

eThink



Using CSS Flexbox to Customize Your Site

By Daniel Hefley (CCA)

What is Flexbox?



Image credit: [learntoprogram](#)

- Flexbox is a new layout mode in CSS₃. The previous version of CSS defined four layout modes:
 - **block layout** for laying out documents
 - **inline layout** for laying out text
 - **table layout** for laying 2D tabular data
 - **positioned layout** for explicit positioning
- Positioning and sizing elements cumbersome with block/floats layout.
- **Flexbox layout** distributes space along a single row or column. This layout makes it easier to design a flexible responsive layout structure without using float or positioning.

The Flex Container & Flex Items

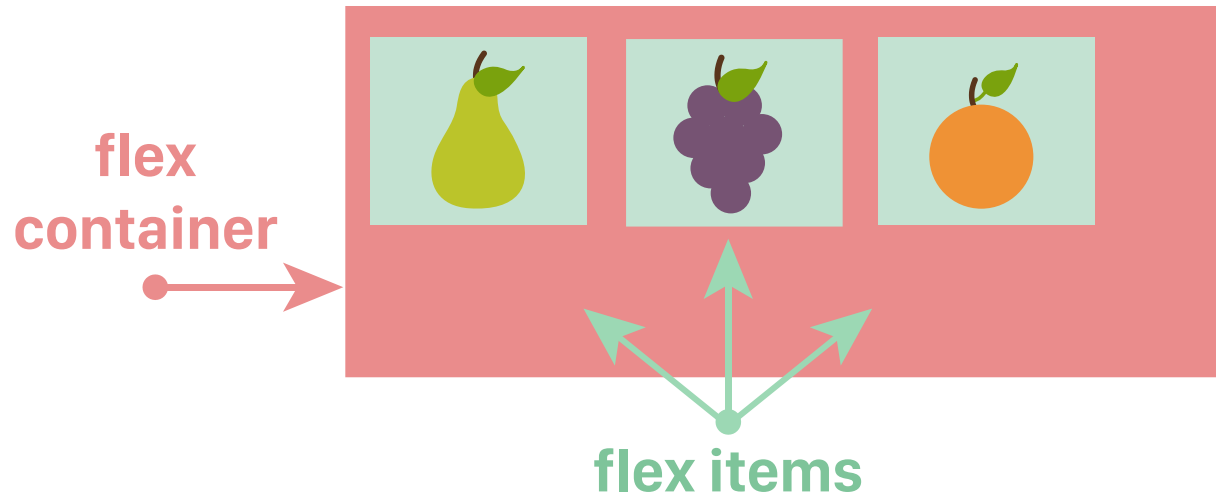


Image credit: [Flexbox Cheatsheet](#)

A Flexbox layout consists of a **flex container** that holds **flex items**.

```
.flex-container {  
  display: flex;  
}
```

```
<div class="flex-container">  
  <div>item1</div>  
  <div>item2</div>  
  <div>item3</div>  
</div>
```

} Flex items

Flex Axes

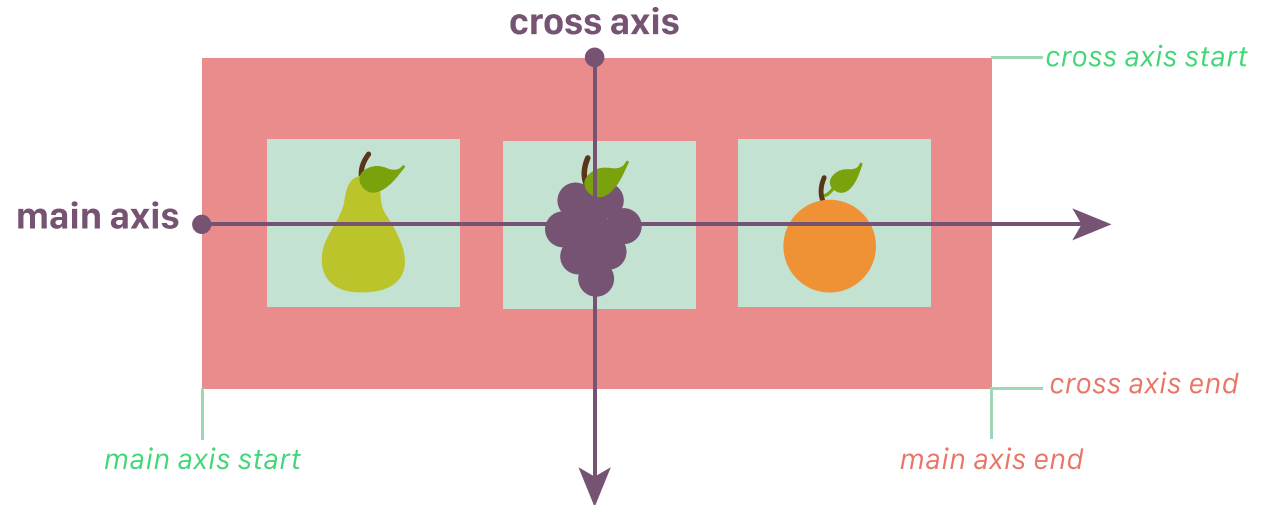


Image credit: [Flexbox Cheatsheet](#)

The flex container can be laid out horizontally or vertically. The direct children of a flex container are laid out along the main axis. These children can “flex” their sizes, growing to fill unused space in the container, or shrinking to avoid overflowing.

The flex container properties are:

[flex-direction](#): defines in which direction the container wants to stack the flex items.

[flex-wrap](#): specifies whether the flex items should wrap or not.

[justify-content](#): used to align the flex items.

[align-items](#): used to align the flex items vertically.

[align-content](#): used to align the flex lines.

The Row & Wrap

```
<!DOCTYPE html>
<html>
<head>
<style>
body {font-family: arial, san-serif;}
.flex-container {
display: flex;
flex-direction: row; /*default value*/
flex-wrap: wrap; /*default value*/
justify-content: center;
background-color: #d95700;
}
.flex-container > div {
background-color: #f1f1f1;
margin: 10px;
padding: 20px;
font-size: 30px;
}
.center-text {text-align: center;}
.extra-text {font-size: 1.5em;}
</style>
</head>
<body>
<div class="flex-container">
  <div class="center-text"><h1>Flex Child 1</h1><p>(Manual Login)</p></div>
  <div class="center-text"><h1>Flex Child 2</h1><p>(SSO Login)</p></div>
</div>
<div class="center-text extra-text">
<p>A Flexible Layout must have a parent element with the display property set to flex.</p>
<p>Direct child element(s) of the flexible container automatically becomes flexible items.</p>
<p>This is the default layout on the login page.</p>
</div>
</body>
</html>
```

Flex Child 1

(Manual Login)

Flex Child 2

(SSO Login)

A Flexible Layout must have a parent element with the `display` property set to `flex`.

Direct child element(s) of the flexible container automatically becomes flexible items.

This is the default layout on the login page.

Flex Child 1

(Manual Login)

Flex Child 2

(SSO Login)

Wrap stacks the manual login on top when viewing on a mobile device.

The Row-reverse & Wrap-reverse

```
<!DOCTYPE html>
<html>
<head>
<style>
body {font-family: arial, san-serif;}
.flex-container {
  display: flex;
  flex-direction: row-reverse;
  flex-wrap: wrap-reverse;
  justify-content: center;
  background-color: #d95700;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  padding: 20px;
  font-size: 30px;
}
.center-text {text-align: center;}
.extra-text {font-size: 1.5em;}
</style>
</head>
<body>
<div class="flex-container">
  <div class="center-text"><h1>Flex Child 1</h1><p>(Manual Login)</p></div>
  <div class="center-text"><h1>Flex Child 2</h1><p>(SSO Login)</p></div>
</div>
<div class="center-text extra-text">
<p>This is how it looks with row-reverse.</p>
</div>
</body>
</html>
```

Flex Child 2

(SSO Login)

Flex Child 1

(Manual Login)

This is how it looks with row-reverse.

Flex Child 2

(SSO Login)

Flex Child 1

(Manual Login)

Wrap-reverse stacks the SSO login on top when viewing on a mobile device.

The Column

```
<!DOCTYPE html>
<html>
<head>
<style>
body {font-family: arial, san-serif;}
.flex-container {
  display: flex;
  flex-direction: column;
  flex-wrap: wrap;
  justify-content: center;
  background-color: #d95700;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  padding: 20px;
  font-size: 30px;
}
.center-text {text-align: center;}
.extra-text {font-size: 1.5em;}
</style>
</head>
<body>
<div class="flex-container">
  <div class="center-text"><h1>Flex Child 1</h1><p>(Manual Login)</p></div>
  <div class="center-text"><h1>Flex Child 2</h1><p>(SSO Login)</p></div>
</div>
<div class="center-text extra-text">
<p>Column direction lays out child items in columnar format.</p>
</div>
</body>
</html>
```

Flex Child 1

(Manual Login)

Flex Child 2

(SSO Login)

Column direction lays out child items in columnar format.

The Column-reverse

```
<!DOCTYPE html>
<html>
<head>
<style>
body {font-family: arial, san-serif;}
.flex-container {
  display: flex;
  flex-direction: column-reverse;
  flex-wrap: wrap;
  justify-content: center;
  background-color: #d95700;
}
.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  padding: 20px;
  font-size: 30px;
}
.center-text {text-align: center;}
.extra-text {font-size: 1.5em;}
</style>
</head>
<body>
<div class="flex-container">
  <div class="center-text"><h1>Flex Child 1</h1><p>(Manual Login)</p></div>
  <div class="center-text"><h1>Flex Child 2</h1><p>(SSO Login)</p></div>
</div>
<div class="center-text extra-text">
<p>Column-reverse direction lays out child items in a reverse columnar format.</p>
</div>
</body>
</html>
```

Flex Child 2

(SSO Login)

Flex Child 1

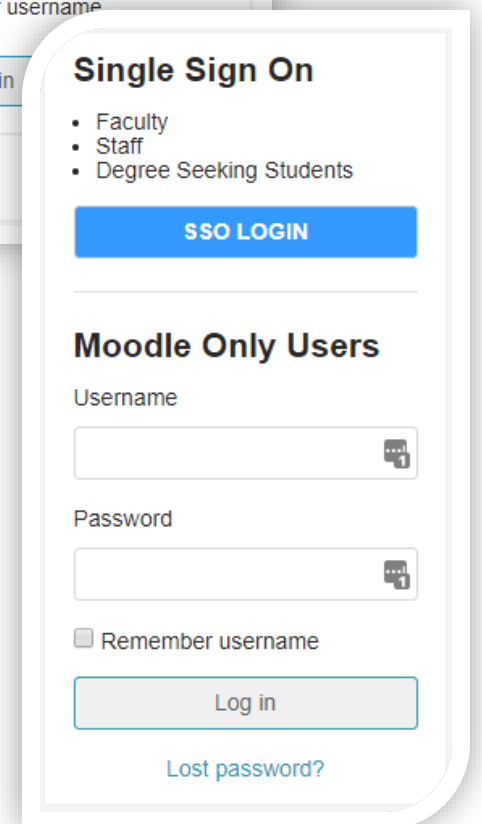
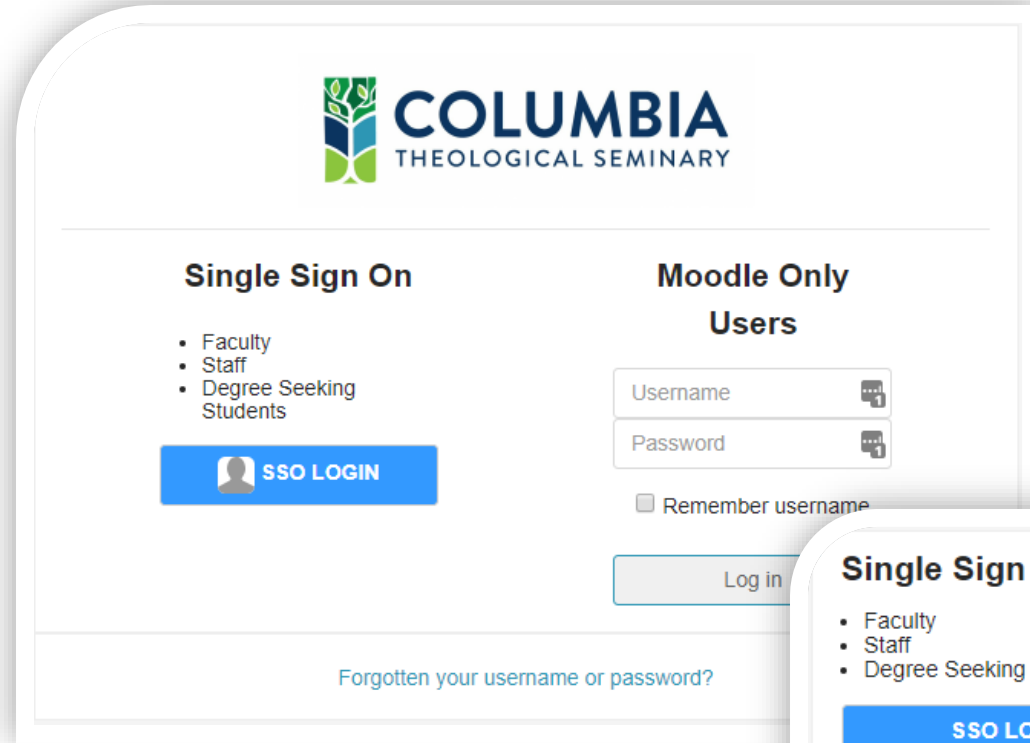
(Manual Login)

Column-reverse direction lays out child items in a reverse columnar format.

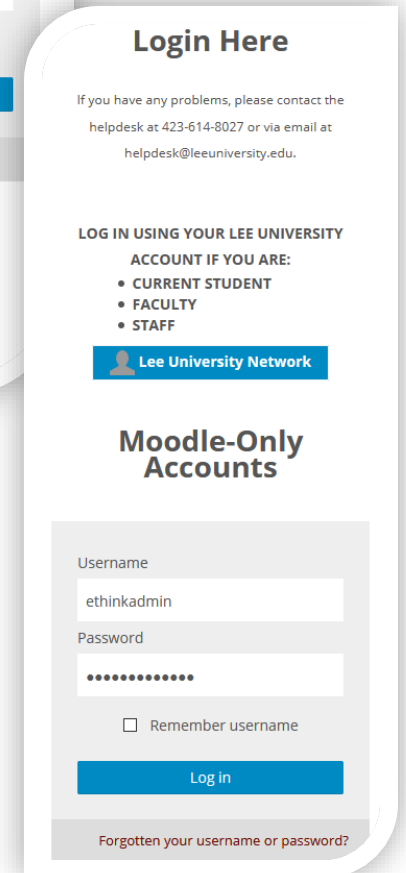
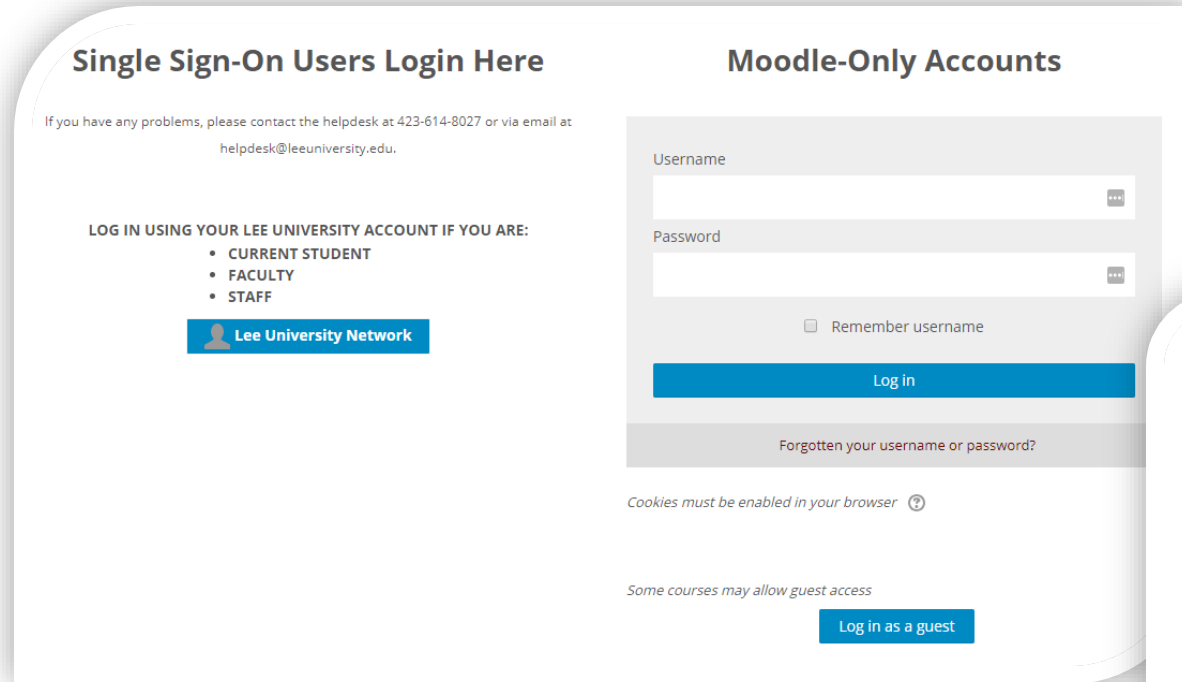
Real World Example 1 – Boost Float

```
/*sso login*/  
#page-login-index .push-md-3 {  
display: block;  
float: left !important;  
left: 10% !important;  
margin-right: auto;  
margin-left: auto;  
text-align: center; }  
/*Moodle login*/  
#page-login-index .push-md-1 {  
display: block;  
float: right !important;  
left: -10% !important;  
margin-right: auto;  
margin-left: auto;  
text-align: center;}
```

```
/*Login block on frontpage*/  
#page-site-index .block-region div.potentialidps {  
position: relative;  
top: -300px;  
border-bottom: 1px solid #E0E0E0;  
padding-bottom: 20px; }  
#page-site-index .block-region .loginform {  
position: relative;  
top: 160px; }  
#page-site-index .block-region .block.block_login div.card-text.content > div:nth-of-type(1) {  
position: relative;  
top: 160px;  
margin-right: auto;  
margin-left: auto;  
text-align: center; }
```



Real World Example 2 – Lambda Flex



```
/*Flip columns on login page*/
#page-login-index #page #page-content div.loginbox.clearfix.twocolumns {
display: -webkit-flex; /* Safari */
-webkit-flex-wrap: wrap-reverse; /* Safari 6.1+ */
display: flex;
flex-wrap: wrap-reverse;
flex-direction: row-reverse;
}
```

Real World Example 3 – Fordson Flex

```
/*Changes to login page*/  
/*Reverse columns on main block*/  
#page-login-index div#page-content.row section#region-  
12 div div.row.justify-content-center div.col-xl-6.col-sm-8  
div.card-block div.card-body div.row.justify-content-md-ce  
flex-direction: row-reverse;  
flex-wrap: wrap-reverse;  
}  
  
#page-login-index div#page-content.row section#region-  
12 div div.row.justify-content-center div.col-xl-6.col-sm-8  
div.card-block div.card-body {  
padding-bottom: 0;  
}  
/*-----*/  
/*Change order of rows in sso login column*/  
#page-login-index div#page-content.row section#region-  
12 div div.row.justify-content-center div.col-xl-6.col-sm-8  
div.card-block div.card-body div.row.justify-content-md-ce  
div:nth-child(2) {  
display: flex;  
flex-direction: column;  
flex-wrap: wrap;  
background-color: rgba(255,255,255,0.5);
```



Browser Support

The current Flexbox spec is fully supported in all modern browsers, including mobile, IE11 and Edge.



Chrome
21+



Opera
12.1+



Firefox
22+



Safari
6.1+



IE
10+

IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8							4.3	
9							4.4	
10	37	42			7.1		4.4.4	
11	38	43	8	30	8.3	8	40	42
Edge	39	44	9	31	9			
	40	45		32				
	41	46						

Thank you!

eThink



- More information –
 - <http://jonibologna.com/flexbox-cheatsheet/>
 - <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
 - https://www.w3schools.com/Css/css3_flexbox.asp
 - <https://cssreference.io/flexbox/>
 - <https://caniuse.com/#feat=flexbox>