Elements in Websites

ELEMENTS IN WEBSITE

Button

(“The Best Way to indicate a button is to use visual cues”)

Text buttons

Text buttons are text labels that fall outside of a block of text. The text should describe the action that will occur if a user clicks or taps a button. Text buttons have a low level of emphasis and are typically used for less important actions. Because text buttons don’t have a container, they don’t distract from nearby content.

Ghost buttons

Outlined buttons (often called “ghost” buttons) are a step up in complexity and emphasis from a text button in button design. They typically indicate actions that are important but not the primary action on a page. Outlined buttons should be exactly that: an outline with no fill surrounding text that indicates an action.

Image not found or type unknown

A picture containing text, screenshot, businesscard, vector graphicsDescription automatically gener

Raised buttons

Raised (or “contained”) buttons are typically rectangular buttons that “lift” from the surface of the screen via use of a drop shadow. The shadow helps indicate that it is possible to click or press the button. Raised buttons can add dimension to mostly flat layouts, and they highlight functionality on busy, wide, or otherwise congested spaces.

Image not found or type unknown

Graphical user interface, textDescription automatically generated with medium confidence

Toggle buttons

Toggle buttons are typically used in button design for one of two reasons: to group related options or to showcase a selected action or setting. For the former, only one option in a group of toggle buttons can be selected and active at a time. Selecting one option deselects any other. For the latter, the toggle button indicates whether an option is active or inactive.

ChartDescription automatically generated

Image not found or type unknown

Floating action buttons

According to Google, “A floating action button (FAB) performs the primary, or most common, action on a screen. It appears in front of all screen content, typically as a circular shape with an icon in its center.” A FAB should perform a constructive action such as creating a new item or sharing the item on screen.

A picture containing electronicsDescription automatically generated

Image not found or type unknown

UX button best practices

Size

The first element to consider when designing in button design is size. You should consider how large a button is in relation to the other elements on the website page. At the same time, you need to make sure that the buttons you design are large enough for people to interact with.

A good rule of thumb comes to us from the MIT Touch Lab. Studies by the Touch Lab say that making your button a minimum of 10mm x 10mm is a great place to start. Keep in mind that with the rise of responsive web, thinking about how the button will resize and percentage widths of buttons has become more important.

Color

The primary action on a page needs to carry a stronger visual weight and have a distinct contrast from its surroundings. It should be the visually dominant button. For instance, adding one color to a grayscale UI draws the eye simply and effectively.

Graphical user interface, text, applicationDescription automatically generated

Image not found or type unknown

Secondary actions (like ‘Cancel’ or ‘Go Back’) should have the weakest visual attraction because reducing the visual prominence of secondary actions minimizes the risk for potential errors while further directing people toward a successful outcome.

Keep in mind that a button isn’t a one-state object. It’s multi-state. Make sure you consider the hover/tap states and active states of the button. Bear in mind that these states should provide enough contrast for people to clearly identify them as different from the default state.

Shape

As far as shape is concerned, a safe bet for website button design is to make buttons square or square with rounded corners, depending on the style of the site or app. Some research suggests that rounded corners enhance information processing and draw our eyes to the center of element.

One suggestion, if you do choose to deviate from traditional button shapes, is to make sure you conduct usability testing on your designs to ensure that people can easily identify the buttons.

Related: Your team needs to make user research a habit

No matter what shape you choose, be sure to maintain consistency throughout your interface controls so that your user will be able to identify and recognize the appropriate UX elements as buttons.

Placement

Regarding UX button placement, try to use traditional layouts and standard UI patterns as much as possible because conventional placement for buttons improves discoverability. Using a standard layout will help users understand the purpose of each element — even if it’s a button without other strong visual signifiers. Combining a standard layout with clean visual design and ample whitespace makes the layout more understandable.

Microcopy

UX button microcopy is often a call-to-action that tells users what action they will complete if they click the button. Strong CTA microcopy has to catch users’ attention quickly and lead them right to the action.

For a more effective call-to-action, keep the number of words at minimum. A few appropriately chosen words are much more effective than a long descriptive phrase. In addition, using action verbs and phrases like “Add to Cart” or “Submit” in CTA microcopy can help you give strong and direct instructions to your users on what to do next.

Date-Picker

Pickers are used to display past, present, or future dates or times. The kind of date (exact, approximate, memorable) you are requesting from the user will determine which picker is best to use. Each picker’s format can be customized depending on location or need.

When to use

Use date and time pickers when you are asking the user for a time or date, or for scheduling tasks.

Variants

Formatting

Anatomy of date pickers

Anatomy of a simple date input and a single date calendar picker

1. Label: Instructs the user what to do with the control.
2. Date field: A text input field where the user can manually type in the date.
3. Icon: calendar icon indicates the calendar menu is available.
4. Calendar: The menu where a date may be selected.
5. Month and year controls: Allows the user to navigate through past and future time frames.
6. Previous and next month controls: Allows the user to move forward or backward one month at a time.
7. Day: Days in the month.
   Anatomy of a time picker
   1. Label: Instructs the user what to do with the control.
   2. Hour and minute field: A text input field where the user types the hours and minutes of the desired time.
   3. AM/PM selector: A select control that allows the user to choose time period.
   4. Timezone selector: A select control that allows the user to set the associated time zone.

Alignment

By default, the pickers have fixed widths. If you are placing the picker inline with other inputs, such as in form, then the widths can be adjusted to match the other inputs. The picker can either increase or decrease in width as needed. If you adjust the size, be aware that pickers have minimum widths and the date content should never horizontal scroll or overflow.

Alignment example

Image not found or type unknown

The calendar itself will always remain a fixed width and is not adjustable. It should always be aligned to the left edge of its assigned text field.

Alignment example

Image not found or type unknown

Sizes

Date picker comes in three sizes: small (32px), medium (40px), and large (48px).

Size example

Image not found or type unknown

Content

Main elements

Label

- Both date and time pickers must be accompanied with labels.
- The labels should be clear and descriptive.
- Range inputs should be being properly labeled with a start and end label.

Date format

- The date format can be displayed differently depending on your location. For example, some countries use month/day/year while other use day/month/year. We will be using day/month/year format
- When using a simple date input include the date format in parentheses inline with the label or as helper text below the label.
- When using calendar picker the date format will be automated if the user selects from the calendar menu.
- Only including the date format as placeholder text inside the field is problematic because it will disappear from view once the user begins typing.

Image not found or type unknown

Alignment example

Time format

- Both the 12-hour and 24-hour time systems are allowed.
- If using the 12-hour format it must be accompanied by an AM/PM selection.
- Use uppercase letters and no periods for the abbreviations AM and PM.
- Specific times should specify a timezone.

Universal behaviors

The behaviors listed in this section are universal across all of the variants. For behaviors that are unique to each variant, see each of the component variant sections below.

States

Validation

Invalid fields should be clearly marked. In pickers with more than one field, the invalid state should only be set on the individual factor that is triggering the error so the user can clearly understand which to address.

Internationalization

Internationalization, also referred to as globalization, refers to software adapting to different languages, regional peculiarities, and technical requirements of a target locale without additional code changes. This means that if the location is known, then formatting of a date or time can automatically change to the acceptable local format. You should always try to design for internationalization.

Simple date input

The simple date input provides the user with only a text field in which they can manually input a date. It allows dates to be entered without adding unnecessary interactions that come with the calendar menu or a dropdown.

The simple date input can include month/year or month/day/year. The formatting may be localized and rearranged in sequence of appearance.

Simple date input

Image not found or type unknown

When to use

Use for memorable dates

Simple date inputs are typically used when the date is known by the user, such as a date of birth or credit card expiration.

Use for approximate dates

Simple date inputs are best for when asking the user for an approximate date instead of an exact date, especially in regards to past dates. For example, when was asking a user when a purchase was made they will most likely easily recall the month and year (November 2019) versus the specific date (November 22, 2019).

States

The simple date input is a text input and has the same interactive state and behaviors. See the style tab for more details.

Simple date input states

Image not found or type unknown

Calendar pickers

Calendar pickers default to showing today’s date when opened and only one month is shown at a time. Calendar pickers allow users to navigate through months and years, however they work best when used for recent or near future dates. If a user needs to input a far distant or future date consider having the calendar default open to a more convenient day.

Keep in mind that some users may find calendar pickers difficult to use. There should always be a simple way to enter dates in a text field when using calendar pickers.

Use for scheduling

Use a calendar picker when the user needs to know a date’s relation to other days such as the day of the week it falls on or its proximity to today. They are optimal for scheduling tasks.

Variants

Single date picker

In a single date picker a user has the option to either manually input a date in the text field or select one specific date from the menu calendar. It requires a day, month, and year to be selected.

Image not found or type unknown

Single date calendar picker

1. Today’s date
2. Hover
3. Day in month
4. Selected day
5. Day in next/previous month

Date range picker

The date range picker functions much like the single date picker but instead of choosing just one date the user can choose a start and end date. For each date in the range, users have the option to manually enter the date in a text field or select the date in the calendar. Each point requires a day, month, and year to be selected.

Image not found or type unknown

Date range calendar picker

1. Day in month
2. Today
3. Selected start date
4. Day in range
5. End date hover and focus
6. Day in next/previous month

Calendar behaviors

Opening the calendar

The calendar can be opened in two ways:

- Clicking the calendar icon on the far right of the field opens the calendar menu.
- When the text field receives focus the calendar menu also appears and remains open until a date is selected or the focus is removed from the picker.

Selecting a date

A date can be selected by:

- Manually entering a date in the text input field.
- Clicking on a date in the calendar menu.
- Navigating to a date by using the Arrow keys and then pressing Enter.

Next and previous month

A user can navigate between the months in a year by:

- Clicking on chevron icons at the top left and right of the calendar.
- Using the Arrow keys to move through the into the next or previous month.

Selecting a year

By default the current date and year appears in the calendar. To navigate to another year the user can do one of the following:

- Manually typing the year in the date text field.
- Clicking the up and down arrows that appear when you focus or hover on the year input in the calendar.
- Selecting then typing into the year input.

Closing the calendar

The calendar can be closed in one of the following ways:

- Selecting a single date or the end date in a range. This will automatically close it.
- Clicking anywhere outside of the calendar menu.
- Removing focus from the picker.
- Pressing Esc.

Selecting a range

There are several ways in which a range can be selected:

- Manually type the start and end dates in the text field.
- Once the calendar is open the first date you click becomes the start date and second date you click becomes the end date.
- Navigating to the start date by using the Arrow keys and pressing Enter. Then continue using the Arrow keys to navigate to a second date and press Enter again.

Min and max dates

In order to constrain the possible selectable dates in a calendar, a min and max date may be set. Once set, only the dates that fall within the min/max range will be selectable with the dates outside of the range being disabled.

Use min and max dates to help prevent user error. If a user cannot select dates in the past when scheduling, then set a min date to today.

Min and max date example

Image not found or type unknown

The dates before today are out of range and disabled.

Time pickers

Time pickers provide the user with a text field in which they can input the hour and minutes. Additionally, they can be accompanied by an AM/PM and a time zone control, both styled as selects.

The time field format should include the hour and minutes, for example 11:30. It may be localized accommodate the 12-hour or 24-hour format.

Use for scheduling

Use the time picker when a specific time needs to be scheduled, such as planning a meeting time.

Time picker example

Image not found or type unknown

Modifiers

Light variant

Use the light prop modifier when placing date pickers on alternate backgrounds. The light prop will change the background color token of the inputs from field-01 to field-02. This is frequently used in form modals or when on a tile.

Light prop example

Image not found or type unknown

Both inputs are displayed in the white theme but the one on the right is using the light prop on the alternate UI background.

Forms

The Components Of Forms

The typical form has following five components:

- Structure. Order of fields and logical connections between individual fields.
- Input fields. Different types of input fields including text fields, password fields, check boxes, radio buttons, sliders.
- Field labels. Labels describe the meaning of input fields.
- Action buttons. When user presses the button, the action is performed (such as submitting the data).
- Feedback. Visual or (and) audio feedback that helps users understand the result of operation. Most apps or sites use messages as a form of visual feedback. Messages notify the user about result, they can be positive (indicating that the form was submitted successfully) or negative (indicating that the data provided by a user required modification).

Forms could also have following components:

- Assistance. Any help that explains how to fill out the form.
- Validation. Automatic check that ensures that user's data is valid.

Form Structure

A form is a conversation between a user and a system. The conversation should be clear and logical.

Only Ask What's Required

Make sure you only ask what you really need. Every extra field you add to a form will affect its conversion rate. Typically, the more data you ask your users to provide the less chances that users will do it. That's why you should always question why and how the information you request from your users is being used.

Order the Form Logically

Details in a the form should be asked logically from a user’s perspective, not the application logic. For example, when you ask billing details, it’s unusual to ask for the address before the full name.

Group Related Information

You should group related information in logical blocks or sets. The flow from one set of questions to the next will better resemble a conversation. Grouping related fields together also helps users make sense of the information that they must fill in. Below is an example for Contact Information.

Image not found or type unknown

Graphical user interface, applicationDescription automatically generated

One Column vs. Multiple Columns

Forms should never consist of more than one column. One of the problems with form fields in multiple columns is that your users are likely to interpret the fields inconsistently. The user will scan the form in Z patterns, and this will slow the speed of comprehension and puzzling the clear path to completion. But if a form is in a single column, the path to completion is a straight line down the page.

Image not found or type unknown

A picture containing text, antennaDescription automatically generated

Input Fields

Input fields are what allow your users to fill in your form. Depending on what information you ask, there are various types of fields — text fields, password fields, dropdowns, check boxes, radio buttons, datepickers and others.

Number of Fields

Try to minimize the number of fields as much as possible. This makes your form less loaded, especially when you request a lot of information from your users.

Graphical user interface, text, applicationDescription automatically generated

Image not found or type unknown

Mandatory vs Optional

Try to avoid optional fields in forms. But if you use them, you should at least clearly distinguish which input fields cannot be left blank by the user. The convention is to use an asterisk (*) or ‘optional’ (which is a preferable for long forms with multiple required fields).

TableDescription automatically generated

Image not found or type unknown

Setting Default Values

You should avoid having a static default unless you believe a large portion of your user’s (e.g. 90%) will select that value. Particularly if it’s a required field. Why? Because you’re likely to introduce errors because people scan forms quickly online — don’t assume they will take the time to parse through all the choices. They simply may skip fields that already has a value.

But smart defaults can make the user’s completion of the form faster and more accurate. For example, pre-select the user’s country based on their geo location data. But still you should use these with caution, because users tend to leave pre-selected fields as they are.

Graphical user interface, applicationDescription automatically generated

Image not found or type unknown

Desktop-only: Make Form Keyboard-friendly

Users should be able to trigger and edit every field using only the keyboard. Power users, who tend to use keyboards heavily, should be able to navigate the form using Tab and make necessary edits, all without lifting their fingers off the keyboard.

Desktop-only: Autofocus for Input Field

Autofocusing a field gives the users an indication and a starting point to quickly begin to fill out the form. You should provide a clear visual indication that the focus has moved there —such as change color, fade in a box. Amazon registration form has both autofocus and visual notification for the user.

Image not found or type unknown

Graphical user interface, applicationDescription automatically generated

Mobile-only: Match the Keyboard With the Required Text Inputs

Mobile app users appreciate apps that provide an appropriate keyboard for text entry (i.e. numeric keyboard for a field that requires numbers). Ensure that this behavior is implemented consistently throughout the app rather than only for certain tasks but not others.

Image not found or type unknown
Graphical user interface, applicationDescription automatically generated

Labels

Clear label text is one of the primary ways to make UIs more accessible. Labels tell the user the purpose of the field and they should remain visbile even after completing the field.

Number of Words

Labels are not help texts. You should use succinct, short and descriptive labels (a word or two) so users can quickly scan your form. Previous version of the Amazon registration form contained a lot of extra words which resulted in slow completion rates.

Image not found or type unknown
Graphical user interface, applicationDescription automatically generated

Current version is much better and has short labels.

Image not found or type unknown
Graphical user interface, applicationDescription automatically generated

Sentence Case vs Title Case

Should it be “Full Name” or “Full name”? Sentence case is slightly easier (and thus faster) for reading than title case. But one thing for sure — you should never use all caps, or else the form would be difficult to read and much harder to quickly scan, as there are no differences in character height any more.

Image not found or type unknown
Graphical user interface, applicationDescription automatically generated

“All Caps” labels are very hard to read.

Alignment of Labels: Left vs Right Aligned vs Top

Matteo Penzo’s 2006 article on label placement implies that forms are completed faster if the labels are on top of the fields. They are good if you want users to eye-scan the form as fast as possible.

DiagramDescription automatically generated

Image not found or type unknown

Left-aligned labels, right-aligned labels and top labels. Image credits: uxmatters

However further research mentioned found no difference between labels above the fields and right-aligned labels.

Top aligned labels. The biggest advantage to top-aligned labels — they make it easier for different sized labels and localized versions to fit easier within the UI (this is especially good for mobile screens with a limited estate).

Left-aligned labels. The biggest disadvantage to left-aligned labels is the slowest completion times. This is likely because of the visual distance between the label and input field. The shorter the label, the further away it is from the input. But slow completion rates aren’t always a bad thing, especially if the form requires important data. If you are asking for things like Driver’s License or Social Security Number, you may implicitly want to slow users down a bit and make sure they enter things correctly.

Graphical user interfaceDescription automatically generated with medium confidence

Image not found or type unknown

Right-aligned labels. The big advantage to right-aligned labels is the strong visual connection between label and input. Because items near each other appear related. This principle of placing related items closer to each other isn’t new; it’s actually the Law of Proximity from Gestalt psychology. For shorter forms, right-aligned labels can have great completion times. Right-aligned labels disadvantage comes from comfortability. Such form will lack that hard left edge, which makes it less comfortable to look at and harder to read.

Chart, box and whisker chartDescription automatically generated

Image not found or type unknown

Takeaway: If you want users to scan out a form fast, put your labels above each field. This layout is easier to scan as the eyes move straight down the page. However if you want your users to read carefully, put the labels to the left of the fields. This layout is read in a slower down and right (Z shape) motion.

Inline Labels (Placeholder Text)

Placeholder text works great for a simple username and password form.

Graphical user interface, text, application, chat or text messageDescription automatically generatec

Image not found or type unknown

But it’s a poor replacement for a visual label for forms that contain a lot of fields because inline labels create usability issues. Once the user clicks on the text box, the label disappears and thus the user cannot double check that what he/she has written is indeed what was meant to be written. Another thing is that when users see something written inside a text box, they may assume that it has already been prefilled in and may hence ignore it.

Graphical user interface, text, application, emailDescription automatically generated

Image not found or type unknown

Good solution for the placeholder text is a floating label. The placeholder text is showing by default, but once an input field is tapped and text is entered the placeholder text fades out and a top aligned label animates in.

Image not found or type unknown

Graphical user interfaceDescription automatically generated

Action Buttons

When clicked, these buttons trigger an action such as submitting the form.

Do not disable Submit button

When designers disable the Submit button, they make their users scan the entire form to find the problem that doesn’t allow them to submit the data. When the Submit button is not disabled, users can click the button and see an error message that directs them towards that place.

Primary vs Secondary Actions

When the form offers primary and secondary actions that do not have a visual distinction between them, users can easily make incorrect decisions. It’s essential to reduce the visual prominence of secondary actions to minimize the risk of potential errors. Visual weight will direct users towards the ‘right’ (most expected) action.

Image not found or type unknown

Graphical user interface, text, application, email, websiteDescription automatically generated

Button Location

Complex forms usually need a back button. If such button is located right below the input fields (like on the first shot) users can accidentally click it. As back button is a secondary action it should be placed out of sight (second form has right location for buttons).

Graphical user interface, application, WordDescription automatically generated

Image not found or type unknown

Left: Wrong back button location. Right: Back button is almost invisible for users (both because location and color contrast with proceed).

Naming Conventions

Avoid generic words such as “Submit” for actions, because they give the impression that the form itself is generic. Instead state what actions the buttons do when clicked, such as ‘Create my FREE account’ or ‘Send me weekly offers’.

Graphical user interface, textDescription automatically generated

Image not found or type unknown

Multiple Action Buttons

Avoid offering more than two action buttons because it might distract users from the goal (submitting the form).

The ‘Reset’ Button Is Pure Evil

Reset button allows users to erase all data in a form. When users don’t know how this button works, click on it and see that all their effort on filling out a form went to drain, it creates a lot of frustration. Thus, Don’t use a ‘reset’ button. This button almost never helps users, but often hurts them.

Graphical user interfaceDescription automatically generated

Image not found or type unknown

Visual Appearance

Design action buttons to look like buttons. Style them in a way that they give users a visual hint on what the element does (for example, it’s nice when a button lifts but it indicates that it is possible to click). For more information about the button you can read article

Graphical user interfaceDescription automatically generated

Image not found or type unknown

Visual Feedback

At the time when the user provided all required data and clicked ‘Submit’ button, it’s extremely important to indicate that the form receives the user feedback. The visual feedback acts as an acknowledgment and prevents the user from clicking the same button again.

Conclusion

Users can be hesitant to fill out forms, so you should make this process as easy as possible. Changes mentioned in this article can significantly increase form usability. Remember that usability testing is simply indispensable in form design. Test early, test often and improve your form design based on the results of testing.

Loading Symbol

Overview

Loading spinners are used when retrieving data or performing slow computations. They notify to the user that their request is being processed. Although they do not provide details about what is occurring on the back-end, they reassure the user that their action is being processed.

Use a loading spinner whenever the wait time is anticipated to be longer than three seconds.

Formatting

Loading spinners may be scaled down if the loading experience is contextual to a certain item on the page.

Small loader

Image not found or type unknown

Placement

The large loading spinner should appear centered over a page or content that it is loading. Please note that the top-level navigation is not included in the page loading overlay.

Large spinner in context example

Image not found or type unknown

Example of a large loading spinner in product context

Disabled State

A disabled state is applied to a component when the user is not allowed to interact with the component due to either permissions, dependencies, or pre-requisites. Disabled states completely remove the interactive function of a component.

Disabled variations

Variation	Description
Default disabled	Cannot be clicked, selected or interacted with. It is not read by a screen reader and takes on the default disabled visual style.
Read-only	The user cannot interact with it but content is still readable and accessible to a screen reader. The visual style should contain no interactive indicators such as $interactive-01, hover states, or text embellishments (i.e., underlines).
Hidden	The component is completely hidden from view. The user does not know the option is there.

Default disabled

A default disabled state is used when a component is temporarily disabled due to dependencies (when one piece of software relies on another one) or pre-requisites. This scenario is a temporary state change that is most commonly triggered by a user’s action or inaction. Once the dependencies have been resolved and/or the pre-requisites have been fulfilled, the default disabled component returns to its enabled state. In a temporarily disabled scenario the component should never fully disappear from the user’s view.

Image not found or type unknown

Default disabled example on the right

Style

Default disabled states are most commonly styled by a decrease in opacity with no hover state change and not-allowed cursor applied. Refer to each individual component for the accurate disabled state.

Attribute	Default disabled style
Component	50% opacity
Text	25% opacity
Icons	50% opacity
Hover	None
Cursor	not-allowed

Image not found or type unknown

Default disabled style examples

Additional warning

An inline warning notification can be shown in cases where a temporarily disabled item affects multiple items or the primary action of the flow. The notification should describe how the user can enable or re-enable the disabled component.

Image not found or type unknown

Additional warning with default disabled example

Read-only

In scenarios where the content of a disabled component or element is still relevant to the user or important to task completion, then the read-only variation is used. This allows the user to read the information but not interact with or change it. Read-only content should always be accessible to a screen reader.

Style

The visual style of the read-only states vary by component but should never contain any interactive indicators such as $brand-01 color usage, hover states, or text embellishments (i.e., underlines).

Hidden

The hidden disabled variation is used when something or someone does not have permission to view, interact with, or take action on an element of the UI. This variation completely hides the component, page, action, etc. from the user's interface. The only way to enable the hidden element and have it resurfaced on the UI is to change the assigned permission.

For example, when a user is the organization owner they are allowed to add members to the organization. Any users that are not an organization owner would not be shown the "Add member" button on a team directory page. Once the user is made an organization owner, then and only then will the button be visible.

Image not found or type unknown

Example of hidden disabled content on the right

Variant	Purpose
Simple date input	Use if the date can be remembered by the user easily, such as a date of birth, and they don't need a calendar to anticipate the dates. It consists only of input fields.
Calendar pickers	Use a calendar picker when the user needs to know a date's relationship to other days or when a date could be variable. It allows the user to view and pick dates from a calendar widget or manually type the date in the text field.
Time picker	Use when asking the user to input a specific time.

Revision #1
Created 29 March 2023 09:29:45 by Dipti Mulay
Updated 29 March 2023 09:38:13 by Dipti Mulay