

5.6 Styling forms

Styling forms with CSS

Web forms are an important part of many websites. A usable form allows the user to quickly and painlessly enter data. Forms require CSS formatting to improve usability.

Figure 5.6.1: HTML form without CSS styling and an improved form with styling.

The figure shows two versions of a registration form side-by-side. The left version is unstyled, with labels ('Name', 'Email', 'Service') positioned above their respective input fields ('text' and 'select'). The right version is styled with CSS, where the labels are aligned to the right of the inputs, and the 'Register' button is a prominent blue rectangle. Below the forms is a horizontal line and a 'Feedback?' link.

PARTICIPATION ACTIVITY

5.6.1: Create a styled form.



The webpage below displays a simple web form with little styling. Add the following CSS to create a more usable web form:

1. Add a `label` selector that makes all labels have the same width and margin.
Since a label is an inline element, the label's width cannot be changed without making a label an inline-block. Also, right-align the label text to improve the reader's ability to mentally link the label to the input field.

```
label {  
    width: 50px;  
    display: inline-block;  
    text-align: right;  
    margin-right: 8px;  
}
```

Render the webpage and observe the labels are equal length and right aligned.

2. Add a selector that gives the text input, email input, and drop-down menu the same consistent width with some padding to increase the size of the inputs. Also, change the border color and radius to give the inputs a softer look.

```
input[type=text], input[type=email], select {  
    box-sizing: border-box;  
    width: 250px;  
    padding: 10px;  
    border: 1px solid #aaa;  
    border-radius: 4px;  
}
```

Render the webpage and observe the inputs are equal size and are spaced out.

3. Add styling to the submit button to create a blue button that looks less like a traditional browser button. Also, change the default mouse cursor to a pointer icon to give the user a visual cue that the button is pressable.

```
input[type=submit] {  
    width: 200px;  
    background-color: #09f;  
    color: white;  
    padding: 15px;  
    margin: 10px 0;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;  
}
```

Render the webpage and observe the Register button is blue and much larger. Moving the mouse over the button changes the pointer icon.

4. To give the user another visual cue that the button is clickable, darken the button color when the mouse hovers over the button.

```
input[type=submit]:hover {  
    background-color: #07d;  
}
```

Render the webpage and verify the Register button's color gets darker when the mouse hovers on the button.

HTML CSS

```
1 form {  
2   font: 13px Arial;  
3   background-color: #eee;  
4   padding: 10px;  
5 }  
6
```

Render webpageReset code**Your webpage**

Name

Email

Service

Expected webpage

Name

Email

Service

▼ View solution

 Explain

--- START FILE: HTML ---

```
<form action="https://wp.zybooks.com/form-viewer.php"  
target="_blank" method="GET">  
  <p>  
    <label for="name">Name</label>  
    <input type="text" id="name" name="fullname">  
  </p>  
  <p>  
    <label for="email">Email</label>
```

```
<input type="email" id="email" name="email">
</p>
<p>
    <label for="service">Service</label>
    <select id="service" name="service">
        <option>Basic</option>
        <option>Prime</option>
        <option>Deluxe</option>
    </select>
</p>

<input type="submit" value="Register">
</form>
```

--- END FILE: HTML ---

--- START FILE: CSS ---

```
form {
    font: 13px Arial;
    background-color: #eee;
    padding: 10px;
}

label {
    width: 50px;
    display: inline-block;
    text-align: right;
    margin-right: 8px;
}

input[type=text], input[type=email], select {
    box-sizing: border-box;
    width: 250px;
    padding: 10px;
    border: 1px solid #aaa;
    border-radius: 4px;
}

input[type=submit] {
    width: 200px;
    background-color: #09f;
    color: white;
    padding: 15px;
    margin: 10px 0;
```

```
border: none;  
border-radius: 4px;  
cursor: pointer;  
  
}  
  
input[type=submit]:hover {  
    background-color: #07d;  
}
```

--- END FILE: CSS ---

[Feedback?](#)**PARTICIPATION ACTIVITY**

5.6.2: Form styles.



- 1) In the example above, the label's width could not be changed until which CSS property/value was set?

Correct

A `<label>` is an inline element without a width, and the `inline-block` value makes an element inline with a width and height.



- `display: inline-block`
- `display: block`
- `display: none`

- 2) What CSS selector selects only text inputs?

- `input`
- `input[type=text]`
- `input[type=text], select`

Correct

The attribute selector selects only `<input type="text">` elements.



- 3) In the example above, what happens to the text and email inputs if `box-sizing: border-box;` is removed from the `input[type=text],`

Correct

By default, the text and email inputs include the border and padding outside the specified width and height values. Thus, the text boxes and drop-down menu are wider than 250px. Adding `box-sizing: border-box;` renders the border and padding *inside* the



`input[type=email],
select` selector's rule?

- Inputs are no longer visible
- Inputs are no longer editable
- Inputs become wider

4) In the example above, what visual cues help the user to know that the blue rectangle with "Register" in the middle is a button that can be pressed?

- The button has
 - rounded corners.
- The pointer icon appears when hovering over the button.
- The pointer icon appears, and the button color changes when hovering over the button.

Correct

The pointer icon and color change lets the user know the area is a clickable button.



[Feedback?](#)

Paragraphs vs. divs

The W3C recommends putting form elements in `<p>` tags, but many leading web frameworks like Bootstrap use `<div>` tags instead. Either container is acceptable, and this material uses both styles.

```
<!-- Example div container -->
<div>
  <label for="nickname">Nickname?</label>
  <input type="text" id="nickname" name="nickname">
</div>

<!-- Example p container -->
<p>
  <label for="nickname">Nickname?</label>
  <input type="text" id="nickname" name="nickname">
</p>
```

Proper use of field labels

Form field labels should be placed uniformly in the same location on a website's web forms. The best places for labels are immediately above or to the left of an input field. Some developers use only the `placeholder` HTML attribute in place of labels to save screen space and reduce clutter, especially on mobile devices. However, usability experts warn that placeholders used as labels can create a number of problems for users and should be avoided.



Improving text input fields

An input field can be further improved. Changing an input's border color or background color focuses the user's attention to the input. Ex: Changing the input border color to red may indicate an error with the input. Adding a commonly recognized icon to an input field can improve the user's ability to recognize the purpose of the input. Ex: Adding a search icon to a search input.

PARTICIPATION ACTIVITY

5.6.3: Augmenting an input.



The webpage below displays a web form on the left and a search box on the right. A partial email address is entered in the email input field. Add the following CSS:

1. Add an `:invalid` selector at the end of the CSS stylesheet that sets an invalid input's border to red to indicate an error:

```
input:invalid {
    border: 2px solid red;
}
```

Render the webpage and observe the email input now has a red border.

2. Add a `:focus` selector that applies styles to an input that has the focus, and change the background color to light blue:

```
input[type=text]:focus, input[type=email]:focus {
    background-color: lightblue;
}
```

Render the webpage and observe that the name and email inputs become light blue when the inputs have the focus. The browser automatically places a border around an input that has the focus.

3. Add a search icon to the search input by adding a `background-image` that is positioned with `background-position`. Set `background-repeat: no-repeat` so the background image only displays once:

```
input[type=search] {
    float: right;
    background-image:
url("https://resources.zybooks.com/WebProgramming/searchiconv1.png");
    background-position: 5px 5px;
    background-repeat: no-repeat;
    padding-left: 40px;
}
```

Render the webpage and observe the search icon in the search input.

HTML CSS

```
8   <p>
9     <label for="email">Email</label>
10    <input type="email" id="email" name="email" value="dara"
11  </p>
12  <p>
13    <label for="service">Service</label>
14    <select id="service" name="service">
15      <option>Basic</option>
16      <option>Prime</option>
17      <option>Deluxe</option>
18    </select>
19  </p>
20
21  <input type="submit" value="Register">
22 </form>
23
```

Render webpage

Reset code

Your webpage

Name

Email

Service

▼ View solution

 Explain

--- START FILE: CSS ---

```
label {  
    display: block;  
}  
  
input[type=text], input[type=email], input[type=search],  
select {  
    box-sizing: border-box;  
    width: 250px;  
    padding: 8px;  
    margin-top: 4px;  
    border: 1px solid #aaa;  
    border-radius: 4px;  
}  
  
input[type=search] {  
    float: right;  
}  
  
input[type=submit] {  
    width: 250px;  
    color: white;  
    background-color: #09f;  
    color: white;  
    padding: 15px;  
}
```

```
margin: 10px 0;  
border: none;  
border-radius: 4px;  
cursor: pointer;  
}  
  
input[type=submit]:hover {  
    background-color: #07d;  
}  
  
form {  
    font: 13px Arial;  
    background-color: #eee;  
    padding: 10px;  
    width: 250px;  
}  
  
input:invalid {  
    border: 2px solid red;  
}  
  
input[type=text]:focus, input[type=email]:focus {  
    background-color: lightblue;  
}  
  
input[type=search] {  
    float: right;  
    background-image:  
url("https://resources.zybooks.com/WebProgramming/searchiconv  
    background-position: 5px 5px;  
    background-repeat: no-repeat;  
    padding-left: 40px;  
}
```

--- END FILE: CSS ---

[Feedback?](#)

PARTICIPATION
ACTIVITY

5.6.4: Augmented inputs.



- 1) Many browsers add a border around an input when the input has the focus.

Correct



The border appears outside the CSS border and helps the user know the input has the focus. The border can be

True False

- 2) The `:focus` selector normally selects more than one element at a time.

 True False

- 3) Changing the email address to `dara@gmail.com` removes the red border.

 True False**Correct**

`:focus` selects the element that has the focus, and only one element may have the focus at a time.

**Correct**

The `:invalid` pseudo-class only applies when the entered text does not appear to be a valid email address.

**Feedback?**

Styling radio buttons and checkboxes

Radio buttons and checkboxes use the styling properties supplied by the browser and may differ between browsers. Styling radio buttons and checkboxes requires hiding the default widget, then displaying a custom widget before each label that changes appearance when checked or focused.

The CSS property **appearance** is used to control a widget's appearance based on the operating system's theme. Setting **appearance** to **none** hides the widget.

To display a custom radio button or checkbox, the `::before` pseudo-element selector and **content** property are used to insert content before the label's content that looks like a radio button or checkbox.

PARTICIPATION ACTIVITY

5.6.5: Styling radio buttons.

**Start**

2x speed

```
input[type=radio] + label {
  cursor: pointer;
}
input[type=radio] {
  appearance: none;
  margin: 0;
}
input[type=radio] + label::before {
  content: "\00a0"; /* Space */
  border-radius: 10px;
  display: inline-block;
```

```
<fieldset>
  <legend>Size:</legend>
  <div>
    <input id="small" type="radio" name="size">
    <label for="small">Small</label>
  </div>
  <div>
    <input id="medium" type="radio" name="size">
    <label for="medium">Medium</label>
  </div>
  <div>
```

```

width: 1em; height: 1em;
text-align: center;
line-height: 1em;
font-size: 0.8em;
margin-right: 10px;
background-color: #aaa;
border: 0.125em solid white;
}


```

```

<input id="large" type="radio" name="size">
<label for="large">Large</label>
</div>
</fieldset>

```

Size:

- Small
- Medium
- Large

Captions ^

1. The default radio buttons are plain looking and cannot be styled.
2. The + selector selects <label> elements that are immediately after <input type="radio"> elements.
3. Changing the label's cursor to a pointer helps the user know the label is clickable.
4. Changing setting appearance to none hides the radio buttons, but screen readers still "see" the radio buttons.
5. ::before inserts the content \00a0, a non-breaking space, before the radio button label's content.
6. Custom radio buttons are created by displaying a single empty space within a circular border.
7. When a radio button is checked, a white bullet with green background replaces the previous content (the space).
8. When a radio button has the focus, a gray shadow displays around the content before the label.

[Feedback?](#)

PARTICIPATION
ACTIVITY

5.6.6: Style the checkboxes.



The webpage below displays three styled radio buttons followed by four unstyled checkboxes. Add the following CSS to style the checkboxes:

1. Add a rule that adds a pointer cursor to the checkboxes' labels:

```

input[type=checkbox] + label {
  cursor: pointer;
  font-size: 1.3em;
}

```

Render the webpage and verify that the cursor changes to a pointer when mousing over the checkbox labels.

2. Add a rule that hides the checkboxes:

```
input[type=checkbox] {
    appearance: none;
    margin: 0;
}
```

Render the webpage and verify the default checkboxes are no longer visible.

3. Add a rule that adds a gray square in front of the checkbox labels:

```
input[type=checkbox] + label::before {
    content: "\00a0"; /* Non-breaking space */
    display: inline-block;
    width: 16px;
    height: 16px;
    text-align: center;
    line-height: 1em;
    font-size: 0.8em;
    margin-right: 10px;
    border: 0.125em solid white; /* Push focus shadow away */
    background-color: #aaa;
}
```

Render the webpage and verify the checkboxes appear as squares.

4. Add a rule to display a white checkmark with green background when a checkbox is checked:

```
input[type=checkbox]:checked + label::before {
    content: "\2713"; /* Checkmark */
    color: white;
    background: green;
}
```

Render the webpage and verify that clicking on a checkbox displays a checkmark in the box.

5. Add a rule to display a gray border around the checkbox that has the focus:

```
input[type=checkbox]:focus + label::before {
    box-shadow: 0 0 0 1px gray;
}
```

Render the webpage and verify that the last clicked checkbox has a gray border.

[HTML](#) [CSS](#)

```
21      <label for="raspberry">Raspberry</label>
22    </div>
23    <div>
24      <input id="grape" type="checkbox" name="flavor" value="g"
25      <label for="grape">Grape</label>
26    </div>
27    <div>
28      <input id="cherry" type="checkbox" name="flavor" value="c"
29      <label for="cherry">Cherry</label>
30    </div>
31    <div>
32      <input id="root-beer" type="checkbox" name="flavor" value="r"
33      <label for="root-beer">Root Beer</label>
34    </div>
35  </fieldset>
36
```

Render webpageReset code**Your webpage**

Size:

Small
Medium
Large

Flavors:

- Raspberry
- Grape
- Cherry
- Root Beer

Expected webpage

Size:

Small
Medium
Large

Flavors:

Raspberry
Grape
Cherry
Root Beer

▼ View solution Explain

--- START FILE: HTML ---

```
<fieldset>
  <legend>Size:</legend>
  <div>
    <input id="small" type="radio" name="size"
value="small">
    <label for="small">Small</label>
  </div>
  <div>
```

```
<input id="medium" type="radio" name="size"
value="medium">
    <label for="medium">Medium</label>
</div>
<div>
    <input id="large" type="radio" name="size"
value="large">
        <label for="large">Large</label>
    </div>
</fieldset>

<fieldset>
    <legend>Flavors:</legend>
    <div>
        <input id="raspberry" type="checkbox" name="flavor"
value="chocolate">
            <label for="raspberry">Raspberry</label>
        </div>
        <div>
            <input id="grape" type="checkbox" name="flavor"
value="grape">
                <label for="grape">Grape</label>
            </div>
            <div>
                <input id="cherry" type="checkbox" name="flavor"
value="cherry">
                    <label for="cherry">Cherry</label>
                </div>
                <div>
                    <input id="root-beer" type="checkbox" name="flavor"
value="rootbeer">
                        <label for="root-beer">Root Beer</label>
                    </div>
    </div>
</fieldset>
```

--- END FILE: HTML ---

--- START FILE: CSS ---

```
body {
    font-family: Arial;
}

input[type=checkbox] + label {
```

```
cursor: pointer;
font-size: 1.3em;
}

input[type=checkbox] {
    appearance: none;
    margin: 0;
}

input[type=checkbox] + label::before {
    content: "\00a0"; /* Non-breaking space */
    display: inline-block;
    width: 16px;
    height: 16px;
    text-align: center;
    line-height: 1em;
    font-size: 0.8em;
    margin-right: 10px;
    border: 0.125em solid white; /* Push focus shadow away
*/
    background-color: #aaa;
}

input[type=checkbox]:checked + label::before {
    content: "\2713"; /* Checkmark */
    color: white;
    background: green;
}

input[type=checkbox]:focus + label::before {
    box-shadow: 0 0 0 1px gray;
}

input[type=radio] + label {
    cursor: pointer;
    font-size: 1.3em;
}

input[type=radio] {
    appearance: none;
    margin: 0;
}

input[type=radio] + label::before {
    content: "\00a0"; /* Non-breaking space */
    border-radius: 10px;
```

```
display: inline-block;
width: 16px;
height: 16px;
text-align: center;
line-height: 1em;
font-size: 0.8em;
margin-right: 10px;
border: 0.125em solid white; /* Push focus shadow away
*/
background-color: #aaa;
}

input[type=radio]:checked + label::before {
content: "\2022";
color: #fff;
background: green;
}

input[type=radio]:focus + label::before {
box-shadow: 0 0 0 1px gray;
}

fieldset {
border-radius: 10px;
margin-bottom: 15px;
}
```

--- END FILE: CSS ---

[Feedback?](#)

PARTICIPATION
ACTIVITY

5.6.7: Styling radio buttons and checkboxes.



- 1) Which label is selected by `input[type=radio] + label`?

- `<input type="checkbox">`
`<label></label>`
- `<input type="radio">`
`<label></label>`
- `<label></label>`
- `<input type="radio">`

Correct

The `+ selector` selects labels immediately after radio buttons.

- 2) In the exercise above, the default radio buttons and checkboxes could have been hidden using the CSS `display:none`. Why is hiding the radio buttons and checkboxes with `display:none` not a good idea?

- Cross-browser support issues.
- Screen readers will think the
- radio buttons or checkboxes are not visible.
- The `display` property should be avoided.

Correct

Developers must take care to create web forms that are accessible to those using assistive technologies.

- 3) Which CSS selector selects only checkboxes that are checked?

- `input:checked`
- `input[type=checkbox]:focus`
- `input[type=checkbox]:checked`

Correct

The `:checked` selector only selects checked items.

- 4) What does the DOM look like after the CSS and HTML below are rendered?

Correct

```
span::before {
    content: "Before";
}
```

```
<span>Test</span>
```

- BeforeTest
- BeforeTest
- TestBefore

The ::before selector inserts the content immediately before the selected element's content.

[Feedback?](#)
CHALLENGE ACTIVITY
5.6.1: Styling forms.


530096.4000608.qx3zqy7

[Start](#)


1



2



3



4

For the <label> tag, set display to inline-block, use a width of 50px, align the text to the left, and add a margin on the right of 4px. [SHOW EXPECTED](#)

[CSS](#)
[HTML](#)

```
1 label {
2
3     /* Your solution goes here */
4
5 }
6 form {
7     font: 10pt Arial;
8     background-color: #eee;
9     padding: 10px;
10}
```

1

2

3

4

[Check](#)
[Next](#)

View your last submission ▾

[Feedback?](#)

Exploring further:

- [CSS Forms](#) from W3Schools
- [An Extensive Guide To Web Form Usability](#) from Smash Magazine
- [Placeholders in Form Fields Are Harmful](#) from Nielsen Norman Group
- [Replacing Radio Buttons Without Replacing Radio Buttons](#) from SitePoint

How was
this
section?



[Provide section feedback](#)