

ENSF 381

Full Stack Web Development

Lecture 10:

CSS Layout Fundamentals

Slides: Ahmad Abdellatif, PhD

Instructor: Novarun Deb, PhD

Outline

- Recap
- Margins
- Padding
- Opacity
- Display

Recap: Division (<div>) element

- A generic container or division that is used to group together and structure content on a web page.
- Serves as a container for other HTML elements.

Recap: <div> - example

```
<!DOCTYPE html>
<html>
<head>
<title>ENSF381</title>
</head>
<body>

<div>
  Welcome to the ENSF381 course.
</div>

</body>
</html>
```

Recap: <div> - example



Containers - example

```
<html>
<head>
  <style>
    .container {
      width: 300px;
      border: 1px solid #ccc;
    }
    .box {
      width: 100%;
      height: 50px;
      background-color: #f0f0f0;
    }
  </style>
  <title>Margins Example</title>
</head>
<body>

  <div class="container">
    <div class="box">Box 1</div>
    <div class="box">Box 2</div>
    <div class="box">Box 3</div>
  </div>
</body></html>
```

Represents a light gray color in hexadecimal color code.

Containers - example



Margins

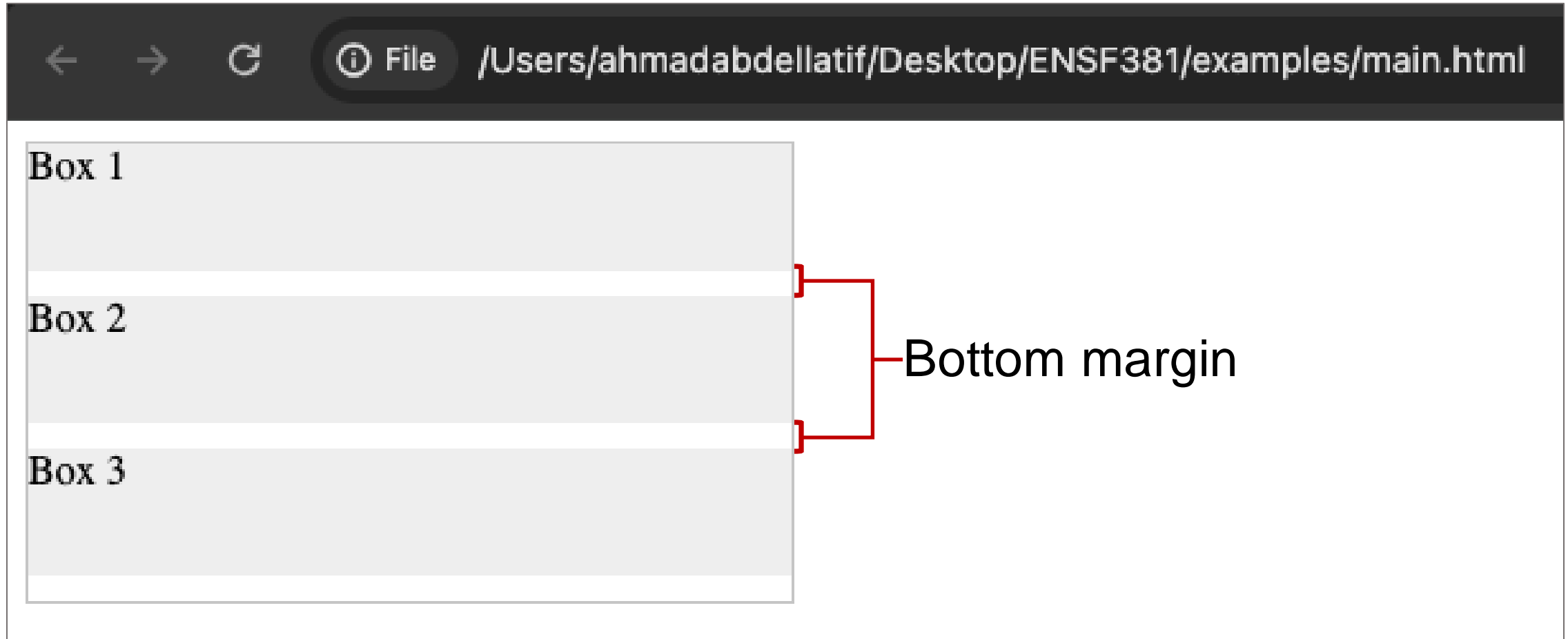
- Refer to the **space around** an element's content.
- Margins create space between an element's border and adjacent elements.
- It can be set for all sides (top, right, bottom, left) or individually. The values can be specified in various units such as pixels, ems, rems, percentages, etc.

Adding margins to the containers - example

```
<html>
<head>
  <style>
    .container {
      width: 300px;
      border: 1px solid #ccc;
    }
    .box {
      width: 100%;
      height: 50px;
      background-color: #f0f0f0;
      margin-bottom: 10px; /* Setting margin for the bottom */
    }
  </style>
  <title>Margins Example</title>
</head>
<body>

  <div class="container">
    <div class="box">Box 1</div>
    <div class="box">Box 2</div>
    <div class="box">Box 3</div>
  </div>
</body></html>
```

Adding margins to the containers - example



Margins - example

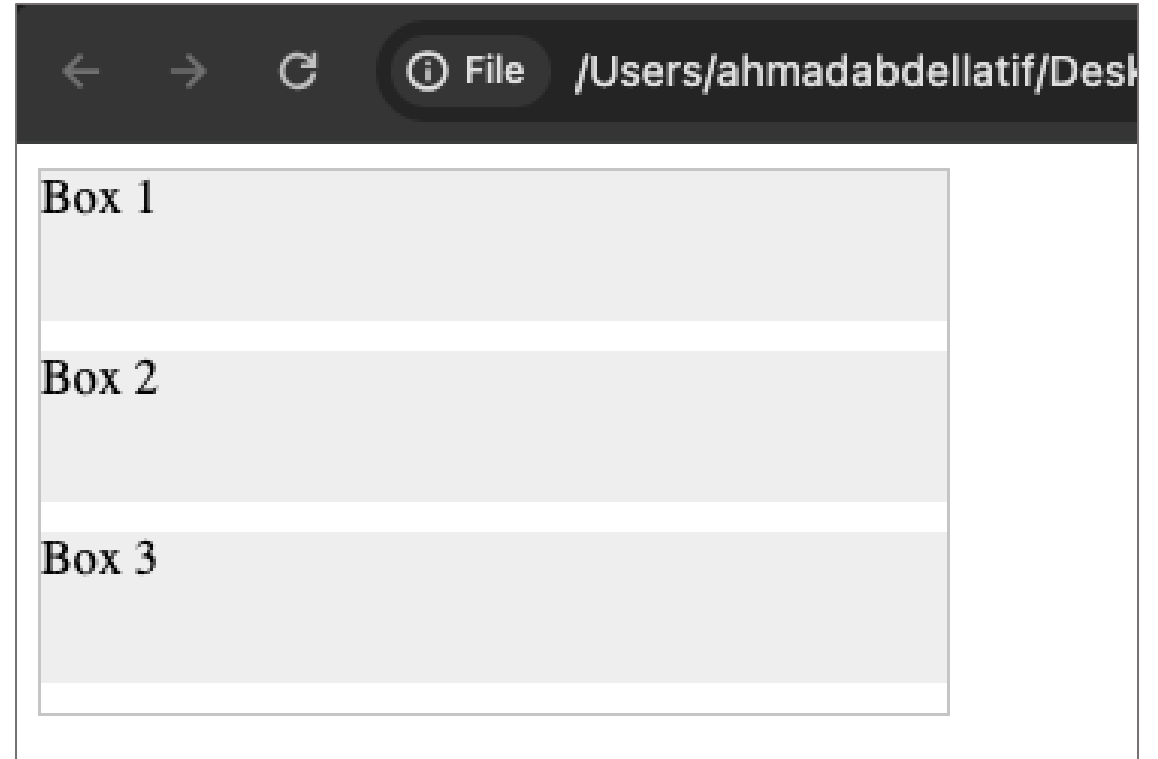
```
<html> <head>
  <style>
    /* CSS styles */
    .container {
      width: 300px;
      margin: 20px; /* Setting margins for all sides */
      border: 1px solid #ccc;
    }
    .box {
      width: 100%;
      height: 50px;
      background-color: #f0f0f0;
      margin-bottom: 10px; /* Setting margin for the bottom */
    }
  </style>
  <title>Margins Example</title>
</head>
<body>

  <div class="container">
    <div class="box">Box 1</div>
    <div class="box">Box 2</div>
    <div class="box">Box 3</div>
  </div>
</body></html>
```

Margins - example



With Margins



Without Margins

Another way to define margins

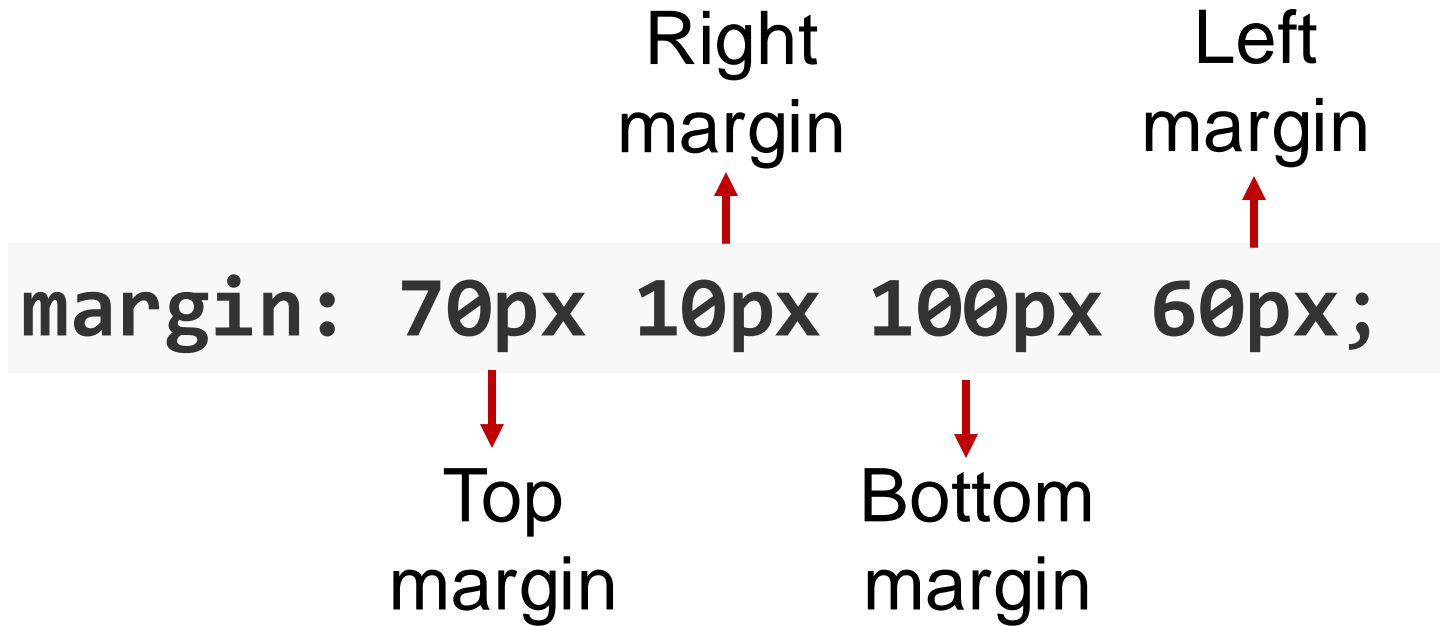
margin: 70px 10px 100px 60px;

Top margin

Right margin

Bottom margin

Left margin



Padding

- The space between an element's content and its border.
- Used to create space within an element and separates the content from the border.
- Padding is important for controlling the internal spacing of an element and ensuring that the content doesn't touch the borders or other adjacent elements.

Padding - example

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .box {
      width: 200px;
      height: 100px;
      border: 1px solid #ccc;
    }
  </style>
  <title>Padding Example</title>
</head>
<body>
  <div class="box">
    This is a box with padding.
  </div>
</body>
</html>
```

Before adding the padding - example



file:///Users/ahmadabdellatif/Desktop/ENSF381/examples/padding.html

This is a box with padding.

Padding - example

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .box {
      width: 200px;
      height: 100px;
      border: 1px solid #ccc;
      padding: 20px; /* Padding for all sides */
    }
  </style>
  <title>Padding Example</title>
</head>
<body>
  <div class="box">
    This is a box with padding.
  </div>
</body>
</html>
```

After adding the padding - example



Margin vs. Padding

Margin

- Represents the space **outside an element's border**.
- It controls the **external** spacing between an element and its neighboring elements.

Padding

- Represents the space **between an element's content and its border**.
- It controls the **internal** spacing within an element, separating the content from the border.

Styling images

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .image {
      width: 20%;
      height: 100px;
      opacity: 0.7; /* Set opacity value between 0 (completely transparent) and 1 (completely opaque) */
    }
  </style>
  <title>Image with Opacity Example</title>
</head>
<body>
  
</body>
</html>
```

Styling images



**Total opacity
(opacity = 1)**



Opacity = 0.7



Opacity = 0.1

Hover effect

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .image {
      width: 20%;
      height: 100px;
      opacity: 0.7; /* Set opacity value between 0 (completely transparent) and 1 (completely opaque) */
    }

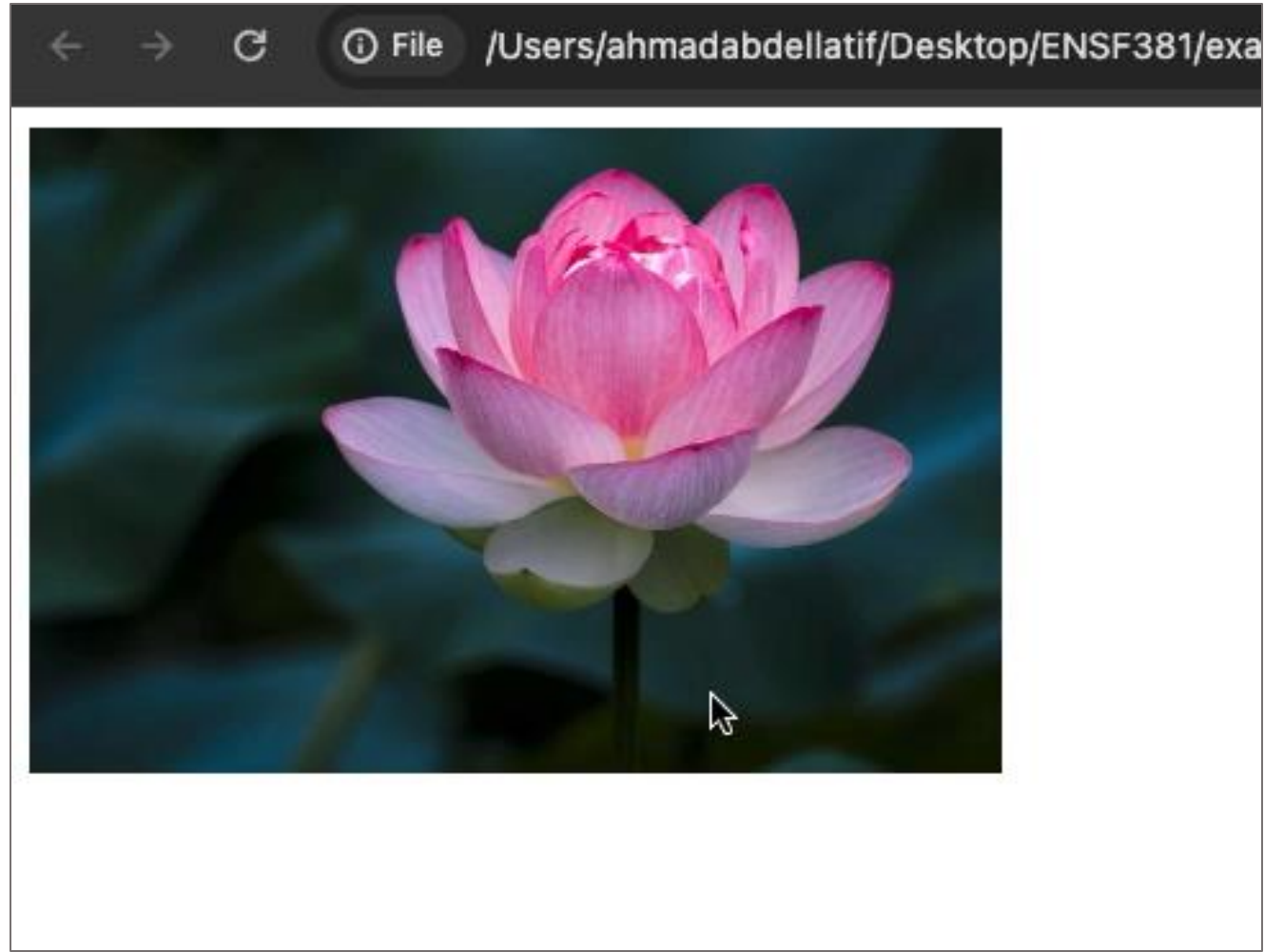
    .image:hover {
      opacity: 1.0;
    }
  </style>
  <title>Image with Opacity Example</title>
</head>
<body>

</body>
</html>
```

Applies styles specifically to an element with the class "image" when the user hovers over it.

Hover effect



 element

A generic inline container that can be used for various purposes, such as applying styles or scripting to specific inline content within a larger block of text.

```
<p>This is a <span style="color: blue;">blue</span> word.</p>
```

[JSBin](#)

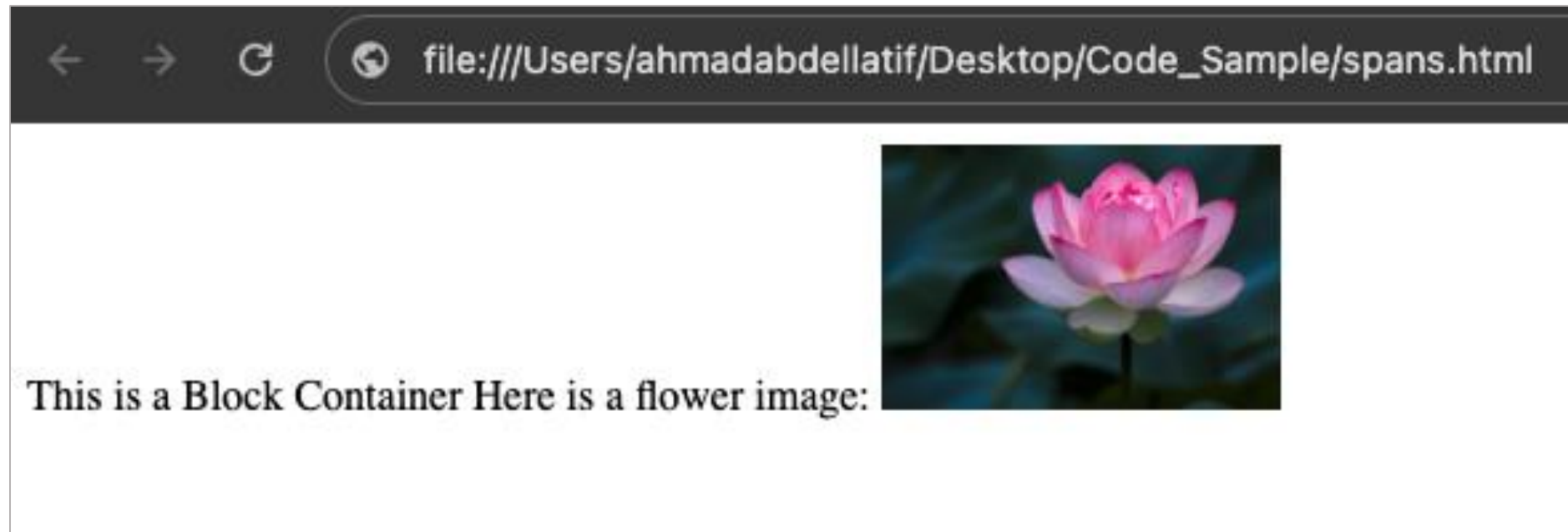


This is a blue word.

Span - example

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      img {
        max-width: 100%;
        height: 100px;
      }
    </style>
    <title>Block Example</title>
  </head>
  <body>
    <div>
      <span>This is a Block Container</span>
      <span>Here is a flower image:</span>
      
    </div>
  </body>
</html>
```

Span - example



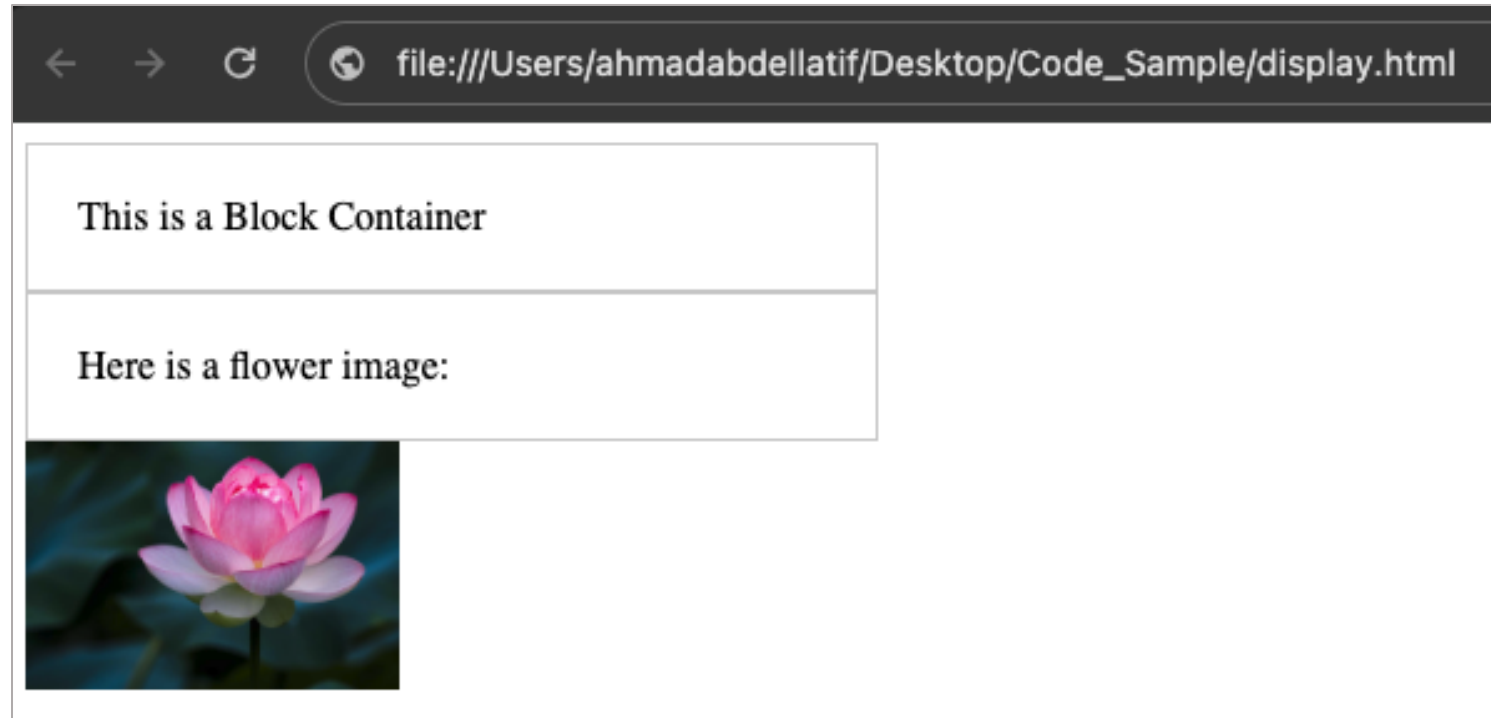
Display property

- Used to define how an HTML element should be displayed on the web page.
- Takes various values, each of which dictates different rendering behaviors:
 - **Block**: starts on a new line and takes up the full width available.
 - **Inline**: It does not start on a new line and only takes up as much width as necessary.
 - **Grid**: allowing child elements to be placed into a grid layout.
 - **Flex**: allowing child elements to be laid out in any direction.

Display property – Block example

```
<!DOCTYPE html>
<html>
<head>
  <style>
    span {
      display: block;
      width: 300px;
      border: 1px solid #ccc;
      padding: 20px;
    }
    img {
      max-width: 100%;
      height: 100px;
    }
  </style>
  <title>Block Example</title>
</head>
<body>
  <div>
    <span>This is a Block Container</span>
    <span>Here is a flower image:</span>
    
  </div>
</body>
</html>
```

Display property – Block example



Display property – Inline example

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .inline-container {
      display: inline;
      border: 1px solid #ccc;
      padding: 10px;
    }

    /* Add some styles for better visibility */
    .inline-container span {
      background-color: #f0f0f0;
      padding: 5px;
      margin-right: 5px;
    }

    a {
      color: #007bff;
    }
  </style>
  <title>Inline Example</title>
</head>
<body>

  <div class="inline-container">
    <span>Inline Container:</span>
    <a href="https://ucalgary.ca">University of Calgary</a>
  </div>
</body>
</html>
```

Applies the specified styles to the `` elements that are descendants of an element with the class "inline-container".

Display property – Inline example



Questions

