Visoko učilište Algebra | Ilica 242 | 10000 Zagreb | Hrvatska
Upisano kod trgovačkog suda u Zagrebu pod MBS: 080664742
Poslovna banka: Zagrebačka banka d.d. Trg bana Josipa Jelačića 10, 10000 Zagreb, Hrvatska
OIB: 14575159920 | matični broj: 2412926 | IBAN: HR7523600001102036972
tel. +385 (0)1 2222 182 | fax. +385 (0)1 2305 004 | info@racunarstvo.hr | www.racunarstvo.hr

Exercise 8 - CSS Flexbox, part I

Intro

Before the Flexbox Layout module, there were four layout modes:

- Block, for sections in a webpage
- Inline, for text
- Table, for two-dimensional table data
- Positioned, for explicit position of an element

The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning.

Flexbox - How does it work?

The Flexbox Layout (Flexible Box) module (a W3C Candidate Recommendation as of October 2017) aims at providing a more efficient way to lay out, align and distribute space among items in a container, even when their size is unknown and/or dynamic (thus the word “flex”).

The main idea behind the flex layout is to give the container the ability to alter its items’ width/height (and order) to best fill the available space (mostly to accommodate to all kind of display devices and screen sizes). A flex container expands items to fill available free space or shrinks them to prevent overflow.

Most importantly, the flexbox layout is direction-agnostic as opposed to the regular layouts (block which is vertically-based and inline which is horizontally-based). While those work well for pages, they lack flexibility (no pun intended) to support large or complex applications (especially when it comes to orientation changing, resizing, stretching, shrinking, etc.).

Note: Flexbox layout is most appropriate to the components of an application, and small-scale layouts, while the Grid layout is intended for larger scale layouts.

Visoko učilište Algebra | Ilica 242 | 10000 Zagreb | Hrvatska
Upisano kod trgovačkog suda u Zagrebu pod MBS: 080664742
Poslovna banka: Zagrebačka banka d.d. Trg bana Josipa Jelačića 10, 10000 Zagreb, Hrvatska
OIB: 14575159920 | matični broj: 2412926 | IBAN: HR7523600001102036972
tel. +385 (0)1 2222 182 | fax. +385 (0)1 2305 004 | info@racunarstvo.hr | www.racunarstvo.hr

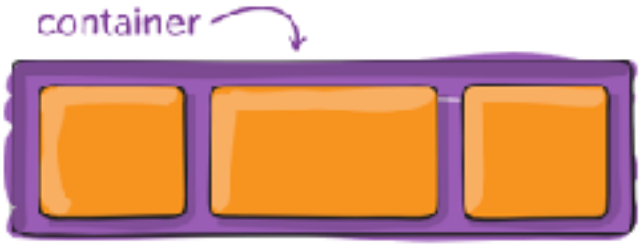
Flexbox properties

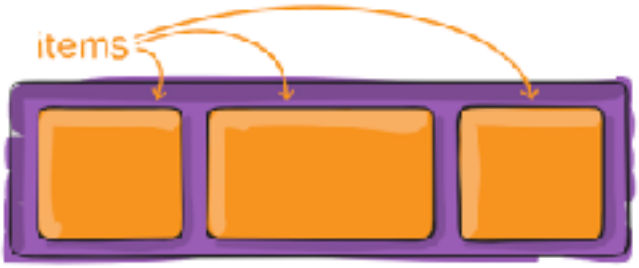
Properties for the Parent (flex container)

display

This defines a flex container; inline or block depending on the given value. It enables a flex context for all its direct children.

```
.container {
  display: flex; /* or inline-flex */
                                    }
```





flex-direction

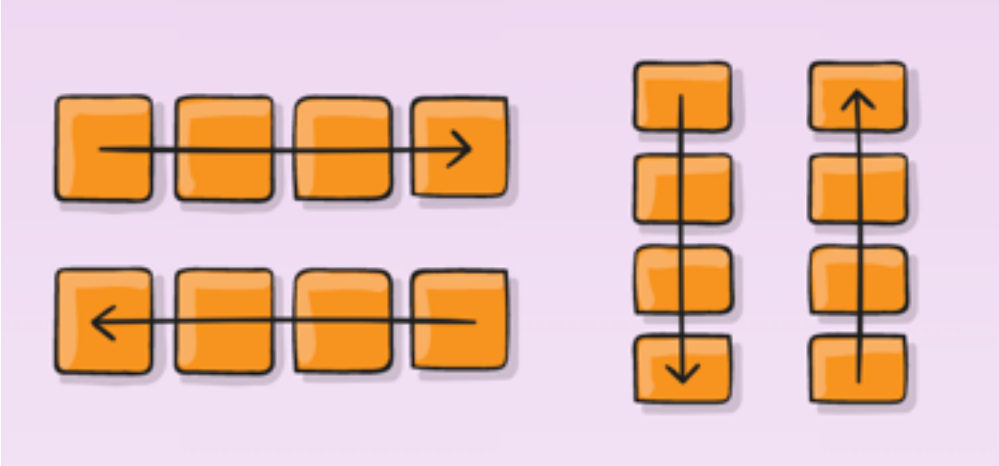
This establishes the main-axis, thus defining the direction flex items are placed in the flex container. Flexbox is (aside from optional wrapping) a single-direction layout concept. Think of flex items as primarily laying out either in horizontal rows or vertical columns.

```
.container {
  flex-direction: row | row-reverse | column | column-reverse;
}
```

- row (default): left to right
- row-reverse: right to left
- column: same as row but top to bottom

Visoko učilište Algebra | Ilica 242 | 10000 Zagreb | Hrvatska
Upisano kod trgovačkog suda u Zagrebu pod MBS: 080664742
Poslovna banka: Zagrebačka banka d.d. Trg bana Josipa Jelačića 10, 10000 Zagreb, Hrvatska
OIB: 14575159920 | matični broj: 2412926 | IBAN: HR7523600001102036972
tel. +385 (0)1 2222 182 | fax. +385 (0)1 2305 004 | info@racunarstvo.hr | www.racunarstvo.hr

- column-reverse: same as row-reverse but bottom to top



order

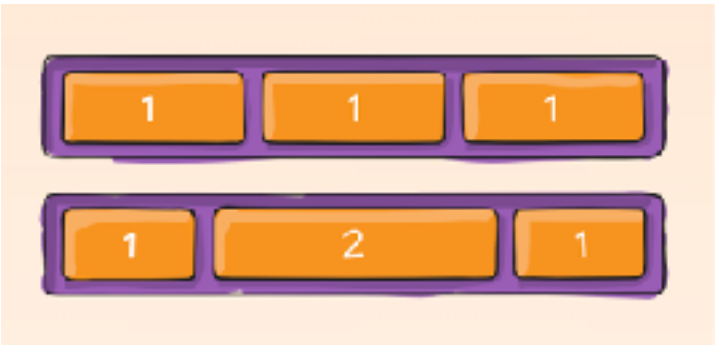
By default, flex items are laid out in the source order. However, the order property controls the order in which they appear in the flex container.

```
.item {
  order: 5; /* default is 0 */
}
```

flex-grow

This defines the ability for a flex item to grow if necessary. It accepts a unitless value that serves as a proportion. It dictates what amount of the available space inside the flex container the item should take up.
If all items have flex-grow set to 1, the remaining space in the container will be distributed equally to all children. If one of the children has a value of 2, that child would take up twice as much of the space either one of the others (or it will try, at least).

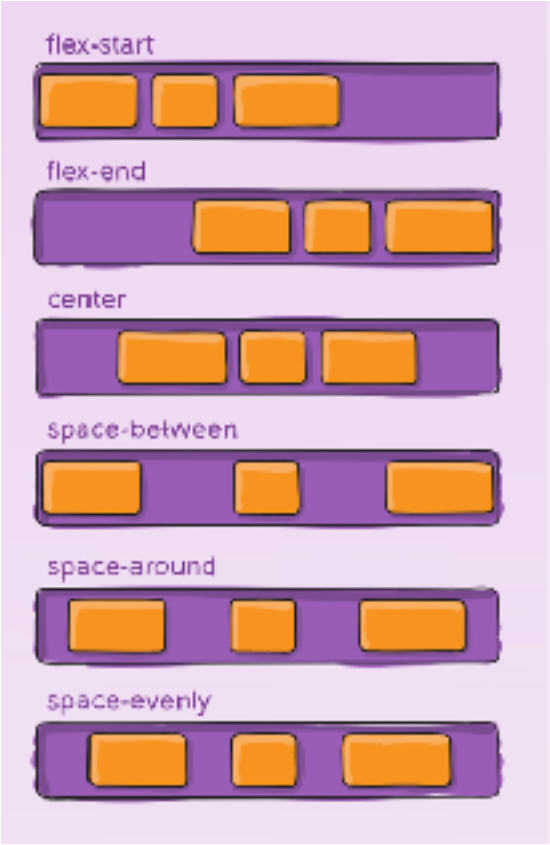
```
.item {
  flex-grow: 4; /* default 0 */
}
```



Visoko učilište Algebra | Ilica 242 | 10000 Zagreb | Hrvatska
Upisano kod trgovačkog suda u Zagrebu pod MBS: 080664742
Poslovna banka: Zagrebačka banka d.d. Trg bana Josipa Jelačića 10, 10000 Zagreb, Hrvatska
OIB: 14575159920 | matični broj: 2412926 | IBAN: HR7523600001102036972
tel. +385 (0)1 2222 182 | fax. +385 (0)1 2305 004 | info@racunarstvo.hr | www.racunarstvo.hr

justify-content

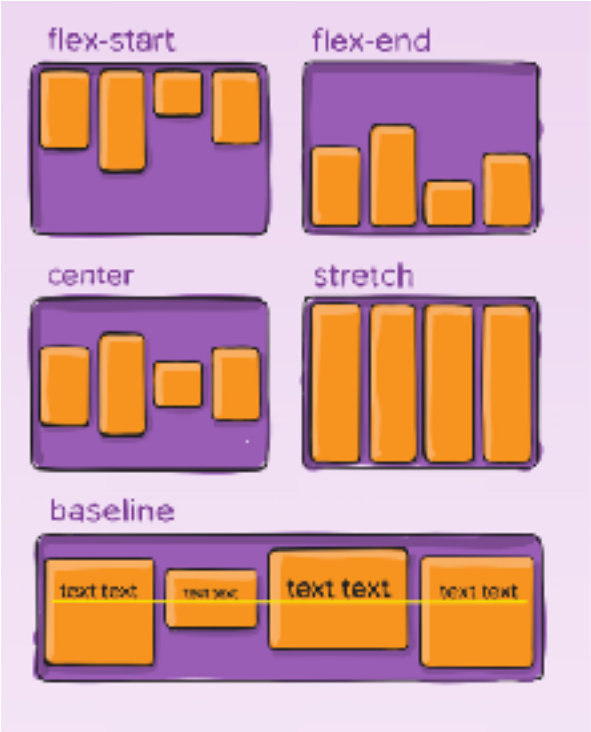
This defines the alignment along the main axis. It helps distribute extra free space leftover when either all the flex items on a line are inflexible, or are flexible but have reached their maximum size. It also exerts some control over the alignment of items when they overflow the line.



align-items

This defines the default behavior for how flex items are laid out along the cross axis on the current line. Think of it as the justify-content version for the cross-axis (perpendicular to the main-axis).

Visoko učilište Algebra | Ilica 242 | 10000 Zagreb | Hrvatska
Upisano kod trgovačkog suda u Zagrebu pod MBS: 080664742
Poslovna banka: Zagrebačka banka d.d. Trg bana Josipa Jelačića 10, 10000 Zagreb, Hrvatska
OIB: 14575159920 | matični broj: 2412926 | IBAN: HR7523600001102036972
tel. +385 (0)1 2222 182 | fax. +385 (0)1 2305 004 | info@racunarstvo.hr | www.racunarstvo.hr



align-content

This aligns a flex container’s lines within when there is extra space in the cross-axis, similar to how justify-content aligns individual items within the main-axis.

Visoko učilište Algebra | Ilica 242 | 10000 Zagreb | Hrvatska
Upisano kod trgovačkog suda u Zagrebu pod MBS: 080664742
Poslovna banka: Zagrebačka banka d.d. Trg bana Josipa Jelačića 10, 10000 Zagreb, Hrvatska
OIB: 14575159920 | matični broj: 2412926 | IBAN: HR7523600001102036972
tel. +385 (0)1 2222 182 | fax. +385 (0)1 2305 004 | info@racunarstvo.hr | www.racunarstvo.hr

Exercise

(1) Let's start by creating a blank HTML document. Create a folder name Exercise8. You can create it on desktop, or your personal network folder.

(2) Inside Exercise8 folder create a new text file index.html If using Windows, make sure file extensions are not hidden and that document contains proper extension. If extension is correct, when the file is double-clicked it will be opened in a web browser.

(3) Open index.html file with a code editor.

(4) When inside text editor, copy basic HTML document skeleton into index.html file. You already learned how to do it :)

(5) Create a list with six items. Add a class flex-container on the ul element and on each li element add a flex-item class. In each list item add a number. Your HTML should look like this:

```
<ul class="flex-container">
  <li class="flex-item">1</li>
  <li class="flex-item">2</li>
  <li class="flex-item">3</li>
  <li class="flex-item">4</li>
  <li class="flex-item">5</li>
  <li class="flex-item">6</li>
</ul>
```

(6) Now create a CSS file and name it style.css

(7) Inside of it, create a flex layout context.

```
.flex-container {
  display: flex;
  justify-content: space-around;
  padding: 0;
  margin: 0;
  list-style: none;
}
```

Visoko učilište Algebra | Ilica 242 | 10000 Zagreb | Hrvatska
Upisano kod trgovačkog suda u Zagrebu pod MBS: 080664742
Poslovna banka: Zagrebačka banka d.d. Trg bana Josipa Jelačića 10, 10000 Zagreb, Hrvatska
OIB: 14575159920 | matični broj: 2412926 | IBAN: HR7523600001102036972
tel. +385 (0)1 2222 182 | fax. +385 (0)1 2305 004 | info@racunarstvo.hr | www.racunarstvo.hr

(8) Let's give it some properties so it looks better. Create small boxes with a tomato red background color. You can do it the way you want, but the point is that the numbers should be aligned in the middle.

```
.flex-item {
  background: tomato;
  padding: 5px;
  width: 200px;
  height: 150px;
  margin-top: 10px;
  line-height: 150px;
  color: white;
  font-weight: bold;
  font-size: 3em;
  text-align: center;
}
```

(9) Now let's create something more complex but familiar. Our first navigation! In html create a list with 4 navigation items at the top of the file. The items should be: Home, About, Products and Contact. Each item should be a link.

```
<ul class="navigation">
  <li><a href="#">Home</a></li>
  <li><a href="#">About</a></li>
  <li><a href="#">Products</a></li>
  <li><a href="#">Contact</a></li>
</ul>
```

(10) Great, now open the css file and select the navigation class and se the. Display to flex. Justify the content to the right end side. Remove the list styling and add a background color so it looks better.

```
.navigation {
  display: flex;
  justify-content: flex-end;
  list-style: none;
  margin: 0;
  background: deepskyblue;
}
```

Visoko učilište Algebra | Ilica 242 | 10000 Zagreb | Hrvatska
Upisano kod trgovačkog suda u Zagrebu pod MBS: 080664742
Poslovna banka: Zagrebačka banka d.d. Trg bana Josipa Jelačića 10, 10000 Zagreb, Hrvatska
OIB: 14575159920 | matični broj: 2412926 | IBAN: HR7523600001102036972
tel. +385 (0)1 2222 182 | fax. +385 (0)1 2305 004 | info@racunarstvo.hr | www.racunarstvo.hr

(11) Ok, now let's make the links look good. Remove the text decoration, add some padding and change the display if needed!

```
.navigation a {
  text-decoration: none;
  display: block;
  padding: 10px;
  color: white;
}
```

(12) We can do more than that! Add a hover state on the links.

```
.navigation a:hover {
  background: #1565C0;
}
```