Lecture - 6
CSS Properties

Course Code: CIS 133 & 133 L

Course Title: Website Development Essential

Course Leader: Md. Faruk Hosen

Overview

- CSS Flexbox
- Media Query
- CSS Grid

CSS Flexbox

CSS flexbox layout allows you to easily format HTML. Flexbox makes it simple to align items vertically and horizontally using rows and columns. Items will "flex" to different sizes to fill the space. It makes responsive design easier.

- The CSS3 flexbox contains flex containers and flex items.

Flexbox specifies how flex items are set inside a flex container. It sets the flex items inside a flex container along a flex line. By default, there is only one flex line per flex container. Everything outside a flex container and inside a flex item is considered as usual.

CSS Flexbox

The flex container properties are:

- **flex-direction**

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

- **flex-wrap**

The flex-wrap property specifies whether the flex items should wrap or not.

- **flex-flow**

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

CSS Flexbox

The flex container properties are:

- **justify-content**

The justify-content property is used to align the flex items.

- **align-content**

The align-content property is used to align the flex lines.

Flexbox Example

```
.flex-container {
    display: flex;
    width: 400px;
    height: 200px;
    background-color: lightpink;
}
.flex-item {
    background-color: brown;
    width: 100px;
    height: 100px;
    margin: 10px;
}
```

```
<div class="flex-container">
  <div class="flex-item">flex item1</div>
  <div class="flex-item">flex item2</div>
  <div class="flex-item">flex item3 </div>
</div>
```

What is Media Query?

CSS Media query is a CSS3 module which is used to adapt to conditions such as screen resolution (e.g. Smartphone screen vs. computer screen).

- It is an extension of media dependent stylesheets used in different media types (i.e. screen and print).
- The most commonly used media feature is "width".
- It uses the **@media** rule to include a block of CSS properties only if a certain condition is true.

What is Responsive Web Design?

The term Responsive Web Design was given by Ethan Marcotte. It facilitates you to use fluid grids, flexible images, and media queries to progressively enhance a web page for different viewing contexts i.e. Desktop, Smartphone, Tablet etc.

Commonly used screen resolutions in different devices:

- Smartphone: 320px
- Tablet: 768px
- Netbook: 1024px
- Desktop: 1600px

Simple use of media query

```
body {
   background-color:yellow;
}
@media only screen and (max-width: 500px) {
   body {
      background-color:green;
   }
}
<body>
<p>If you resize the browser window and the width of this
document is less than 500 pixels, the background-color is
"green", otherwise it is "yellow"</p>
</body>
```

CSS Grid

- A grid can be defined as an intersecting set of horizontal lines and vertical lines. CSS Grid layout divides a page into major regions. Grid property offers a grid-based layout system having rows and columns. It makes the designing of web pages easy without positioning and floating.

- Similar to the table, it enables a user to align the elements into rows and columns. But compare to tables, it is easy to design layout with the CSS grid. We can define columns and rows on the grid by using **grid-template-rows** and **grid-template-columns** properties.

Grid Container

- We can define the grid container by setting the **display** property to **grid** or **inline-grid** on an element.
- Grid container contains grid items that are placed inside rows and columns.

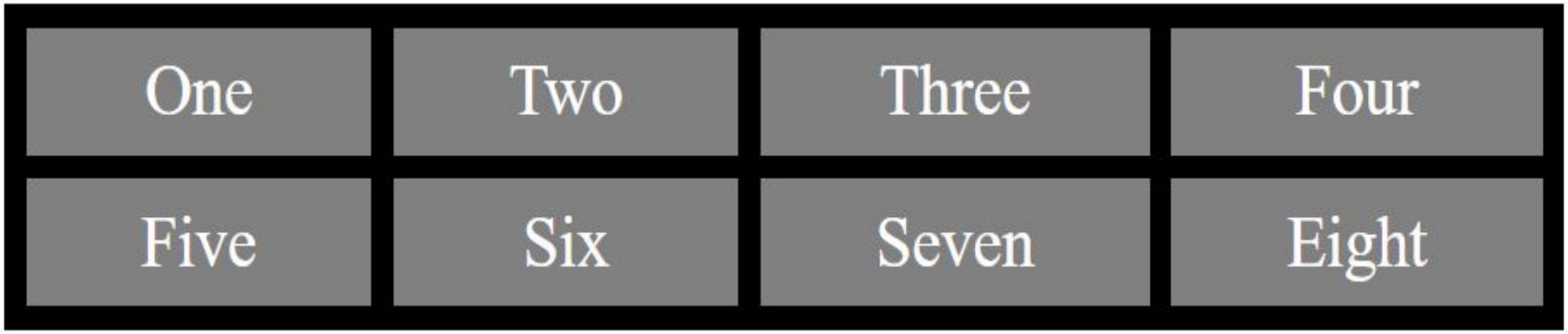
Some of the properties:

- **grid-template-columns**: It is used to specify the size of the columns.
- **grid-template-rows**: It is used to specify the row size.
- **grid-auto-rows**: It is used to specify the automatic size of the rows.
- **grid-auto-columns**: It is used to specify the automatic size of the columns.

Example

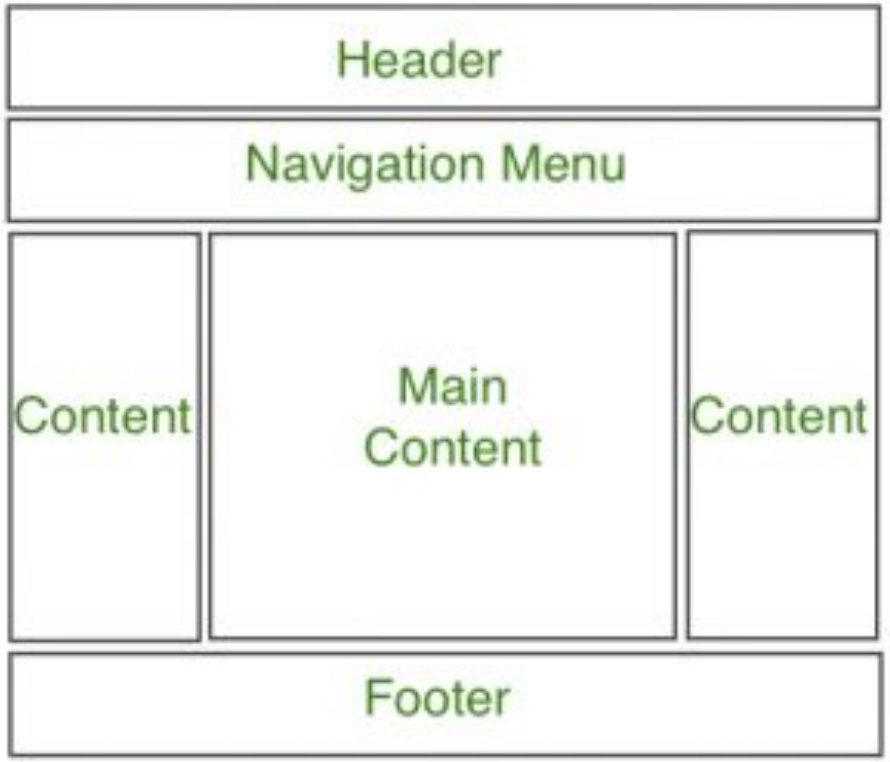
```
.main {
        display: grid;
        grid: auto auto / auto auto auto auto;
        grid-gap: 10px;
        background-color: black;
        padding: 10px;
    }
    .num {
        background-color: grey;
        text-align: center;
        color: white;
        padding: 10px 10px;
        font-size: 30px;}
```

```
<div class="main">
    <div class="num">One</div>
    <div class="num">Two</div>
    <div class="num">Three</div>
    <div class="num">Four</div>
    <div class="num">Five</div>
    <div class="num">Six</div>
    <div class="num">Seven</div>
    <div class="num">Eight</div>
</div>
```



CSS Layout

A website can be divided into various sections comprising of header, menus, content and footer based on which there are many different layout design available for developer. Different layouts can be created by using **div** tag and use CSS property to style it.



CSS Layout

Header Section:

The header section is generally placed either at the top of the Website or just below a top navigation menu. It often comprises of the name of the Website or the logo of the Website.

Example:

```
<div class = "header">
        <h2 style =
"color:white;">
            This is header
section
        </h2>
    </div>
    <br>

    <center
style="font-size:200%;">
        Remaining Section
    </center>
```

```
<style>
  .header {
        background-color:
green;
        padding: 15px;
        text-align: center;
       }
</style>
```

CSS Layout

Navigation Menu:

A Navigation Bar/Menu is basically a list of links that allows visitor to navigate through the website comfortably with easy access.

Example:

```
<div class = "nav_menu">
      <a href = "#">Algo</a>
      <a href = "#">DS</a>
       <a href ="#">AI</a>
</div>
```

```
<style>
            .nav_menu {
               overflow: hidden;
               background-color: #333;}
          .nav_menu a {
               float: left;
               display: block;
               color: white;
               text-align: center;
               padding: 14px 16px;
               text-decoration: none;}
          .nav_menu a:hover {
               background-color: white;
               color: green;}
</style>
```

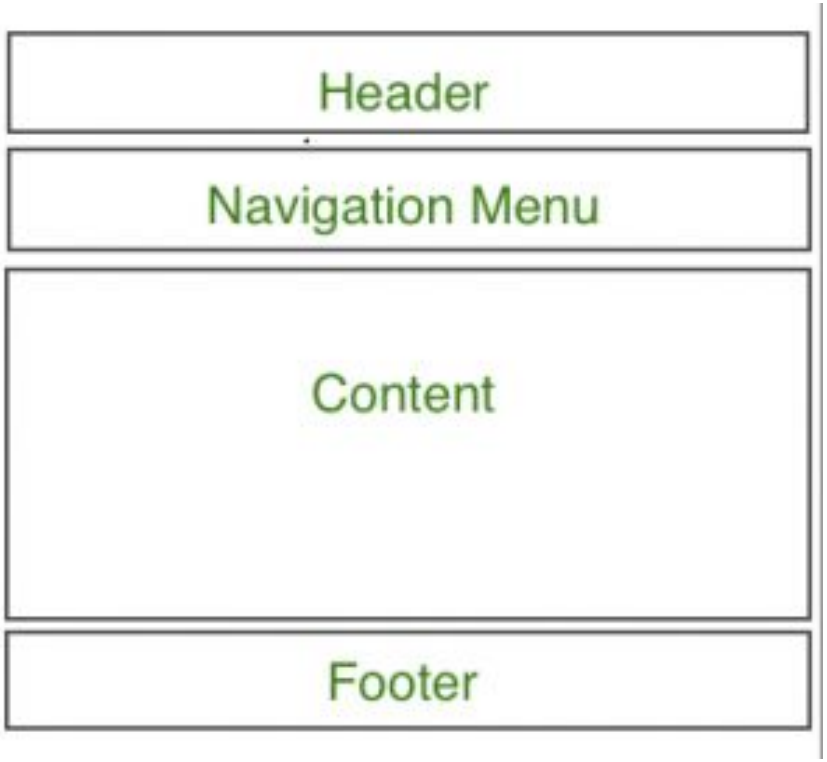
CSS Layout

Content Section:

The content section is the main body of the website. The user can divide content section in n-column layout.

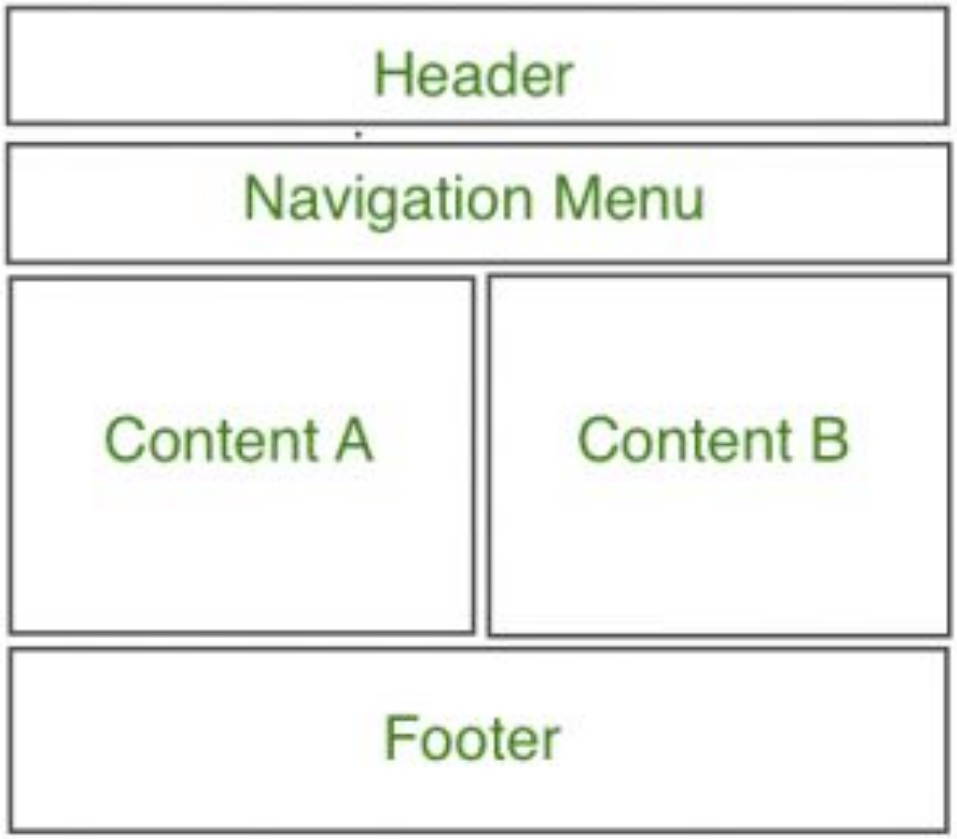
The most common layouts are:

1-Column Layout: It is mostly used for mobile layout



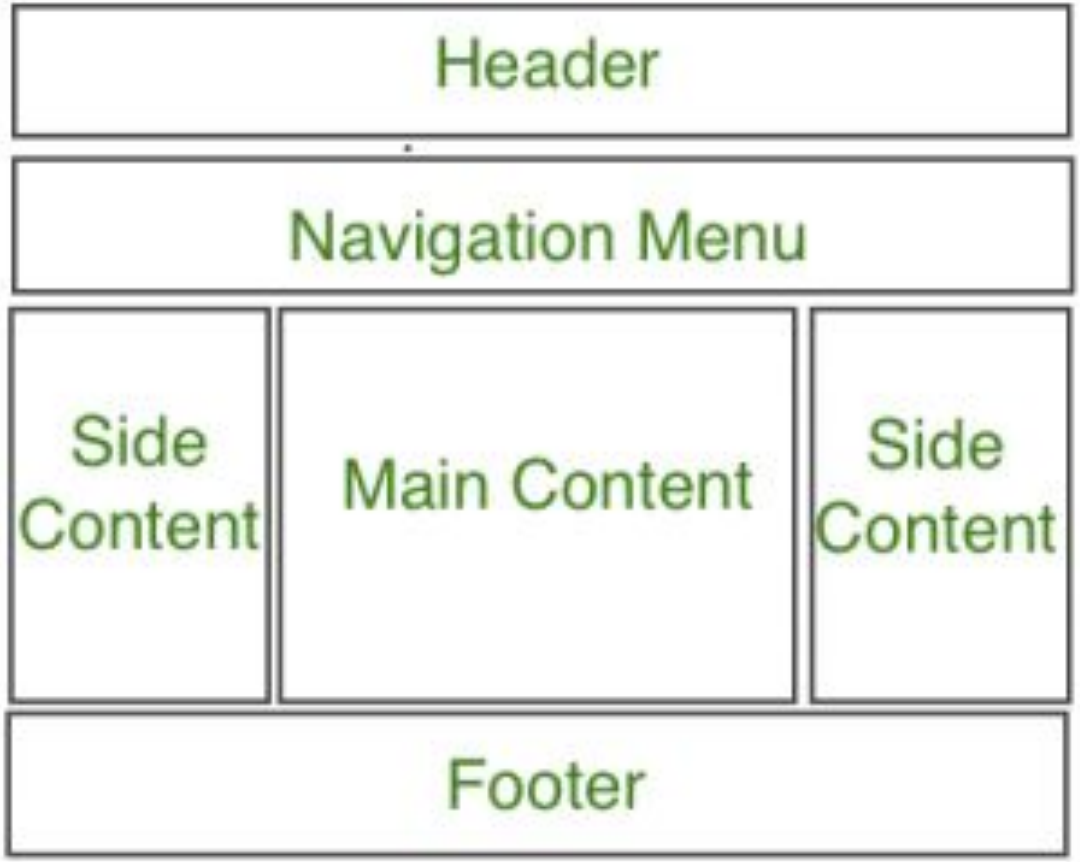
CSS Layout

2-Column Layout: This website layout is mostly used for tablets or laptops.



CSS Layout

3-Column Layout: This website layout is mostly used for desktops.



CSS Layout

The user can also create a responsive layout where the layout will get changed as per screen size. Consider the below example where if width of screen is more than 600px then there will be 3-column layout and if width of screen is between 400px to 600px then there will be 2-column layout and if screen size less than 400px then 1-column layout will display.

Footer Section:

A footer section is placed at the bottom of the webpage and it generally consists of information like contact info, copyrights, About us etc.