Review: Flexbox

You should be proud of yourself! You have learned the most important properties of flexbox. Flexbox is an art and a science; you can use it to make laying out multiple elements a piece of cake. You know everything necessary to begin using it in your own projects.

- `display: flex` changes an element to a block-level container with flex items inside of it.
- `display: inline-flex` allows multiple flex containers to appear inline with each other.
- `justify-content` is used to space items along the major axis.
- `align-items` is used to space items along the cross axis.
- `flex-grow` is used to specify how much space (and in what proportions) flex items absorb along the major axis.
- `flex-shrink` is used to specify how much flex items shrink and in what proportions along the major axis.
- `flex-basis` is used to specify the initial size of an element styled with `flex-grow` and/or `flex-shrink` .
- `flex` is used to specify `flex-grow` , `flex-shrink` , and `flex-basis` in one declaration.
- `flex-wrap` specifies that elements should shift along the cross axis if the flex container is not large enough.
- `align-content` is used to space rows along the cross axis.
- `flex-direction` is used to specify the major and cross axes.
- `flex-flow` is used to specify `flex-wrap` and `flex-direction` in one declaration.
- Flex containers can be nested inside of each other by declaring `display: flex` or `display: inline-flex` for children of flex containers.

Let's apply a few of the properties you've learned to arrange one section of the web page in the browser to the right!