

Flexbox

MMP 350 Fall 2020

CSS Flexbox Layout

While CSS Grid is used for imposing a layout on the entire webpage, the Flexbox layout (Flexible Box) is used to create a flexible or responsive layout in one section with multiple elements. Advantages of Flexbox include:

- Flexbox can be used for horizontally or vertically oriented layouts.
- Page content can be laid out in any direction, including left, right, downwards and even upwards.
- Items can “flex” their sizes and re-align themselves based on the container.
- Easier equal-column layouts.

flexbox vs block

Before Flexbox, there were four layout modes:

- Block - for sections in a webpage
- Inline - for text
- Table - for 2-D data
- Positioned - for explicit position of elements

Compare the menu below as a block vs a flex.

```
<div class="menu">  
  <div class="menu-item">Home</div>  
  <div class="menu-item">Profile</div>  
  <div class="menu-item">About</div>  
  <div class="menu-item">Users</div>  
</div>
```

```
.menu { display: block; }
```

Home
Profile
About
Users

```
.menu { display: flex; }
```

HomeProfileAboutUsers

Flex container

A Flexbox grid is comprised of two parts, the container and the items within it.

One set of rules is used to set up the container, or parent, while rules are added to individual items, or children, to determine their behavior.

Elements with `display: flex` will attempt to fill the available space with the content of the child elements.

```
<div class="menu">
  <div class="menu-item">Home</div>
  <div class="menu-item">Profile</div>
  <div class="menu-item">About</div>
  <div class="menu-item">Users</div>
</div>
```

```
.menu { display: flex; }
```

HomeProfileAboutUsers

justify-content

The child elements in a flex container can be aligned to the space using the justify-content rule.

.menu { justify-content: flex-start; }

HomeProfileAboutUsers

.menu { justify-content: flex-end; }

HomeProfileAboutUsers

.menu { justify-content: center; }

HomeProfileAboutUsers

.menu { justify-content: space-between; }

Home Profile About Users

.menu { justify-content: space-around; }

Home Profile About Users

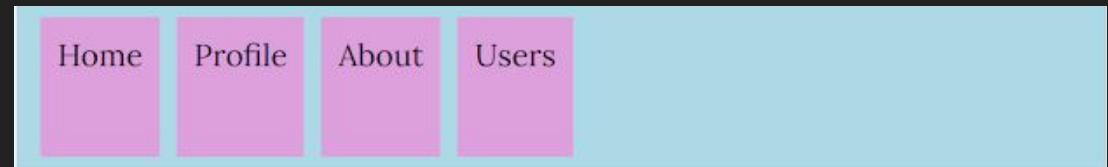
.menu { justify-content: space-evenly; }

Home Profile About Users

align-items

align-items is used to determine the alignment of the children in the opposing axis of the justify-content.

```
.menu { align-items: stretch; }
```



```
.menu { align-items: flex-start; }
```



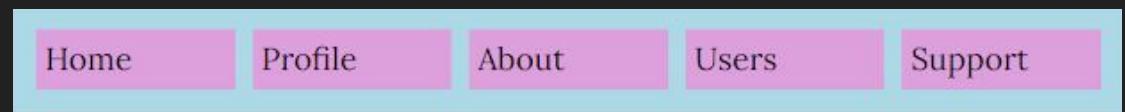
```
.menu { align-items: flex-end; }
```



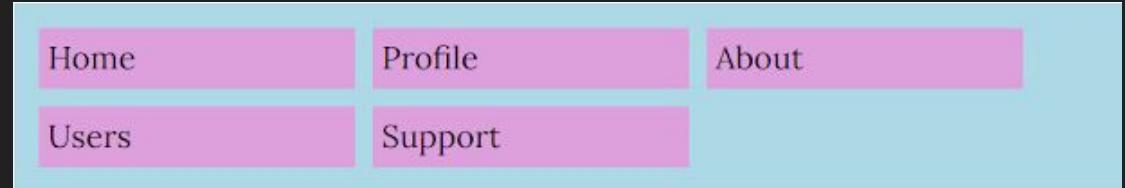
flex-wrap

flex-wrap determines whether or not content will wrap to a new column or row when it overflows the space of the parent element.

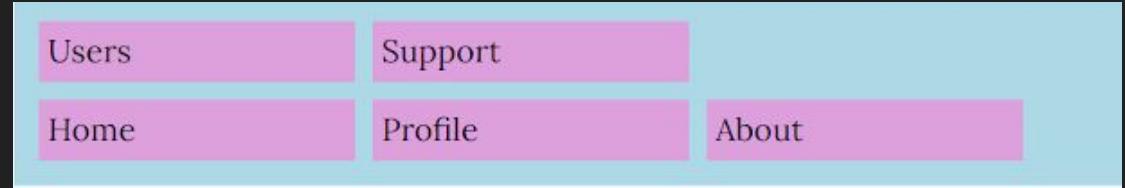
```
.menu { flex-wrap: nowrap; }
```



```
.menu { flex-wrap: wrap; }
```



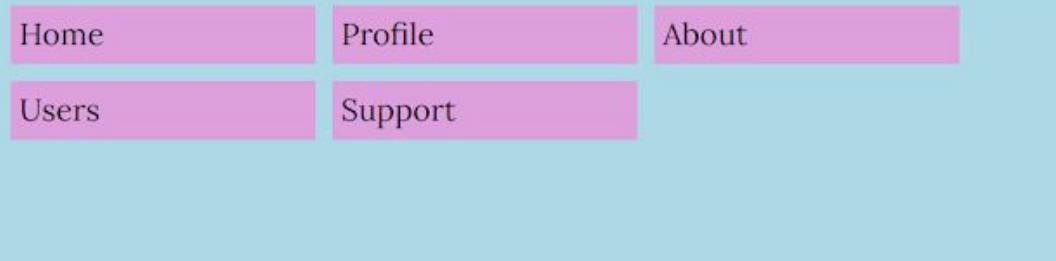
```
.menu { flex-wrap: wrap-reverse; }
```



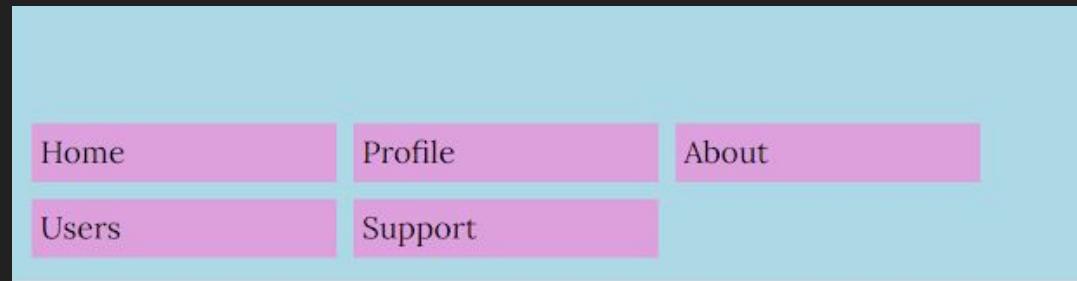
align-content

align-content is used to determine the alignment of all the content as a group within the parent container.

```
.menu { align-content: flex-start; }
```



```
.menu { align-content: flex-end; }
```



flex-direction: column

Most of the time Flex Box is used to layout items in horizontal space, but all of the same rules can be applied to a vertical layout by changing the flex-direction. The default is row, which distributes elements left to right in the layout, column distributes elements from top to bottom, and it can be reversed.

```
.menu { flex-direction: column; }
```

Home
Profile
About
Users
Support

```
.menu { flex-direction: column-reverse; }
```

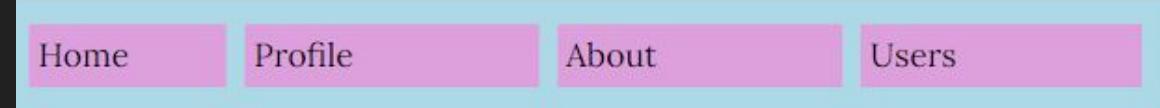
Home
Profile
About
Users
Support

flex-grow

Flex Box rules can also apply to individual items within the children, although it is not necessary to set these rules for the flex box to work.

flex-grow specifies how much space a menu item can take up in a layout relative to other menu items. An item with the value 2 will get twice as much space as an item with the value 1, if it's available.

```
.menu-item { flex-grow: 2; }  
#home { flex-grow: 1; }
```



flex-basis

flex-basis specifies the default size of an element before space is distributed among all the elements. The default is auto and any of the basic CSS units can be used, such as px, em or %.

```
.menu-item { flex-basis: 100px; }  
#home { flex-basis: 2em; }
```

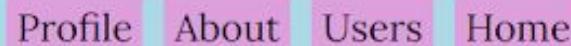


flex-order

flex-order specifies the order of an item. By default they are ordered by the order in the HTML document, but flex-order can be used to adjust the order.

Elements with the same order number will appear in the order of the HTML document as well. If you have 4 elements with orders 2 1 1 1, they will appear as 1 1 1 2.

```
.menu-item { order: 1; }  
#home { order: 2; }
```



Profile About Users Home

align-self

align-self is used to change the alignment of one item.

```
.menu-item { align-items: flex-start; }  
#home { align-self: flex-end; }
```



Flexbox Resources

Flexbox was introduced in 2017 so designers are still experimenting with it. You can read more about Flexbox here:

- [MDN Flexbox](#)
- [CSS-Tricks: a guide to flexbox](#)
- [Sitepoint: A friendly intro to Flexbox](#)
- [W3Schools: Flexbox](#)