



Students:  
Section 5.1 is a part of 1 assignment:  
**Homework Assignment #8: More CSS (Take 1)**

Includes: PA CA  
Due: 04/13/2024, 11:59 PM  
PDT

## 5.1 Flexbox

### 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.

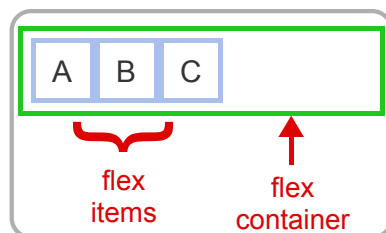
A **flex container** is an element that has the CSS property `display` set to `flex` to create a block-level flex container or `inline-flex` to create an inline flex container. Ex: `<div style="display: flex">`. Flex containers hold flex items. A **flex item** is a child element of a flex container that is positioned and sized according to various CSS flexbox properties.

#### PARTICIPATION ACTIVITY

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

**Start** ☐ 2x speed

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



```
/* flex container */
#container {
  display: flex;
  border: 1px green solid;
  padding: 5px;
}

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

Captions ▼

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

## 5.1.2: 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

[Feedback?](#)

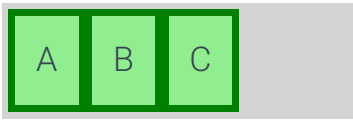
## Flex container properties

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

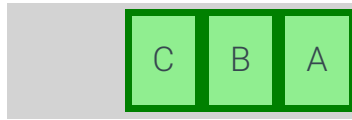
- The ***flex-direction*** property defines the direction of flex items within the container using

values:

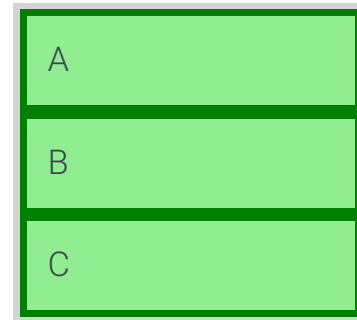
row (default)



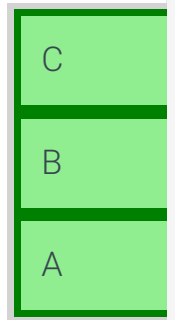
row-reverse



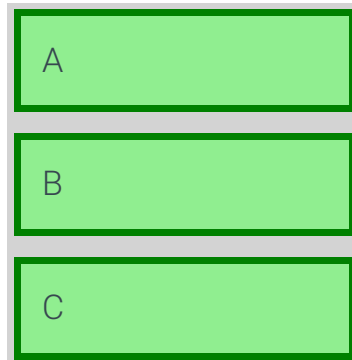
column



column-reverse



- The **gap** property defines the space between flex items. Ex: **gap: 10px;** puts a 10px gap between all items.



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

flex-start (default)



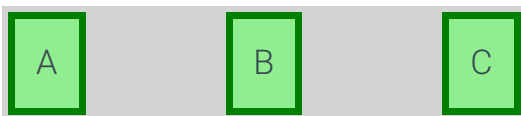
flex-end



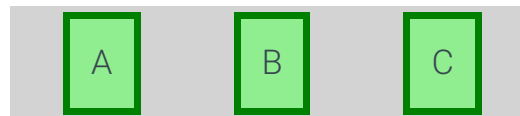
center



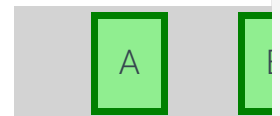
space-between



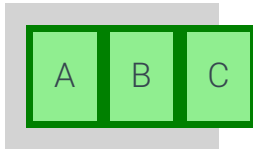
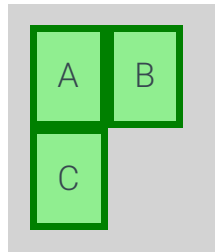
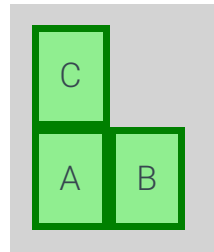
space-around



space-evenly



- 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**

## 5.1.3: Flex container properties for photos.



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

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

1. Add a **gap** property to the flex container to put a 20px gap between the photos.
2. Add a **flex-direction** property to the flex container to display the photos in reverse.
3. Add a **justify-content** property to the flex container to display the photos centered in the browser window.
4. Add a **flex-wrap** property to the flex container so the photos wrap to the next row when the browser width is reduced.

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

**HTML****CSS**

```
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

[Feedback?](#)

## 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.

### PARTICIPATION ACTIVITY

#### 5.1.5: Changing flex item properties.

☐ 2x speed

```
<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>
```

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

Home Products About

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

Home Products About

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

Home Products About

Captions ▼

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

## 5.1.6: Flex item properties.



Refer to the webpage below.

```
<!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>
```

1) How many flex containers exist in the webpage?

- ☐ 0
- ☐ 1
- ☐ 2

2) The heading "ACME Widgets" appears \_\_\_\_ the navigation links in the browser.

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

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 webpage
- ☐ Entire width of the webpage

- 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;`

[Feedback?](#)

## 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

#### 5.1.7: Flexbox layout using the flex property.

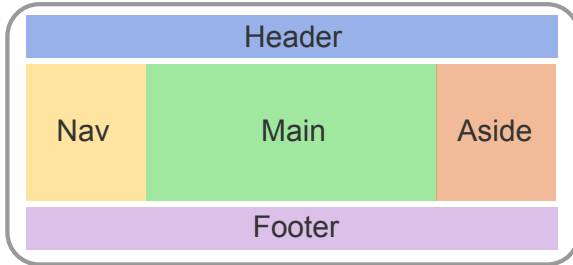
[Start](#)☐ 2x speed

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

```
#container {  
  display: flex;  
}  
  
nav {  
  flex: 0 1 20%;  
}
```

```
</div>

<footer>Footer</footer>
</body>
```



```
main {
  flex: 0 1 60%;
}

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

flex-grow      flex-shrink      flex-basis

Captions ▼

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

## 5.1.8: 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 webpage?

```
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 webpage?



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

- ☐ 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.



4) How does the change below affect the webpage?

```
#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.

[Feedback?](#)

**PARTICIPATION  
ACTIVITY**

5.1.9: 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.

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 webpage](#)[Reset code](#)

Your webpage

# Grammy Nominees for Album of the Year

## 2020

- *Folklore* by Taylor Swift
- *Future Nostalgia* by Dua Lipa
- *Hollywood's Bleeding* by Post Malone
- *Chilombo* by Jhené Aiko
- *Black Pumas (Deluxe Edition)* by Black Pumas
- *Everyday Life* by Coldplay
- *Djesse Vol.3* by Jacob Collier
- *Women In Music Pt. III* by HAIM

## 2019

[► View solution](#)[Feedback?](#)

CHALLENGE  
ACTIVITY

## 5.1.1: Flexbox.



550544.4142762.qx3zqy7

Start



1



2



3

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 }
```

1

2

3

Check

Next

[Feedback?](#)

Exploring further:

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

How was  
this  
section?



Provide section feedback

Activity summary for assignment: Homework Assignment #8:

0 / 91  
points

More CSS (Take 1)

Due: 04/13/2024, 11:59 PM PDT

0 / 91 points submitted to canvas

Completion details ^

Section 5.1      0 / 24 points ^



Participation activities



Challenge activities

- 5.1.1    0 / 1 point
- 5.1.2    0 / 3 points
- 5.1.3    0 / 1 point
- 5.1.4    0 / 3 points
- 5.1.5    0 / 1 point
- 5.1.6    0 / 6 points
- 5.1.7    0 / 1 point
- 5.1.8    0 / 4 points
- 5.1.9    0 / 1 point

5.1.1    0 / 3 points

Next section

Section 5.2      0 / 22 points v

Section 5.3      0 / 19 points v

Section 5.4      0 / 26 points v