



GIFT School of Engineering and Applied Sciences

Fall 2022

**CS-120 Introduction to Information and
Communication Technology**

Lab-13 Manual

Introduction to Flex Boxes

v1
04-12-2022

Lab Objectives:

This lab helps students to get familiarize with basic properties of Flexbox
Module like:

- Flex Containers
- Flex Direction
 - ❖ Row
 - ❖ Row-reverse
 - ❖ Column
 - ❖ Column-reverse
- Flex wrap
 - ❖ Wrap
 - ❖ No-wrap
 - ❖ Wrap-reverse
- Justify Contents
 - ❖ Flex-start
 - ❖ Flex-end
 - ❖ Center
 - ❖ Space-between
 - ❖ Space-around
- Flex order

Flexbox

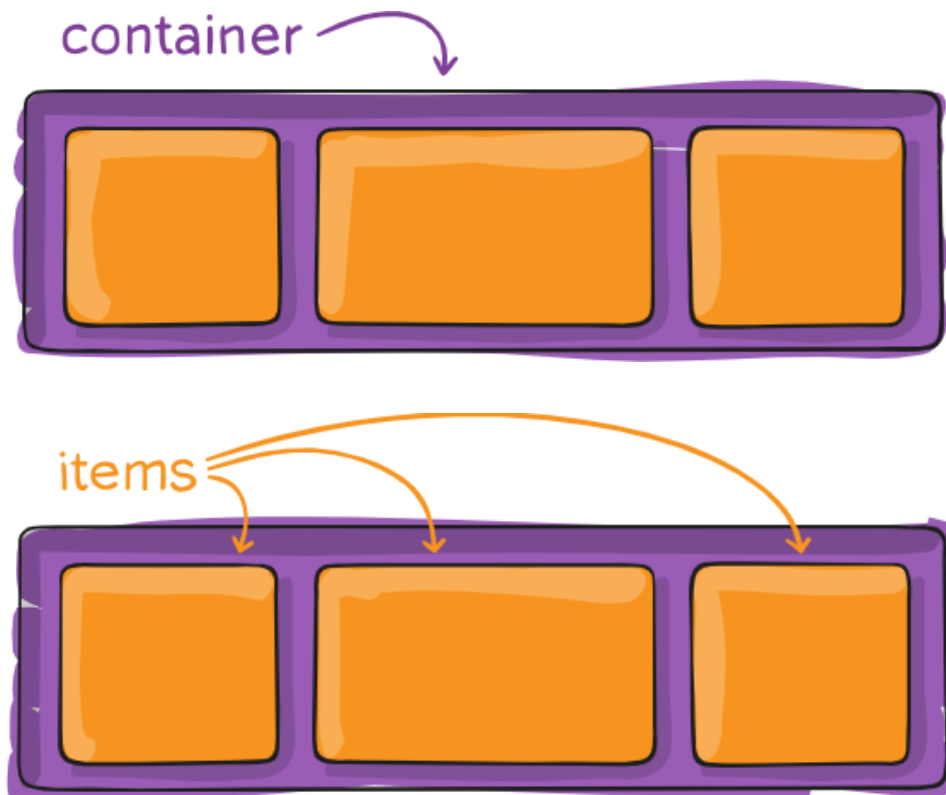
Flexbox (flexible box) is a layout mode of CSS3. Using this mode, you can easily create layouts for complex applications and web pages. Unlike Floats, Flexbox layout gives complete control over the direction, alignment, order, size of the boxes.

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.

Flex Container:



To use Flexbox in your application, you need to create/define a flex container using the **display** property.

```
display: flex;
```

On passing this value to the display property, a block level flex container will be created. It occupies the full width of the parent container (browser).

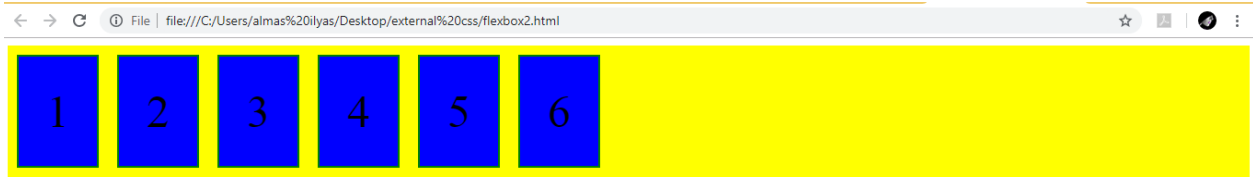
The following example demonstrates how to create a block level flex container. Here, we are creating six boxes and we have used the flex container to hold them.

```
<body>
  <style>
    .flex-container{
      display: flex;
      background-color: yellow;
    }
    .flex-item{
      border: 2px solid green;
      background-color: blue;
      font-size: 50px;
      padding: 30px;
      margin: 10px;
    }
  </style>

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

</body>
```

Save the document and open in browser



Flex direction:

The **flex-direction** property defines in which direction the container wants to specify the flex items.

This property accepts four values –

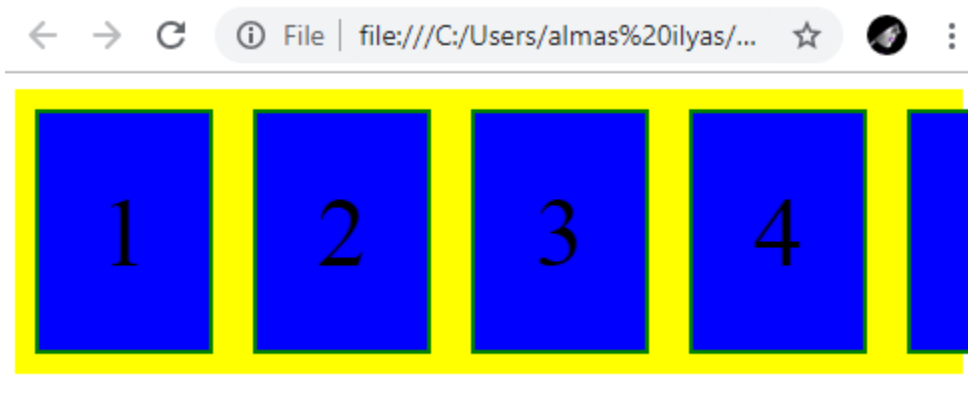
- ❖ **Row**
Arranges the elements of the container horizontally from left to right
- ❖ **Row-reverse**
Arranges the elements of the container horizontally from right to left.
- ❖ **Column**
Arranges the elements of the container vertically from left to right.
- ❖ **Column-reverse**
Arranges the elements of the container vertically from right to left.

Task:

Try all four values of property flex-direction and experience the difference.

Flex-wrap:

Generally, in case of insufficient space for the container, the rest of the flex items will be hidden as shown below.



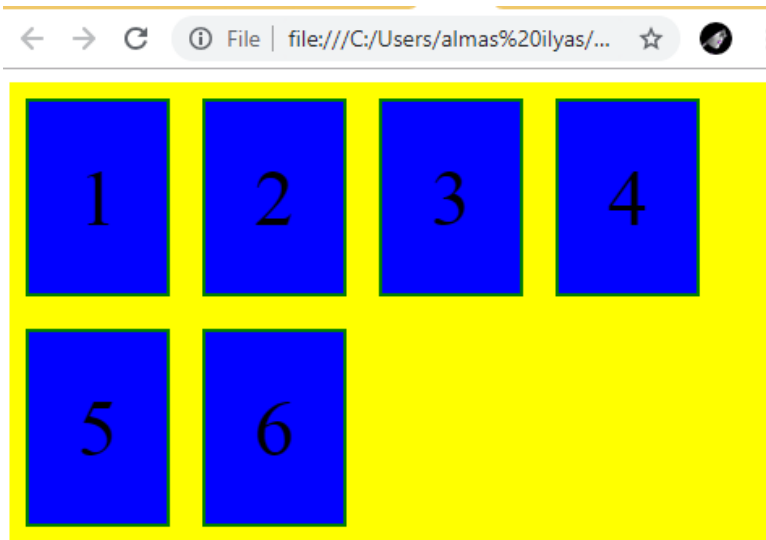
The **flex-wrap** property is used to specify the controls whether the flex-container is single-line or multi-line.

This property accepts the following values –

- ❖ **wrap** – In case of insufficient space for them, the elements of the container (flex items) will wrap into additional flex lines from top to bottom.

```
.flex-container{  
    display: flex;  
    flex-wrap: wrap;  
    background-color: yellow;  
}
```

Save the document and refresh the browser



- ❖ **wrap-reverse** – In case of insufficient space for them, the elements of the container (flex-items) will wrap into additional flex lines from bottom to top.

Task:

Try wrap-reverse property and experience the difference.

Flex-flow:

The **flex-flow** property is a shorthand property for setting both the **flex-direction** and **flex-wrap** properties.

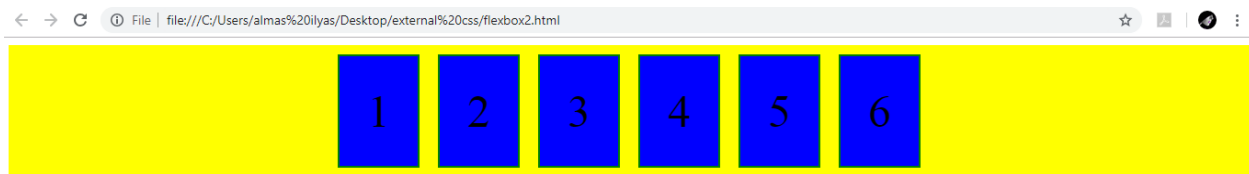
```
.flex-container{  
  
    display: flex;  
    flex-flow: row wrap;  
    background-color: yellow;  
}
```

Justify-content:

Justify-Content property is used to align flex items. For example if you want to align content at center.

```
.flex-container{  
  
    display: flex;  
    flex-flow: row wrap;  
    justify-content: center;  
    background-color: yellow;  
}
```

Save the document and refresh the browser.



- ❖ This property accepts the following values –
- ❖ **flex-start** – The flex-items are placed at the start of the container.
- ❖ **flex-end** – The flex-items are placed at the end of the container.
- ❖ **center** – The flex-items are placed at the center of the container, where the extra space is equally distributed at the start and at the end of the flex-items.
- ❖ **space-between** – The extra space is equally distributed between the flex-items.
- ❖ **space-around** – The extra space is equally distributed between the flex items such that the space between the edges of the container and its contents is half as the space between the flex-items.

Task:

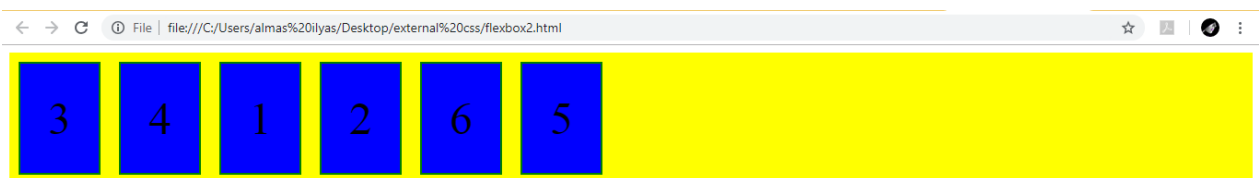
Try remaining four values of property Justify-content and experience the difference.

Flex order property:

The order property specifies the order of the flex items.

```
<div class="flex-container">
  <div class="flex-item" style="order: 3">1 </div>
  <div class="flex-item" style="order: 4">2 </div>
  <div class="flex-item" style="order: 1">3 </div>
  <div class="flex-item" style="order: 2">4 </div>
  <div class="flex-item" style="order: 6">5 </div>
  <div class="flex-item" style="order: 5">6 </div>
</div>
```

Save the document and refresh the browser.



Flex-grow property:

The flex-grow property specifies how much a flex item will grow relative to the rest of the flex items.

```
<div class="flex-container">
  <div class="flex-item" style="flex-grow: 1">1 </div>
  <div class="flex-item" style="flex-grow: 1">2 </div>
  <div class="flex-item" style="flex-grow: 1">3 </div>
  <div class="flex-item" style="flex-grow: 3">4 </div>
  <div class="flex-item" style="flex-grow: 1">5 </div>
  <div class="flex-item" style="flex-grow: 1">6 </div>
</div>
```

Save the document and refresh the browser.



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, the remaining space would take up twice as much space as the others.

Task:

- ❖ Open sublime text.
- ❖ Set "Flexbox" the title of your page.
- ❖ Create heading of pets.
- ❖ Create Flexbox container and add 6 images as flex items.
- ❖ Set images in item to width=350px ,height=200px; and margin=10px
- ❖ Set images in item-1 to width=500px ,height=200px and margin=10px
- ❖ Use flex-wrap property and justify the content to center.
- ❖ Apply following hover properties to images

```
img:hover{  
    transform: scale(1.2);  
    transition: all 2s;  
}
```

Your page should display as shown below.



Go forward one page (Alt+ Right Arrow)
Right-click or pull down to show history

Pets



The End
