DIGITAL INSIGHT
an NCR company

CSS3 Pattern Library

⚛ React Pattern Library

Latest Release (1.5) ⌄

Getting Started

Style ⌄

Layout ⌄

UI Elements ⌃

   Buttons

   Drop Down

   Toggle

   Radio Buttons

   Check Boxes

   Modals

   Tool Tips

   Input Fields

Patterns ⌄

Resources ⌄

Native mobile guide

SSO widgets/pages

Contribution

FAQ / Contact

Input Fields

Design guideline	Code view

Visual Specifications

There are two input field sizes default and small (font size stays the same for both sizes).

Default is 42px text box height. Small is 35px text box height and is used when creating a complex form with space constrictions.

Default

Placeholder text — 42px
10px

Small

Placeholder text — 35px

Inner Shadow CSS: 1px 1px 1px 0 rgba(0,0,0,0.45);

Normal

Placeholder text remains until user types the first character.

Label
18px

Placeholder

Label text: Arial 16px #333333
Input field text: Arial 16px #666666
Input field Fill: #FFFFFF
Stroke: #CCCCCC 1 px

Focus

Text input

Input field text: Arial 16px #333333
Stroke: #666666 2px

Error

Text input

⚠ 7px | 11px Error message.
7px

Error text: Arial 14px #C90202
Stroke: #C90202 2px

Text input

⚠ Warning message.

Warning text: Arial 14px #333333

Ajax save when field goes out of focus

The loading icon disappears after the input is saved.

Text input

◌ Saving

Saving text: Arial 14px #333333

Success message shows up and fades away after a delay.

Text input

✔ Success

Success text: Arial 14px #2E6F03

Input field with drop down

Placeholder ▼

Drop down fill: #EDEDED
Drop down Stroke: #CCCCCC 1px

Interaction Details

Label placement

For reduced completion time the label is placed above the field.

Label

Placeholder

Error placement

Inline error messages should highlight the field in red, and give the error message below the field. If there is a need to save vertical space then the message can be right aligned to the field.

Input value

⚠ Error message goes here.

Form completion best practices

- Fewest input fields possible to achieve task

- Smart defaults

- Inline validation

- Forgiving inputs (i.e. multiple ways people enter phone numbers)

- When possible, use field length as an affordance

- When possible, validate the field on "blur" when user clicks out of the field

Accessibility

1. Indicate: focus; the easiest way to achieve this is to not remove default styling for the outline property.

2. Close buttons: images that function as a button (such as an image of an "X" to represent "Close") include a text
   Style hover and focus the same way, so mouse and keyboard-only users receive consistent feedback.

3. Identify errors with more than color alone (ex. also use an icon and text).

4. All images must have an alt attribute (even if the function of the image is purely decorative and alt=""). When an image has meaning to a sighted user, convey this with the alt attribute if possible. Avoid images that contain text; if they do, the alt value must contain the same string of text.4. Style hover and focus the same way, so mouse and keyboard-only users receive consistent feedback.

5. Validation: required elements get the required attribute and aria-required="true"; items that fail validation get aria-invalid="true". Error feedback within a form gets the alert role to provide the information immediately to the user. A role of alert is an assertive live region, meaning the message will be delivered to the user immediately. When identifying failed validation, consider placing :focus on the item that needs correction. One of the biggest failures with forms is not sending keyboard focus to the field with an error or the list of error messages--a user may assume the form is broken; however, using the alert role delivers the information immediately to assistive technology, so it's not necessary to bring :focus to the alert node.