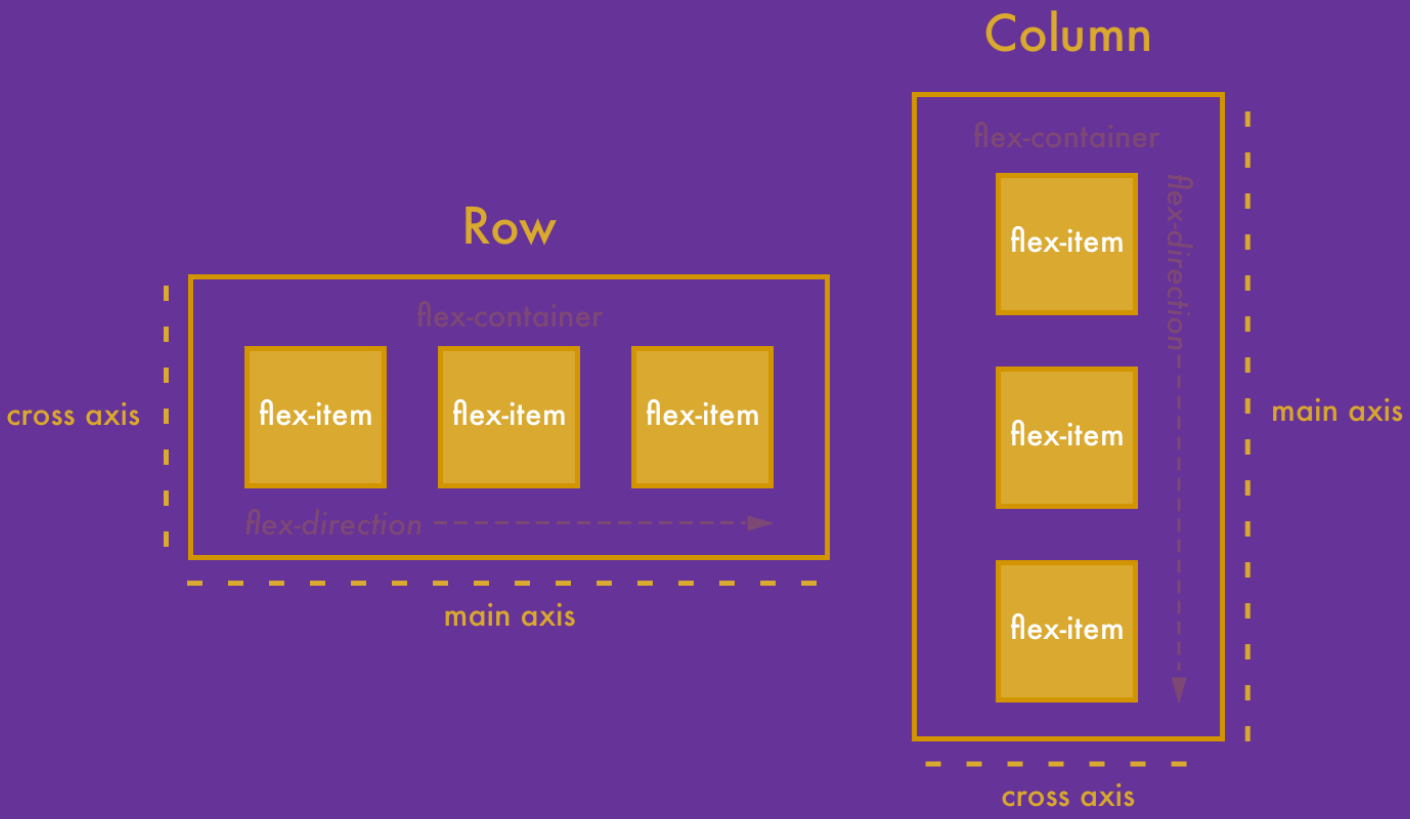
CASCADING STYLE SHEET (CSS)

Class 11
CSS Flexbox

Introduction

Flexbox is a one-dimensional layout method for arranging items in rows or columns. It is designed to distribute space dynamically and align content efficiently.



Key features of Flexbox

Flex Containers and Items: Flexbox works with a container and items inside it. The container sets the layout rules, and the items follow these rules.

Axis Control: Flexbox lets you control the layout direction (horizontal or vertical) and alignment.

Responsive Design: Flexbox makes it easy to create layouts that adjust to different screen sizes.

Alignment and Justification: Flexbox provides properties like justify-content and align-items to align items within the container.

CASCADING STYLE SHEET (CSS)

Basics

To start using Flexbox, apply "`display: flex`" to a container.

```
<div class="container">
  <div>Item 1</div>
  <div>Item 2</div>
  <div>Item 3</div>
</div>
```

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

This element is called the flex container, and stores the sub-elements which are known as flex items

Flex Container Properties

The flex container properties are:

1. Flex Direction

It defines in which direction the flex elements would be displayed. It takes values like row, column or "reverse" too.

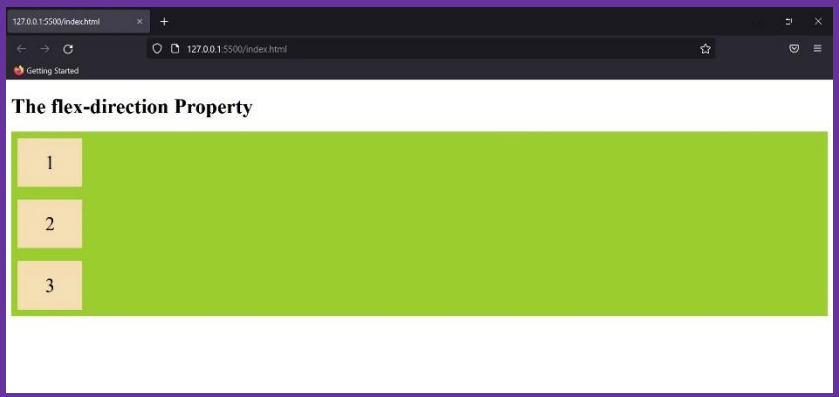
Example:

CASCADING STYLE SHEET (CSS)

index.html

```
<!DOCTYPE html>
<html>

<head>
    <style>
        .flex-container {
            display: flex;
            flex-direction: column;
            background-color: yellowgreen;
        }

        div {
            background-color: wheat;
            width: 100px;
            margin: 10px;
            text-align: center;
            line-height: 75px;
            font-size: 30px;
        }
    </style>
</head>

<body>

    <h1>The flex-direction Property</h1>
    <span class="flex-container">
        <div>1</div>
        <div>2</div>
        <div>3</div>
    </span>

</body>
```

Output:



2. Flex Wrap

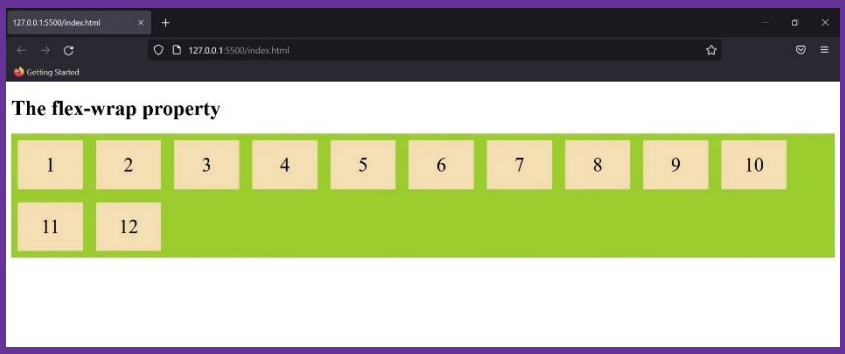
By using this property we can make our elements responsive for different screen sizes.

Example:

CASCADING STYLE SHEET (CSS)

```
.flex-container {
            display: flex;
            flex-direction: row;
            background-color: yellowgreen;
            flex-wrap: wrap;
        }
```

Output:



You can use flex flow as a short to add both these properties. Eg: `{flex-flow: row wrap;}`

3. Justify Content

This property is used to set the position of content or rather align content along the main axis/horizontal axis.

Example:

```
.flex-container {
            display: flex;
            flex-direction: row;
            background-color: yellowgreen;
            justify-content: center;
        }
```

Output:

It can take other values too like “space-around” which distributes flex items equally spaced in the container. Other properties include `flex-end, flex-start, space-between,` etc. (These could be seen in vs code when the justify-content property is selected).

CASCADING STYLE SHEET (CSS)

4. Align Items

Just like the justify-content property, align-items define the alignment of the flex container but along the cross-axis/vertical axis.

Example:

```
.flex-container {
            display: flex;
            height: 200px;
            flex-direction: row;
            background-color: yellowgreen;
            align-items: center;
        }
```

Output:

5. Align Content

This property is very similar to align item but here rather than the flex items, the content present in the item is selected for the property.

Example:

```
.flex-container {
            display: flex;
            height: 200px;
            flex-direction: row;
            background-color: yellowgreen;
            align-content: center;
        }
```

Output:

CASCADING STYLE SHEET (CSS)

Flex Items Properties

The flex item properties are:

1. Order:

As the name suggests, this property sets the order in which the flex items are shown.

Example:

```
<div style="order: 4;">1</div>
<div style="order: 3;">2</div>
<div style="order: 1;">3</div>
<div style="order: 5;">4</div>
<div style="order: 2;">5</div>
```

Output:

2. Flex Grow & Shrink

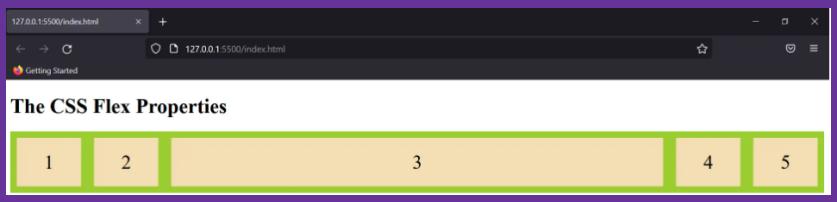
Decides the relative size of flex items. By default, this property is zero and thus items have the same size.

CASCADING STYLE SHEET (CSS)

Example:

```
<div>1</div>
<div>2</div>
<div style="flex-grow: 3;">3</div>
<div>4</div>
<div>5</div>
```

Output:



We can also use flex-shrink to decrease size of an element.

3. Align Self

This property allows default alignment to be overridden for the individual flex items. Try adding inline CSS to see how this property is used.