.

Horizontal spacers

Horizontal spacers are used to space out elements horizontally.



[2 Take action 2](#)

[1.5 Keep me signed in](#)

We value flexibility for layout spacing to ensure that individual experience needs can be met. Consider the following when deciding what spacing to use:

- We recommend spacing similar content closer together than dissimilar content.
- Due to the nature of smaller screen sizes on mobile devices, we recommend limiting horizontal spacing at smaller breakpoints.

Usage

Responsive spacing

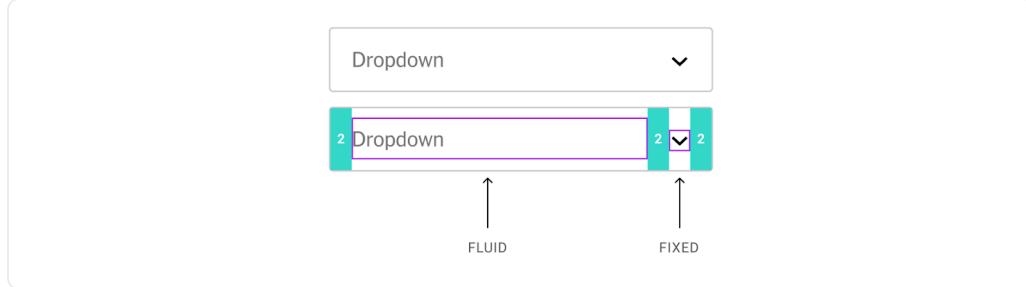
Spacing can be changed between different [breakpoints](#). For example, decreasing the vertical space between elements on smaller screen sizes can decrease page height and make scrolling less exhaustive.

MOBILE

DESKTOP

Component padding

Within a component, content is defined by fixed, fluid, and max width and height properties. Spacers are used alongside content with these properties to define content's relationship to the component it's within.



Implementation

Design

Our spacer components are available in our [Figma library](#).

The screenshot shows the Figma interface with three main panels:

- LIBRARY**: Shows a sidebar with navigation icons (Cover, Libraries, Updates) and a search bar. Below are lists of available libraries: "Hyphen Component Library" (selected), "Ink Component Library Dark", "Material Design 2.0 Theme", and "Material Design Theme Kit".
- ASSET PANEL**: Shows a search bar and a tree view of components under "Hyphen Component Library": "Grid", "Text Link", and "Button". The "Button" section is expanded, showing three button variants with the text "Take an action".
- VARIANTS**: A detailed view of a single button variant. It shows the button's dimensions (131x44), orientation (0°), and rotation (3). The "Filled" state is selected. The "Width" is set to "Content" and "Height" to "Standard". The "Icon" field is empty. The "State" is set to "Enabled". The "Auto Layout" settings show a grid with 10 columns and a mixed row height. The "Resizing" section is also visible.

Engineering

Spacing tokens are available as reusable code in the Hyphen codebase. All tokens and components are available to preview on [Storybook](#).

States

States communicate the status of UI elements to the user. They must have clear affordances to distinguish them from one another and the surrounding UI, and be applied consistently across components for user recognition.



We are currently maintaining but no longer updating the Hyphen Design System.

To migrate off the Hyphen system, [we're here to help](#). Hyphen will be decommissioned in approximately one year, in July 2024.

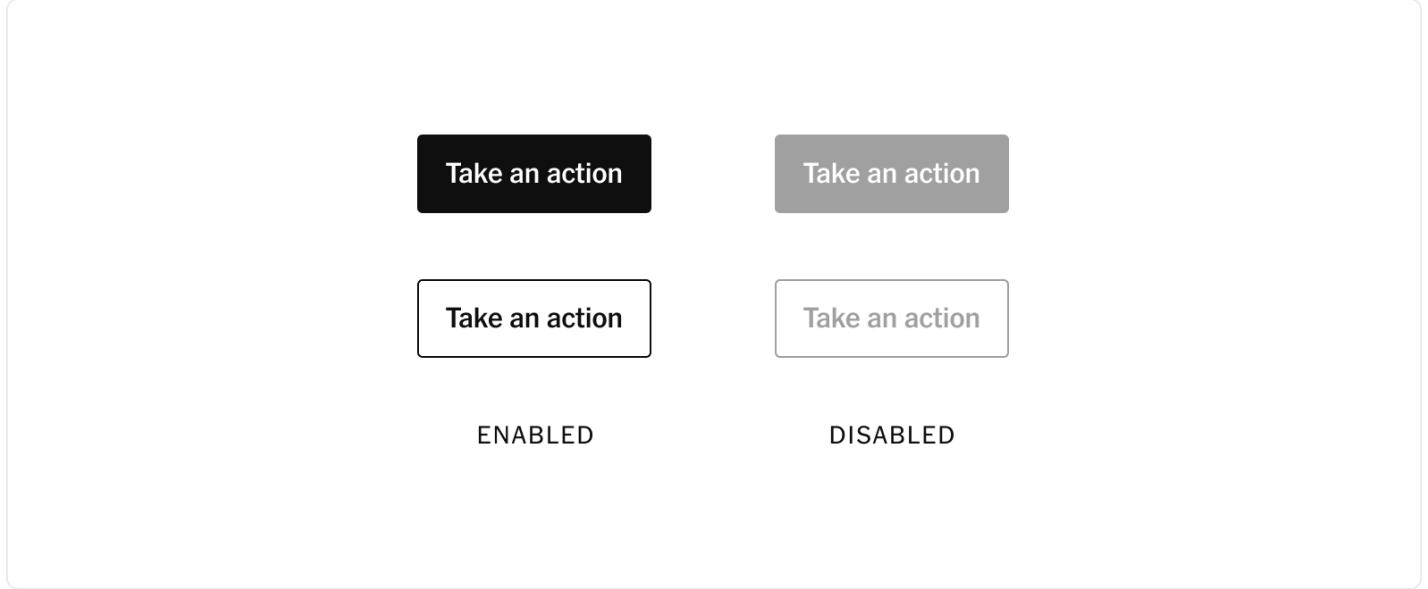
Enabled & disabled

Enabled

An enabled state communicates an interactive component or element. This is generally the default state of a component.

Disabled

A disabled state communicates a non-interactive component or element; they don't change state when tapped or hovered over. We achieve a disabled look by reducing the opacity of the component to 40%.

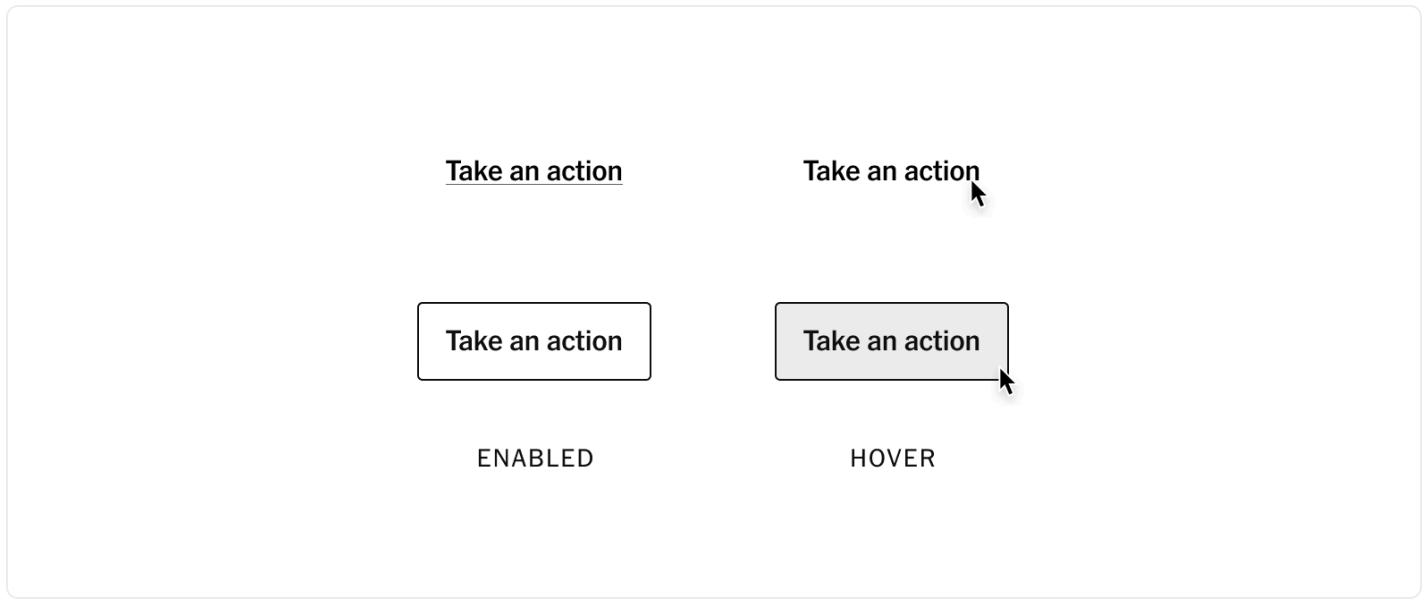


In progressive flows, where a user must input information before proceeding, it is better to hide the forward-moving button than apply a disabled state. Reducing the number of components on a screen helps focus the user on the necessary task.

Hover & pressed

Hover

A hover state communicates when a user has placed a cursor above an interactive element. We represent this change differently for certain components to ensure that we provide enough distinction between the enabled state.



For component backgrounds, we achieve a hover look by adding an overlay that changes its color. We provide two overlays: one to be used on dark-colored backgrounds and one to be used on light-colored backgrounds.

Light Background Overlay Hover

#121212 at 8%

Dark Background Overlay Hover

#FFFFFF at 16% opacity

Take an action

Dark Background Overlay Hover

Background Primary Inverse

Pressed

A pressed state communicates a user-initiated tap or click by a cursor, keyboard, or voice input method. We represent this change differently for certain components to ensure that we provide enough distinction between the enabled state on tap and the hover state on click.

Take an action

ENABLED

HOVER

PRESSED

For component backgrounds, we achieve a pressed look by adding an overlay that changes its color. We provide two overlays: one to be used on dark-colored backgrounds and one to be used on light-colored backgrounds.

Light Background Overlay Pressed
#121212 at 12%

Dark Background Overlay Pressed
#FFFFFF at 20% opacity

Take an action

Dark Background Overlay Pressed
Background Primary Inverse

Focused

A focused state communicates when a user has highlighted an element. Only one focus state may exist at a time on a screen. On the web it can specifically refer to highlighting an element using an input method such as a keyboard or voice.

Take an action

Take an action

Take an action

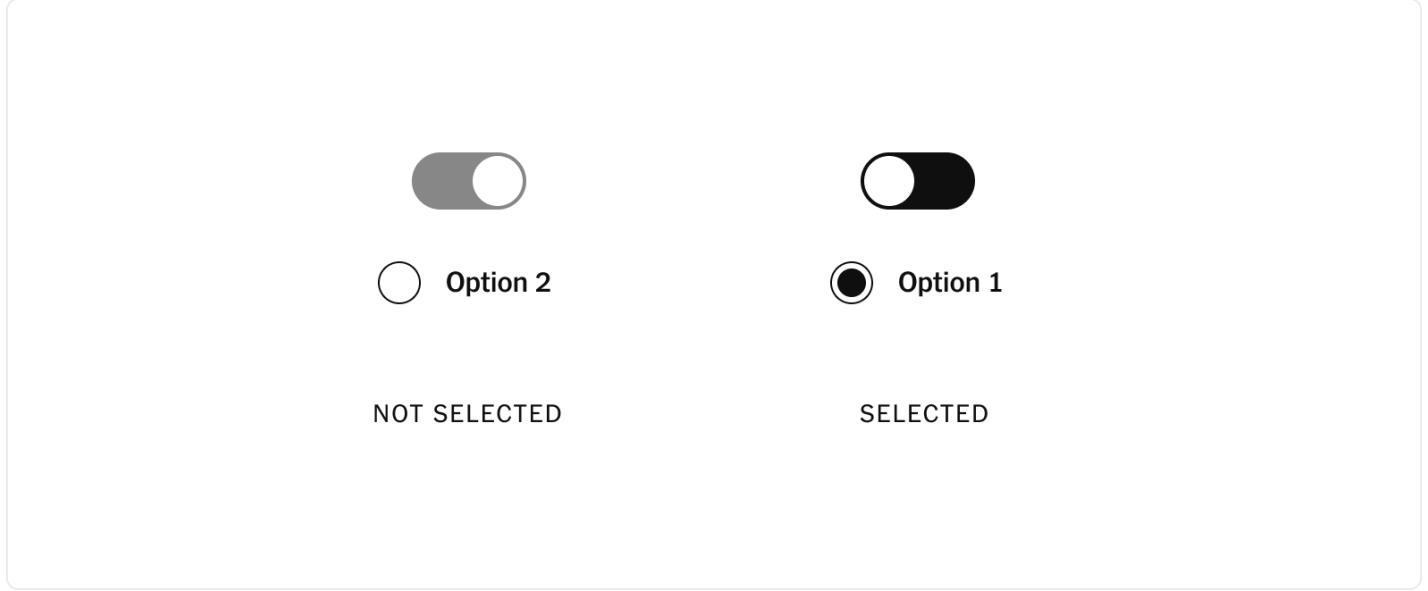
Take an action

NOT FOCUSED

FOCUSED

Selected

A selected state indicates the user-selection of one or more options. They are binary in nature, only existing as selected or not.



Loading & processing

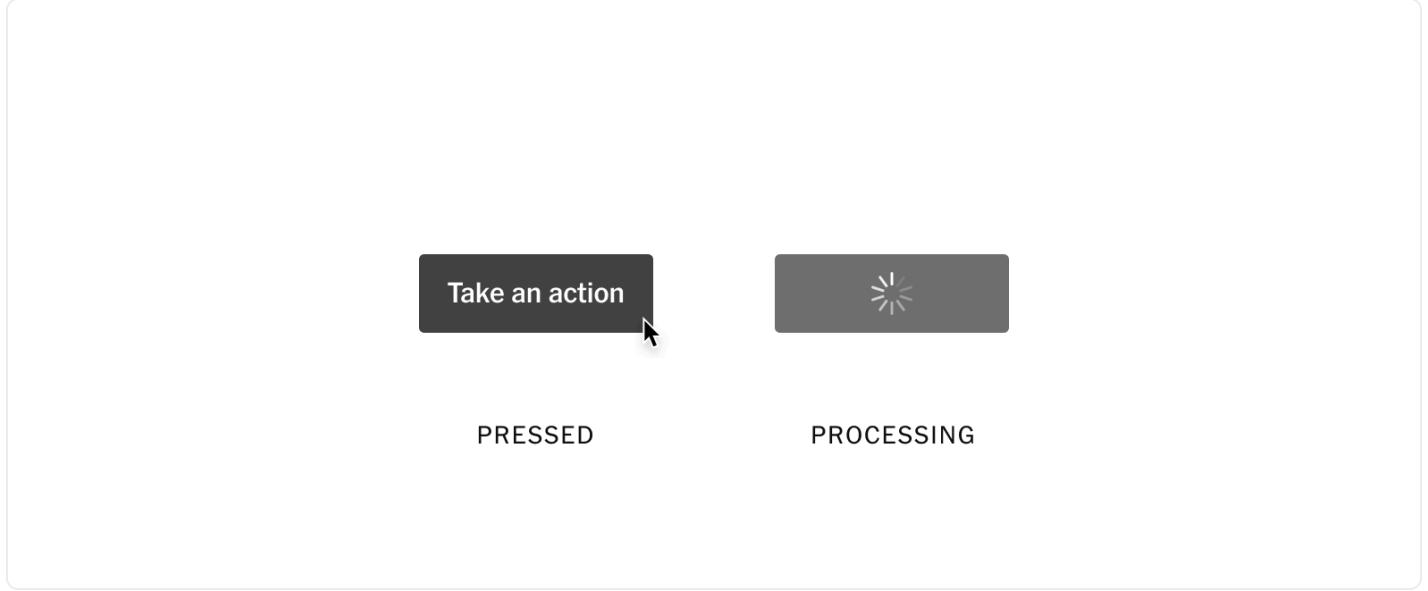
Loading

Loading refers to when a user is navigating between pages and the content needs to load in. This can be achieved at the component level or the page level.



Processing

Processing occurs when the user performs an action that takes time for the server to respond such as saving, verifying, or adjusting preferences. This is achieved at the component level.



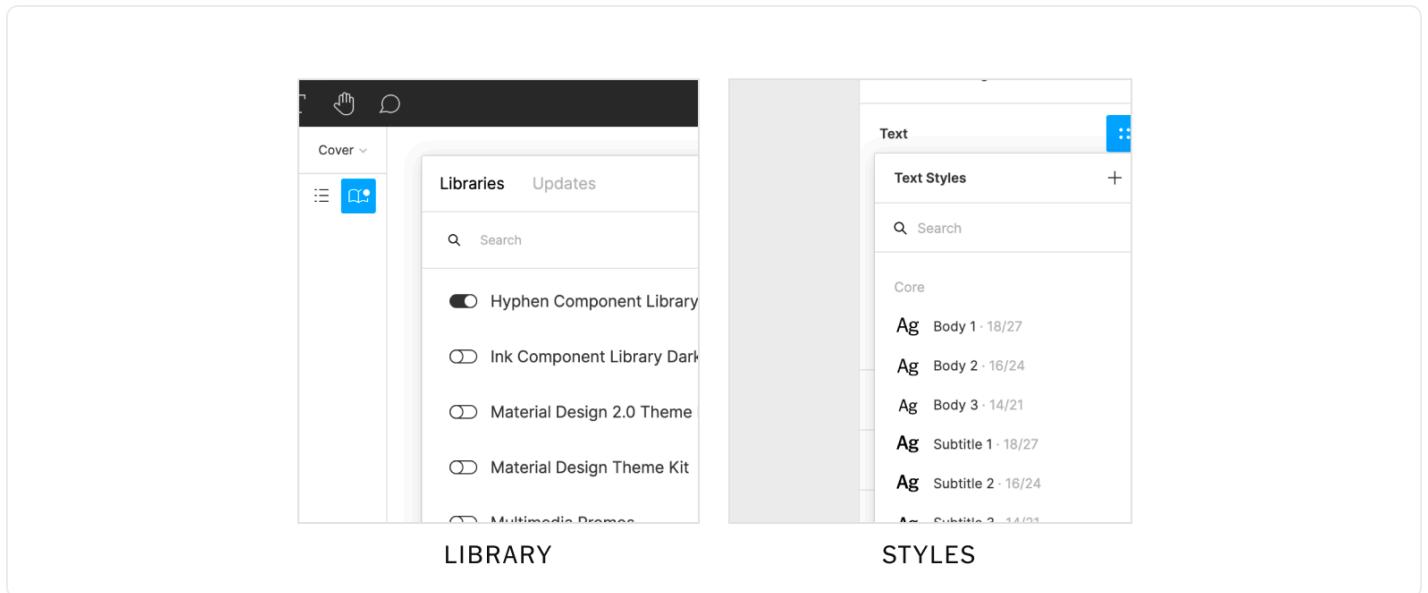
Contribution

We hope to grow and extend our guidance and standards for states to meet current and future product needs. If there is a specific state that you are working on that you would like to contribute back to the system, reach out to us on Slack!

Implementation

Design

Components have their own states but follow the same general guidance found here. The colors used for States can be found in the Color Styles Panel in [Figma](#).



Engineering

States are available as reusable code in the Hyphen codebase. All tokens and components are available to preview on [Storybook](#).

Text Link

Text links are interactive text that navigates a user from one experience to another.



We are currently maintaining but no longer updating the Hyphen Design System.

To migrate off the Hyphen system, [we're here to help](#). Hyphen will be decommissioned in approximately one year, in July 2024.

Type

Default

Default text links are used for standalone actions. They are always visually represented by an underline.

Take an action



Best of Late Night

As Needed | [See latest](#)



Cooking

Weekdays | [See latest](#)



Five Weeknight Dishes

Tertiary action

Directional

Directional text links are used for standalone navigational actions. They are always visually represented by a chevron symbol.

[Take an action ›](#)



DAILY

Coronavirus Briefing

[See the latest ›](#)

Navigational action

Inline

Inline links are used within blocks of content. They are always represented by an underline and the Content Accent color.

Take [an action](#) on this.

All of our advocates are currently occupied. You can try again soon or [contact us](#) another way.

Inline with content



Looking to make a block of content work like a link? Use [LinkBox](#) to extend the clickable area of a hyperlink to a group of elements. [More details in Storybook](#)

Style

In addition to the underline and symbol treatments, text links can apply color and typography treatments for certain use cases.

Color

The default color for standalone text links is the Content Primary palette color. We recommend using the Text Interactive Inline palette color for inline text links, to provide better distinction from the other text content.

All of our advocates are currently occupied. You can try again soon or [contact us](#) another way.

✓ Stronger contrast

All of our advocates are currently occupied. You can try again soon or [contact us](#) another way.

✗ Weaker contrast

Typography

Text links can be used with any of our [core type styles](#). We recommend limiting standalone text links to use Title, Subtitle, or Body styles to achieve the right hierarchy with other content.

Payment method

Mastercard *123

Edit

Body 2

Take an action ›

Subtitle 2

Content

Inline text link interaction is always second to content. They should be written as short, concise descriptions of their destination. Avoid non-informative phrases such as "click here" that don't make sense out of context.

All of our advocates are currently occupied. You can try again soon or [contact us](#) another way.

All of our advocates are currently occupied. You can try again soon or [click here](#) to contact us.

✓ Description of its destination

✗ Non-informative phrase

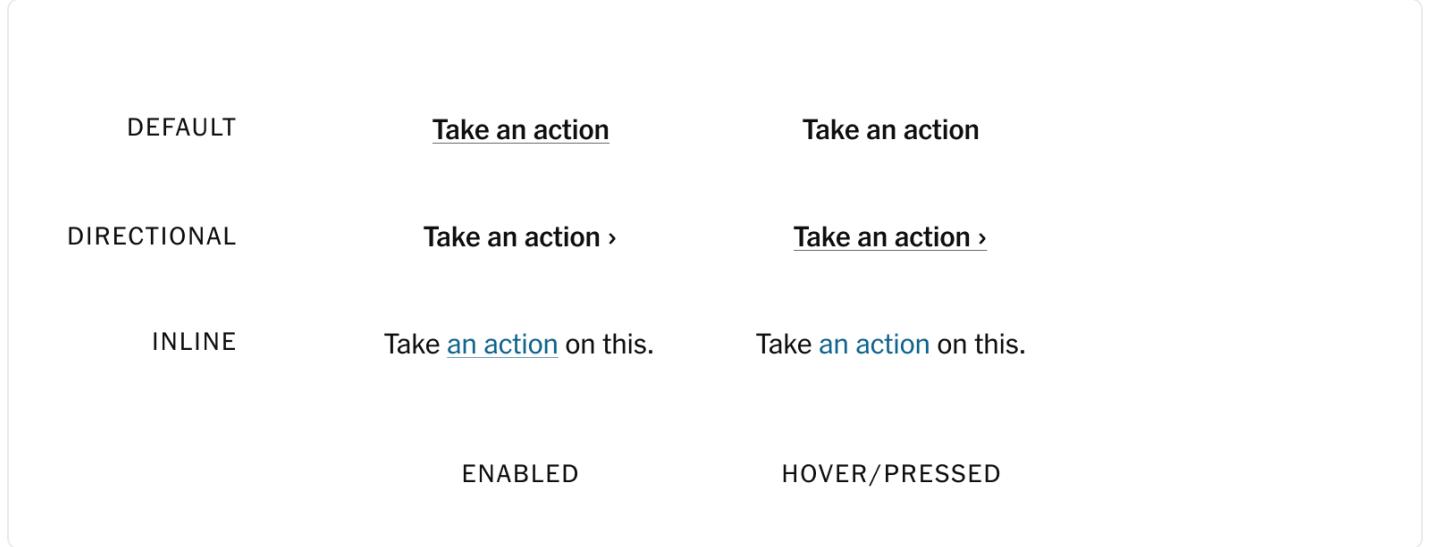
Size

The width of a Text Link is determined by the length of the content. The height of a Text Link is determined by the height of the type style's line height.

[Take an action](#) ↑ Line height

States

Default and **inline** text links have an underline, which goes away on hover and press. **Directional** text links do the reverse; an underline appears on hover/press.



Implementation

Design

Link components are available for designers in our [Figma library](#).

The screenshot shows the Figma interface with three main panels:

- LIBRARY:** Shows a list of components and libraries, with 'Hyphen Component Library' selected.
- ASSET PANEL:** Shows a list of assets under 'Hyphen Component Library', including 'Grid', 'Text Link', and 'Button'. The 'Button' section displays four variations of a 'Take an action' button.
- VARIANTS:** A detailed view of one button variant, showing its dimensions (131x44), rotation (0°), and transform (3). It also shows clipping, fill type (Filled), width (Content), height (Standard), icon (none), state (Enabled), auto layout, and resizing options.

Engineering

Link components are available as reusable code in the Hyphen codebase. All tokens and components are available to preview on [Storybook](#).

Typography

Typography is the usage of different visual treatments to display information and actions with words. Our system prioritizes accessibility and clarity for our users.

⚠ We are currently maintaining but no longer updating the Hyphen Design System. We recommend using [TPL Typography](#) for any new projects.

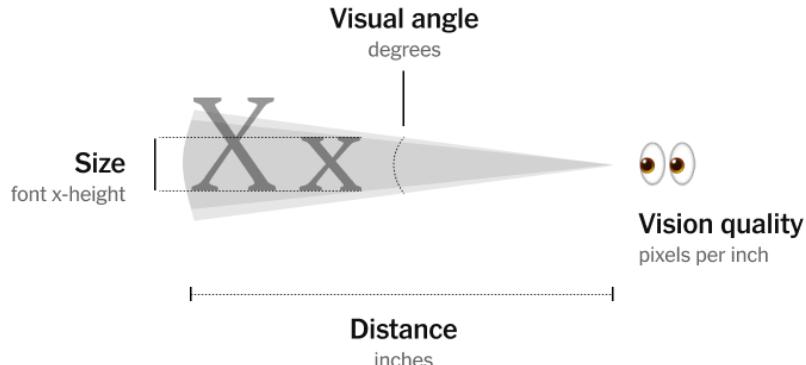
To migrate off the Hyphen system, [we're here to help](#). Hyphen will be decommissioned in approximately one year, in July 2024.

Accessibility

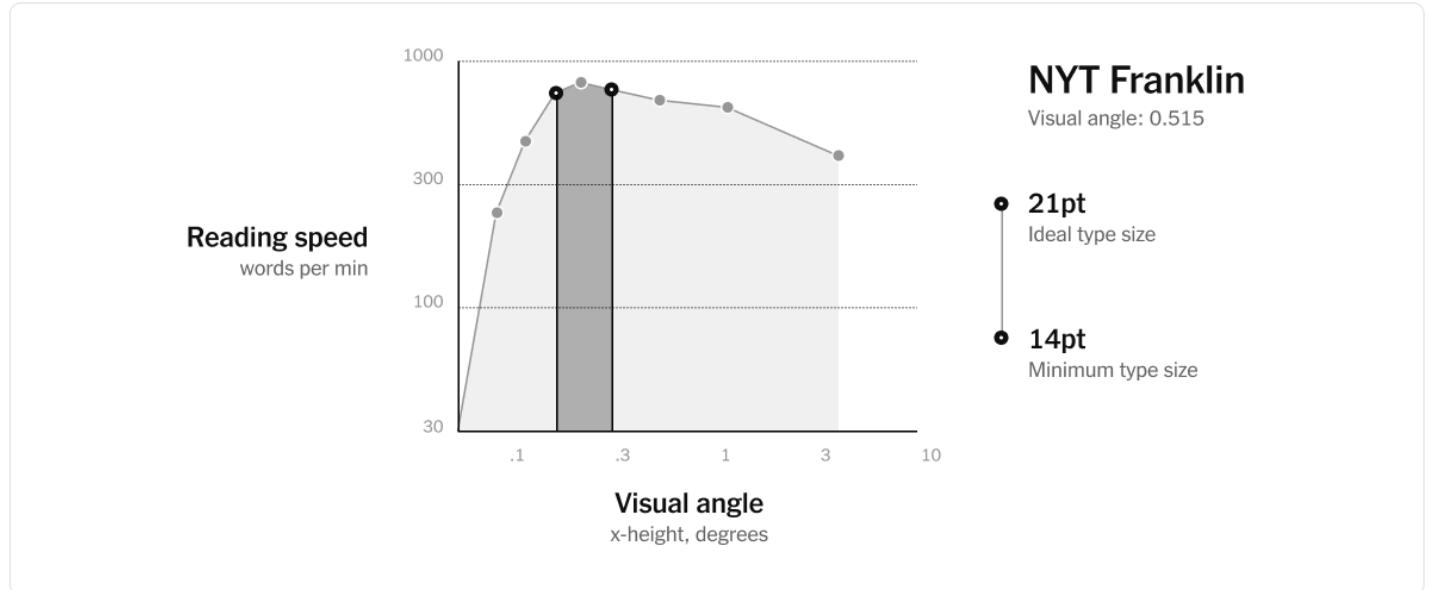
Type accessibility can be understood as type legibility, determined by **size** and **color**.

Size

We based our definition of legibility on [2011 research published in the Journal of Vision](#). The study found a correlation between the degree of visual angle and reading speed.

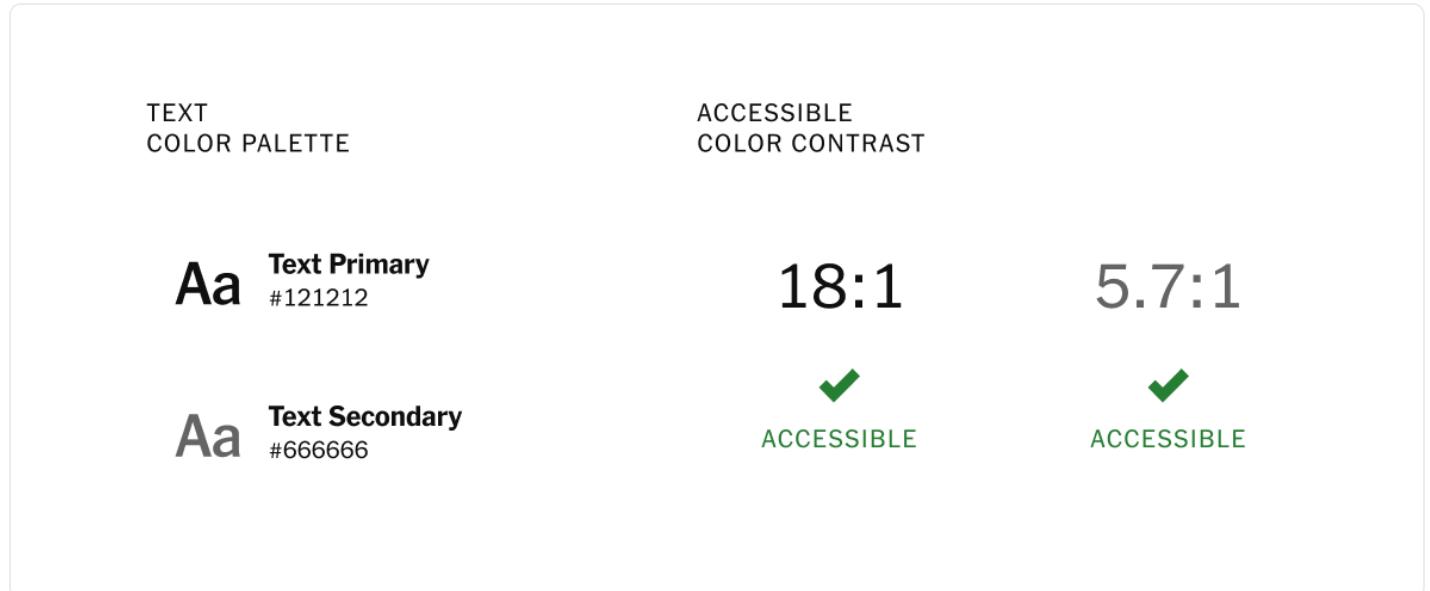


The visual angle (defined by the font size's x-height, the distance between the eye and the device, and the overall pixel quality) combined with the maximum and minimum thresholds for reading speed allow us to quantify the minimum and ideal type sizes for any given font. Our system baselines above the minimum illegible font size and creates a hierarchy that revolves around the ideal type size.



Color

Type color contrast accessibility is defined by WCAG 2.1 guidelines and is used across the industry. To ensure that you are providing the right color contrast for typography, utilize our [color palette system](#).



Fonts

We utilize two main typefaces in Hyphen: NYT Franklin and NYT Cheltenham.

NYT Franklin

Franklin should be used with functional and instructional language content. Products such as Account should use mostly Franklin within its interfaces to align with its utilitarian purpose.

WEIGHTS

Franklin Extra Bold
Franklin Bold
Franklin Semibold
Franklin Medium
Franklin Light

Email newsletters

You're currently signed up to with your email **youraddress**

Information content

Email address

evan.maeda@nytimes.com

Continue

Interface components

NYT Cheltenham

Cheltenham should be used with conversational language content or to represent the voice of the company. Because Cheltenham is such an important part of our brand identity, we use it to help anchor that identity in utilitarian experiences.

WEIGHTS

Cheltenham Extra Bold
Cheltenham Bold
Cheltenham Semibold
Cheltenham Medium
Cheltenham Light
Cheltenham Extra Light

The New York T

Billing Hist

Page headers

Step 1 of 4

Please tell us why you'd like to cancel your New York Tim subscription.

Conversation language content

Other fonts

Although we mainly use Franklin and Cheltenham, we use other Times typefaces to maintain familiarity with experiences outside customer experiences. For example, NYT Karnak is used infrequently for representing certain products and brands such as NYT Cooking, NYT Games, and Wirecutter.

Set

Our typography set was made to have a clear hierarchy, legibility, and readability when a device is around 12" from a person's eye. This distance is the average standard for mobile and desktop device usage.

CORE TYPE STYLES

Headline 1

NYT Franklin Semibold
Size: 26px
Line height: 34px

Title 1

NYT Franklin Bold
Size: 18px
Line height: 27px

Subtitle 1

NYT Franklin Semibold
Size: 18px
Line height: 27px

Body 1

NYT Franklin Medium
Size: 18px
Line height: 27px

Headline 2

NYT Franklin Bold
Size: 22px
Line height: 28px

Title 2

NYT Franklin Bold
Size: 16px
Line height: 24px

Subtitle 2

NYT Franklin Semibold
Size: 16px
Line height: 24px

Body 2

NYT Franklin Medium
Size: 16px
Line height: 24px

Title 3

NYT Franklin Bold
Size: 14px
Line height: 21px

Subtitle 3

NYT Franklin Semibold
Size: 14px
Line height: 21px

Body 3

NYT Franklin Medium
Size: 14px
Line height: 21px

SPECIALTY TYPE STYLES

Display 1

NYT Cheltenham Extra Light
Size: 50px
Line height: 56px

• Legal

NYT Franklin Medium
Size: 12
Line height: 18px

Display 2

NYT Cheltenham Light
Size: 40px
Line height: 48px

Display 3

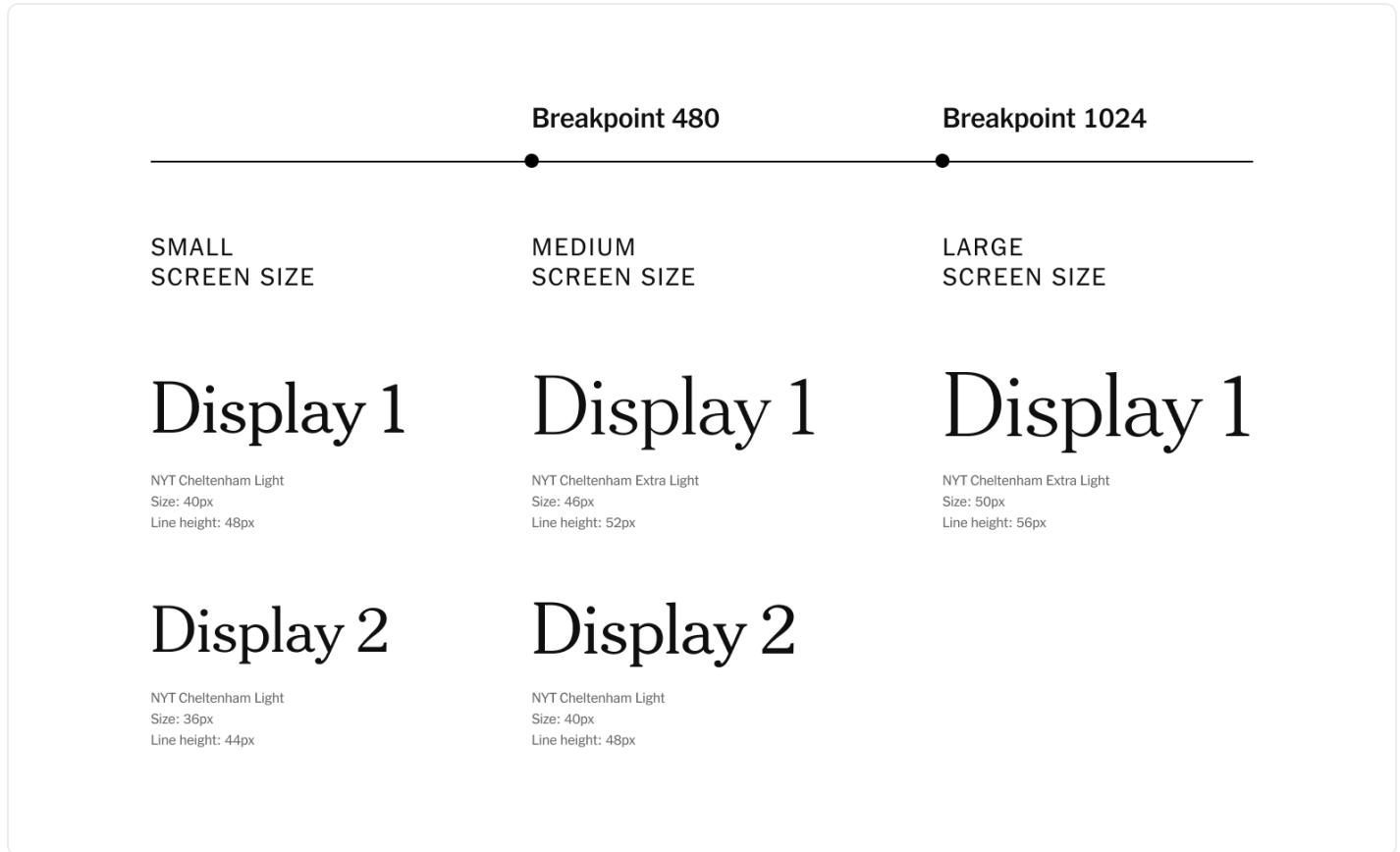
NYT Cheltenham Book
Size: 30px
Line height: 40px

● Falls below legibility standards, use sparingly

● These styles change in size and line height at different breakpoints

Responsive sizing

To maintain hierarchy, we have provided responsive sizing for all styles. However, not all styles change at every size. Below are the type styles that change based on our [defined breakpoints](#).



Styles

The typography system is broken down into two style groups: **core** and **specialty** styles.

Core type styles

The core typography styles were selected for the ideal hierarchy.

Headline

Headline styles are used to label entire sections of content or as calls to action.

- They can be used alone or with Subtitle or Body
- Avoid using long strings or dynamic content in this style

Headline 2

Email newsletters

Body 2

You're currently signed up to receive the following newsletters and communications with your email **youraddresshere@email.com**.

Title & Subtitle

Title and Subtitle styles are used as the primary and secondary labels of UI.

- They can be used alone, together, or with Body
- You can use longer strings and dynamic content in this style

Email address

evanmaeda@gmail.com

Subtitle 2

Payment method

Body 2

Mastercard *123

Recipient's email address

uma.nagesh@nytimes.com

Title 3

Enter recipient's email address

se

Body

Body is used for long strings of content.

- They should pair with Headline, Title, or Subtitle so it isn't floating by itself
- They should be used for user-generated text, like within text inputs.

California Consumer Privacy Act

We use cookies and similar methods to recognize users when they visit our site, remember their preferences. We also use them to measure and analyze how users interact with our content. We use this information for campaign effectiveness, target ads and analyze search queries. You can learn more about these methods, including how to manage them, by viewing our cookie policy.

✓ Good hierarchy paired with Title/Subtitle styles

California Consumer Privacy Act

We use cookies and similar methods to recognize users when they visit our site, remember their preferences. We also use them to measure and analyze how users interact with our content. We use this information for campaign effectiveness, target ads and analyze search queries. You can learn more about these methods, including how to manage them, by viewing our cookie policy.

✗ Weak contrast paired with other Body styles

Specialty type styles

The specialty typography styles are used for specific products or use cases. See the guidance below for appropriate usage.

Display

Display is intended for use at large sizes for editorial content and page headers.

The New York Times

Display 2

Display 1

Billing History

Step 1 of 4

Please tell us why you'd like to renew your New York Times subscription.

Legal

Legal is used for disclaimers and other legal copy. It falls below our legibility standards, so we recommend using it sparingly.



No, cancel my subscription

When you cancel, we will stop charging your account the following billing cycle. Each billing cycle is 4 weeks. Your will continue until the end of your current billing cycle.

✓ Legal disclaimer text

Email newsletters

We offer over X emails on a variety of topics. Find the newsletters that interest you, and remove any that don't.

Email newsletters

We offer over X emails on a variety of topics. Find the newsletters that interest you, and remove any that don't.

Email newsletters

We offer over X emails on a variety of topics. Find the newsletters that interest you, and remove any that don't.

✗ Used as a smaller body size

Contribution

We have structured our typography system to enable evolution and contribution to meet current and future product needs.

Adding new styles

We recognize that new use cases will emerge that our typography system may not consider or be able to cover in its current state. Consider the following when determining if an addition is necessary:

- Core type styles were created with flexibility and breadth in mind. To ensure that we maintain consistency, we do not recommend additions to our core type styles.
- Specialty type styles are designed to grow as new use cases. Feel free to request a new style to meet your use case.

Updating existing styles

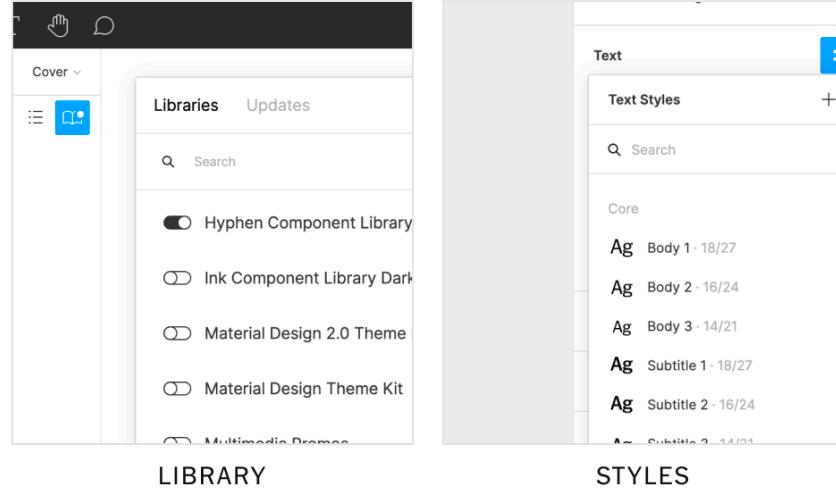
We welcome feedback and recommendations for our existing type set and styles. Consider the following when determining if an update is necessary:

- **Core type styles** make up the majority of our screens. Consider how a change might impact multiple products and experiences if updated.
- **Specialty type styles** have specific use cases and can be easily added to. Consider whether adding a new specialty type style or updating an existing one is more beneficial.

Implementation

Design

Typography styles are available for designers in our [Figma library](#).



Engineering

Typography styles are available as reusable code in the Hyphen codebase. All tokens and components are available to preview on [Storybook](#).

Piano

 We are currently maintaining but no longer updating the Piano Design System.

To migrate off the Piano system, [we're here to help](#). Piano will be decommissioned in approximately one year, July 2024.

The Piano design system was documented entirely in Storybook. [The Piano Storybook can be found here](#).