

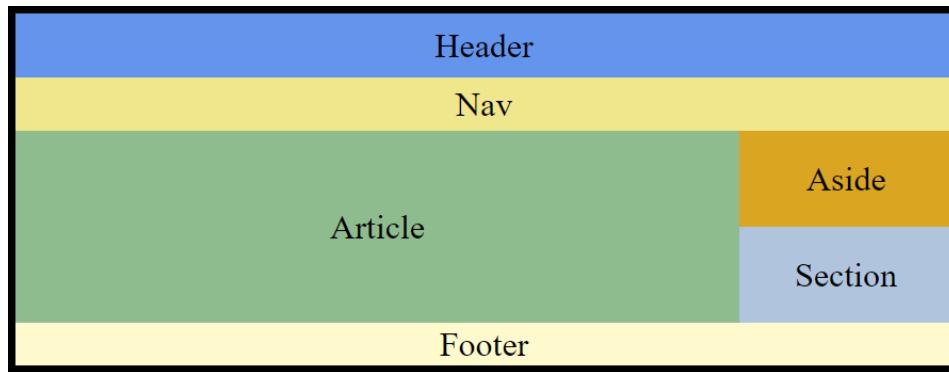
5.1 Flexbox

Fluid layout

CSS is instrumental in creating a page layout that is visually appealing. Many websites use the same layout on every web page for consistency. The layout normally contains the following sections:²

- Header at the top that identifies the website
- Navigational links at the top or left side that present the main links for navigating the website
- Various sections that include related content
- Footer at the bottom that contains contact information, copyright, author name, etc.

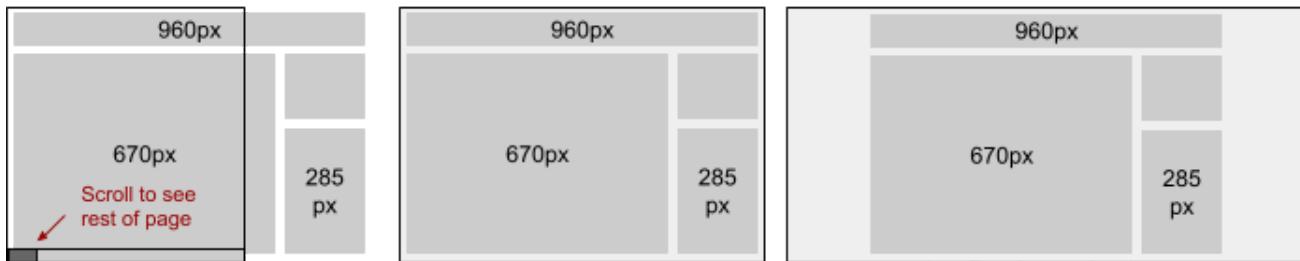
Figure 5.1.1: Typical web page layout.



Some web pages use a **fixed layout**, which uses a fixed-width container to envelop the web page contents. Ex: The figure below shows a web page at the top using a fixed layout where the entire contents fit inside 960px. Resizing the browser does not change the width of the web page contents.

A web page that uses a **fluid layout** allows the page contents to fill the browser, sometimes by using percentages for widths. Ex: The figure below shows a web page at the bottom using a fluid layout where the contents always fit the browser's width. Fluid layouts make better use of the available space than fixed layouts and do not produce a horizontal scroll bar when the browser is resized.

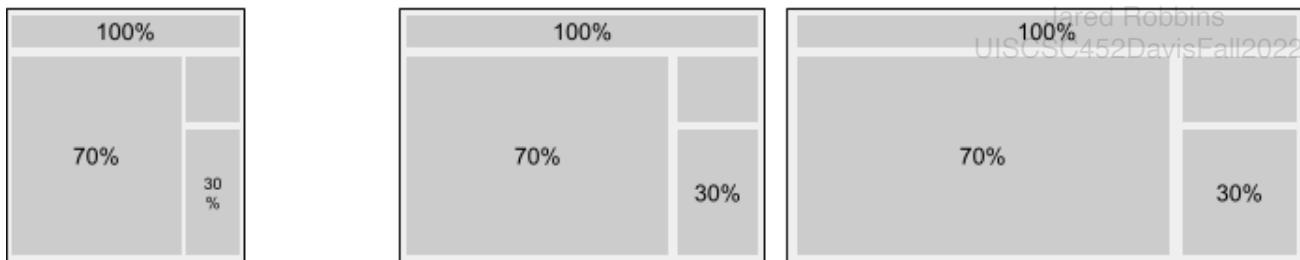
Figure 5.1.2: Fixed vs. fluid layout.

Fixed

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

**Fluid****PARTICIPATION ACTIVITY**

5.1.1: Create a fluid layout.



The web page below displays a **wireframe**, a blueprint for a web page that shows how future content will be arranged. The HTML uses common containers like `<header>`, `<footer>`, etc., and the CSS determines the height, color, etc. of the containers.

The web page can be resized by grabbing the right edge of the rendered web page and moving the mouse left and right. The web page uses a fixed layout, so the layout does not change sizes as the browser is resized.

Make the following HTML and CSS modifications to create a fluid layout that is optimized for a desktop screen:

1. Remove the `width: 700px;` from the CSS `body` selector. Render the web page and verify that the page now fills the browser width. Resize the rendered web page to see the contents resize to fill the available space.

2. To move the Section directly under the Aside, place the Aside and Section inside a `<div>`:

```
<div>
  <aside>Aside</aside>
  <section>Section</section>
</div>
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Then add the following CSS rule to float the `<div>` to the right of the Article:

```
div {  
    float: right;  
    width: 200px;  
    height: 180px;  
}
```

Render the web page and observe the Section is now directly under the Aside, and both are to the right of the Article. However, the Aside is so tall that not enough space exists for the Section.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

3. Change the Aside's `line-height` from 180px to 90px so the Aside's height (90px) plus the Section's height (90px) fits within the `<div>`'s height (180px). Render the web page and verify the Aside and Section fit next to the Article.
4. Some web pages do not look good when expanded beyond a certain width. Add `max-width: 700px;` to the `body` selector to limit the maximum size of the content. Render the web page and observe how the content does not go beyond 700 pixels. When the browser is less than 700 pixels wide, the content automatically adjusts.

HTML CSS

```
1 <body>  
2   <header>Header</header>  
3   <nav>Nav</nav>  
4   <aside>Aside</aside>  
5   <section>Section</section>  
6   <article>Article</article>  
7   <footer>Footer</footer>  
8 </body>  
9
```

Render webpage

Reset code

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Your webpage

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

PARTICIPATION ACTIVITY

5.1.2: Fluid layout.



- 1) A fluid layout widens to fit the browser's width.



- True
- False

- 2) The example above uses the `float` property to place two containers next to each other.



- True
- False

- 3) The `<div>` element from the example above always occupies the same percent of the browser width.



- True
- False

- 4) In the example above, changing the `div` CSS selector width to 30% makes the `<div>` resize when the browser width is resized.



- True
- False



- False
- 5) The `max-width` property should never be used in a fluid layout.
- True
 False

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Flexbox container and items

The **Flexible Box** or **flexbox** is a CSS layout mode that provides an efficient way to lay out elements in a container so the elements behave predictably when the container is resized or viewed on different screen sizes. Many developers find the flexbox layout easier to use than floating elements when creating fluid layouts.

A **flex container** is an element that has the CSS property `display` set to `flex` to create a block-level flex container or `inline-flex` to create an inline flex container. Ex:

`<div style="display: flex">`. Flex containers hold flex items. A **flex item** is a child element of a flex container that is positioned and sized according to various CSS flexbox properties.

PARTICIPATION ACTIVITY

5.1.3: Flexbox example renders three div elements on the same row.



Animation content:

Two blocks of code are displayed, the first in HTML:

```
<div id="container">
    <div>A</div>
    <div>B</div>
    <div>C</div>
</div>,
```

the second in CSS:

```
/* flex container */
#container {
    display: flex;
    border: 1px green solid;
    padding: 5px;
}
/* flex items */
#container > div {
    padding: 10px;
    border: 1px blue solid;
}
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

A screen is displayed below. The first scene shows A, B, and C

vertically aligned on the left without CSS. Next the flex container block is highlighted and A, B, and C display left-aligned in a row. Lastly the flex items block is highlighted and A, B, and C are formatted.

Animation captions:

1. Without any CSS, the A, B, and C div elements display vertically, each filling the browser width.
2. Setting the CSS display property to "flex" makes the outer div the flex container. The flex items now display on the same row.
3. The flex items have padding and blue borders.

PARTICIPATION ACTIVITY

5.1.4: Flexbox container and items.



Refer to the animation above.

- 1) The flex container has ____ flex item(s).

- 1
- 2
- 3



- 2) Removing ____ from the `#container` rule displays all flex items vertically on separate rows.

- `display: flex;`
- `border: 1px green solid;`
- `padding: 5px;`



- 3) Flex items appear ____ within a flex container by default.

- left aligned
- centered
- right-aligned



©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Flex container properties

Several CSS properties modify the default behavior of a flex container:

- The **flex-direction** property defines the direction of flex items within the container using values:

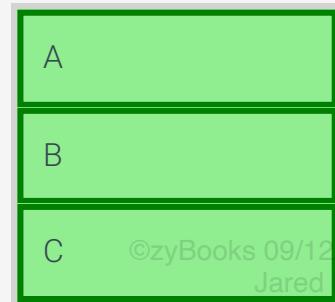
row (default)



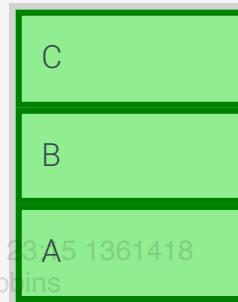
row-reverse



column

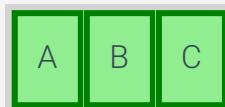


column-reverse

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

- The **justify-content** property justifies the flex items within the container using values:

flex-start (default) flex-end



center



space-between

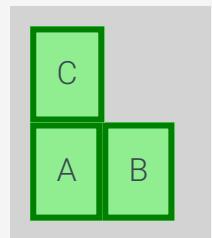
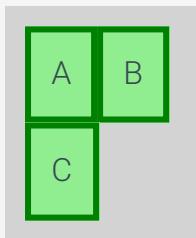


- The **flex-wrap** property determines if or how flex items wrap onto multiple rows when the container is not wide enough to hold all items, using values:

nowrap (default) wrap



wrap-reverse

**PARTICIPATION ACTIVITY**

5.1.5: Flex container properties for photos.



The web page below uses a flexbox to display three photos with captions.

Make the following CSS modifications to the flex container. After each modification, render the web page to verify the modification works properly.

- Add a **flex-direction** property to the flex container to display the photos in reverse.
- Add a **justify-content** property to the flex container to display the photos centered in the browser window.
- Add a **flex-wrap** property to the flex container so the photos wrap to the next row when the browser width is reduced.

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Note: The window that renders the web page can be resized using the small handle in the bottom-right corner.

```
1 <div id="container">
2   <div class="photo">
3     
9     
15    
  
  
  
</div>
```

- 1) What **flex-direction** value



displays Madrid on the right side of the grid container with all three images on the same row?

- row**
 - row-reverse**
 - column-reverse**
- 2) What **justify-content** value puts an equal amount of space on either side of the three images but leaves no space between the images?
- center**
 - space-between**
 - space-around**
- 3) What **flex-wrap** value makes the Islamabad image appear below Madrid if the grid container is not wide enough to show all three images on the same row?
- nowrap**
 - wrap**
 - wrap-reverse**

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022



Flex item properties

A flex item's width is determined by the combination of three CSS properties:

- The **flex-basis** property sets the initial length of a flex item. The values can be **auto** (the default), a percentage, or a length unit. The default value **auto** makes the flex item the same initial length as the content.
- The **flex-grow** property sets a proportion that determines how much of the available container space should be assigned to the item. The default is 0, meaning the size should be based on the item's content.
- The **flex-shrink** property sets a proportion that determines the item's minimum size. The default is 1, meaning the size should shrink at the same rate as other items when the container width shrinks. A value of 0 means the item should not change sizes when the container width shrinks.

Jared Robbins
UISCSC452DavisFall2022



Animation content:

A code block of HTML is displayed:

```
<nav>
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="products.html">Products</a></li>
    <li><a href="about.html">About</a></li>
  </ul>
</nav>
```

Next three code blocks of CSS are shown. The first block shows the following:

```
nav ul {
  display: flex;
  list-style-type: none;
  padding: 0;
}
```

The second:

```
nav li {
  flex-grow: 1;
  background-color: gold;
  text-align: center;
}
```

And the third:

```
nav li {
  flex-basis: 100px;
  flex-shrink: 0;
  background-color: gold;
  text-align: center;
}
```

Below each CSS code block there is a screen displaying the output. The first shows Home, Product, and About in a row across the top. The second first shows the three nav links left-aligned in a row when flex-grow is 0, then portions the nav links evenly when flex-grow is 1. The third box shows the three nav links changing container size when the browser is resized when flex-shrink is 0, then flex-shrink is changed to 1, preventing the containers from resizing.

Animation captions:

1. A website's navigation links are displayed in an unordered list.
2. Making the ul element a flex container places the nav links on the same row.

3. By default, the li elements have flex-basis:auto and flex-grow:0, so li elements are only as wide as the item's content.
4. Changing flex-grow from the default 0 to 1 gives all li elements the same proportion. The elements fill the flex container.
5. Replacing "flex-grow:1" with "flex-basis:100px" makes each li element 100px wide.
6. Resizing the browser changes the container size. When the container shrinks, the li elements shrink to fill the available space.
7. Changing flex-shrink from the default 1 to 0 prevents the li elements from shrinking when the browser is resized.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins
UIISCSC452DavisFall2022

PARTICIPATION
ACTIVITY

5.1.8: Flex item properties.



Refer to the web page below.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins
UIISCSC452DavisFall2022

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>ACME Widgets</title>
    <style>
        body {
            font-family: Arial;
        }

        header {
            display: flex;
            justify-content: space-between;
        }

        header h1 {
            margin: 6px;
        }

        nav ul {
            display: flex;
            list-style-type: none;
            padding: 0;
            margin: 0;
            width: 500px;
        }

        nav li {
            background-color: gold;
            margin: 5px;
            padding: 10px;
            text-align: center;
        }

        nav a {
            color: black;
            text-decoration: none;
        }
    </style>
</head>
<body>
    <header>
        <h1>ACME Widgets</h1>
        <nav>
            <ul>
                <li><a href="index.html">Home</a></li>
                <li><a href="products.html">Products</a></li>
                <li><a href="services.html">Services</a></li>
                <li><a href="about.html">About</a></li>
            </ul>
        </nav>
    </header>
</body>
</html>
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

- 1) How many flex containers exist in the web page?

 0

1 2

- 2) The heading "ACME Widgets" appears _____ the navigation links in the browser.

- to the left of
- to the right of
- above

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

- 3) What is the **flex-basis** for the **<nav>** flex item?

- auto
- 0
- 1

- 4) How wide is the **<nav>** flex item?

- Just wide enough to hold all the navigation links
- Half the width of the web page
- Entire width of the web page

- 5) In addition to the CSS below, _____ must be added to the **nav li** rule so the li elements fill half the row.

```
nav {  
    flex-basis: 50%;  
}
```

- flex-grow: 0;**
- flex-grow: 1;**
- flex-grow: 50%;**

- 6) Adding _____ to the **header h1** rule prevents the heading from shrinking when the browser is resized.

- flex-shrink: none;**
- flex-shrink: 1;**
- flex-shrink: 0;**

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

The flex property

The shorthand property **flex** specifies **flex-grow**, **flex-shrink**, and **flex-basis** together. Ex: **flex: 0 1 auto;** is the same as **flex-grow: 0; flex-shrink: 1; flex-basis: auto;**.

PARTICIPATION
ACTIVITY

5.1.9: Flexbox layout using the flex property.



©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Animation content:

Two blocks of code are shown. The HTML:

```
<body>
  <header>Header</header>
  <!-- Flexbox layout -->
  <div id="container">
    <nav>Nav</nav>
    <main>Main</main>
    <aside>Aside</aside>
  </div>
  <footer>Footer</footer>
</body>
```

and the CSS:

```
#container {
  display: flex;
}

nav {
  flex: 0 1 20%;
}

main {
  flex: 0 1 60%;
}

aside {
  flex: 0 1 20%;
}
```

In the CSS **flex** property, the **0** controls **flex-grow**, the **1** controls **flex-shrink**, and the percentage controls **flex-basis**. Below is a screen displaying header spanning the width at the top of body, footer spanning the width at the bottom of body, and between the two is **<nav>** occupying 20% of the left side of body, main occupying 60% of the middle of body, and aside occupying the remaining 20% of body on the right side.

Animation captions:

1. <header> and <footer> span the entire width of <body>, but the <div> is a flex container that displays the flex items on the same row.
2. <nav>, <main>, and <aside> all have flex-grow = 0, so all three flex items' width should be based on each item's content.
3. <nav>, <main>, and <aside> all have flex-shrink = 1, so all three flex items shrink at the same rate when the browser is resized.
4. <nav> occupies 20% of the row, <main> occupies 60%, and <aside> occupies 20%. $20\% + 60\% + 20\% = 100\%$ of the row.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

PARTICIPATION ACTIVITY**5.1.10: Flex item properties.**

Refer to the animation above.

- 1) Which property makes the Main flex item wider than Nav and Aside?

- flex-grow**
- flex-basis**
- flex-shrink**



- 2) How do the changes below affect the web page?

```
nav { flex: 1 1 auto; }
main { flex: 3.5 1 auto; }
aside { flex: 1 1 auto; }
```



- All three flex items are equal width.
- Nav and Aside are wider than Main.
- The layout will be nearly the same as before.

- 3) How do the changes below affect the web page?



```
#container { display: flex;
            flex-direction:
            row;
            justify-content:
            center; }
nav { flex: 0 1 auto; }
main { flex: 0 1 auto; }
aside { flex: 0 1 auto; }
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

- All three flex items are default width and centered horizontally in the flex container.
- Nav and aside are wider than Main.
- All three flex items will be default width, aligned to the left.

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

- 4) How does the change below affect the web page?



```
#container { display: flex;  
flex-direction: column; }
```

- No changes.
- All three flex items are stacked on top of one another.
- The direction of the flex items is reversed.

PARTICIPATION
ACTIVITY

5.1.11: Flexbox practice.



The webpage displays three years of Grammy Award nominations for Album of the Year. Each year's nomination is placed in a `<section>` element.

Alter the webpage to display the three sections in a single row:

1. Add a single `<div>` element that surrounds all three sections.
2. Add the proper CSS so the `<div>` becomes a flex container and displays each section on a single row.
3. Add a `flex` property to the `section` rule that sets `flex-grow` to 0, `flex-shrink` to 1, and `flex-basis` to 250px.
4. Add a `flex-wrap` property that makes the sections wrap to the next row when the browser is resized.

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

After adding the HTML and CSS above, render the webpage. Resize the rendered webpage's window and verify that the sections wrap to the next row when the window isn't wide enough to display the sections side-by-side.

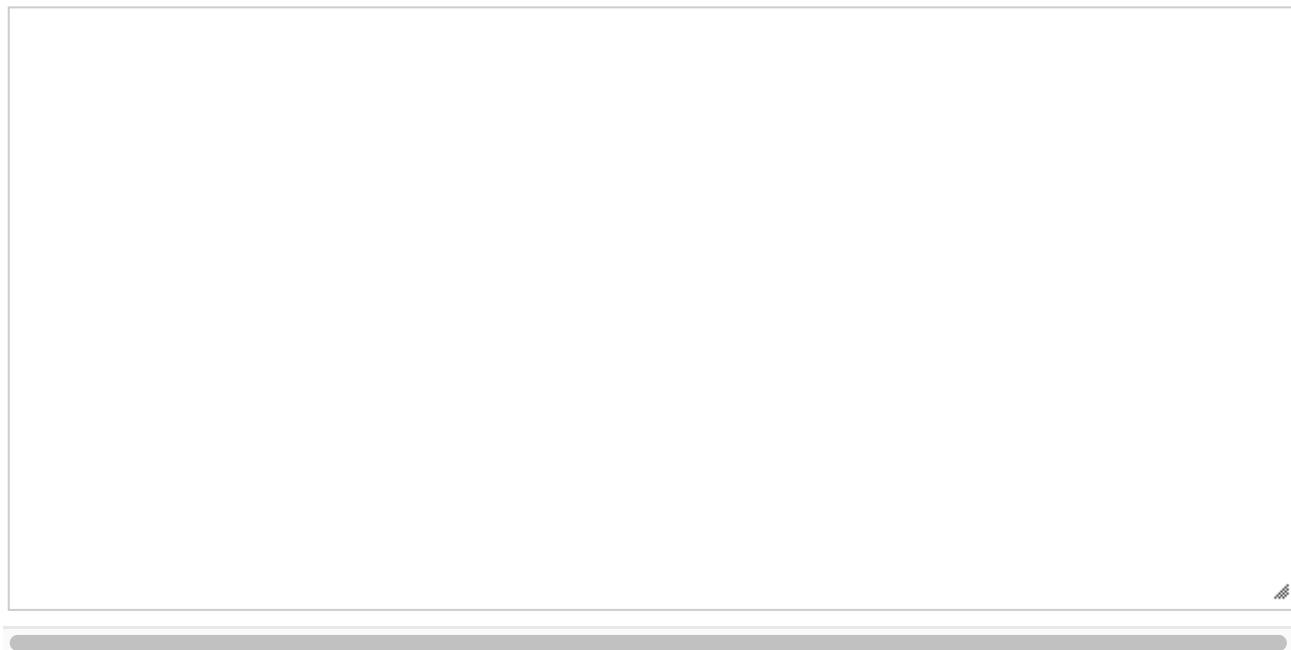
HTML

CSS

```
1 <h1>Grammy Nominees for Album of the Year</h1>
2
3 <section>
4   <h2>2020</h2>
5   <ul>
6     <li><cite>Folklore</cite> by Taylor Swift</li>
7     <li><cite>Future Nostalgia</cite> by Dua Lipa</li>
8     <li><cite>Hollywood's Bleeding</cite> by Post Malone</li>
9     <li><cite>Chilombo</cite> by Jhené Aiko</li>
10    <li><cite>Black Pumas (Deluxe Edition)</cite> by Black Pumas</li>
11    <li><cite>Everyday Life</cite> by Coldplay</li>
12    <li><cite>Djesse Vol.3</cite> by Jacob Collier</li>
13    <li><cite>Women In Music Pt. III</cite> by HAIM</li>
14  </ul>
15 </section>
16 <section>
```

Render webpageReset code

Your webpage



CHALLENGE ACTIVITY

5.1.1: Flexbox.



©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

422366.2722836.qx3zqy7

Start

For the element with class container, set display to flex and width to 105px.

SHOW EXPECTED

CSS

HTML

```
1 .container {  
2  
3     /* Your solution goes here */  
4  
5 }  
6 .container > div {  
7     background-color: lightblue;  
8     border: 2px solid blue;  
9     padding: 6px;  
10 }
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

1

2

3

Check

Next

Exploring further:

- [HTML Layouts](#) from W3Schools
- [CSS Flexbox](#) from W3Schools

5.2 Grid layout

Grid container and grid items

Grid layout is a CSS layout mode that divides a web page into a rectangular grid in which to position page elements. Grid layout is ideal for designing two-dimensional web page layouts.
©zyBooks 09/12/22 23:15 1361418
Jared Robbins

A **grid container** is an element that has the CSS property `display: grid` set to `grid` to create a block-level grid container or `inline-grid` to create an inline grid container. Ex:

`<div style="display: grid">`. A **grid item** is a child element of a grid container that is by default placed into a single grid cell.

The **grid-template-columns** property defines the grid container's number of columns and optionally the width of each column. Ex: `grid-template-columns: 50px 90px auto auto;` specifies 4

values that create 4 columns: the first is 50px wide, the second is 90px wide, and the third and fourth columns are automatically sized to fit the remainder of the grid width.

PARTICIPATION ACTIVITY**5.2.1: Grid layout example.****Animation content:**

Two blocks of code are shown. The HTML:

```
<div id="grid-container">
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
    <div>5</div>
    <div>6</div>
</div>
```

and the CSS:

```
#grid-container {
    display: grid;
    grid-template-columns: auto auto auto;
}

#grid-container > div {
    text-align: center;
    background: lightgreen;
    border: 3px solid green;
    padding: 20px;
}
```

The CSS `grid-template-columns` property determines how many columns will be displayed based on how many property values there are. A screen is shown divided initially into 3 columns based on `grid-template-columns`, then the 6 div elements are displayed as 3 columns and 2 rows.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Animation captions:

©zyBooks 09/12/22 23:15 1361418

1. The "display: grid;" declaration makes the `<div>` with id `grid-container` a block-level grid.
2. The `grid-template-columns` property is assigned 3 "auto" values, so the grid container will contain 3 equally-sized columns.
3. Six `<div>` elements are children of the grid container, so each element becomes a grid item. The `div` child selector puts a green border around each grid item.

PARTICIPATION

ACTIVITY

| 5.2.2: Grid layout basics.



Refer to the animation above.

- 1) If `#grid-container` is modified, how many columns will the grid have?



```
#grid-container {  
    display: grid;  
    grid-template-columns: auto  
    auto;  
}
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

- 2
- 3
- 4

- 2) If `#grid-container` is modified, how wide is the second column?



```
#grid-container {  
    width: 600px;  
    display: grid;  
    grid-template-columns: 200px  
    auto;  
}
```

- 200px
- 400px
- 600px

- 3) If `#grid-container` is modified, how wide is the second column?



```
#grid-container {  
    width: 600px;  
    display: grid;  
    grid-template-columns: 20%  
    auto;  
}
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

- 120px
- 480px
- 600px

- 4) If the grid container has 10 grid items, what `grid-template-columns`



value creates a grid with 2 rows and 5 columns?

```
#grid-container {
    width: 600px;
    display: grid;
    grid-template-columns:
    _____;
}
```

- 5
- 200px auto
- 200px auto auto auto 200px

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Controlling the grid container

The default behavior of a grid container can be modified with various CSS properties:

- The **grid-gap** property defines the gap between each grid row and column. Ex:
`grid-gap: 10px 20px;` puts a 10px gap between each row and a 20px gap between each column.



- The **grid-template-rows** property defines the height of each row. Ex:
`grid-template-rows: 20px 40px;` makes the first row 20px tall and the second row 40px tall.



- The **justify-content** property horizontally aligns the grid items inside the grid container using values:
 - **start** - Aligns grid flush with the grid container's starting edge.



©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

- **end** - Aligns grid flush with the grid container's ending edge.



- **center** - Aligns grid in the center of the grid container.



- **stretch** - Stretches the grid items to fill the grid container width.



- **space-around** - Places equal spacing between grid items with half the space on either side of the grid container.



- **space-between** - Places equal spacing between grid items with no space on either side of the grid container.



- **space-evenly** - Places equal spacing between grid items, including the sides of the grid container.



- The **align-content** property vertically aligns the grid items inside the grid container using values:

- **start** - Aligns grid flush with the grid container's starting edge.



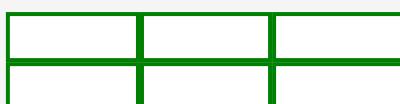
- **end** - Aligns grid flush with the grid container's ending edge.



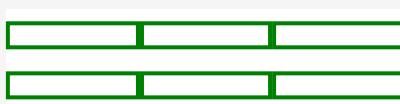
- **center** - Aligns grid in the center of the grid container.



- **stretch** - Stretches the grid items to fill the grid container height.



- **space-around** - Places equal spacing between grid items with half the space on either side of the grid container.



©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

- **space-between** - Places equal spacing between grid items with no space on either side of the grid container.



- **space-evenly** - Places equal spacing between grid items, including the sides of the grid container.



©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

The **justify-content** and **align-content** properties have no effect unless the grid width or height is less than the grid container's width or height.

PARTICIPATION ACTIVITY

5.2.3: Modify the grid container.



Refer to the HTML and CSS below:

```
<style>
#grid-container {
  width: 300px;
  height: 70px;
  background: mediumseagreen;
  display: grid;
  grid-template-columns: auto auto auto;
}

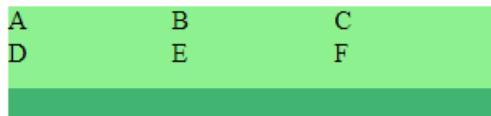
#grid-container > div {
  background: lightgreen;
}
</style>

<body>
  <div id="grid-container">
    <div>A</div>
    <div>B</div>
    <div>C</div>
    <div>D</div>
    <div>E</div>
    <div>F</div>
  </div>
</body>
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

- 1) Add a CSS declaration to **#grid-container** so the first row is 20 pixels tall and the second row is 30 pixels tall.





`grid-gap: 20px;`

Check

Show answer

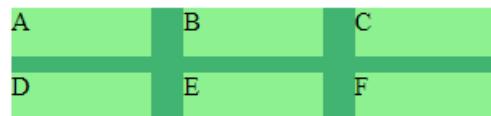
©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022



- 2) Put a 10 pixel gap between each row and 20 pixel gap between each column.



`grid-gap:`

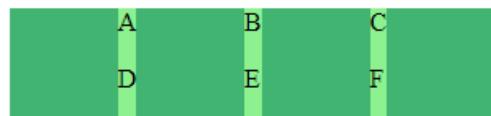
`10px 20px;`

Check

Show answer



- 3) Place an equal amount of horizontal space between each grid item, including the grid edges.



`justify-content:`

`space-around;`

Check

Show answer



- 4) Center each grid item vertically in the grid.



`align-items: center;`

Check

Show answer

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022



Controlling grid item placement

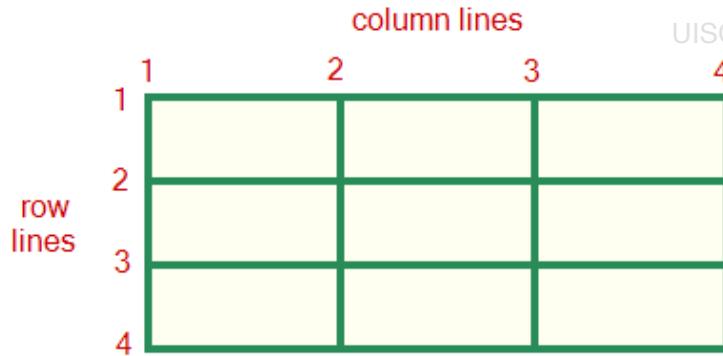
A grid item by default appears in a single row and column based on the ordering of the grid item within the grid container. However, grid items may be positioned at specific grid locations using the column line and row line numbers as illustrated in the figure below.

Figure 5.2.1: Row and column lines.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022



A grid item may be placed in a specific row or column or span multiple rows and/or columns using various following CSS properties:

- The **grid-row** property lists the grid item's starting and ending row line numbers. Ex:
`grid-row: 1 / 3;` makes the grid item start at row line 1 and end at row line 3, so the grid item spans 2 rows.
- The **grid-column** property lists the grid item's starting and ending column line numbers. Ex:
`grid-column: 1 / 4;` makes the grid item start at column line 1 and end at column line 4, so the grid item spans 3 columns.
- The **grid-area** property lists the grid item's starting and ending row and column numbers. Ex:
`grid-area: 1 / 2 / 3 / 4;` makes the grid item start at row line 1 and column line 2 and end at row line 3 and column line 4, so the grid item spans 2 rows and 2 columns.

PARTICIPATION ACTIVITY

5.2.4: Rearrange the grid.



Add the following CSS modifications to alter the grid:

1. Add the following CSS to move the A item to the 1st column and 3rd row:

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

```
#grid-item-a {
    grid-row: 3 / 4;
}
```

Render the web page and observe the A item is previously where the G item was located.

2. Add the following CSS to make the B item occupy the first 3 rows of the 3rd column:

```
#grid-item-b {  
    grid-column: 3 / 4;  
    grid-row: 1 / 4;  
}
```

Render the web page and observe that B is on the right edge of the grid, and a 4th row is created with an empty grid cell below B.

3. Add the following CSS to make the C item occupy the first 3 rows and 2 columns:

```
#grid-item-c {  
    grid-area: 1 / 1 / 4 / 3;  
}
```

Render the web page and observe the web page matches the expected web page.

HTML CSS

```
1 <body>  
2   <div id="grid-container">  
3     <div id="grid-item-a">A</div>  
4     <div id="grid-item-b">B</div>  
5     <div id="grid-item-c">C</div>  
6     <div>D</div>  
7     <div>E</div>  
8     <div>F</div>  
9     <div>G</div>  
10    <div>H</div>  
11    <div>I</div>  
12  </div>  
13 </body>
```

Render webpage

Reset code

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Your webpage**Expected webpage**

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

PARTICIPATION ACTIVITY

5.2.5: Modify the grid layout.



Refer to the HTML and CSS below:

```
<style>
#grid-container {
    width: 300px;
    background: mediumseagreen;
    display: grid;
    grid-template-columns: auto auto auto;
    grid-gap: 5px 5px;
}

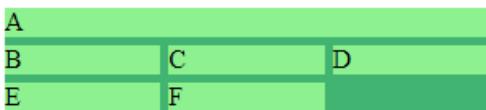
#grid-container > div {
    background: lightgreen;
}
</style>

<body>
    <div id="grid-container">
        <div id="grid-item-a">A</div>
        <div id="grid-item-b">B</div>
        <div id="grid-item-c">C</div>
        <div>D</div>
        <div>E</div>
        <div>F</div>
    </div>
</body>
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

- 1) Make the A item occupy the entire top row.





```
#grid-item-a {
    grid-column:
    [ ] ;
}
```

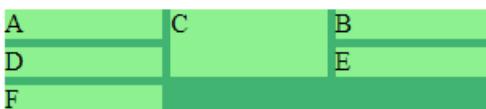
Check**Show answer**

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

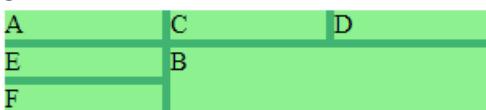
- 2) Make the C item occupy the top 2 rows in the 2nd column.



```
#grid-item-c {
    grid-column: 2 / 3;
    grid-row:
    [ ] ;
}
```

Check**Show answer**

- 3) Make the B item occupy 2 rows and 2 columns in the bottom-right grid corner.



```
#grid-item-b {
    grid-area:
    [ ] ;
}
```

Check**Show answer**

Naming grid items

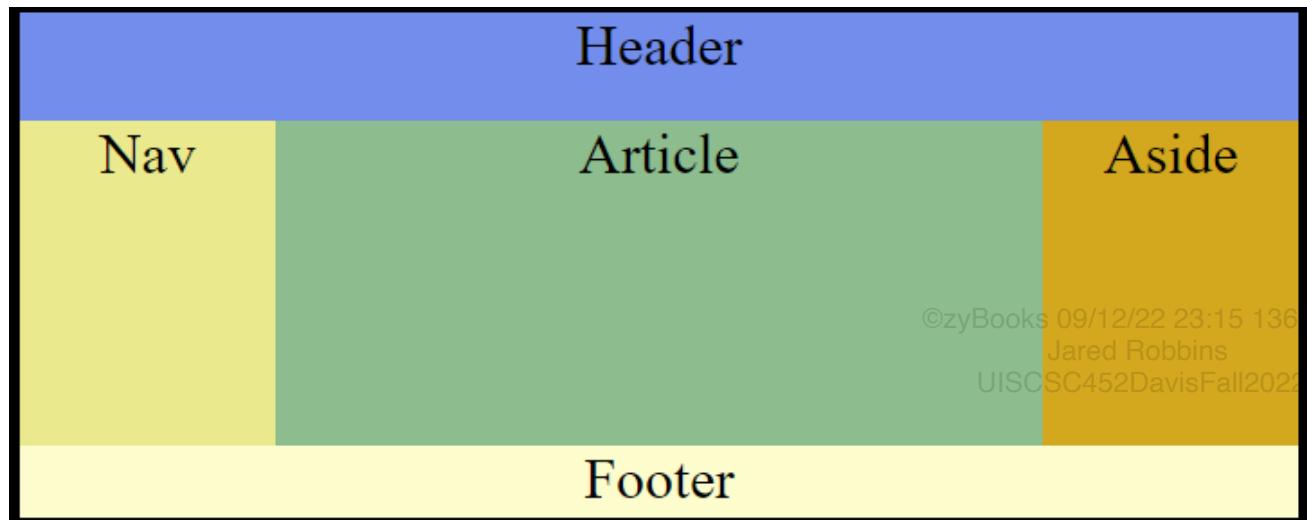
©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Grid items may be assigned names with the `grid-area` property. The grid container's **`grid-template-areas`** property specifies the grid layout using the named grid items.

Figure 5.2.2: Layout goal for the Participation Activity below.

**PARTICIPATION ACTIVITY**

5.2.6: Grid layout using named grid items.



Add the following CSS modifications to create a grid layout for the web page that matches the figure above:

1. Add the following CSS to the **body** selector to change the <body> into a grid container with 3 columns and 3 rows:

```
body {  
  ...  
  display: grid;  
  grid-template-columns: 150px auto 150px;  
  grid-template-rows: 60px 180px 40px;  
}
```

Render the web page and observe that the five grid items occupy 2 rows.

2. Use the **grid-area** property to name each of the 5 grid items:

©zyBooks 09/12/22 23:15 136 418
Jared Robbins
UIISCSC452DavisFall2022

```
header {
    background: cornflowerblue;
    grid-area: head;
}

nav {
    background: khaki;
    grid-area: nav;
}

article {
    background: darkseagreen;
    grid-area: article;
}

aside {
    background: goldenrod;
    grid-area: aside;
}

footer {
    background: lemonchiffon;
    grid-area: foot;
}
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

3. Use the `grid-template-areas` property on the grid container to layout the grid items. Each row is specified in a single string with the grid item names.

```
body {
    ...
    grid-template-areas:
        "head head head"
        "nav article aside"
        "foot foot foot";
}
```

Render the web page, which should look like the figure above. The Header occupies all 3 columns on the top row; the Nav, Article, and Aside occupy the 3 columns in the second row; and the Footer occupies all 3 columns on the bottom row.

HTML CSS

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

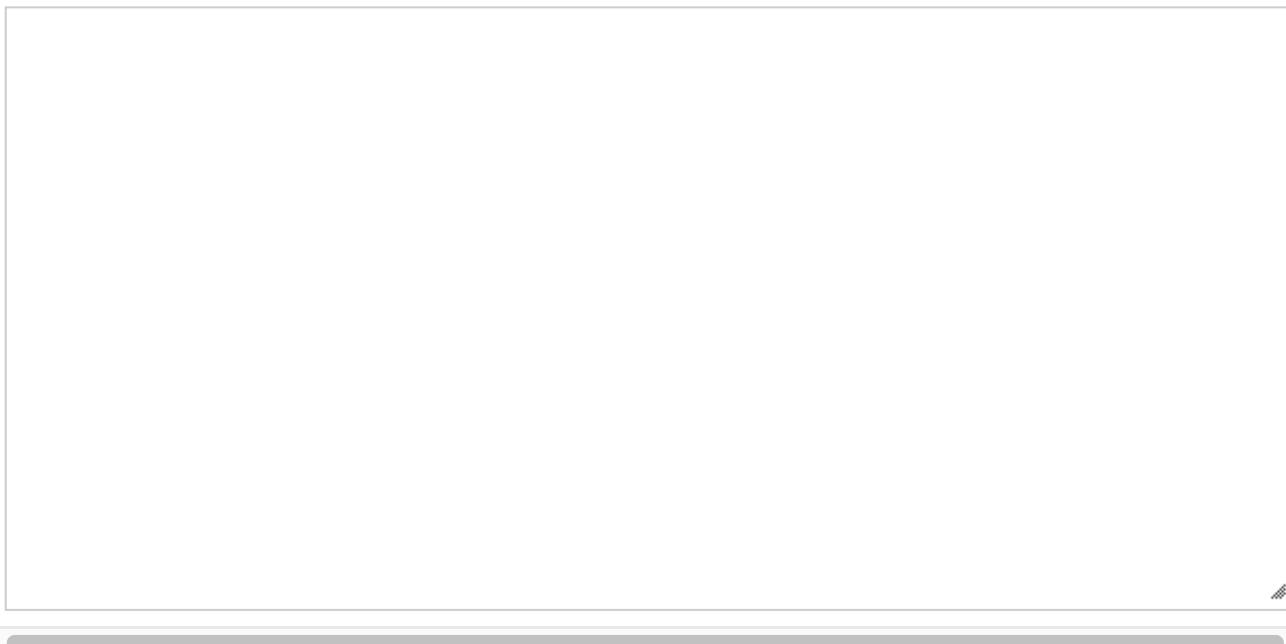
```
1 <body>
2   <header>Header</header>
3   <nav>Nav</nav>
4   <article>Article</article>
5   <aside>Aside</aside>
6   <footer>Footer</footer>
7 </body>
8
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

Render webpage

Reset code

Your webpage



PARTICIPATION ACTIVITY

5.2.7: Named grid items.



©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

Refer to the Participation Activity above.

1) What `grid-template-areas` value makes the Article span 2 columns?



- "head head head"
"nav article aside"
"foot article foot"

"head article head"
 "nav article article aside"
 "foot foot foot"

"head head head"
 "nav article article"
 "foot foot foot"

- 2) A single period can be used in a `grid-template-areas` value to represent an empty grid cell. Which `grid-template-areas` value places an empty grid cell under Nav?

". head head"
 "nav article aside"
 "foot foot foot"

"head head head"
 "nav . aside"
 "foot foot foot"

"head head head"
 "nav article aside"
 ". foot foot"

- 3) What is the correct way to name the new grid item `<section>`?

```
section {  
    grid-area: ____;  
}
```

"mysection"
 mysection
 'mysection'

CHALLENGE ACTIVITY

5.2.1: Grid layout.

422366.2722836.qx3zqy7

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Start

For the element with class container, set display to grid and `grid-template-columns` to display 2 columns that are both 80px wide. **SHOW EXPECTED**

CSS HTML

1 | .container {

```
2
3 /* Your solution goes here */
4
5 }
6 .container > div {
7   background-color: lightblue;
8   border: 2px solid blue;
9   padding: 6px;
10}
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

1

2

3

4

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

Exploring further:

- [CSS Grid Layout Module from W3Schools](#)
- [CSS Grid Layout Browser Support from caniuse.com](#)

5.3 Positioning elements

The position property

The CSS **position** property gives developers more control over where elements should appear in the browser. **position** has four possible values:

- **static** - **Static positioning** is the default positioning
- **relative** - **Relative positioning** positions the element relative to the element's default position
- **fixed** - **Fixed positioning** positions the element relative to the viewport in a fixed location
- **absolute** - **Absolute positioning** positions the element relative to the nearest positioned ancestor

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022



Animation content:

Two blocks of code are shown. The CSS:

```
#content {  
    border: solid 2px blue;  
    color: blue;  
    position: relative;  
    left: 20px;  
    top: 10px;  
}
```

and the HTML:

```
<div>Before</div>  
<div id="content">Content</div>  
<div>After</div>  
}
```

A screen is displayed with the Content div. When the property position is added to the CSS the div element's position becomes relative to its normal position and changes based on the properties left and top. A negative value in the property left makes the div move to the left and a positive value moves div to the right. A negative value in the property top makes the div move down and a positive value moves div up.

Animation captions:

1. The "Content" <div> displays in the default location.
2. Adding relative positioning to #content does not change the "Content" <div> position until "left" and/or "top" properties are specified.
3. "left: -20px" moves the left edge 20 pixels left from the default location.
4. "left: 20px" moves the left edge 20 pixels to the right of the default location.
5. Negative values for "top" move the element up, and positive values move the element down.

PARTICIPATION ACTIVITY

5.3.2: Relative and static positioning.



- 1) Where is the image located relative to the image's default location?

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

```

```

- 30 pixels to the right
- 30 pixels to the left
-



No change

- 2) Where is the image located relative to the image's default location?

```

```

- 30 pixels higher
- 30 pixels lower
- No change

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022



- 3) Where is the image located relative to the image's default location?

```

```

- 20 pixels to the right and 30 pixels higher
- 20 pixels to the left and 30 pixels lower
- No change

Fixed positioning

Fixed positioning places the element at a fixed location in the viewport, and scrolling does not move the element. A **viewport** is the visible area of a web page. The fixed element is detached from the normal flow of elements in the page and is layered on top of the page contents.

PARTICIPATION
ACTIVITY

5.3.3: Fixed positioning.



Animation content:

Two blocks of code are shown. The CSS:

```
#content {  
border: solid 2px blue;  
color: blue;  
position: fixed;  
left: 60px;  
top: 50px;  
}
```

and the HTML:

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

```
<div>Before</div>
&ly;div id="content">Content</div>
<div>After</div>
}
```

A screen is displayed with the Content `<div>`. When the property position is added to the CSS the `<div>` element's position becomes relative to the browser window. Using the properties left and top `<div>` is positioned on the screen. A negative value in the property left makes the `<div>` move to the left and a positive value moves `<div>` to the right. A negative value in the property top makes the `<div>` move down and a positive value moves `<div>` up.

Animation captions:

1. The "Content" `<div>` displays in the default location.
2. Adding fixed positioning to `#content` detaches the "Content" `<div>` so the `<div>` is layered on top of the underlying content.
3. "left: 60px" moves the `<div>`'s left edge 60 pixels to the right of the browser's left edge.
4. "top: 50px" moves the `<div>`'s top edge 50 pixels below the browser's top edge.

PARTICIPATION ACTIVITY

5.3.4: Fixed positioning.



Refer to the CSS below.

```
.special {
  position: fixed;
  left: 100px;
  top: 25px;
}
```

- 1) All elements using the "special" class are displayed 100 pixels from the browser's left edge and 25 pixels from the browser's top edge.

- True
- False

- 2) All elements using the "special" class scroll with the page contents.

- True
- False

- 3) The text "123" is displayed on top of

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022



"ABC".

```
<span
  class="special">ABC</span>
<span
  class="special">123</span>
```

- True
- False

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Absolute positioning

Absolute positioning is similar to fixed positioning except:

1. The position is based on the nearest positioned ancestor element that uses fixed, absolute, or relative positioning. If no positioned ancestor element exists, the element is positioned relative to the document body.
2. An absolute-positioned element scrolls with the document unless an ancestor element is using fixed positioning.

Figure 5.3.1: Cheer is absolute positioned inside a positioned ancestor (left) and relative to the document body (right).

```
#container {
  border: solid 2px green;
  position: relative;
  height: 60px;
  width: 150px;
}

#cheer {
  color: red;
  position: absolute;
  left: 10px;
  top: 25px;
}
```

```
<div id="container">
  <div id="cheer">Go, fight, win!
</div>
</div>
```

Go, fight, win!

```
#container {
  border: solid 2px green;
  /* No positioning */
  height: 60px;
  width: 150px;
}

#cheer {
  color: red;
  position: absolute;
  left: 10px;
  top: 25px;
}
```

```
<div id="container">
  <div id="cheer">Go, fight, win!
</div>
</div>
```

Go, fight, win!

PARTICIPATION ACTIVITY**5.3.5: Absolute positioning.**

Refer to the CSS below.

```
.special {  
    position: absolute;  
    left: 100px;  
    top: 25px;  
}
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

- 1) The `` is displayed 100 pixels from the browser's left edge and 25 pixels from the browser's top edge.



```
<body>  
    <span  
        class="special">Special</span>  
</body>
```

- True
 False

- 2) Elements using the "special" class that do not have a positioned ancestor will scroll with the page contents.



- True
 False

- 3) If the "container" class uses fixed positioning, the `` will not scroll with the page contents.



```
<div class="container">  
    <span  
        class="special">Special</span>  
</div>
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

- 4) If the "container" class uses static positioning, the `` is positioned relative to the `<div>`.



```
<body>
  <div class="container">
    <span
      class="special">Special</span>
  </div>
</body>
```

- True
- False

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

z-index property

When a relative, absolute, or fixed element is placed on top of another positioned element, the element that is specified last in the HTML is placed on top. However, the CSS **z-index** property is used to specify a relative distance that orders the appearance of elements. Elements with higher **z-index** values are placed on top of elements with lower **z-index** values.

On the left side of the figure below, the browser renders the square elements in the order the elements appear in the HTML: The orange square is rendered first, and the green square is rendered last. The right side of the figure shows how the ordering changes using the **z-index** property: The orange square has the largest **z-index** and therefore appears on top.

Figure 5.3.2: No z-index is used on the left, but z-index changes the rendered order on the right.

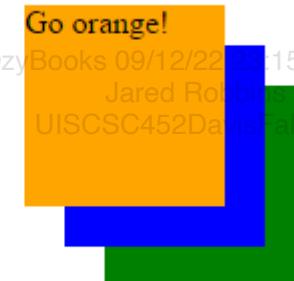
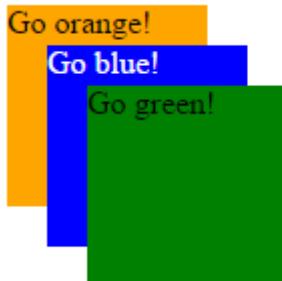
```
div {
  width: 100px;
  height: 100px;
  position: absolute;
}
#orange {
  background-color: orange;
  left: 10px;
  top: 10px;
}
#blue {
  background-color: blue;
  color: white;
  left: 30px;
  top: 30px;
}
#green {
  background-color: green;
  left: 50px;
  top: 50px;
}
```

```
div {
  width: 100px;
  height: 100px;
  position: absolute;
}
#orange {
  background-color: orange;
  z-index: 3;
  left: 10px;
  top: 10px;
}
#blue {
  background-color: blue;
  color: white;
  z-index: 2;
  left: 30px;
  top: 30px;
}
#green {
  background-color: green;
  z-index: 1;
  left: 50px;
  top: 50px;
}
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

```
<div id="orange">Go orange!
</div>
<div id="blue">Go blue!</div>
<div id="green">Go green!
</div>
```

```
<div id="orange">Go orange!
</div>
<div id="blue">Go blue!</div>
<div id="green">Go green!
</div>
```


PARTICIPATION ACTIVITY
5.3.6: z-index.


Refer to the figure above.

- 1) In the example on the right, what **z-index** value would make the green square appear on top of the orange and blue squares?

- 1
- 2
- 4



- 2) If all three squares are given the same **z-index** value of 5, which square appears on top?

- orange
- blue
- green



©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

PARTICIPATION ACTIVITY
5.3.7: Positioning practice.


The webpage below displays the iconic "I ❤ NY" logo. Use the **position** and **z-index** properties to make the webpage render like the expected webpage:

1. Use relative positioning in the `img` CSS rule to place the t-shirt image 10 pixels further to the right of the image's default location.
2. Use absolute positioning in the `.first`, `.heart`, and `.last` CSS rules to place "I", "❤" and "NY" in the correct configuration on top of the t-shirt.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UIISCSC452DavisFall2022

HTML CSS

```
1 <span class="first words">I</span>
2 <span class="heart">&hearts;</span>
3 <span class="last words">NY</span>
4
5 <div>
6   
8
```

Render webpage

Reset code

Your webpage

Expected webpage

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

**CHALLENGE
ACTIVITY****5.3.1: Positioning elements.**

422366.2722836.qx3zqy7

Start

For the element with id of content, set the position to relative, the top to 5 pixels, and the left to -7 pixels. **SHOW EXPECTED**

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

CSS**HTML**

```
1 #content {  
2  
3     /* Your solution goes here */  
4  
5 }
```

1

2

3

4

Check**Next**

Exploring further:

- [CSS Layout - The position Property](#) from W3Schools

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

5.4 Special effects

Text shadows

CSS3, the latest version of cascading style sheets, adds a number of properties that control a web page's visual presentation. Some improvements include shadows, rounded corners, image borders, and color gradients. Shadows are added to text using the CSS property **text-shadow**, which accepts four values:

- **offset-x** - Horizontal pixel offset of shadow
- **offset-y** - Vertical pixel offset of shadow
- **blur-radius** - Optional shadow blur (default is 0)
- **color** - Optional shadow color (default is usually the current CSS color)

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

The figure below illustrates how the four property values change the shadow.

Figure 5.4.1: Examples of different text-shadow values.

```
<p style="text-shadow: 5px 5px">Example 1</p>
<p style="text-shadow: 5px 5px 1px;">Example 2</p>
<p style="text-shadow: -5px -5px 1px green;">Example 3</p>
<p style="text-shadow: 0 0 3px red;">Example 4</p>
<p style="text-shadow: 0 0 3px red, 0 0 6px purple;">Example 5</p>
```

Example 1

Example 2

Example 3

Example 4

Example 5

PARTICIPATION
ACTIVITY

5.4.1: Text shadows.



- 1) Positive **offset-x** and **offset-y** make the shadow appear to the right and below the text, but negative values make the shadow appear to the left and above the text.

True

False

- 2) The **offset-x** and **offset-y** must be a non-zero value.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022



True False

- 3) A shadow with `blur-radius:4px` is less blurry than a shadow with `blur-radius:2px`.

 True False

- 4) Multiple shadows can apply to the same text.

 True False

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022



Box shadows

The CSS property **box-shadow** adds a shadow to the box around an element using the following properties:

- `inset` - Optional value that draws the shadow inside the box (default is outside the box)
- `offset-x` - Horizontal pixel offset of shadow
- `offset-y` - Vertical pixel offset of shadow
- `blur-radius` - Optional shadow blur (default is 0)
- `spread-radius` - Positive value causes shadow to grow, negative values to shrink (default is 0)
- `color` - Optional shadow color (default is usually the current CSS color)

Figure 5.4.2: Examples of different box-shadow values.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

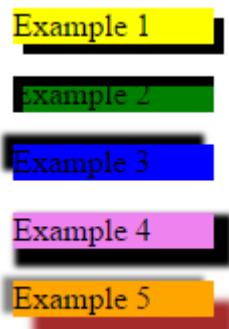
```
p {  
    width: 100px;  
}  
  
#example1 {  
    background-color: yellow;  
    box-shadow: 5px 5px;  
}  
  
#example2 {  
    background-color: green;  
    box-shadow: inset 5px 5px;  
}  
  
#example3 {  
    background-color: blue;  
    box-shadow: -5px -5px 3px;  
}  
  
#example4 {  
    background-color: violet;  
    box-shadow: 5px 5px 3px 4px;  
}  
  
#example5 {  
    background-color: orange;  
    box-shadow: -5px -2px 3px gray, 10px 10px 5px brown;  
}
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

```
<p id="example1">Example 1</p>  
<p id="example2">Example 2</p>  
<p id="example3">Example 3</p>  
<p id="example4">Example 4</p>  
<p id="example5">Example 5</p>
```



©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

PARTICIPATION ACTIVITY**5.4.2: Box shadows.**

- 1) The **box-shadow** property creates a shadow for text.

- True
 False





2) If the **box-shadow** uses the value **inset**, then the shadow appears inside the box.

- True
- False

3) A zero **spread-radius** makes the shadow the same size as the box.

- True
- False

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

PARTICIPATION
ACTIVITY

5.4.3: Shadow practice.



HTML CSS

```
1 <div class="card">
2   <p class="question">Q: Who invented the WWW?</p>
3   <p class="answer">A: Tim Berners-Lee</p>
4 </div>
5
6 <div class="card">
7   <p class="question">Q: When was the first website published?</p>
8   <p class="answer">A: 1991</p>
9 </div>
10
11 <div class="card">
12   <p class="question">Q: What web browser did most people use in the ea
13   <p class="answer">A: Internet Explorer</p>
14 </div>
15
```

Render webpage

Reset code

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Your webpage

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Rounded corners

An element border's corners can be rounded using the CSS property **border-radius**, which is assigned one to four radius values.

- Single value - All four corners are equally rounded
- Two values - First value is top-left and bottom-right corners, second value is top-right and bottom-left corners
- Three values - First value is top-left, second is top-right and bottom-left, third is bottom-right
- Four values - First value is top-left, second is top-right, third is bottom-right, fourth is bottom-left

Each corner may also be assigned a radius using four CSS properties:

- `border-top-left-radius`
- `border-top-right-radius`
- `border-bottom-left-radius`
- `border-bottom-right-radius`

PARTICIPATION ACTIVITY

5.4.4: Rounded corners.



©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Match the square with the CSS that produces the square's rounded corners.



A



B



C



D

Mouse: Drag/drop. Refresh the page if unable to drag and drop.

D C B A

border-radius: 40px 20px 10px 5px;

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

border-radius: 40px 20px; C452DavisFall2022

border-top-left-radius: 20px;
border-bottom-right-radius: 50px;

border-radius: 15px;

Reset

Border images

The CSS property **border-image** renders an element's border using sections of an image. The border image takes the place of any border properties specified by **border-style**. The following CSS properties are specified by **border-image** all at once:

- **border-image-source** - Image URL
- **border-image-slice** - Image section size
- **border-image-repeat** - "repeat" to repeat the image section, "round" to repeat the image section but resize the image if needed to fit, or "stretch" to stretch an image section

A border image will not appear unless the **border-width** property is non-zero and **border-style** is set to any legal value except **none** and **hidden**.

PARTICIPATION
ACTIVITY

5.4.5: Try different border-image values.

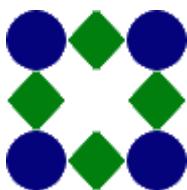


The borderv2.png image is used to display a border image around the <div> in the webpage below. The blue circles and green diamonds in the image are each 30 x 30 pixels.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022



Change the following CSS property values to see the effect on the border image:

1. Change the image slice size from 30 to 15 in the CSS property **border-image**. Render the page and observe how half the circle and half the diamond is used to render the border.
2. Change the **border-image** slice size from 15 to 60. Render the page and observe how a 60×60 pixel section (2/3 of the image) is used to render the border corners. Since borderv1.png is only 90×90 pixels, an unused 60×60 pixel section does not exist, so the border sides are empty.
3. Change the **border** size from 15px to 30px. Render the webpage and observe how the border size increased.
4. Change the **border-image** slice size back to 30, and change "round" to "repeat". Render the page and observe how the green diamonds are repeated but do not fit perfectly on the left and right sides.
5. Finally, change the **border-image** "repeat" to "stretch". Render the page and observe how the green diamonds stretch to fill the border.

HTML **CSS**

```
1 <div id="example">
2   Example using a border image.
3 </div>
4
```

Render webpage

Reset code

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Your webpage

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

PARTICIPATION ACTIVITY

5.4.6: Border images.



Refer to the CSS below.

```
border-image: url(some-border.png) 50 repeat;
```

- 1) 50 × 50 pixel sections of some-border.png are used to create the border image.

- True
 False

- 2) If some-border.png is 50 × 50 pixels, then the border will have empty sides.

- True
 False

- 3) If some-border.png is 150 × 150 pixels, the border image section is stretched on the sides.

- True
 False

- 4) To display the 50 × 50 pixel sections evenly around an element's border



©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022



image, the element's border should be specified as: `border: 50 solid;`.

- True
- False

CSS3 browser support

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Most modern browsers support CSS3, but some CSS3 properties require vendor prefixes to work on certain browsers. A **vendor prefix** is a prefix added to an experimental or nonstandard CSS property that only works on a specific browser type. Typical vendor prefixes are:

- `-webkit-` for Chrome, Safari, and newer versions of Opera
- `-moz-` for Firefox
- `-ms-` for Internet Explorer
- `-o-` for older versions of Opera

The following CSS specifies a `border-image` property for WebKit and Opera browsers:

```
#borderimg {  
    -webkit-border-image: url(border.png) 30 round; /* Safari 3.1-5 */  
    -o-border-image: url(border.png) 30 round; /* Opera 11-12.1 */  
    border-image: url(border.png) 30 round;  
}
```

Linear gradients

A CSS background may use gradient colors that transition from one color to another. Two CSS gradients exist:

1. Linear gradient - A gradient that follows a straight line
2. Radial gradient - A gradient that radiates outward into an ellipse

The CSS function **`linear-gradient(color1, color2)`** creates a linear gradient that transitions from `color1` to `color2` when moving from the top edge to the bottom edge. Additional colors can be supplied to the function. Ex: `linear-gradient(red, green, blue, yellow)` transitions from red to green to blue to yellow when moving from top to bottom.

To change the gradient's direction, the first argument to `linear-gradient` can be a direction or an angle:

- **Direction** - A direction of `left`, `right`, `top`, or `bottom` with the word `to` in front. Ex: `to left` creates a linear gradient that moves from right to left, and `to bottom right` goes from the top-left corner to the bottom-right corner.
- **Angle** - A CSS angle that points in the direction of the linear gradient. The angles `0deg`, `90deg`, `180deg`, and `270deg` correspond to `to top`, `to right`, `to bottom`, and `to left`, respectively.

©zyBooks 09/12/22 23:15 1361418

UISCSC452DavisFall2022

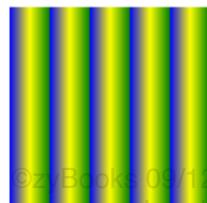
The **`repeating-linear-gradient()`** function repeats a linear gradient where the color values are supplied an optional percent. The percentage value after the last color is the percent of the gradient's total length the repeating gradient should occupy. Ex:

`repeating-linear-gradient(red, yellow 10%)` means the red to yellow gradient occupies 10% of the gradient's total length and is repeated to fill the entire background.

Figure 5.4.3: Examples of linear gradients.

```
#example1 {
    background: linear-gradient(green, yellow);
}
#example2 {
    background: linear-gradient(to bottom right, black, red);
}
#example3 {
    background: linear-gradient(45deg, red, orange, yellow, green, blue,
indigo, violet);
}
#example4 {
    background: repeating-linear-gradient(to right, blue, yellow, green 20%);
}
```

```
<div id="example1"></div>
<div id="example2"></div>
<div id="example3"></div>
<div id="example4"></div>
```



©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

PARTICIPATION ACTIVITY

5.4.7: Linear gradients.



- 1) What direction creates the gradient below?



```
background: linear-  
gradient(_____, orange,  
red);
```

**Check****Show answer**

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

- 2) What angle (direction of red arrow) creates the gradient below?

```
background: linear-  
gradient(_____, blue,  
green);
```

**Check****Show answer**

- 3) What color and percent creates the repeating linear gradient that ends in white?

```
background: repeating-  
linear-gradient(black,  
_____);
```

**Check****Show answer**

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Radial gradients

A radial gradient is created with the CSS function **`radial-gradient(color1, color2)`**, which creates an ellipse-shaped gradient that begins with `color1` in the center and ends with `color2` on the perimeter. More than two colors may be specified. A percentage or length can be placed after a color to give more emphasis to the color. Ex: `radial-gradient(red 10%, yellow 30%)` gives more emphasis to red and yellow than the default rendering.

The ellipse shape of a radial gradient fits the gradient's bounding rectangle. However, a circular radial gradient can be created with the `circle` argument. Ex:

©zyBooks 09/12/22 23:15 1361418

`radial-gradient(circle, red, yellow)` creates a circle gradient.

Jared Robbins
UISCSC452DavisFall2022

Figure 5.4.4: Examples of radial gradients.

```
#example1 {
    background: radial-gradient(red, orange);
}
#example2 {
    background: radial-gradient(red, orange 50%);
}
#example3 {
    background: radial-gradient(red 20%, orange 50%);
}
#example4 {
    background: radial-gradient(circle, red 20%, orange 50%);
}
```



PARTICIPATION ACTIVITY

5.4.8: Radial gradient.



- 1) A radial gradient is always an ellipse or circle.

- True
- False



- 2) The radial gradient below has a blue interior and a green exterior.

```
radial-gradient(green, blue);
```

- True
-

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022



False

- 3) What arguments to `radial-gradient()` create the radial gradient below?



- black, white, red
- black 40%, white, red

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UIISCSC452DavisFall2022

Positioning radial gradients

A radial gradient's ellipse or circle is centered by default in the enclosing rectangle, but the center position can be specified using "`at centerX centerY`" where `centerX` and `centerY` specify a distance or percentage. Ex: `radial-gradient(at 50px 10px, yellow, green)` specifies a center that is 50px from the left edge and 10px from the top.

By default, a radial gradient's shape reaches to the farthest corner of the containing rectangle. An `extent` keyword describes the size of the radial gradient's shape:

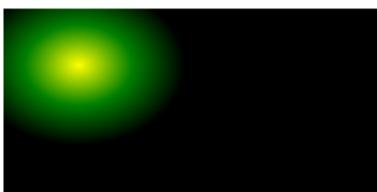
- **`closest-side`** - Circle touches the rectangle's side closest to the circle's center. Ellipse touches the vertical and horizontal sides closest to the ellipse's center.
- **`farthest-side`** - Circle touches the rectangle's side farthest from the circle's center. Ellipse touches the vertical and horizontal sides farthest from the ellipse's center.
- **`closest-corner`** - Circle or ellipse touches the corner closest to the shape's center.
- **`farthest-corner`** - Circle or ellipse touches the corner farthest from the shape's center. (Default behavior.)

PARTICIPATION ACTIVITY

5.4.9: Positioned radial gradients.



Match the background with the radial gradient CSS that produced the background.



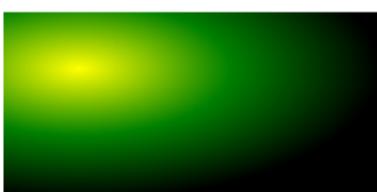
A.



B.



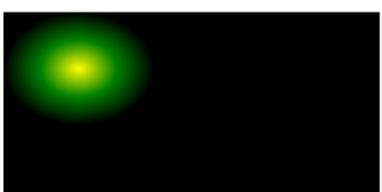
C.



D.



E.



F.

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

Mouse: Drag/drop. Refresh the page if unable to drag and drop.

F C B D A E

radial-gradient(at 60px 100px, yellow, green, black)

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

radial-gradient(yellow, green, black)

radial-gradient(closest-corner at 20% 30%, yellow, green, black)

radial-gradient(farthest-side at 20% 30%, yellow, green, black)

radial-gradient(closest-side at 20% 30%, yellow, green, black)

radial-gradient(farthest-corner at 20% 30%, yellow, green, black)

Reset

PARTICIPATION
ACTIVITY

5.4.10: Gradient practice.



The web page below displays an advertisement with a background produced by the CSS function `repeating-radial-gradient()`. Make the following modifications to the HTML and CSS so the rendered web page resembles the expected web page:

1. Add a radial gradient background to the `<body>` using any colors you prefer, and position the ellipse close to the bottom-right corner.
2. Create two more advertisements like the ads in the expected web page. Choose whatever fonts and colors you prefer. One ad should have a linear gradient background and the other a repeating linear gradient background.

HTML CSS

```
1 <div id="ad1">Vote this Tuesday!</div>  
2
```

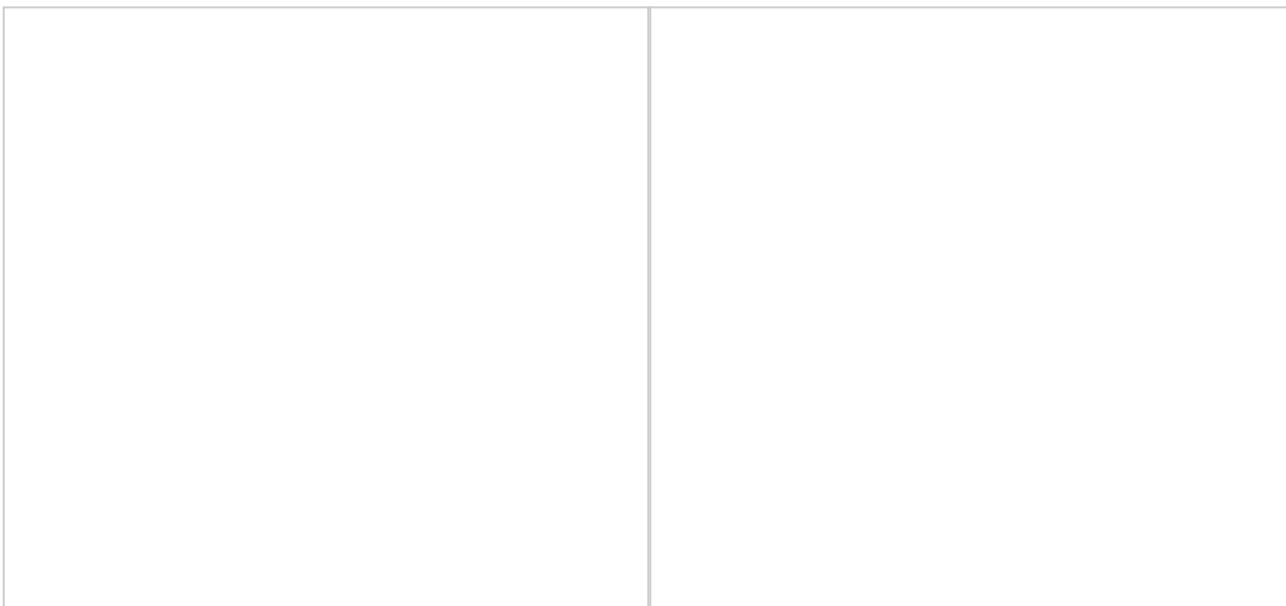
©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

Render webpage

Reset code

Your webpage

Expected webpage



CHALLENGE
ACTIVITY

5.4.1: Special effects.



422366.2722836.qx3zqy7

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

Start

For the <p> tag, set the text-shadow to be blue with an offset-x of 5px and offset-y of 12px.

SHOW EXPECTED

CSS HTML

```
1 p {  
2  
3 /* Your solution goes here */  
4  
5 }
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

1

2

3

4

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

Exploring further:

- [CSS3 text-shadow Property](#) from W3Schools
- [CSS3 box-shadow Property](#) from W3Schools
- [CSS3 Rounded Corners](#) from W3Schools
- [CSS3 border-image Property](#) from W3Schools
- [CSS3 Gradients](#) from W3Schools

5.5 Animation

Animations and keyframes

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Developers often use JavaScript and JavaScript libraries like jQuery to produce animations. However, CSS can also be used to produce animations. A **CSS animation** transforms an element's styles over a set time period, producing an animation. CSS animations have three advantages over JavaScript animations:

1. CSS animations do not require any JavaScript code.

2. CSS animations often put less load on the computer and can use techniques to produce smoother animations when the computer's resources are limited.
3. CSS animations allow the browser to more efficiently control animations and stop animations from running in browser tabs that are not visible.

A CSS animation's behavior is defined with the **@keyframes** rule, which contains a keyframe list. A **keyframe list** has a name and contains the keyframes or the properties and values that will be animated. A keyframe list contains two keyframe selectors:

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

- **from** - The animation starting state that lists the CSS properties and values that apply when the animation begins
- **to** - The animation ending state that lists the CSS properties and values that the "from" values become by the time the animation ends

Percentages may be used to specify keyframes at various points during the animation. Ex: 0% is equivalent to **from** and 100% is equivalent to **to**. The value 50% indicates the animation state at the halfway point.

To create an animation, two CSS properties must be defined:

- **animation-name** - Names the keyframe list associated with the animation
- **animation-duration** - Length of the animation in seconds (s) or milliseconds (ms)

An animation begins immediately when the browser renders the web page unless an **animation-delay** is used to delay the start of the animation.

PARTICIPATION ACTIVITY

5.5.1: Animating the background color.



Animation content:

Two blocks of code are shown. The CSS:

```
div {  
    width: 100px;  
    height: 100px;  
    background-color: orange;  
    animation-name: changeColors;  
    animation-duration: 1s;  
    animation-delay: 2s;  
}  
  
@keyframes changeColors {  
    from { background-color: blue; }  
    to { background-color: green; }  
}
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

and the HTML:

```
<div></div>
```

A screen shows the div element, displaying as an orange box. The div container includes animation properties. The browser waits 2 seconds before playing the change color animation because of the animation-delay property. Then the animation will play over 1 second, as assigned by the animation-duration, and in that time will change from blue to green. The div element is initially orange, so when the animation starts it becomes blue.

Animation captions:

1. "animation-name" names the keyframe list associated with the animation.
2. "animation-duration" specifies the animation will last 1 second.
3. "animation-delay" tells the browser to wait 2 seconds before starting the animation.
4. After the 2 second delay, the animation begins with "from", so the background is initially blue.
5. During the 1 second duration, "to" indicates the background color becomes green.
6. The background becomes orange again after the animation completes.

PARTICIPATION ACTIVITY

5.5.2: Keyframes and animation.



- 1) A valid keyframe list must define the starting (`from`) and ending (`to`) animation states.

- True
 False



- 2) A property that is listed in the `from` keyframe selector but not the `to` keyframe selector will not be animated.

- True
 False



- 3) An animation without an `animation-delay` property begins immediately.

- True
 False

©zyBooks 09/12/22 23:15 136148
Jared Robbins
UISCSC452DavisFall2022



- 4) If `animation-duration` is assigned the value 0s, the animation occurs very quickly.



True False**PARTICIPATION ACTIVITY****5.5.3: Percentages for keyframes.**

©zyBooks 09/12/22 23:15 1361418

UISCSC452DavisFall2022

The web page below shows a smiley face that moves to the right while changing colors to blue, then moves back to the left while changing colors back to red. Add two keyframes to the animation:

1. 25% through the animation, make the smiley face appear 200 pixels to the right and 100 pixels below the smiley face's starting location. The smiley face should also become yellow.
2. 75% through the animation, make the smiley face appear against the left side of the web page and 100 pixels below the smiley face's starting location. The smiley face should also become green.

Rendering the web page should animate the smiley face down and to the right, up, down and to the left, and up again while changing colors.

HTML **CSS**

```
1 <div id="smiley">#9786;</div>
2
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins

UISCSC452DavisFall2022

Render webpage**Reset code**

Your webpage

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Timing, iteration count, and direction

In the smiley face animation above, each transition from one keyframe to the next began with a slow start, then fast, then a slow end. The ***animation-timing-function*** property controls an animation's speed between keyframes. Several timing functions are available:

- **ease** - Slow start, then fast, then slow end (default)
- **linear** - Same speed throughout
- **ease-in** - Slow start
- **ease-out** - Slow end
- **ease-in-out** - Slow start and end
- **cubic-bezier(n1,n2,n3,n4)** - Specify numbers that control speed based on a Bezier curve

Other animation properties include:

- ***animation-iteration-count*** - Indicates the number of times the animation will run. The value `infinite` runs the animation repeatedly without stopping. Ex:
`animation-iteration-count: 3` runs the animation three times.
- ***animation-direction*** - Indicates animation direction
 - **normal** - Normal direction (default)
 - **reverse** - Reverse direction
 - **alternate** - Alternate between normal and reverse
 - **alternate-reverse** - Alternate between reverse and normal
- ***animation*** - Shorthand property indicating the animation name, duration, timing function, delay, iteration count, and direction. Ex: `animation: move 3s linear 2s infinite normal`.

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

PARTICIPATION ACTIVITY

5.5.4: Keyframes and timing functions.



- 1) Which keyframe selector is equivalent to `to`?

- 0%
- 100%
- 50%

- 2) Which keyframe selector specifies the animation state when the animation is three quarters finished?

- 0%
- 50%
- 75%

- 3) Which timing function makes the animation progress at the same speed the entire time?

- linear
- ease
- cubic-bezier

- 4) How many times will the animation below run?

```
animation: move 4s ease 1s 2
reverse;
```

- once
- twice
- infinite

- 5) What color is `#thing`'s font right as the animation completes?

```
#thing {
  animation: changeColors 4s
  ease 1s 2 reverse;
}
@keyframes changeColors {
  0% { color:red; }
  50% { color:blue; }
  100% { color:green; }
}
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022



- red
- blue
- green

Transitions

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

A **CSS transition** animates an element's transition from one state to another when an element's CSS property changes value. Ex: A transition may animate an element getting wider when the element's width is increased. Transitions are commonly used with the `:hover` pseudo-class to trigger an animation when the user mouses over an element.

Transitions differ from CSS animations in two ways:

1. Transitions execute when an element's property values are changed, unlike CSS animations that execute at a particular time.
2. Transitions provide less control over the animation than CSS animations.

The **transition** property defines a transition by specifying one or more CSS properties and each property's transition duration.

PARTICIPATION
ACTIVITY

5.5.5: Transitioning the width and height when hovering.



Animation content:

There are 2 blocks of code. The CSS:

```
div {  
    width: 100px;  
    height: 100px;  
    background-color: purple;  
    transition: width 1s, height 1s;  
}  
  
div:hover {  
    width: 120px;  
    height: 120px;  
}
```

and the HTML:

```
<div></div>
```

A screen shows the div element, a purple box that is 100 by 100 pixels. When the mouse hovers over the box it expands to 120 by 120 pixels over 1 second. When the mouse stops hovering, the div returns to 100 by 100

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

pixels over 1 second. The 1 second time interval is determined by the transition property. The div:hover class controls the hover animation.

Animation captions:

1. When the width property or height properties are changed, the transition to the new values will be animated over 1 second.
2. The width and height properties are increased to 120px when the mouse hovers over the <div>.
3. So, when the mouse cursor hovers over the <div>, the width and height transitions from 100px to 120px, animated over 1 second.
4. When the cursor no longer hovers over the <div>, the transition from 120px to 100px is animated over 1 second.

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

The **transition-timing-function** property controls the speed of the transition. Several timing functions are available, and all complete in the same amount of time:

- **ease** - Slow start, then fast, then slow end (default)
- **linear** - Same speed throughout
- **ease-in** - Slow start
- **ease-out** - Slow end
- **ease-in-out** - Slow start and end
- **cubic-bezier(n1,n2,n3,n4)** - Specify numbers that control speed based on a Bezier curve

The **transition-delay** property delays the transition's start.

PARTICIPATION ACTIVITY

5.5.6: Transitions.



- 1) A transition can animate one or more CSS properties.
 True
 False



- 2) The **transition** property below makes the width take 3 seconds longer than the height to complete the transition.



```
transition: width 3s, height 1s;
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

- True
- False





- 3) The #div1 transition takes longer to complete than the #div2 transition.

```
div {  
    width: 100px;  
    height: 100px;  
    transition: width 1s;  
}  
  
#div1 { transition-timing-  
function: ease-in; }  
#div2 { transition-timing-  
function: linear; }
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

- True
- False

- 4) According to the CSS below, a div element would not decrease in size until 500 milliseconds after the cursor hovered over the div.

```
div {  
    width: 100px;  
    height: 100px;  
    transition: width 1s;  
    transition-delay: 500ms;  
}  
  
div:hover {  
    width: 80px;  
}
```

- True
- False

- 5) The CSS below causes a paragraph to disappear when the mouse hovers over the paragraph.

```
p {  
    transition: opacity 500ms;  
}  
  
p:hover {  
    opacity: 0;  
}
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

- True
- False

Transformations

The **transform** property applies a 2D or 3D transformation to an element. A **transformation** is a graphical operation that alters the position, shape, or orientation of an object. The **transform** property is assigned a transformation function. A selected number of 2D transformation functions are summarized in the table below.

Transformations are used in animations and transitions to create engaging web pages.

Table 5.5.1: Selected 2D transformation functions.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Function	Description	Example
<code>translate(x, y)</code>	Moves an element on the x-axis x distance and along the y-axis y distance	<i>/* Moves right 10px and up 20px */</i> translate(10px, -20px)
<code>scale(x, y)</code>	Increases (values > 1) or decreases (values < 1) the width and height by the x and y multiplier	<i>/* Halves the width, doubles the height */</i> scale(0.5, 2)
<code>rotate(angle)</code>	Rotates clockwise by angle	<i>/* Rotates clockwise 45 degrees */</i> rotate(45deg)

PARTICIPATION ACTIVITY

5.5.7: `translate()`, `scale()`, and `rotate()` transformation functions.



Animation content:

There are two blocks of code. The CSS:

```
#example1 {
    background-color: red;
    transform: translate(30px, -10px);
}

#example2 {
    background-color: yellow;
    transform: scale(1.2, 0.5)
}

#example3 {
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

```
background-color: green;  
transform: rotate(45deg)  
}
```

and the HTML:

```
<div id="example1"></div>  
<div id="example2"></div>  
<div id="example3"></div>
```

©zyBooks 09/12/22 23:15 1361418

A screen shows a red, yellow, and green box aligned vertically. The translate() function moves the red div up and to the right. The scale() function widens and flattens the yellow div. The rotate() function turns the green div to the right.

Jared Robbins
UISCSC452DavisFall2022

Animation captions:

1. All three squares are displayed at default locations.
2. translate() moves the square 30 pixels to the right and 10 pixels up.
3. scale() multiplies the yellow square's width by 1.2 and the height by 0.5, so the square is 20% wider and 50% shorter.
4. rotate() rotates the green square 45 degrees clockwise.

PARTICIPATION ACTIVITY

5.5.8: Animations, transitions, and transformations practice.



The web page below displays a welcome message and a large button. When the cursor hovers over the button, the font color turns red, and the button grows larger. When the button is pressed, the background color is darkened.

The button transition is defined in the `.button` class and specifies that "all" properties should be animated in a transition in 100 milliseconds using the "ease-in-out" timing function. The `:hover` pseudo-class is used to scale the button 5% larger and change the font color, and the `:active` pseudo-class is used to change the background color.

Make the following modifications to the CSS so the rendered web page behaves like the expected web page:

1. Add an animation that runs as soon as the page is rendered that moves the text onto the screen. Use the `translate()` function to move the welcome message to -300px initially, then to 60px half way through the animation, then back to 0px at the end.
2. Add an animation that wiggles the button one second after the page renders. Use the `rotate()` function to initially rotate 0 degrees, then 3 degrees, then -3 degrees, and then 0 degrees. The animation should run twice.
3. Experiment with different timing functions. The expected web page uses linear timing for the two animations listed above.

4. Finally, add a transformation to the `:active` pseudo-class that uses the `scale()` function to scale the button size down 5%.

HTML CSS

```
1 <div id="welcome">Enter the exciting world of fantasy sports!</div>
2 <div class="button">Play Now!</div>
3
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Render webpage

Reset code

Your webpage

Expected webpage

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

PARTICIPATION
ACTIVITY

5.5.9: Transformations in transitions and animations.



- 1) `translate()`, `scale()`, and
`rotate()` are ____ functions.



- transition transformation
- translation
- 2) Which function moves an element 20 pixels to the left and 5 pixels down?
- `translate(-20, 5)`
- `translate(20px, -5px)`
- `translate(-20px, 5px)`



©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

- 3) What does the element using the keyframes below look like at the end of the animation?

```
@keyframes example {
  from { transform:
rotate(-45deg) scale(0.5, 0.5);
}
  to   { transform:
rotate(45deg) scale(1.5, 1.5);
}
}
```

- Rotated 45 degrees only
- Rotated -45 degrees and scaled smaller
- Rotated 45 degrees and scaled larger
- 4) What does the <p> look like when the mouse clicks the element?



```
p {
  transition: all 1s;
}
p:hover {
  transform: translate(10px,
0);
}
p:active {
  transform: scale(2, 2);
}
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

- Scaled larger
- Translated 10 pixels to the right
- No change

Exploring further:

- [CSS3 Animations](#) from W3Schools
- [CSS3 Transitions](#) from W3Schools
- [CSS3 2D Transforms](#) from W3Schools
- [CSS3 3D Transforms](#) from W3Schools

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

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 side-by-side HTML forms. The left form is unstyled, consisting of three text input fields stacked vertically with labels 'Name', 'Email', and 'Service' to their left. Below these is a dropdown menu labeled 'Basic' and a 'Register' button. The right form is styled with CSS, featuring the same elements but with improved readability. The labels are aligned to the right of their respective input fields. The 'Service' dropdown has a small arrow indicating it's a dropdown. The 'Register' button is a prominent blue rectangle with white text.

PARTICIPATION ACTIVITY

5.6.1: Create a styled form.

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022



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;
}
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

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 changing the color to blue and changing the appearance to look 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 action="https://wp.zybooks.com/form-viewer.php" target="_blank"
2   <p>
3     <label for="name">Name</label>
4     <input type="text" id="name" name="fullname">
5   </p>
6   <p>
7     <label for="email">Email</label>
8     <input type="email" id="email" name="email">
9   </p>
10  <p>
11    <label for="service">Service</label>
12    <select id="service" name="service">
13      <option>Basic</option>
14      <option>Prime</option>
15      <option>Deluxe</option>
16    </select>
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Render webpage**Reset code****Your webpage****Expected webpage****PARTICIPATION
ACTIVITY****5.6.2: Form styles.**©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

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

- display: inline-block**
- display: block**

`display: none`

- 2) What CSS selector selects only text inputs?

`input`

`input[type=text]`

`input[type=text], select`



- 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], 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.



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>
```

zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

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.



```
1 <input type="search" placeholder="Search">
2
3 <form action="https://wp.zybooks.com/form-viewer.php" target="_blank">
4   <p>
5     <label for="name">Name</label>
6     <input type="text" id="name" name="fullname">
7   </p>
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>
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Render webpageReset code

Your webpage

PARTICIPATION ACTIVITY

5.6.4: Augmented inputs.



©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

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

- 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

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

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.

Animation content:

There are two blocks of code. The CSS:

```
input[type=radio] + label {  
    cursor: pointer;  
}  
  
input[type=radio] {  
    position: absolute;  
    height: 1px; width: 1px;  
    clip: rect(0 0 1 1);  
}  
  
input[type=radio] + label::before {  
    content: "\00a0"; /* Non-breaking space */  
    border-radius: 10px;  
    display: inline-block;  
    width: 1em; height: 1em;  
    text-align: center;
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

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

input[type=radio]:checked + label::before {
    content: "\2022"; /* Bullet */
    color: white;
    background: green;
}

input[type=radio]:focus + label::before {
    box-shadow: 0 0 0 1px #888;
}
```

and the HTML:

```
<p>Size:</p>
<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>
    <input id="large" type="radio" name="size">
    <label for="large">Large</label>
</div>
```

A screen shows three radio buttons labeled Small, Medium, and Large. cursor: pointer makes the cursor change when the user hovers over the radio button or the label. The input[type=radio] block positions the radio buttons off of the screen. To create custom radio buttons, add an empty space before the labels with a circular border, this is shown in the input[type=radio] + label::before CSS block. The input[type=radio]:checked + label::before CSS block changes the custom button to green empty space with a white dot in the middle when clicked. When a radio button has the focus, input[type=radio]:focus + label::before shows a grey shadow around the empty space.

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Animation 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.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

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:

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

```
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;
}
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

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](#)

```
1 <fieldset>
2   <legend>Size:</legend>
3   <div>
4     <input id="small" type="radio" name="size" value="small">
5     <label for="small">Small</label>
6   </div>
7   <div>
8     <input id="medium" type="radio" name="size" value="medium">
9     <label for="medium">Medium</label>
10  </div>
11  <div>
12    <input id="large" type="radio" name="size" value="large">
13    <label for="large">Large</label>
14  </div>
15 </fieldset>
```

©zyBooks 09/12/23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

[Render webpage](#)[Reset code](#)**Your webpage****Expected webpage**

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

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">`

- 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

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

avoided.



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

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

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022



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

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

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

- BeforeTest
- BeforeTest
- TestBefore

CHALLENGE ACTIVITY

5.6.1: Styling forms.



422366.2722836.qx3zqy7

Start

For the `<label>` tag, set `display` to `inline-block`, use a width of `40px`, align the text to the left, and add a margin on the right of `6px`. **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 }
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

1

2

3

4

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

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

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

5.7 Sass

CSS preprocessors

Sass is a popular CSS preprocessor that uses CSS-like syntax to build complex CSS stylesheets. Other popular CSS preprocessors, like Less and Stylus, offer similar and unique features with different syntax.

The [Sass website](#) has instructions on installing the Sass preprocessor on a variety of operating systems. Some developers prefer to run the Sass preprocessor from the command line or from an application like Koala. The Sass preprocessor compiles a Sass file (.scss) into a CSS (.css) file.

Sass version 3 introduced a new syntax called **Sassy CSS (SCSS)**, which uses semicolons and brackets like CSS. Some online references still refer to the old Sass syntax which relies on indentation and has no brackets.

PARTICIPATION ACTIVITY

5.7.1: Compiling SCSS into CSS.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Animation content:

There are two blocks of code. The SCSS:

```
$font-face: Arial, sans-serif;  
$font-color: #0ff;  
body {
```

```
font: 12pt $font-face;
color: $font-color;
}
called styles.scss and the CSS
body {
  font: 12pt Arial, sans-serif;
  color: #0ff;
}
called styles.css.
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

The command-line tool sass styles.scss styles.css compiles styles.scss into styles.css. During compilation the SCSS variables that begin with \$ are inserted into the CSS.

Animation captions:

1. A .scss file contains SCSS syntax.
2. The sass command-line tool compiles styles.scss and outputs the resulting CSS to styles.css.
3. Variables begin with a \$ and are set like CSS properties.
4. The value of the variables \$font-face and \$font-color are inserted into the resulting CSS.

PARTICIPATION
ACTIVITY

5.7.2: Sass CSS preprocessor.



- 1) SCSS is syntactically different than the original Sass syntax.

- True
 False



- 2) The sass command-line tool creates a .scss file from a .css file.

- True
 False



- 3) The SCSS below results in CSS that sets a web page's background color to blue.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

```
$theme-color: blue;
body {
  background-color: $theme-
color;
}
```



True
 False

- 4) An advantage to using an SCSS variable to store a color value used multiple times in a stylesheet is that if the color needs to be changed in the future, only the variable needs to be changed.

 True
 False

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022



Nesting

Selectors may be nested in Sass to create child selectors that only apply to the parent selector. In the figure below, the `strong` child selector is nested in a `.notes` parent selector, creating a `.notes strong` selector in the resulting CSS.

Figure 5.7.1: Selector nesting.

```
// scss

.notes {
  font-size: smaller;

  strong {
    color: green;
  }
}
```

```
/* Resulting CSS */

.notes {
  font-size: smaller;
}

.notes strong {
  color: green;
}
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

The `&` character is used to reference the parent selector from a child selector's properties.

Figure 5.7.2: Referencing the parent with &.

```
// scss

a {
  text-decoration: none;
  &:hover {
    color: blue;
  }
}
```

```
/* Resulting CSS */

a {
  text-decoration: none;
}
a:hover {
  color: blue;
}
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

A number of CSS properties begin with the same prefix. Ex: `font-family`, `font-size`, and `font-weight` all begin with the same `font` prefix. Sass allows properties that share the same prefix to be nested under the prefix.

Figure 5.7.3: Property nesting.

```
// scss

p {
  font: {
    family: Arial;
    size: 12pt;
    weight: bold;
  }
}
```

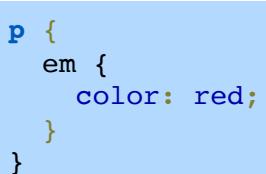
```
/* Resulting CSS */

p {
  font-family: Arial;
  font-size: 12pt;
  font-weight: bold;
}
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022



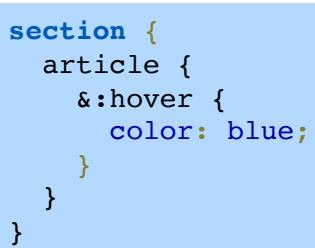
Select the CSS that results from the given SCSS.

1)  

```
p {
  em {
    color: red;
  }
}
```

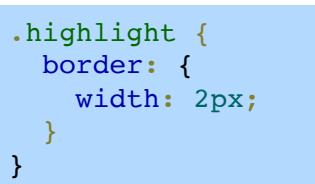
- p em {
 color: red;
}
- p + em {
 color: red;
}
- p, em {
 color: red;
}

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

2)  

```
section {
  article {
    &:hover {
      color: blue;
    }
  }
}
```

- section:hover {
 color: blue;
}
- section hover {
 color: blue;
}
- section article:hover {
 color: blue;
}

3)  

```
.highlight {
  border: {
    width: 2px;
  }
}
```

- .highlight {
 border: 2px;
}
- .highlight {
 border-width: 2px;
}

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

```
○ .highlight border {  
    width: 2px;  
}
```

Variables and arithmetic

SassScript is a set of extensions to CSS that allow properties to use variables, arithmetic, and functions. SassScript also provides basic control directives for performing conditional logic and looping.

A SassScript variable begins with a \$ and can store one of the following data types:

- Number - Any number that is optionally followed by a CSS unit. Ex: 3, 5.1, 20px
- String - "Double", 'single', or unquoted strings. Ex: "red", 'red', red
- Color - Color name or value. Ex: green, #00ff00, rgb(0,255,0)
- Boolean - true or false
- Null - null
- List of values - Separated by spaces or commas. Ex: 10px 20px 30px 40px, Helvetica, Arial, sans-serif
- Map - Key/value pairs. Ex: (111:red, 222:blue)

Basic arithmetic like addition, subtraction, multiplication, and division may be performed on numbers and numeric variables. Ex: 20px + 15 = 35px. Arithmetic on color values results in the red, green, and blue values being added, subtracted, multiplied, or divided one at a time. Ex: #0011aa + #bb2244 results in 00 + bb = bb, 11 + 22 = 33, and aa + 44 = ee; so the final value is #bb33ee.

PARTICIPATION
ACTIVITY

5.7.4: Performing SassScript arithmetic.



Animation content:

There are two blocks of code. The SCSS:

```
$width: 300;  
$size: 20pt;  
$color: #220055;  
p {  
    width: $width - 50px;  
    font-size: $size * 0.9;  
    line-height: $size / 2 + 14  
    color: $color * 2;  
}  
and the CSS
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

```
p {
  width: 250px;
  font-size: 18pt;
  line-height: 24pt;
  color: #4400aa;
}
```

When the SCSS is compiled to the CSS, the calculations for properties are performed. The width property in CSS is calculated by \$width - 50px or 300 - 50px to get 250px. The font-size property in CSS is calculated by multiplying \$size * 0.9 or 20pt * 0.9 to get 18pt. The line-height property in CSS is calculated by \$size / 2 + 14 or 20pt / 2 + 14 to get 24pt. The color property in CSS is calculated by multiplying \$color * 2 or #220055 * 2 to get #4400aa. The multiplication is done by multiplying each color value by 2, to get red 2 * 22, to get green 2 * 00, and to get blue 2 * 55.

Animation captions:

1. \$width and \$size store numbers, but \$color stores a color.
2. 50px is subtracted from the value of the \$width variable.
3. The variable \$size is multiplied by 0.9.
4. \$size is divided by 2, and then 14 is added to the result.
5. Multiplying \$color by 2 results in red, green, and blue components each being multiplied by 2.

PARTICIPATION ACTIVITY

5.7.5: Variables and arithmetic.



What is \$value?

1) **\$value: 20px - 15;**

Check

Show answer

2) **\$value: 20pt + (10 / 2) ;**

Check

Show answer

3) **\$value: #ff1150 - #001120;**

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022



4) `$value: #ff1150 + 2;`

©zyBooks 09/12/22 23:15 1361418
 Jared Robbins
 UISCSC452DavisFall2022

Functions

SassScript includes a large number of utility functions.

Table 5.7.1: Some SassScript functions.

Function	Description	Example
<code>lighten(color, amount)</code>	Returns a <code>color</code> lightened by an <code>amount</code> between 0% and 100%	<code>/* Returns #d00 */ \$color: lighten(#a00, 10%);</code>
<code>invert(color)</code>	Returns the inverse (negative) of a <code>color</code>	<code>/* Returns #5ff */ \$color: invert(#a00);</code>
<code>to-upper-case(string)</code>	Returns <code>string</code> using all uppercase characters.	<code>/* Returns "BEHOLD!" */ \$message: to-upper-case("Behold!");</code>
<code>round(number)</code>	Returns a <code>number</code> rounded to the nearest whole number	©zyBooks 09/12/22 23:15 1361418 <code>/* Returns 21px */ \$width: round(20.5px);</code>
<code>random(limit)</code>	Returns a random integer between 1 and <code>limit</code> (inclusive)	

```
/* Returns a  
number between 1  
and 5 */  
\$width: random(5)  
* 20px;
```

**PARTICIPATION
ACTIVITY**

5.7.6: SassScript functions.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022



What is \$value?

1) **\$value: lighten(black, 20%);**

- #000
- white
- #333

2) **\$value: invert(white);**

- black
- white
- gray

3) **\$value: round(16.4pt);**

- 16.4pt
- 16pt
- 17pt

4) **\$value: random(3) * 100px;**

- 0, 1, 2, or 3
- 1, 2, or 3
- 100px, 200px, or 300px

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

**Mixins**

A **mixin** is set of reusable styles and is defined by the **@mixin** directive. A **directive** is an extension to the CSS at-rules, which are statements that begin with the @ character. Mixins may take arguments, which give mixins the ability to customize the styles that the mixin defines. Mixins are included in a document using the **@include** directive.

PARTICIPATION



Animation content:

There are two blocks of code. The SCSS:

```
@mixin cool-font {  
    font: italic small-caps 16pt cursive;  
    color: darkgreen;  
}  
  
@mixin highlight($color, $width) {  
    border: $width solid $color;  
    box-shadow: 4px darken($color, 20%);  
}  
  
.special {  
    background-color: #eee;  
    @include cool-font;  
    @include highlight(red, 2px);  
}
```

and the CSS:

```
.special {  
    background-color: #eee;  
    font: italic small-caps 16pt cursive;  
    color: darkgreen;  
    border: 2px solid red;  
    box-shadow: 4px #990000;  
}
```

When the SCSS is compiled to the CSS, cool-font's attributes are transferred directly to the CSS, and highlight's attributes are first filled with the two arguments then transferred to the CSS.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Animation captions:

1. Two mixins are defined: cool-font and highlight.
2. The special class includes the cool-font mixin.
3. The highlight mixin is included with two arguments, red and 2px, which are assigned to \$color and \$width.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022



Given the mixins below, match the SCSS with the resulting CSS.

```
@mixin shadow-font {  
  font-size: 12pt;  
  text-shadow: 2px 2px blue;  
}  
  
@mixin pretty-border($img, $size) {  
  border: 10px solid transparent;  
  padding: 20px;  
  border-image: url($img) $size round;  
}
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Mouse: Drag/drop. Refresh the page if unable to drag and drop.

```
div {  
  @include shadow-font;  
}
```

```
div {  
  @include pretty-border(  
    "border.png", 30);  
}
```

```
div {  
  @include shadow-font;  
}  
ol {  
  @include shadow-font;  
}
```

```
div {  
  @include pretty-border(  
    "border.png", 30);  
  @include shadow-font;  
}
```

```
div {  
  font-size: 12pt;  
  text-shadow: 2px 2px blue;  
}
```

```
div {  
  border: 10px solid transparent;  
  padding: 20px;  
  border-image: url("border.png") 30 round  
}
```

```
div {  
  border: 10px solid transparent;  
  padding: 20px;  
  border-image: url("border.png") 30 round  
  font-size: 12pt;  
  text-shadow: 2px 2px blue;  
}
```

```
div {
```

```
        font-size: 12pt;
        text-shadow: 2px 2px blue;
    }
ol {
    font-size: 12pt;
    text-shadow: 2px 2px blue;
}
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins
UISCSC452DavisFall2022**Reset**

Control directives and expressions

Sass contains other features *including*:

- Control directives, like `@if` and `@for`, that support conditional styling and looping
- Ability to import SCSS and Sass files using the `@import` directive
- Ability to extend the styles in a class with the `@extend` directive
- Ability to write custom functions

See the Sass website's [documentation](#) for more details.

Exploring further:

- [Sass](#)
- [Koala](#)
- [Less](#)
- [Stylus](#)

©zyBooks 09/12/22 23:15 1361418

Jared Robbins
UISCSC452DavisFall2022

5.8 Example: Styled Restaurant Reviews

Styling the home page

This section adds CSS to the example restaurant review website from an earlier section. The restaurant review website displays user reviews of local restaurants.

Home page.

The index.html file is the home page, and the external stylesheet styles.css is used by all website's pages to present a consistent look. Some special characteristics of the HTML below include:

Jared Robbins
UISCSC452DavisFall2022

- A subtle gray text shadow on the "Restaurant Reviews" header, produced with the `text-shadow` property.
- The use of an unordered list to create the navigation links. The bullets are removed from the list with the CSS `list-style-type:none`, and the `` elements are displayed side by side by changing the `` elements to inline elements with `display:inline`.
- Featured restaurant sections with a rounded border, produced using the CSS `border-radius` property.
- A flexbox to put the stars and description to the right of the restaurant photo.

Images from [Wikimedia.org](#)

[index.html](#) [styles.css](#)

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>Restaurant Reviews</title>
6     <link rel="stylesheet" href="styles.css">
7   </head>
8   <body>
9     <header>
10       <h1>Restaurant Reviews</h1>
11       <nav>
12         <ul>
13           <li><a href="index.html">Home</a></li>
14           <li><a href="addreview.html">Add Review</a></li>
15           <li><a href="about.html">About</a></li>
16 
```

[Render webpage](#)

[Reset code](#)

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Your webpage

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

PARTICIPATION ACTIVITY

5.8.1: Home page.



- 1) Increasing the first two numbers in the `text-shadow` property value for the header moves the text shadow further away from the text.

True
 False



- 2) The link text to Main Street Cafe and Greek House appears red when the mouse hovers on top of the links.

True
 False



- 3) The navigation link text appears red when the mouse hovers on top of the links.

True
 False



©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

- 4) Adding `border-radius:10px;` to the `nav li` rule adds rounded corners



to the red rectangle that appears around navigation links.

- True
- False

5) The flex container has 3 flex items.

- True
- False

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

6) Changing the `flex-direction` property to `row-reverse` puts the restaurant photo on the right side of the stars and description.

- True
- False



Show an advertisement

The home page may be modified to display an advertisement in the upper-right corner by using fixed positioning. To draw the user's attention to the advertisement, some CSS animations can drop the advertisement onto the web page after a short delay. The `animation-*` properties in the figure below use the keyframe `lowerAd` to lower the advertisement onto the screen by changing the advertisement's `top` property from `-80px` to `40px` over a 1 second duration. The `animation-fill-mode: forwards` declaration makes the advertisement retain the `top: 40px` value from the animation.

Figure 5.8.1: HTML and CSS for the advertisement.

```
<div id="advertisement">
  <a href="about.html">Add your advertisement here!
</a>
</div>
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

```
@keyframes lowerAd {  
    from { top: -80px; }  
    to { top: 40px; }  
}  
  
#advertisement {  
    animation-name: lowerAd;  
    animation-duration: 1s;  
    animation-delay: 2s;  
    animation-fill-mode: forwards;  
    position: fixed;  
    right: 30px;  
    top: -80px;  
    width: 100px;  
    background: linear-gradient(lightgreen, white);  
    box-shadow: 5px 5px;  
    padding: 10px;  
    border-radius: 4px;  
}
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

The advertisement can expand in size when the mouse is hovering over the advertisement by using the CSS `transform` property to scale the advertisement larger. The CSS `transition` property animates the expanding advertisement over 200 ms.

Figure 5.8.2: CSS for expanding advertisement.

```
#advertisement {  
    animation-name: lowerAd;  
    animation-duration: 1s;  
    animation-delay: 2s;  
    animation-fill-mode: forwards;  
    position: fixed;  
    right: 30px;  
    top: -80px;  
    width: 100px;  
    background: linear-gradient(lightgreen,  
white);  
    box-shadow: 5px 5px;  
    padding: 10px;  
    border-radius: 4px;  
    transition: all 200ms ease-in-out;  
}  
  
#advertisement:hover {  
    transform: scale(1.2, 1.2);  
}
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Animated advertisement.

After the page is rendered, the advertisement falls onto the web page after a 2 second delay. Click the "Render web page" button to reset the animation. Place the mouse on top of the advertisement to see the advertisement grow.

©zyBooks 09/12/22 23:15 1361418
index.html styles.css Jared Robbins
UISCSC452DavisFall2022

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>Restaurant Reviews</title>
6     <link rel="stylesheet" href="styles.css">
7   </head>
8   <body>
9     <header>
10       <h1>Restaurant Reviews</h1>
11       <nav>
12         <ul>
13           <li><a href="index.html">Home</a></li>
14           <li><a href="addreview.html">Add Review</a></li>
15           <li><a href="about.html">About</a></li>
16 
```

Render webpage

Reset code

Your webpage

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022



- 1) Which CSS property should be modified to make the advertisement drop immediately when the page is rendered?

- animation-name
- animation-duration
- animation-delay

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022



- 2) What CSS property is needed to make the advertisement slide in from the right?

```
@keyframes lowerAd {  
    from { ____: -100px; top:  
40px; }  
    to { ____: 30px; top: 40px;  
}  
}
```

- left
- right
- bottom

- 3) What CSS makes the advertisement scale and tilt right when the mouse hovers over the ad?

```
#advertisement:hover {  
    transform: scale(1.2, 1.2)  
    ____;  
}
```

- translate(-5px, 5px)
- scaleZ(-1.5)
- rotate(5deg)

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Styling the detail pages

The detail page for each restaurant displays additional restaurant details and user reviews. Two CSS rules are added to styles.css to place the overall rating stars on top of the restaurant photo:

- **.restaurant-photo** - Adds relative positioning to the section that displays the restaurant photo and overall rating stars
- **.stars** - Positions overall rating stars on top of the restaurant photo with absolute positioning

Figure 5.8.3: CSS for styling the reviews and overall rating stars.

```
.restaurant-photo {
    position: relative;
}

.stars {
    position: absolute;
    top: 220px;
    left: 10px;
    width: 180px;
}
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

The user reviews are layed out with a grid layout. The **.review** rule creates the grid container with 2 columns and lays out the grid using named grid items. The grid items are assigned names in the classes **.review-name**, **.review-date**, **.review-rating**, and **.review-details**.

Figure 5.8.4: Orange grid lines show the grid layout for each review.

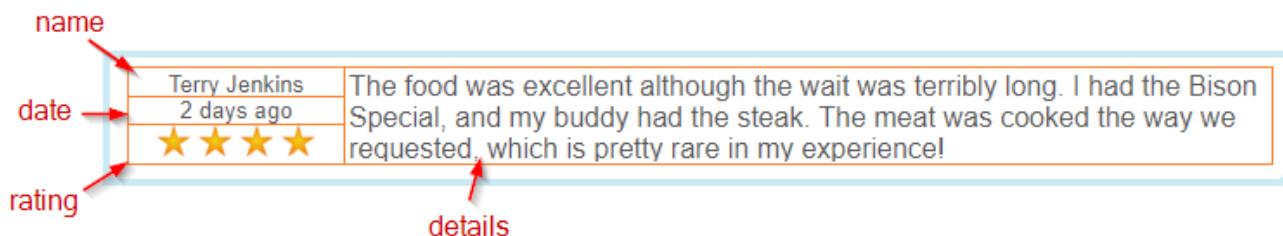


Figure 5.8.5: CSS for the user rating grid layout.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

```
.review {  
    display: grid;  
    grid-template-columns: 130px  
auto;  
    grid-template-areas:  
        "name details"  
        "date details"  
        "rating details";  
    border-radius: 5px;  
    background: white;  
    padding: 8px 8px;  
    margin: 10px 0;  
    font-size: smaller;  
}  
  
.review-name {  
    grid-area: name;  
    text-align: center;  
}  
  
.review-date {  
    grid-area: date;  
    text-align: center;  
}  
  
.review-rating {  
    grid-area: rating;  
    text-align: center;  
}  
  
.review-rating > img {  
    width: 20px;  
}  
  
.review-details {  
    grid-area: details;  
    font-size: larger;  
}
```

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Restaurant review details.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

mainstreetcafe.html

styles.css

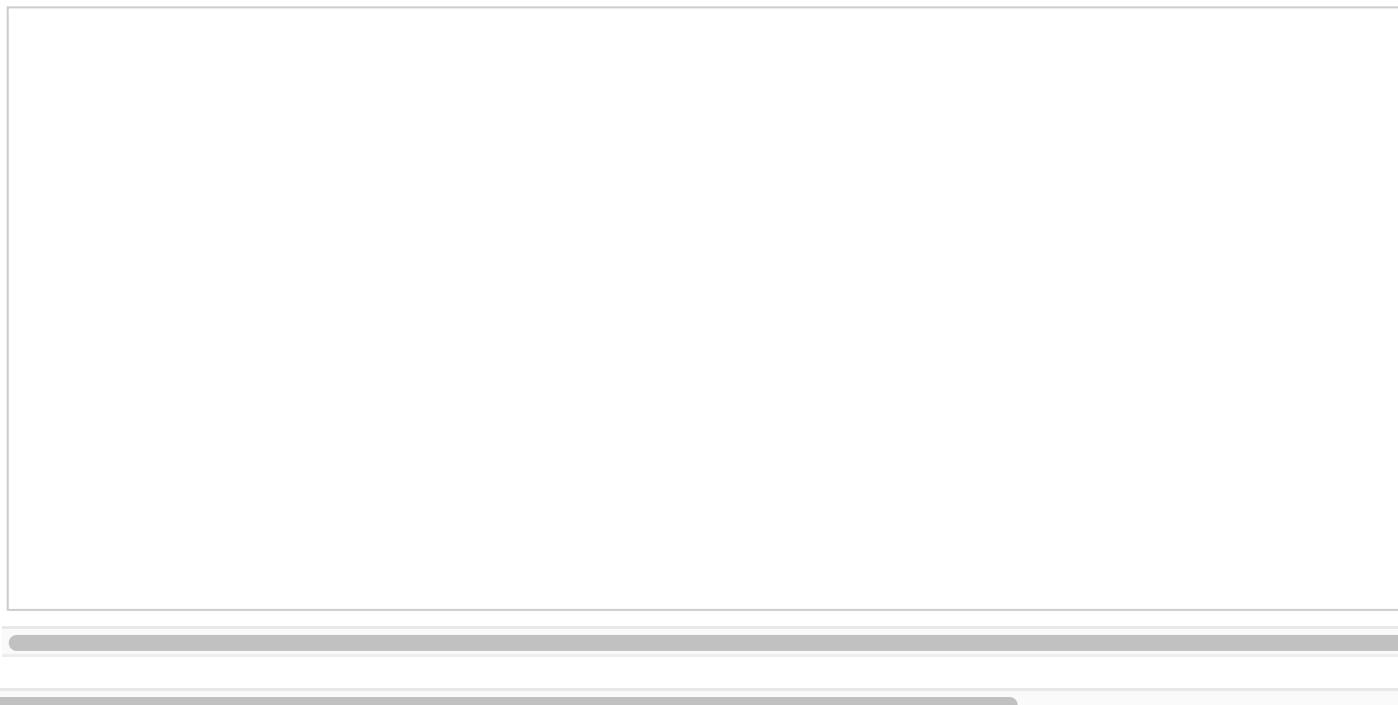
```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>Restaurant Reviews: Main Street Cafe</title>
6     <link rel="stylesheet" href="styles.css">
7   </head>
8   <body>
9     <header>
10       <h1>Restaurant Reviews</h1>
11       <nav>
12         <ul>
13           <li><a href="index.html">Home</a></li>
14           <li><a href="addreview.html">Add Review</a></li>
15           <li><a href="about.html">About</a></li>
16 
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Render webpage

Reset code

Your webpage



PARTICIPATION ACTIVITY

5.8.3: Detail page.

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022



- 1) The reviews would have a blue background if `background:white` was removed from the `.review` rule.

- True
 False





2) The `top` and `left` values in the `.stars` rule position the stars relative to the browser's top and left edges.

- True
- False

3) Changing the `.stars` rules declaration from `position: absolute` to `position: fixed` changes nothing about the rendered web page.

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

- True
- False

4) The star images only appear on top of the restaurant image because the star images are defined later in the HTML.

- True
- False

5) The grid layout used to lay out the user reviews has 2 total grid items.



- True
- False

6) Which CSS property must be modified to change the number of rows the review detail spans?



- `grid-template-columns`
- `grid-template-areas`

Styling the form

The form on the addreview.html page is improved by fixing the width of the form labels, adding a margin between rows, and adding padding, sizing, and better borders to the form widgets.

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

PARTICIPATION ACTIVITY

5.8.4: Styling the form.



Match the letter to the missing CSS.

```

label {
  width: 100px;
  display: __A__;
  text-align: right;
  margin-right: 8px;
}

form div {
  margin-bottom: 8px;
}

input, select, textarea{
  width: 350px;
  padding: 10px;
  border: 1px solid #aaa;
  border-radius: 4px;
  box-sizing: __B__;
}

input[__C__], input[type=date] {
  width: auto;
}

button {
  width: 100px;
  background-color: var(--accent-color);
  color: white;
  padding: 15px;
  margin: 10px 0;
  border: none;
  border-radius: 4px;
  cursor: __D__;
}

button:hover {
  background-color: var(--accent-dark-color);
}

```

©zyBooks 09/12/22 23:15 1361418
 Jared Robbins
 UISCSC452DavisFall2022

Mouse: Drag/drop. Refresh the page if unable to drag and drop.

B **D** **A** **C**

pointer

©zyBooks 09/12/22 23:15 1361418
 Jared Robbins
 UISCSC452DavisFall2022

border-box

inline-block

type=number

Reset

Styling the form.

[addreview.html](#) [styles.css](#)

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>Restaurant Reviews - Add Review</title>
6     <link rel="stylesheet" href="styles.css">
7   </head>
8   <body>
9     <header>
10       <h1>Restaurant Reviews</h1>
11       <nav>
12         <ul>
13           <li><a href="index.html">Home</a></li>
14           <li><a href="addreview.html">Add Review</a></li>
15           <li><a href="about.html">About</a></li>
16 
```

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022[Render webpage](#)[Reset code](#)

Your webpage

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

5.9 LAB: CSS practice

Create a web page for the following poem/song, entitled "Yesterday":

Yesterday, All those backups seemed a waste of pay. Now my database has gone away. Oh I believe in yesterday.

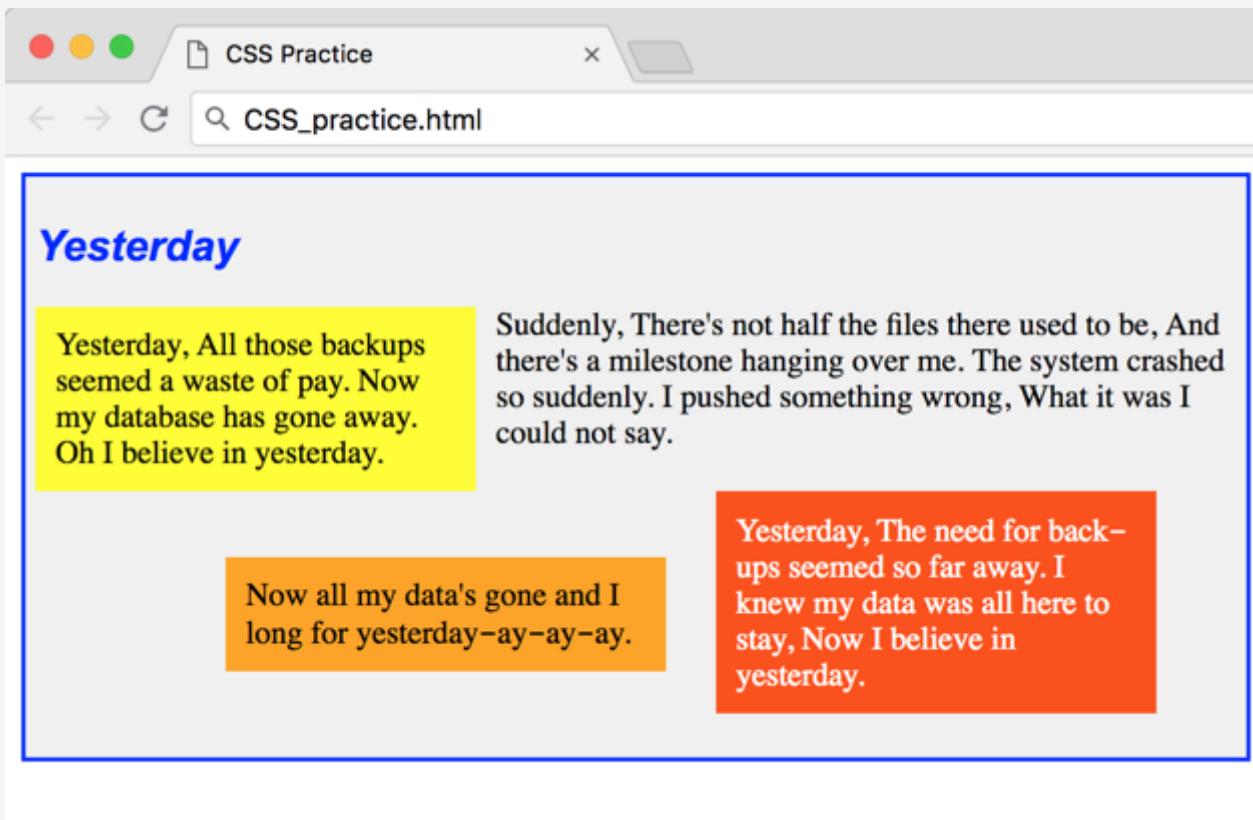
Suddenly, There's not half the files there used to be, And there's a milestone hanging over me. The system crashed so suddenly. I pushed something wrong, What it was I could not say.

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

Now all my data's gone and I long for yesterday-ay-ay-ay.

Yesterday, The need for back-ups seemed so far away. I knew my data was all here to stay, Now I believe in yesterday.

Use CSS to format the web page so it looks exactly like the screenshot below:



Requirements

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UISCSC452DavisFall2022

1. Use either an embedded stylesheet or inline styles for all formatting.
2. Use a `<div>` for each of the 4 verse containers and the outer container.
3. The title "Yesterday" uses an `<h2>` tag with a 16pt Arial font that is blue and italicized. Do not use an `<i>` tag.

4. The entire song is embedded in a container that is 600 x 280 pixels, has a two pixel solid blue border, and a five pixel padding. The background color is #F0F0F0.
5. The outer <div>'s first child is the <h2>. The 4 child <div>s follow, declared in the order listed below.
6. The yellow section:
 - has a background color of "yellow"
 - floats to the left
 - has a 200 pixel width
 - has a 10 pixel padding
 - has a right margin of 10 pixels, with all other margins set to 0
7. The section with no background color has no CSS styling.
8. The orange section:
 - has a background color of "orange"
 - is positioned at (110px, 200px) with absolute positioning
 - has a 200 pixel width
 - has a 10 pixel padding
9. The orange-red section:
 - has a background color of "rgb(255, 80, 0)"
 - has white text
 - floats to the right
 - has a 200 pixel width
 - has a 10 pixel padding
 - has a right margin of 40 pixels, a top margin of 20 pixels, and all other margins set to 0
10. Do not use
 anywhere in your HTML.

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

422366.2722836.qx3zqy7

LAB ACTIVITY

5.9.1: LAB: CSS practice

0 / 10



Submission Instructions

Downloadable files

[index.html](#)

[Download](#)

Upload your files below by dragging and dropping into the area or choosing a file on your hard drive.

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022

index.html

Drag file here

or

[Choose on hard drive.](#)

[Submit for grading](#)Coding trail of your work [What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Latest submission

No submissions yet

5.10 LAB: Tic-tac-toe game with grid layout



This section's content is not available for print.

5.11 LAB: Position the playing cards



This section's content is not available for print.

5.12 LAB: Animating the answer

©zyBooks 09/12/22 23:15 1361418

Modify the given HTML file to produce an animation, resulting in the web page below.

Jared Robbins
UISCSC452DavisFall2022**Question: $\frac{1}{4} + \frac{1}{2} = \frac{3}{4}$**

Perform the following to the embedded stylesheet:

- Add a keyframe called `moveFraction` that does the following:

- At 0% sets the font color to yellow and uses `transform` property to translate to position (-355px, 60px) so the fraction is off the left side of the screen.
 - At 50% translates to (0px, 60px) so the fraction is just below and right of the question.
 - At 100% sets font color to red and translates to (0px, 0px) so the fraction is located next to the = sign.
- In the `#answer` rule start the `moveFraction` animation 1 second after the CSS loads, and make the animation duration 2 seconds.
 - In the `#answer` rule use `animation-fill-mode: forwards` to keep the fraction from resetting back to the original location after the animation completes.
 - Add a `transition` property to the `#question` rule to perform a `transform` over 0.6 seconds with the `ease-in-out` timing function.
 - Add a `transform` property to the `#question:hover` rule to scale by 0.9.

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Verify that when the page loads, the 3/4 fraction moves from off the screen to the right, just under the question, then up next to the question. When the mouse hovers over the question, the text should shrink some until the mouse is moved off the text.

422366.2722836.qx3zqy7

**LAB
ACTIVITY****5.12.1: LAB: Animating the answer**

0 / 10

**Submission Instructions**

Downloadable files

[question.html](#)[Download](#)

Upload your files below by dragging and dropping into the area or choosing a file on your hard drive.

question.html

Drag file here

or

[Choose on hard drive.](#)**Submit for grading**

©zyBooks 09/12/22 23:15 1361418

Jared Robbins

UISCSC452DavisFall2022

Coding trail of your work

[What is this?](#)

History of your effort will appear here once you begin working on this zyLab.

Latest submission

No submissions yet

©zyBooks 09/12/22 23:15 1361418

Jared Robbins
UIISCSC452DavisFall2022

5.13 LAB: Recipe with flexbox



This section's content is not available for print.

©zyBooks 09/12/22 23:15 1361418
Jared Robbins
UIISCSC452DavisFall2022