

ENSF 381

Full Stack Web Development

Lecture 11:

Flexbox

Slides: Ahmad Abdellatif, PhD

Instructor: Novarun Deb, PhD

Outline

- Flexbox
- Flexbox properties (direction, wrap, and order)
- Responsive layout

What is Flexbox

- A layout model in CSS designed to create efficient and predictable layouts for user interfaces.
- Distribute space and align items within a container, even **when the size of the items is unknown or dynamic.**
- It provides an easy and clean way to design complex layouts.

Advantages of Flexbox

- Allows designers to control the visual order of flex items, independent of their order in the source code.
- Allows items within a container to grow or shrink to fill the available space.
- Flexbox is well-suited for creating responsive designs that adapt to different screen sizes and devices.
- Minimize the code needed to achieve complex layouts.

Key concepts

- Flex Container: an element becomes a flex container when you set its display property to **flex**.
- The children of a flex container become **flex items**.
- Flex Direction: identify the direction of the items.
- Order Property: Specifies the order in which a flex item appears in the flex container.

Flexbox - example

```
<!DOCTYPE html>
<html>
<head>
  <style>
```

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

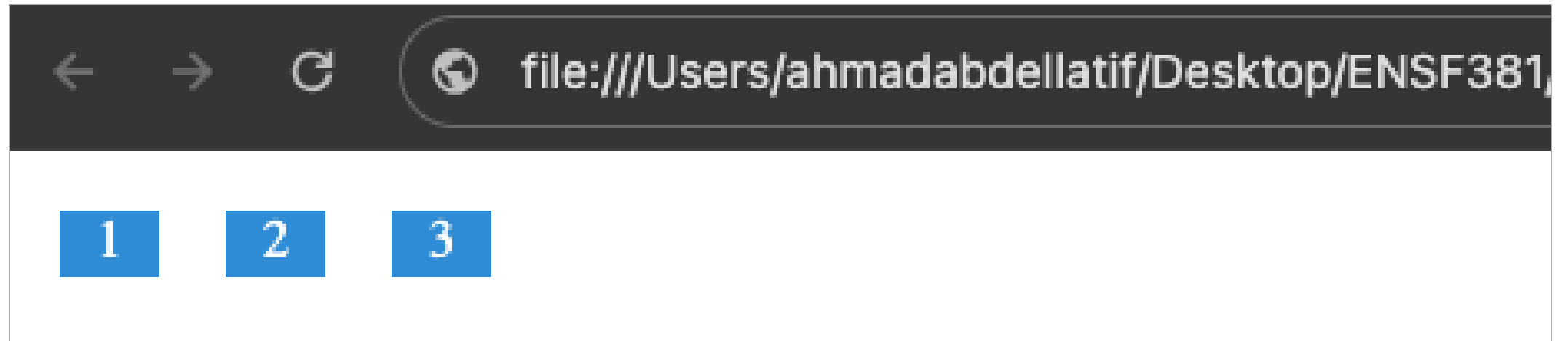
```
.flex-item {
  width: 30px;
  height: 20px;
  margin: 10px;
  background-color: #3498db;
  color: #ffffff;
  text-align: center;
}
```

Set the horizontal alignment of the text content within an element.

```
</style>
</head>
<body>
```

```
<div class="flex-container">
  <div class="flex-item">1</div>
  <div class="flex-item">2</div>
  <div class="flex-item">3</div>
</div>
</body> </html>
```

Flexbox - example



Flex direction

- Used to define the direction of the main axis along which the flex items are placed inside a flex container.
- It determines whether the items should be arranged horizontally or vertically.
- It can take one of four values:
 - **row** (default): arranges the flex items along the main axis from left to right in a horizontal line.
 - **row-reverse**: this value is similar to **row**, but it arranges the flex items from right to left in a horizontal line.
 - **column**: arranges the flex items along the main axis from top to bottom in a vertical line.
 - **column-reverse**: arranges the flex items from bottom to top in a vertical line.

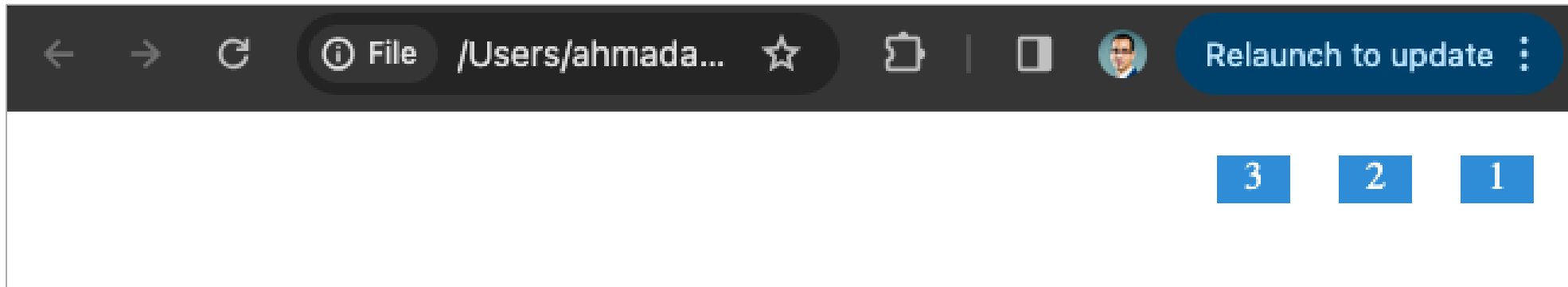
Flex direction – example (row reverse)

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .flex-container {
      display: flex;
      flex-direction: row-reverse;
    }

    .flex-item {
      width: 30px;
      height: 20px;
      margin: 10px;
      background-color: #3498db;
      color: #ffffff;
      text-align: center;
    }
  </style>
</head>
<body>

<div class="flex-container">
  <div class="flex-item">1</div>
  <div class="flex-item">2</div>
  <div class="flex-item">3</div>
</div>
</body></html>
```

Flex direction – example (row reverse)



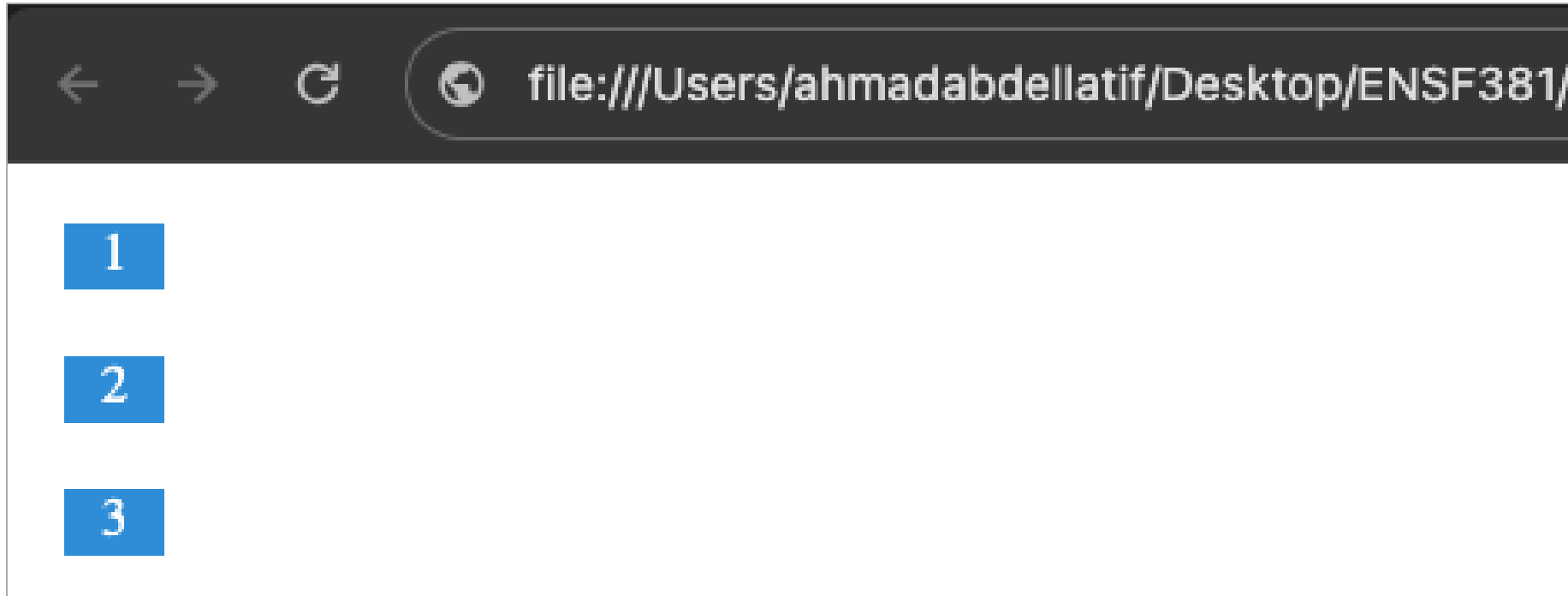
Flex direction – example (column)

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .flex-container {
      display: flex;
      flex-direction: column;
    }

    .flex-item {
      width: 30px;
      height: 20px;
      margin: 10px;
      background-color: #3498db;
      color: #ffffff;
      text-align: center;
    }
  </style>
</head>
<body>

<div class="flex-container">
  <div class="flex-item">1</div>
  <div class="flex-item">2</div>
  <div class="flex-item">3</div>
</div>
</body></html>
```

Flex direction – example (column)



Flex wrap

- Controls whether the flex container should wrap its contents onto multiple lines or stay on a single line.
- Helps manage the layout of flex items when there is insufficient space along the main axis.
- It can take one of three values:
 - **nowrap** (default): all items are forced onto a single line.
 - **wrap**: allows the flex container to wrap its items onto multiple lines if necessary. Items will flow onto the next line **when they exceed the available space** along the main axis.
 - **wrap-reverse**: this value is similar to **wrap**, but it reverses the wrapping direction.

Flex wrap – example (wrap)

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .flex-container {
      display: flex;
      flex-wrap: wrap;
    }

    .flex-item {
      width: 100px;
      height: 100px;
      margin: 10px;
      background-color: #3498db;
      color: #ffffff;
      text-align: center;
      line-height: 100px;
    }
  </style>
</head>
<body>

<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></html>
```

Controls the height of each line of text within the element.

Flex wrap – example (wrap)



Flex wrap – example (wrap)

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .flex-container {
      display: flex;
      flex-wrap: wrap;
      width: 400px;
    }

    .flex-item {
      width: 100px;
      height: 100px;
      margin: 10px;
      background-color: #3498db;
      color: #ffffff;
      text-align: center;
      line-height: 100px;
    }
  </style>
</head>
<body>
<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></html>
```


Flex wrap – example (wrap)

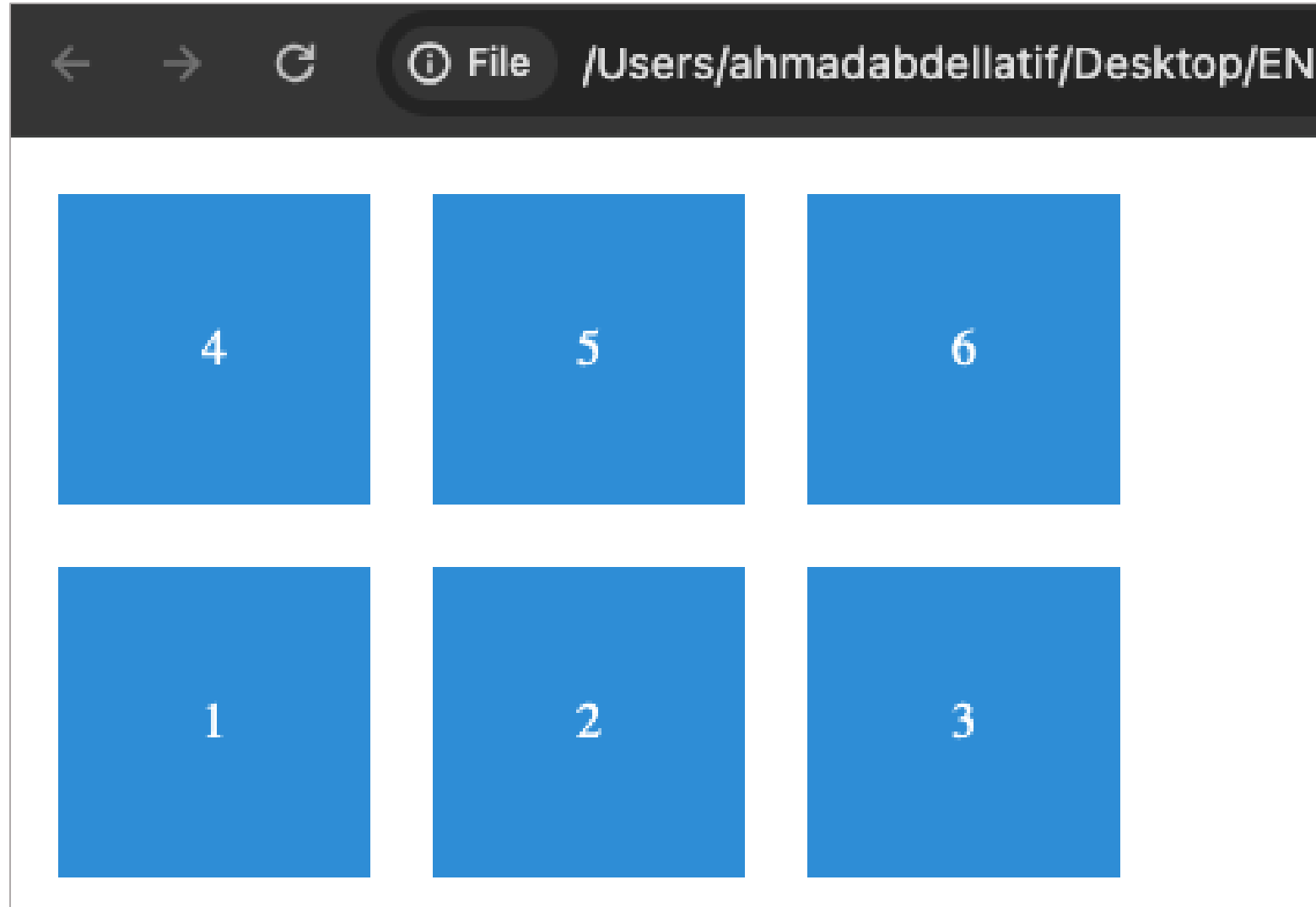


Flex wrap – example (wrap-reverse)

```
<!DOCTYPE html>
<html>
<head>
  <style>
    .flex-container {
      display: flex;
      flex-wrap: wrap-reverse;
      width: 400px;
    }

    .flex-item {
      width: 100px;
      height: 100px;
      margin: 10px;
      background-color: #3498db;
      color: #ffffff;
      text-align: center;
      line-height: 100px;
    }
  </style>
</head>
<body>
<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></html>
```

Flex wrap – example (wrap-reverse)



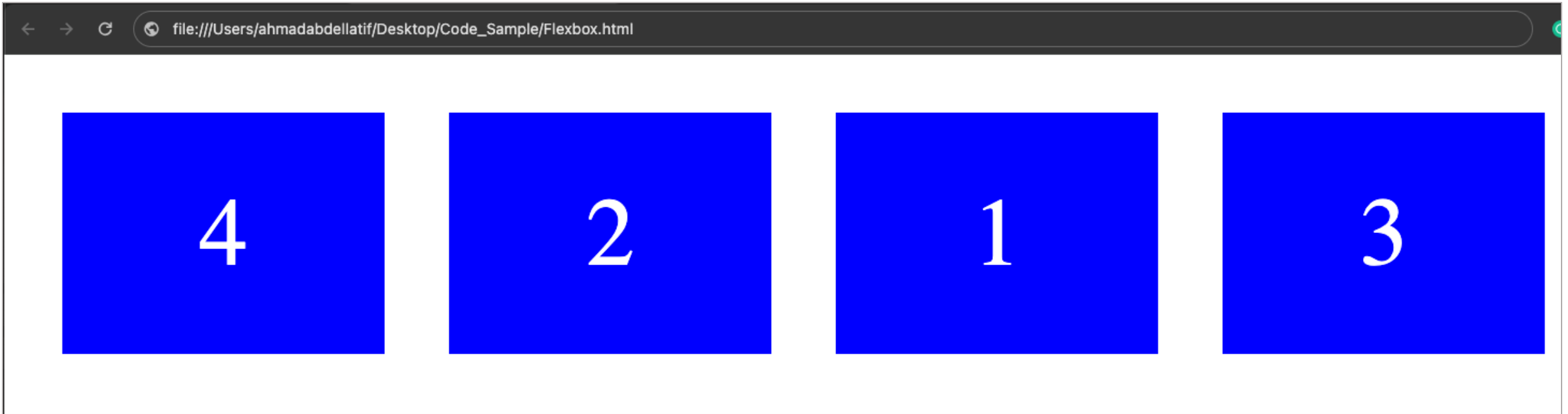
Order property - example

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
}

.flex-container div {
  background-color: blue;
  color: white;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<div class="flex-container">
  <div style="order: 3">1</div>
  <div style="order: 2">2</div>
  <div style="order: 4">3</div>
  <div style="order: 1">4</div>
</div>
</body></html>
```

Order property: used to specify the visual order of a flex item within its flex container.

Order property - example



Responsive layout

A design approach that ensures a web page or application looks and functions well on various devices and screen sizes.

The goal of responsive design is to provide an **optimal user experience**, regardless of whether the user is using a desktop computer, laptop, tablet, or smartphone.

Responsive layout cont.

Media Queries: rules that apply styles based on the characteristics of the device or viewport, such as screen width, height, or device orientation.

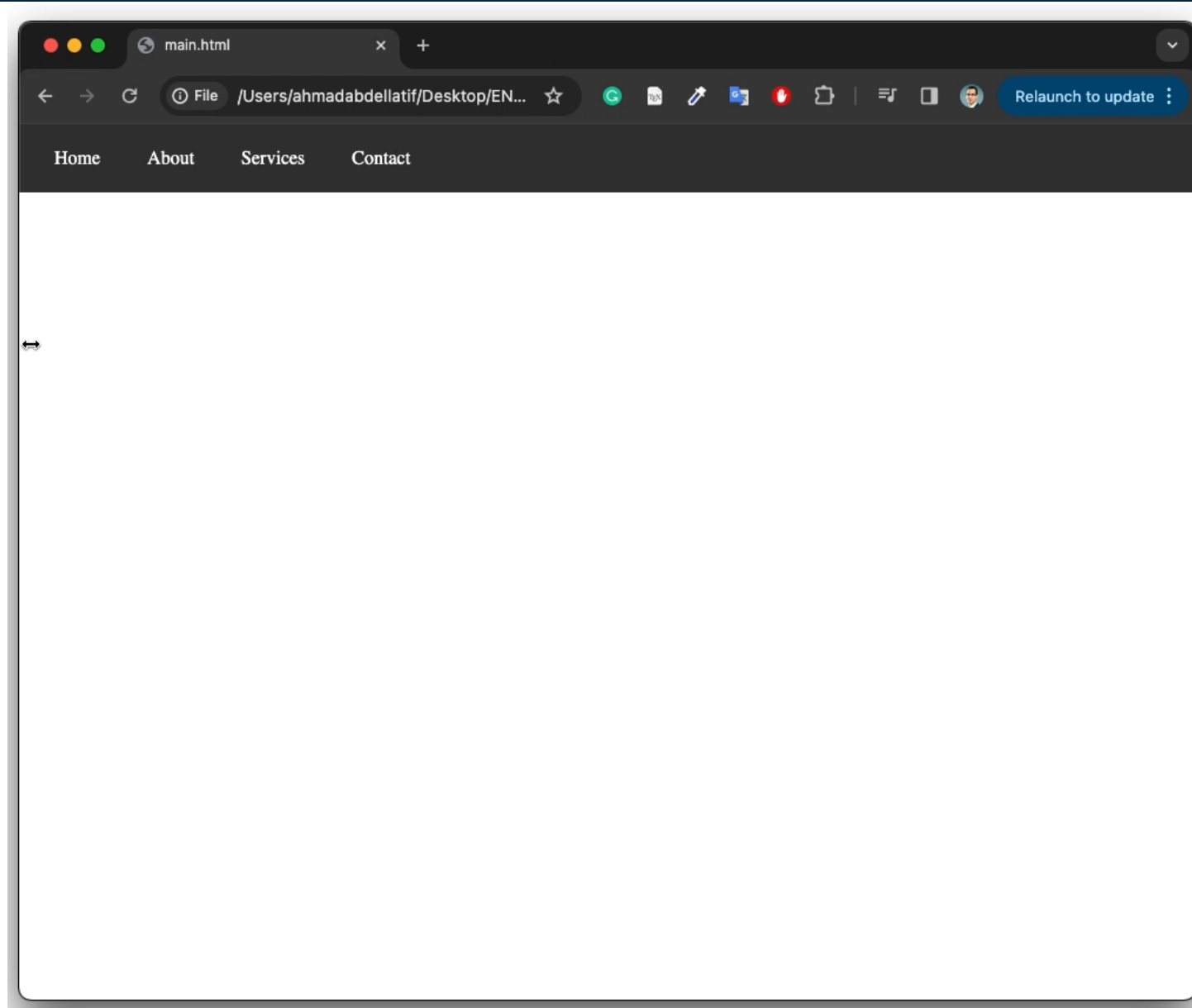
```
@media (max-width: 600px) {  
    /* Styles for small screens */  
}
```

Responsive layout - example

```
<!DOCTYPE html>
<html><head>
  <style>
    body {
      margin: 0;
    }
    div {
      display: flex;
      background-color: #333;
      color: #fff;
      padding: 10px;
    }
    .nav-item {
      margin: 0px 10px;
    }
    @media (max-width: 800px) {
      div {
        flex-direction: column;
        align-items: center;
      }
      .nav-item {
        margin: 5px 0px;
      }
    }
  </style>
</head> <body>
<div>
  <div class="nav-item">Home</div>
  <div class="nav-item">About</div>
  <div class="nav-item">Services</div>
  <div class="nav-item">Contact</div>
</div>
</body> </html>
```

shorthand property: it applies a margin of 0 pixels to the top and bottom of the element and a margin of 10 pixels to both the right and left sides.

Responsive layout - example



Questions

