Learn

Getting Started

What's New

Help & Contributing

Use

Foundations

Components

Discover What's Next

Foundations in Progress

Components in Progress

Patterns & Templates in Pr...

+ Add Page

Unity | Design+Code System

Unity streamlines code and design into an efficient and connected system, freeing up time to explore new ideas for impactful customer solutions.

> ***"Each individual piece of all that we do is what makes this system whole."*** **— Nielsen Unity**

Learn

Help & Contributing

How to contact us for questions, report issues, contribute and make requests.

What's New

See what's new in our site, design kits and code.

Getting Started

Learn how to get started as a designer or developer with these useful references.

Use

Foundations

Foundations establish our brand, structure, continuity and quality in our products.

Components

Components are reusable building blocks that quickly snap prototypes or code together.

Discover What's Next

Patterns & Templates in Progress

Discover what a pattern is and what will be added to a new Pattern section.

Foundations in Progress

These pages will be added to Unity. Meanwhile see links to Storybook for existing code and content.

Components in Progress

These pages will be added to Unity. Meanwhile see links to Storybook for existing code.

Learn

How to get started, get help, contribute, report an issue and find out what's new.

- **Getting Started** for designers and developers using our system.
- **What's New** on the site, in code and in Figma.
- **Help & Contributing** to get help with design or development, and learn how to contribute or make a request.

Have Questions? Contact us in these Gspaces.

Design: Unity | Design Questions & Feedback | **Code and Development:** GDS(Unity) | Components Dev Community

Use

Find examples, variants, guidance, accessibility standards, responsiveness, code previews, and props.

- **Foundations** establish brand and continuity through writing, icons, typography and more.
- **Components** are the basic reusable building blocks used when designing and coding our applications.

Discover What's Next

Link directly to Storybook for in-progress components that exist in code and see what's next.

- **Foundations in Progress**
- **Components in Progress**
- **Patterns & Templates in Progress**

Copyright © 2024 The Nielsen Company (US), LLC. All Rights Reserved.

On this page

"Each individual piece of all that we do is what makes this system whole." — Nielsen Unity

Learn

Use

Discover What's Next

Learn

Getting Started

What's New

Help & Contributing

Use

Foundations

* Accessibility

* Colors

Icon Libraries

* Layout

* Logos

* Navigation

* Tokens

* Typography

* Writing Guidelines

Components

Accordions

Action Icons

Breadcrumbs

Buttons

Button Groups

Checkboxes

* Cards

Chips

Getting Started

Learn how to get started as a designer or developer with these useful references.

Designers

- **Design Kits** (Figma) - Add these 9 libraries to Figma to begin designing and prototyping with our system.

What are Design Kits?
Design Kits are libraries of components that designers use in our Figma tool to quickly create concepts, prototypes, and design deliverables without starting from scratch for each element like select boxes or navigation bars.

Developers

For now, use Storybook to learn how to get started using our libraries.

- **Infra Frontend Layer** (Gitlab) - Use this code, service and API layer for the frontend layer of your application.
- **Code Component Library** (Gitlab) - Snap in functionality to your application with code from this reusable library.
- **MAF API Documentation** - Keep up-to-date on all API services and utilities in our platform.

Copyright © 2024 The Nielsen Company (US), LLC. All Rights Reserved.

On this page

Designers

Developers

* Drag and Drops
* Empty States
Inputs
Input Fields
Text Areas
* Input Gro...
* Input Call...
* Labels
Links
Lists
List Callouts
Loaders
Menus
Messages
* Modals
Pagination
* Popouts
Progress Bars
Range Sliders
Radios
* Radio Tile Buttons
Scroll Bars
* Search
Selects
* Side Panels

Links

The Links component is used to create hyperlinks with various styles and behaviors for users to navigate to related content.

Link Text

When to Use

The Links component should be used whenever you need to create clickable text that navigates to a different page or location. It is commonly used in navigation menus, buttons, and text links within content. The variant property allows you to customize the appearance of the link based on the context or visual hierarchy.

Code Preview

Primary Link

Primary Underlined Link

Secondary Link

100% Show Code

Properties

aria-label `string`

children `Node[]`

className `string`

data-maf-aria-hidden `boolean`

data-maf-aria-hidden-exact `boolean`

data-selector `string`

data-track-ids `string`

externalAs `string`

fontSize `"s0" | "s100" | "s200" | "s300" | "s400" | "s500" | "s600" | "s700" | "s800"`

fontWeight `"regular" | "bold"`

for `string`

href `string`

id `string`

onBlur `Event`

onClick `Event`

onFocus `Event`

onKeyDown `Event`

onMouseOver `Event`

ref `Event`

role `string`

style `Event`

tabIndex `number`

target `"_blank" | "_parent" | "_self" | "_top"`

variant `"primary" | "secondary" | "primaryUnderlined"`

Anatomy

Design Principles

- Links are used to navigate to pages, not to trigger actions or events, except when the text is actually a label to an action icon.
- Linked text should always clearly represent the link's purpose and destination.
- Avoid using vague or generic labels for links. Do not use the words "Click Here", "Find It", or similar phrases. They don't inform the user about the link.
- Assume links will open in new tabs, unless the link is inside of a table.
- Links are often used in tables to link to a detail page.

Do's and Don'ts

Do's

- Use if you need the user to open a web page.
- Write the link so it's clear to the user what they will be viewing once they click.

Don'ts

- Never show text in blurple-600 unless it's a link.
- Don't link whole paragraphs and avoid wrapping.

Variants

Light and dark mode

Link Text Link Text Link Text Link Text Link Text Link Text

Link Text Link Text Link Text Link Text Link Text Link Text

Sizes

0, 100-800

Link Text Link Text Link Text Link Text Link Text Link Text Link Text Link Text Link Text

Types

- Primary, secondary (always underlined), regular and bold
- Primary underlined, regular and bold

Link Text Link Text Link Text Link Text Link Text Link Text

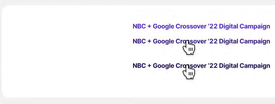
States

- Default, hovered and clicked

Link Text Link Text Link Text

Behaviors

Default, hover and click to navigate



Responsiveness

- Link width should always be dependent on the length of text that comprises the link

Accessibility

Keyboard Navigation

- A link is focused by the Tab key
- A link is activated by the Enter key

Screen Reader Support

Expected to read out:

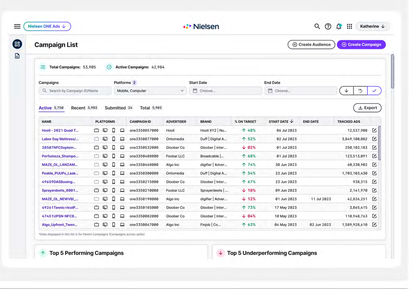
- Type to begin filtering focused search input
- Navigate between options with the Arrows keys
- Select an option with the Enter key

WAI-ARIA Roles

Apply `role="link"`

More accessibility information | example

Common Examples



Other Names

Click, select, hover, hyperlink, url

Related

Buttons, Button Groups, Action Icons

Copyright © 2024 The Nielsen Company (US), LLC. All Rights Reserved.

On this page
When to Use
Code Preview
Properties
Anatomy
Design Principles
Do's and Don'ts
Do's
Don'ts
Variants
Sizes
Types
States
Behaviors
Responsiveness
Accessibility
Keyboard Navigation
Screen Reader Support
WAI-ARIA Roles
Common Examples
Other Names
Related

Learn
Getting Started
What's New
Help & Contributing
Use
Foundations
Components
Accordions
Action Icons
Breadcrumbs
Buttons
Button Groups
Checkboxes
* Cards
Chips
* Context Switchers
* Data Visualizati...
* Date Pickers
Display Icons
* Drag and Drops
* Empty States
Inputs
* Labels
Links

Buttons

Buttons are interaction elements with a hierarchy of styles to guide users to a primary call to action on a page. They allow users to take action by selecting the button with click, touch, or enter.

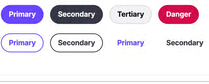
When to Use

Buttons are used for triggering an action or navigating to a different section of an application. Buttons should be used sparingly and should be visually distinct to draw attention.

Common scenarios are:

- Submitting forms
- Triggering actions or events, such as filters or in modals
- Highlighting important calls-to-action in a hierarchy

Code Previews



Properties

aria-label	`string`
autoFocus	`boolean`
children	`Node[]`
className	`string`
data-maf-aria-hidden	`boolean`
data-maf-aria-hidden-exact	`boolean`
data-selector	`string`
data-track-ids	`string`
disabled	`boolean`
externalAs	`string`
fontSize	`"x0" | "x100" | "x200" | "x300" | "x400" | "x500" | "x600" | "x700" | "x800" | "x900" | "x1000"`
fontWeight	`"regular" | "bold" | "semiBold"`
icon	`object`
onBlur	`Event`
onClick	`Event`
onFocus	`Event`
onKeyDown	`Event`
onMouseOver	`Event`
ref	`Event`
role	`string`
roundedCorners	`"left" | "right" | "all" | "none"`
size	`"tiny" | "compact" | "regular" | "jumbo"`
style	`Event`
tabIndex	`number`
type	`string`
variant	`"primary" | "secondary" | "tertiary" | "danger"`
view	`"solid" | "outlined" | "link"`

Anatomy

Design Principles

- **Minimal** - not too many to a page.
- **Discoverable and accessible** - placement in the upper right for left-right languages for all page-, table-, and container-level actions.
- **Used within a hierarchy** - establishing primary, secondary and tertiary actions.
- **Identifiable and understandable** - text and icons (when used) are clear, concise, predictable and contextual.
  - Users must easily understand which interface parts are static and which are clickable through the use of actionable colors, labels and icons (when applicable).

Do's and Don'ts

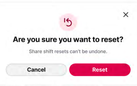
Do's

Hierarchy and Placement

- Use one primary button on a page. In some cases, two may appear (e.g. steppers may have Back/Next buttons).
- Put solid buttons to the right of outlined buttons.
  - If there are several secondary buttons, put them into a button group next to the primary button.
- Use tertiary and transparent buttons sparingly, mainly for lower hierarchy actions or menus or for cancel or close actions in modals.

Colors and Sizes

- Be consistent in button sizes throughout the application.
- Use a solid primary danger button and a solid secondary button in errors or destructive messages



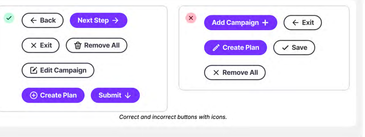
Example of proper use of the danger button in a dialog.

Labels

- Be as short as possible and use simple terms that translate well.
- Have a clear and distinct meaning to reflect the context of what will happen. Avoid generic commands such as "OK".
- Strive for either a single verb, or a verb and a noun. Examples: "View Details".
  - In button groups with action buttons, using only a single noun like "Views", "Options" or "Layout" is acceptable.
  - Using only the noun, like Summary, or Details is acceptable for links or buttons with actions lower in hierarchy.
  - Format button labels in title case.
    - Examples: "Export File", "Add to Cart", "Submit", "Export". If you are unsure of the proper title case usage, consult TitleCaseConverter.com.
- Be specific for the label on the submission button of a stepper, do not just use "Save". Save should be reserved for steppers that allow saving in progress.

Icons

- Icons are **not** required for buttons unless a button will navigate laterally or produce a menu of options.
  - Backward and forward arrows are used when moving back out of an expanded view from a dashboard or in a stepper to move between steps.
- Use icons in buttons sparingly within a page to emphasize primary calls to action, such as add or create, or to differentiate actions in longer button groups.
- Only use icons when they have a strong association with the label text and action and use them consistently for the same thing.



Correct and incorrect buttons with icons.

- Icons are placed before the label in the direction of reading, unless a button opens a menu or indicates a forward navigation, it is after the label in the direction of reading.
- Place menu buttons in button groups on the end.
- Icon-only buttons can be used in the case of space restrictions and only for common actions that are not primary, such as trash, edit, or export.
  - If the label is hidden, a tooltip on hover should be used to show the label.

Examples showing proper use of tooltips with icon-only buttons.

Disabled State

- Buttons are disabled in forms until all required fields are completed, ideally, using inline validation for each field in the form.
- Change the cursor to `cursor:not-allowed` and display a tooltip on hover, tap or click that explains what the user needs to do to enable the button.
  - The tooltip message specify all required elements on the page or stepper.
  - The tooltips must be triggered by the screenreader to make the disabled button accessible (see accessibility below).
- The final step in a stepper can be disabled at the top of all steps following the guidelines above, unless the stepper offers the option to actively save a draft.
- A form submit button will be temporarily disabled to prevent double clicking while the system takes the user to the next page.
- Button groups are not disabled, if they don't apply, they don't appear.

Menu Buttons (see Button Groups)

Don'ts

- Don't put more than three buttons next to each other. Instead, use button groups.
- Don't mix button sizes in a set. Use consistent button sizes throughout your application.
- Don't use buttons to duplicate the functionality of links, accordions, tabs, or tiles.
- Don't override button appearances, types, styles, states or behaviors.
- Don't use cancel as the secondary button in a dialog if the primary action is to cancel to avoid confusion. Instead, use a specific label, like "Cancel Report" and "Close" as the secondary option.

Examples of improper multiple buttons: too many individual buttons, mixed sizes or two primary buttons.

On this page
When to Use
Code Previews
Properties
Anatomy
Design Principles
Do's and Don'ts
Do's
Don'ts
Variants
Size
Hierarchy and Color
Background
Icons
Shape
Behaviors
Responsiveness
Hug Contents
Fill Container
Accessibility
Keyboard Interactions
Screen Reader Support
WAI-ARIA Roles
Common Examples
Other Names