עמותת תפוח

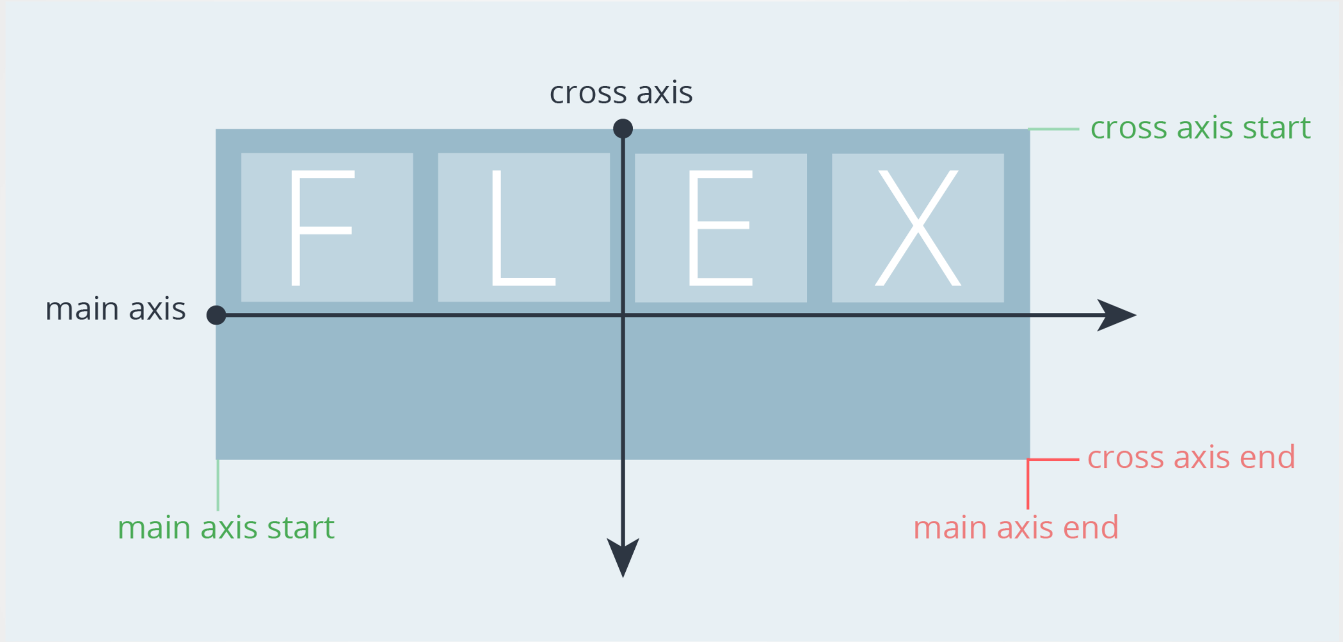
APPLESEEDS

ACADEMY

جمعية تفاح

מכשירים לחיים בעולם טכנולוגי

Flexbox


cross axis
cross axis start
F
L
E
X
main axis
cross axis end
main axis start
main axis end

Display Flex

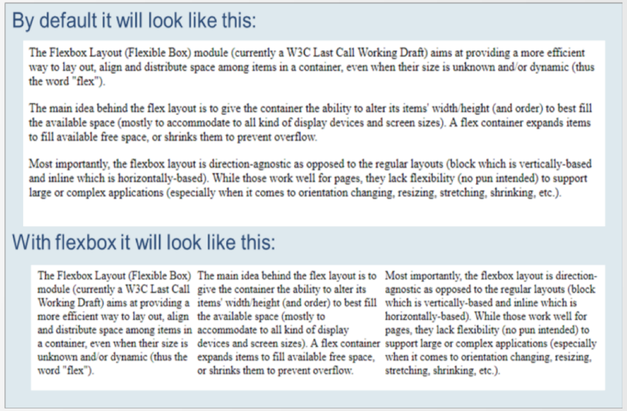
- Flex container has flex items (display: flex; makes a flex container)
- Each flex item takes space according to its content (by default)
- The flex container properties are: flex-direction, flex-wrap, flex-flow, justify-content, align-items, align-content

HTML

```html
<div class="flex-container">
    <p>The Flexbox Layout</p>
    <p>The main idea</p>
    <p>Most importantly</p>
</div>
```

CSS

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



By default it will look like this:

The Flexbox Layout (Flexible Box) module (currently a W3C Last Call Working Draft) 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.).

With flexbox it will look like this:

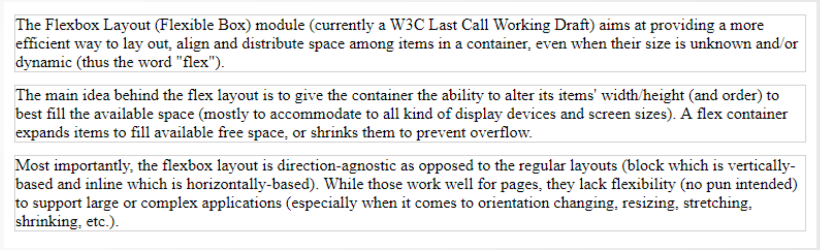
The Flexbox Layout (Flexible Box) module (currently a W3C Last Call Working Draft) aims at providing a more efficient way to lay out, align and distribute space among items in a container, even when their size is unknown and/or dynamic (thus the word "flex").	The main idea behind the flex layout is to 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.).

Flex Direction

CSS

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

- Flex default layout is a row, but we can also set it to column.
- Changing the direction to column changes the main and cross axis.

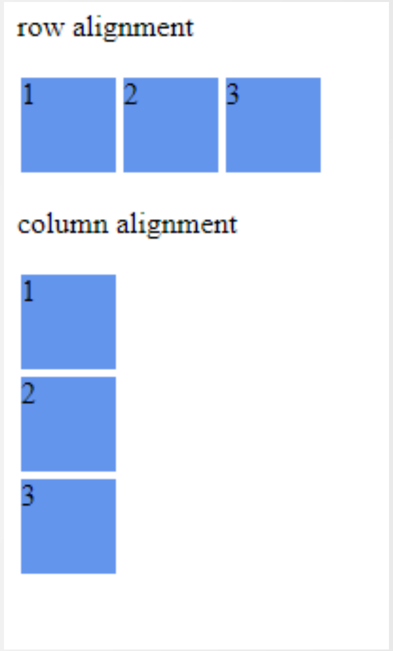

The Flexbox Layout (Flexible Box) module (currently a W3C Last Call Working Draft) aims at providing a more efficient way to lay out, align and distribute space among items in a container, even when their size is unknown and/or dynamic (thus the word "flex").

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.).

Exercise 1

- Create 2 arrangements of boxes
  - A row of 3 boxes
  - A column of boxes

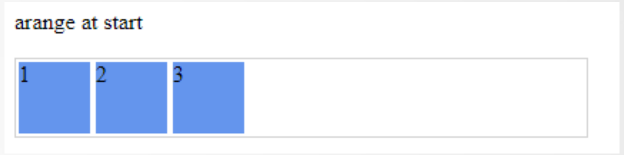


Justify Content

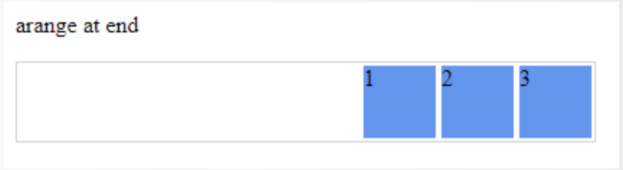
css

```
justify-content: flex-start;

justify-content: flex-end;

justify-content: space-evenly;
```

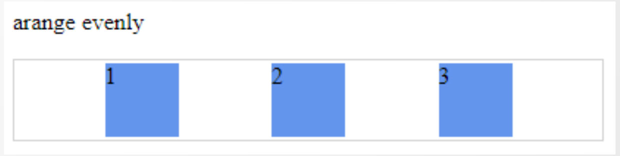
- Applies on the main axis
- flex-start (default)


arange at start

- flex-end

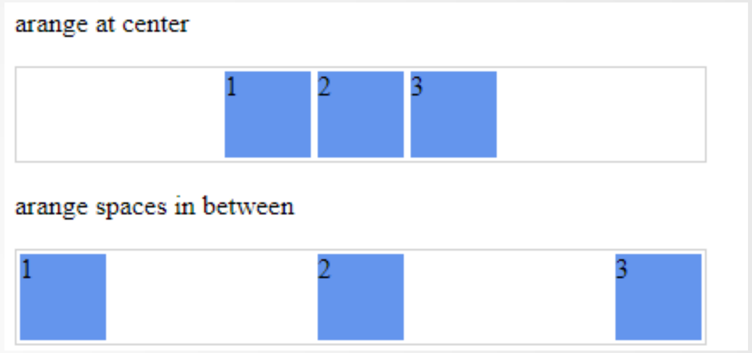

arange at end

- space-evenly


arange evenly

Exercise 2

- Create 2 arrangements of boxes
  - All boxes at the center
  - There are space between the boxes

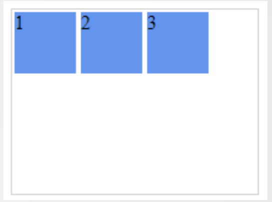


Align Items

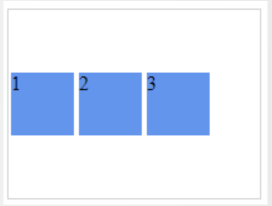
CSS

```css
align-items: flex-start;


align-items: center;


align-items: flex-end;
```

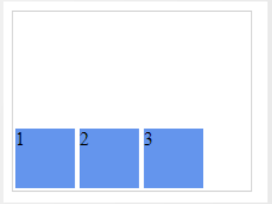
- Applies on the cross axis
- flex-start (default)



- center

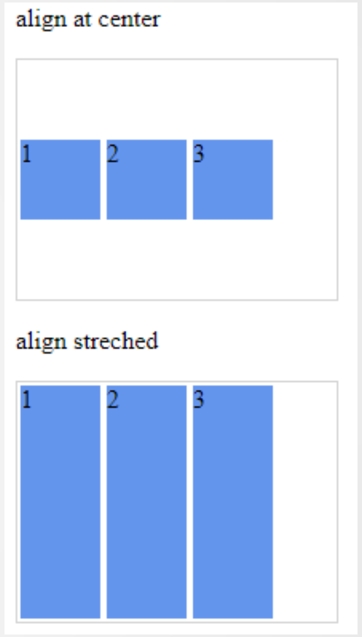


- flex-end



Exercise 3

- Create 2 arrangements of boxes
  - all boxes vertically aligned at the center
  - all boxes are vertically stretched across the container

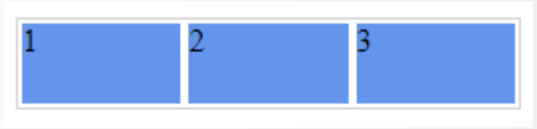


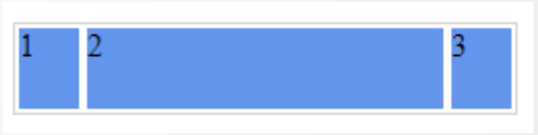
Items Modification

css

```css
.flex-item {
  flex-grow: 1;
}

.flex-item2 {
  flex-grow: 1;
}
```

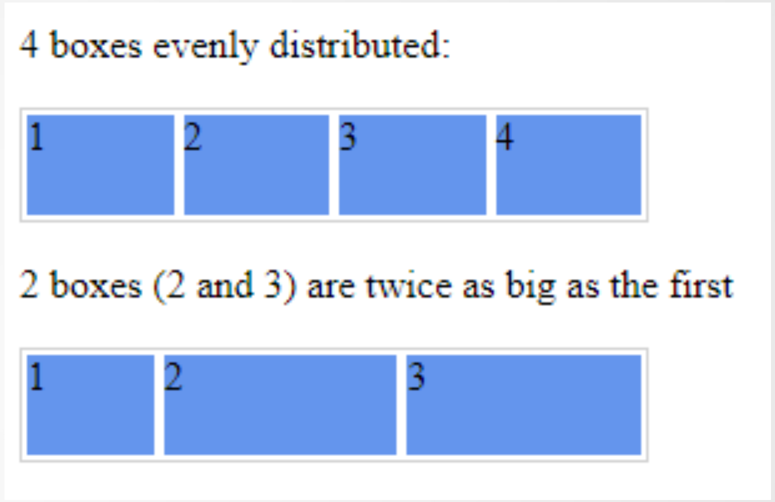
- By default each item takes the space of its content (and stretched all the way).
- flex-grow allows to relatively set the item width (default is 0).





Exercise 4

- Create 2 arrangements of boxes
  - 4 boxes evenly distributed
  - 2 boxes (2 and 3) are twice as big as the first


4 boxes evenly distributed:

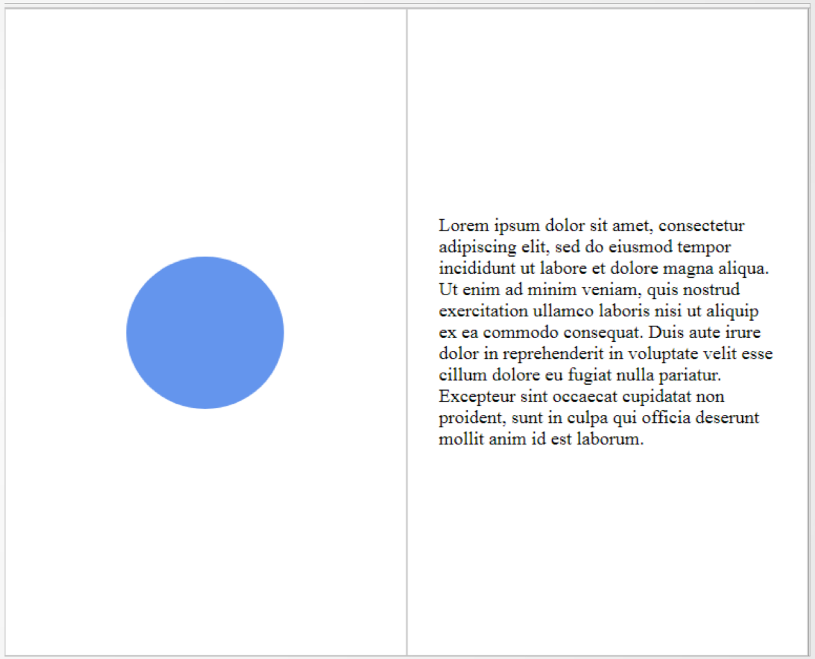
2 boxes (2 and 3) are twice as big as the first

Cheat Sheet

```
.flex-container {
  display: flex | inline-flex;
  flex-direction: row | column;
  justify-content: flex-start | flex-end | center;
  justify-content: space-between | space-around | space-evenly;
  align-items: flex-start | flex-end | center | baseline | stretch;
  align-content: flex-start | flex-end | center;
  align-content: space-between | space-around | stretch;
  --------------------------------------
  flex-wrap: nowrap | wrap;
}

.flex-item {
  flex-grow: <number>; /* default 0 */
  --------------------------------------
  flex-shrink: <number>; /* default 1 */
  flex-basis: <length> | auto; /* default auto */
  order: <integer>; /* default is 0 */
}
```

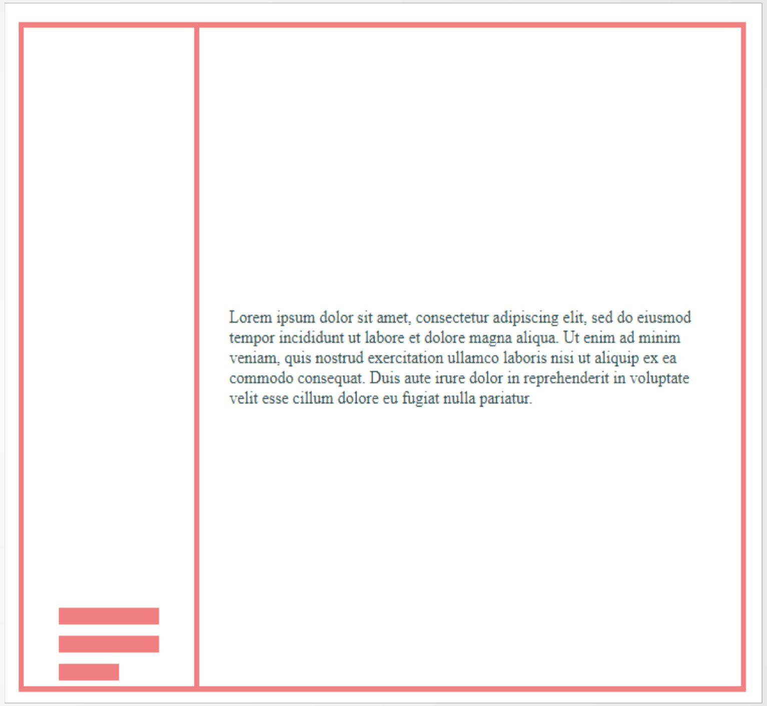
Exercise 5

- 2 Sections Layout


Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Exercise 6

- Left menu layout



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.