Flex

THE ULTIMATE FLEXBOX CHEAT SHEET

Before you dive right into the cheat sheet, **brush up on the fundamentals.**

# Table of Contents

**Note: If any of the examples are not working you will need to upgrade your browser.**

# Create a flex container

The first step in any flexbox layout is to create a flex container. To do that, set the display property to flex. In Safari, you will still have to use the -webkit prefix.

```
.flexcontainer {
   display:-webkit-flex;
   display: flex;}
```

# Put flex items into a row

Flex items are the children of a flex container. They are positioned along a main axis and a cross axis. The main axis is horizontal by default, so the items flow into a row. You can flip the main axis by setting flex-direction to column, it's set to row by default.

```
/*On the flex container*/.flexcontainer {
   display:-webkit-flex;
   display: flex;-webkit-flex-direction: row;
   flex-direction: row;}
```

# Put flex items into a column

```
.flexcontainer {
   display:-webkit-flex;
   display: flex;-webkit-flex-direction: column;
```

```
    flex-direction: column;}
```

# Move flex items to the top

How you move flex items to the top depends on the orientation of the main axis. If it's vertical, you set align-items. If it's horizontal, you set justify-content.

```
.flexcontainer {-webkit-flex-direction: column;
    flex-direction: column;-webkit-justify-content: flex-start;
    justify-content: flex-start;
```

```
.flexcontainer {
    display:-webkit-flex;
    display: flex;-webkit-flex-direction: row;
    flex-direction: row;-webkit-align-items: flex-start;
    align-items: flex-start;}
```

# Move flex items to the left

Moving items to the left or right also depends on the orientation of the main axis. If flex-direction is set to row, then you set justify-content, otherwise if it's set to column you set align-items

```
.flexcontainer {
    display:-webkit-flex;
    display: flex;-webkit-flex-direction: row;
    flex-direction: row;-webkit-justify-content: flex-start;
    justify-content: flex-start;}
```

```
.flexcontainer {
    display:-webkit-flex;
    display: flex;-webkit-flex-direction: column;
    flex-direction: column;-webkit-align-items: flex-start;
    align-items: flex-start;}
```

# Move flex items to the right

```
.flexcontainer {
    display:-webkit-flex;
    display: flex;-webkit-flex-direction: row;
    flex-direction: row;-webkit-justify-content: flex-end;
    justify-content: flex-end;}
```

```
.flexcontainer {
    display:-webkit-flex;
    display: flex;-webkit-flex-direction: column;
    flex-direction: column;-webkit-align-items: flex-end;
    align-items: flex-end;}
```

# Center everything

Vertical and horizontal centering within a flexcontainer is trivial. Just set justify-content and/oralign-items to center. Again, it depends on the orientation of the main axis, so whether flex-direction is set to row or column.

```
.flexcontainer {
    display:-webkit-flex;
    display: flex;-webkit-flex-direction: row /* works with row or column */
```

```
    flex-direction: row;-webkit-align-items: center;
    align-items: center;-webkit-justify-content: center;
    justify-content: center;}
```

# Grow a flex item X times as big as other flex items

You can define how an individual flex item grows and shrinks relative to other flex items in the container. To do this set the flex property on each flex item you want to grow or shrink.

```
.bigitem {/* This will be twice as big as the small item. */-webkit-flex:200;
    flex:200;}.smallitem {-webkit-flex:100;
    flex:100;}
```

# Wrap flex items into multiple rows

At the moment browser support for wrapping flex items is limited to webkit browsers and IE11. Firefox does not support wrapping yet. (Bad Firefox, BAD!)

```
/* On the flex container */.flexcontainer {
    display:-webkit-flex;
    display: flex;-webkit-align-items: center;
    align-items: center;-webkit-justify-content: center;
    justify-content: center;/* You can set flex-wrap and flex-direction individually */-webkit
-flex-direction: row;
    flex-direction: row;-webkit-flex-wrap: wrap;
    flex-wrap: wrap;/* Or do it all in one line with flex flow */-webkit-flex-flow: row wrap;
    flex-flow: row wrap;/* tweak the where items line up on the row *//* valid values are: fle
x-start, flex-end, space-between, space-around, stretch */-webkit-align-content: flex-end;
    align-content: flex-end;}
```

# Wrap flex items into multiple columns

```
    display:-webkit-flex;
    display: flex;-webkit-align-items: center;
    align-items: center;-webkit-justify-content: center;
    justify-content: center;-webkit-flex-flow: column wrap;
    flex-flow: column wrap;-webkit-align-content: stretch;
    align-content: stretch;}
```

# Remove the space from wrapped rows or columns

The align-content property gives you options for distributing the space around wrapped rows or columns. The options are flex-start, flex-end, space-between, space-around and stretch. To simply remove the space around wrapped columns, set align-content to center.

```
.flexcontainer {
    display:-webkit-flex;
    display: flex;-webkit-align-items: center;
    align-items: center;-webkit-justify-content: center;
    justify-content: center;-webkit-flex-flow: column wrap;
    flex-flow: column wrap;-webkit-align-content: center;
    align-content: center;}
```
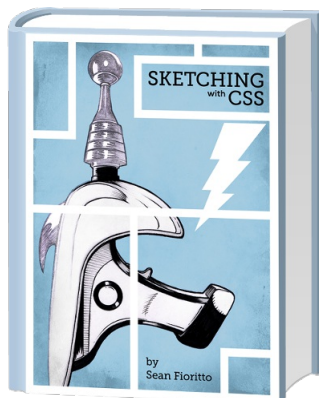
# Pin an element to one side of the flex container

You can control the align-items value for individual elements with align-self. You can also use margins to move individual elements up or down or left of right. e.g. on a column layout you can move an individual flex item all the way to the left of the container by setting margin-right: auto.

```
/* On the flex item to pin */.left {-webkit-align-self: flex-start;
  align-self: flex-start;}.right {
  margin-left:auto;}
```

# Other flexbox resources

- [A free class on flexbox (based on this cheat sheet)](#)
- [The latest spec](#)
- [CSS-Tricks guide to flexbox](#)
- [Mozilla guide to flexbox](#)
- [Smashing Magazine tutorial](#)

# If you liked this, you should see what's in the book.



Blow away your clients with your fast turnaround time. Bend CSS layouts to your will. Show your clients comps they can actually click on.

Get a free chapter and an 8 day email course on how to design in the browser.

**Email Address**

**Add Flexbox to your bag of tricks**

✕

## Add Flexbox to your bag of tricks

Sign up for my free email course on flexbox and get five interactive video tutorials, the Ultimate Flexbox Cheat Sheet, a 22 page PDF walkthrough of flexbox and tips for how you can incorporate this modern technology into your workflow today.