COSI 152A

Web Application Development

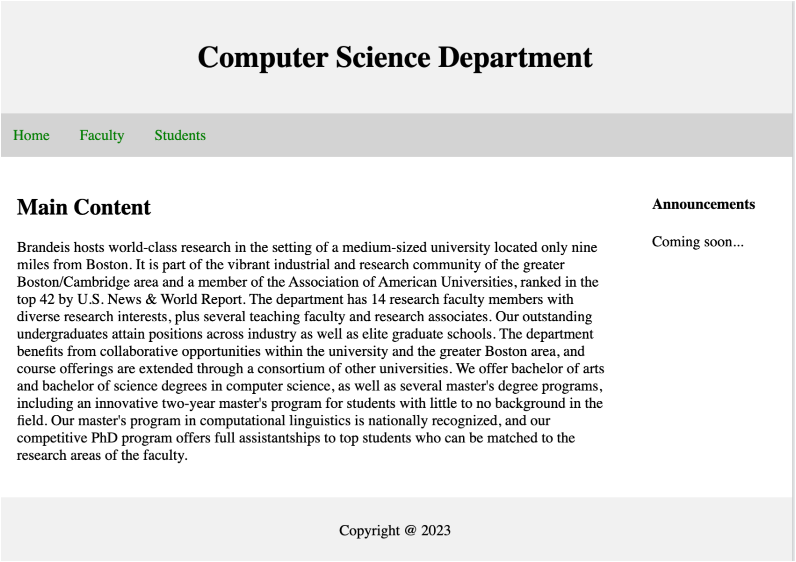
Page Layout

Page layout

- The <div> tag is a generic block level element that can be used for any division or section of your page.
- The <span> tag is a generic inline element for specifying any range of text inside a box.
- By using these fundamental display tags, combined with CSS selectors we can write powerful and reusable CSS rules to style and layout pages.

Motivation for page sections

- Want to be able to style individual elements, groups of elements, sections of text or sections of the page.
- Want to create complex page layouts.



Computer Science Department

Main Content

Brandeis hosts world-class research in the setting of a medium-sized university located only nine miles from Boston. It is part of the vibrant industrial and research community of the greater Boston/Cambridge area and a member of the Association of American Universities, ranked in the top 42 by U.S. News & World Report. The department has 14 research faculty members with diverse research interests, plus several teaching faculty and research associates. Our outstanding undergraduates attain positions across industry as well as elite graduate schools. The department benefits from collaborative opportunities within the university and the greater Boston area, and course offerings are extended through a consortium of other universities. We offer bachelor of arts and bachelor of science degrees in computer science, as well as several master's degree programs, including an innovative two-year master's program for students with little to no background in the field. Our master's program in computational linguistics is nationally recognized, and our competitive PhD program offers full assistantships to top students who can be matched to the research areas of the faculty.

Announcements

Coming soon...

Copyright @ 2023

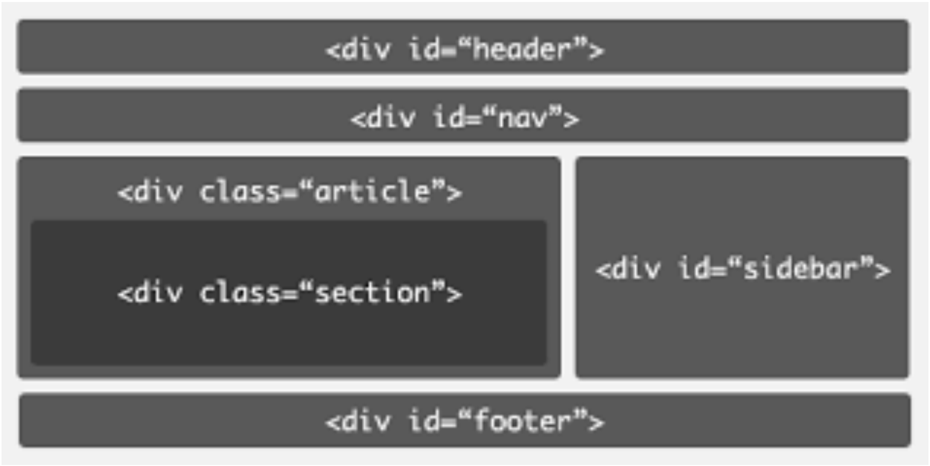
Sections of a page: <div> vs <span>

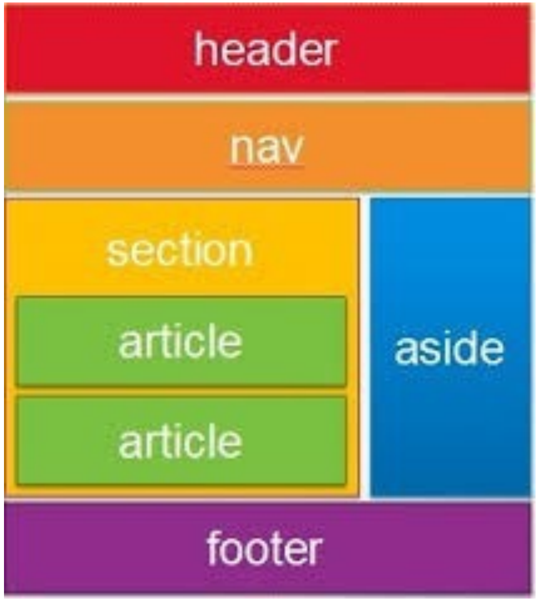
- <div> is a block element
- <span> is an inline element
- They have no onscreen appearance, but you can apply styles to them

```
<div class="shout">
   <h2>Hello</h2>
   <p class="special">See our specials!</p>
   <p>We'll beat<span class="shout"> all prices!</span></p>
</div>
```

HTML5 tags for page sections

- Serve the same purpose as div – more semantic and descriptive than divs
  - Section can be section of an article or section of document containing articles



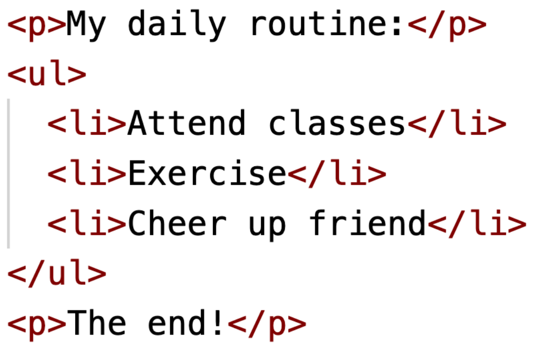


CSS Layout

- CSS page layout techniques allow us to take elements contained in a web page and control where they're positioned relative to the following factors:
  - Their default position in normal layout flow
  - The other elements around them
  - Their parent container
  - And the main viewport/window

Normal Flow

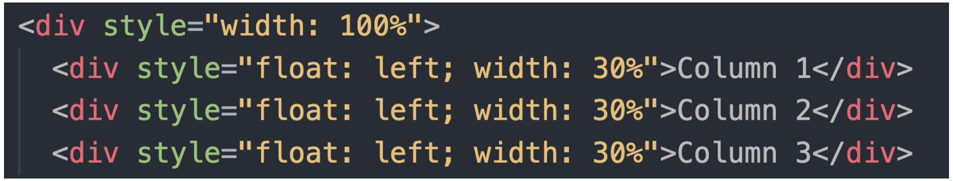
- Normal flow is how the browser lays out HTML pages by default when you do nothing to control page layout.
  - elements stacked on top of one another.



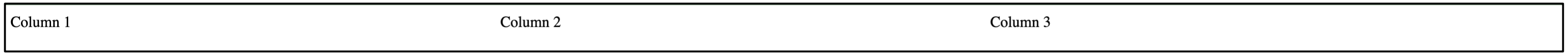
```
<p>My daily routine:</p>
<ul>
  <li>Attend classes</li>
  <li>Exercise</li>
  <li>Cheer up friend</li>
</ul>
<p>The end!</p>
```

Float

- Float is a CSS positioning property that allows an element to be placed alongside the others and “float” to the left or right of its parent container.



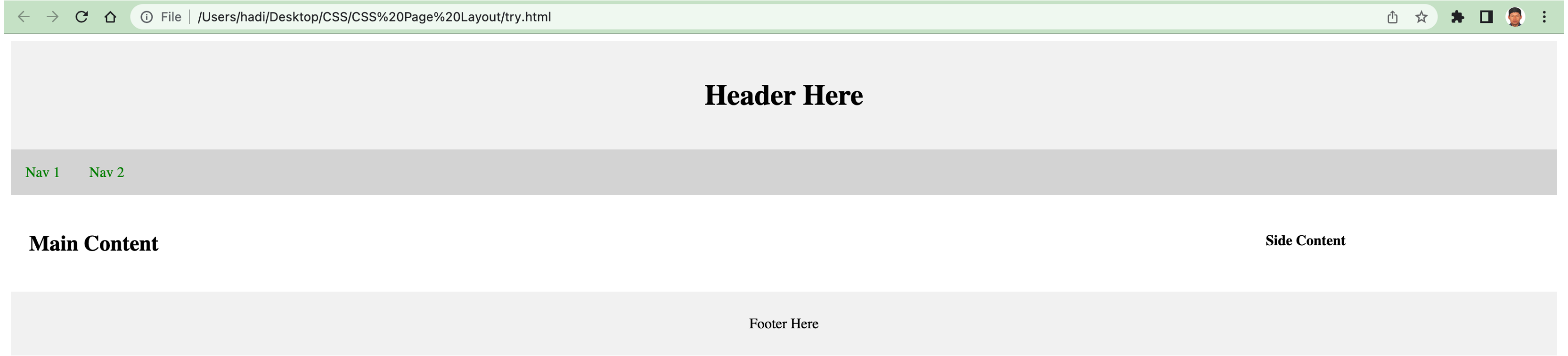
```
<div style="width: 100%">
  <div style="float: left; width: 30%">Column 1</div>
  <div style="float: left; width: 30%">Column 2</div>
  <div style="float: left; width: 30%">Column 3</div>
```



Column 1 Column 2 Column 3

Web Layout Sample

- Let’s create following web layout



Header Here

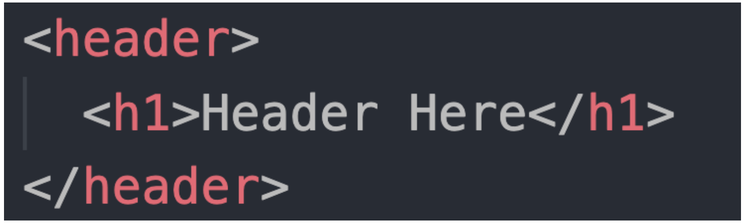
Nav 1 Nav 2

Main Content

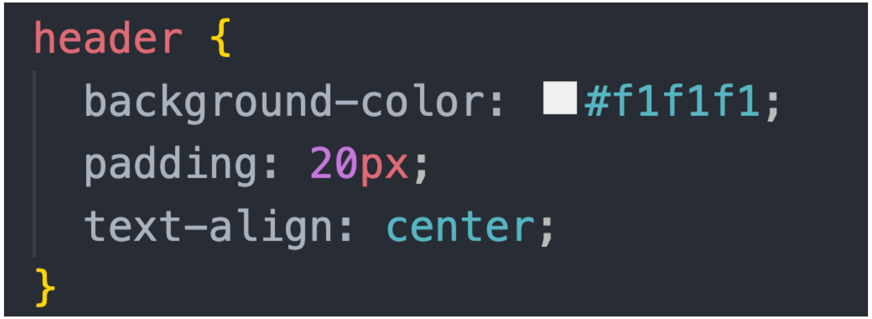
Side Content

Footer Here

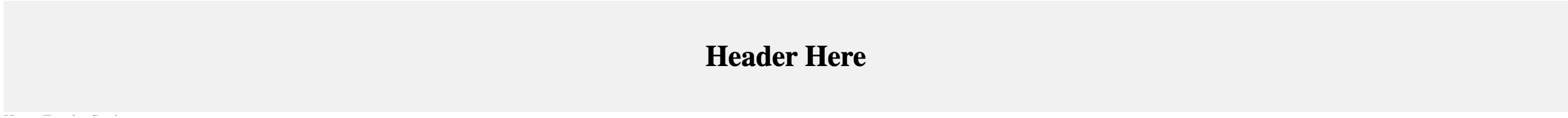
Header: <header>



```html
<header>
  <h1>Header Here</h1>
</header>
```

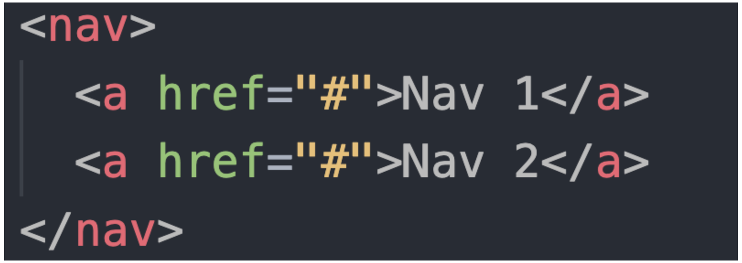


```css
header {
  background-color: ⬜#f1f1f1;
  padding: 20px;
  text-align: center;
}
```

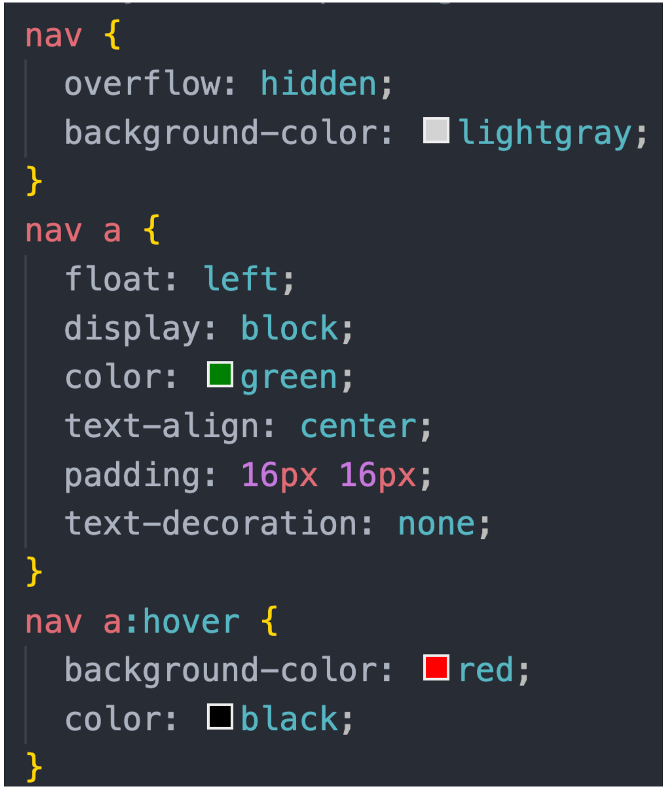


Header Here

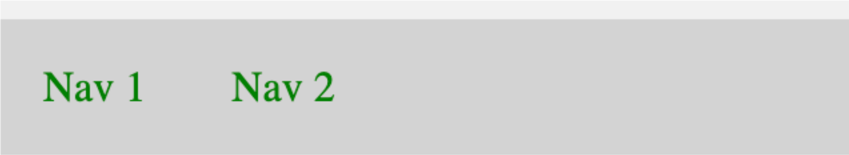
Navigation Menu: <nav>



```html
<nav>
  <a href="#">Nav 1</a>
  <a href="#">Nav 2</a>
</nav>
```

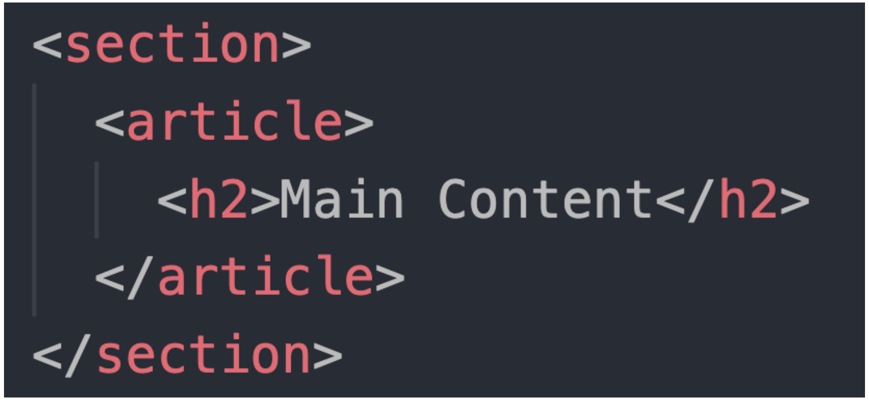


```css
nav {
  overflow: hidden;
  background-color: ⬜lightgray;
}
nav a {
  float: left;
  display: block;
  color: 🟩green;
  text-align: center;
  padding: 16px 16px;
  text-decoration: none;
}
nav a:hover {
  background-color: 🟥red;
  color: ⬛black;
}
```

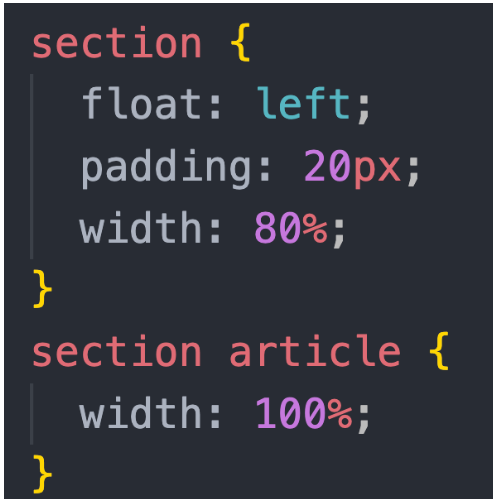


Nav 1 Nav 2

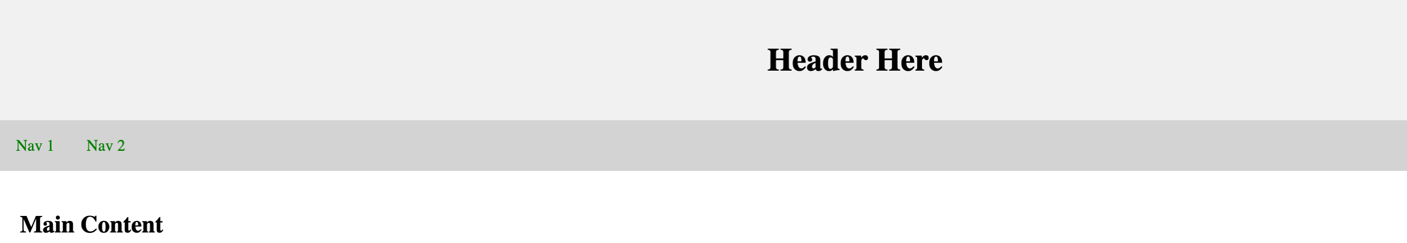
Section: <section>



```html
<section>
  <article>
    <h2>Main Content</h2>
  </article>
</section>
```



```css
section {
  float: left;
  padding: 20px;
  width: 80%;
}
section article {
  width: 100%;
}
```

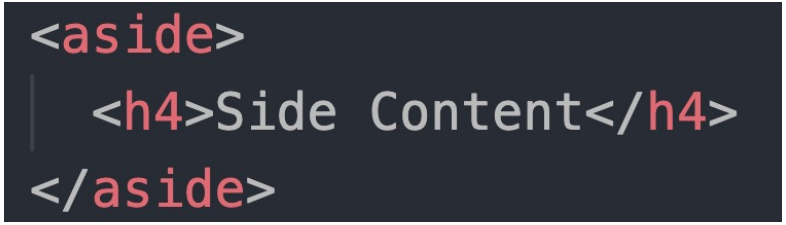


Header Here

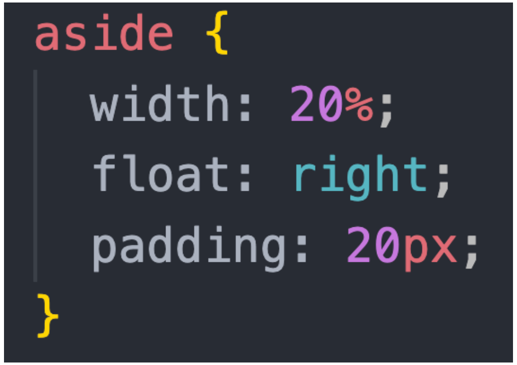
Nav 1 Nav 2

Main Content

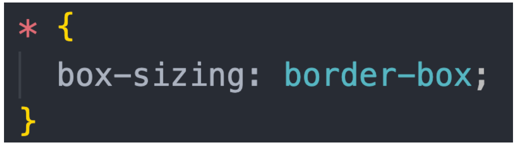
Side section: <aside>



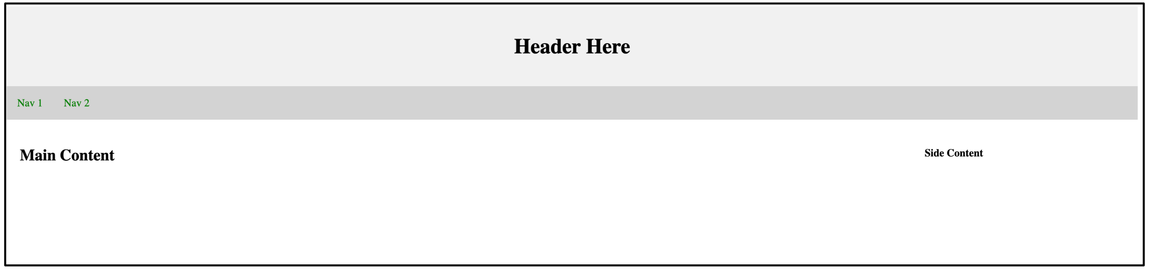
```
<aside>
  <h4>Side Content</h4>
</aside>
```



```
aside {
  width: 20%;
  float: right;
  padding: 20px;
}
```



```
* {
  box-sizing: border-box;
}
```



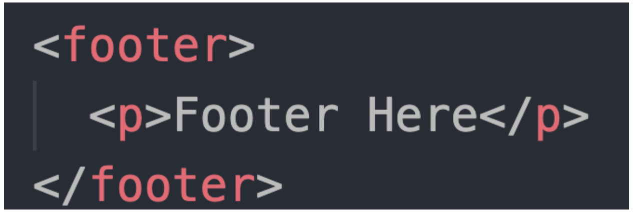
Header Here

Nav 1 Nav 2

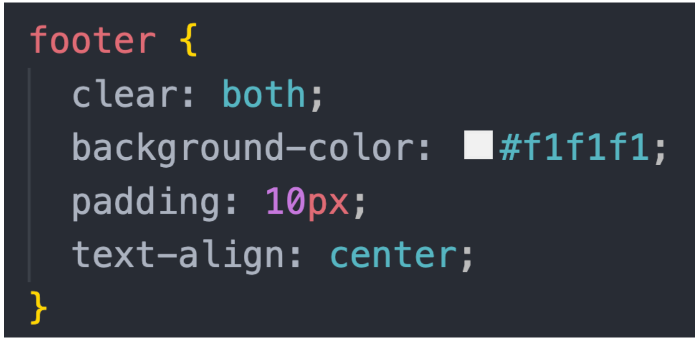
Main Content

Side Content

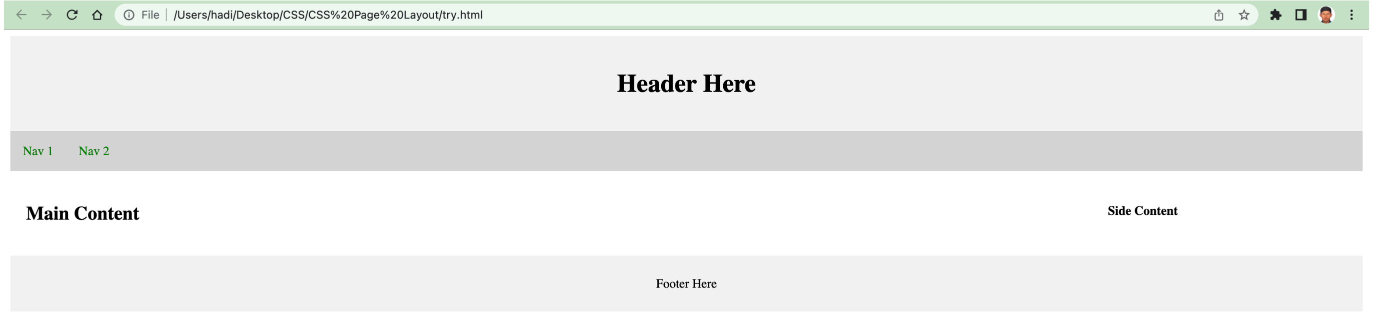
Footer: <footer>



```html
<footer>
  <p>Footer Here</p>
</footer>
```



```css
footer {
  clear: both;
  background-color: #f1f1f1;
  padding: 10px;
  text-align: center;
}
```



Header Here

Nav 1 Nav 2

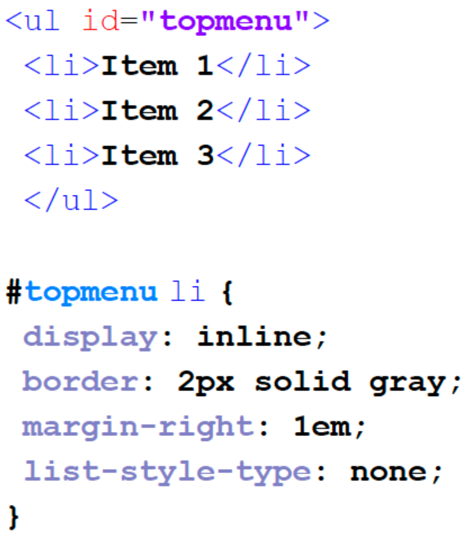
Main Content

Side Content

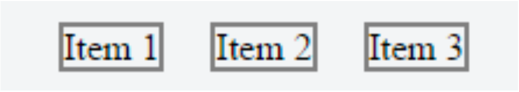
Footer Here

The display Property

- The standard values of display property such as block, inline or inline-block can change how elements behave in normal flow.



```
<ul id="topmenu">
 <li>Item 1</li>
 <li>Item 2</li>
 <li>Item 3</li>
 </ul>

#topmenu li {
 display: inline;
 border: 2px solid gray;
 margin-right: 1em;
 list-style-type: none;
}
```



```
Item 1  Item 2  Item 3
```

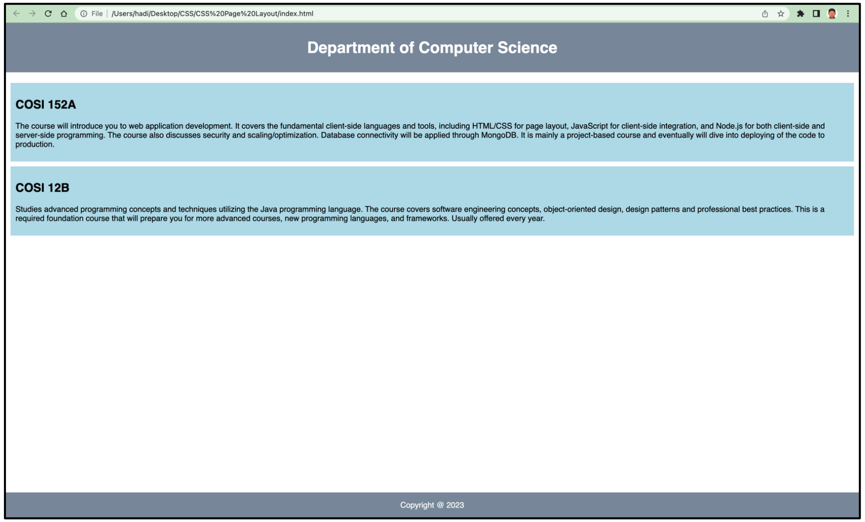
Layout Methods

- Layout methods that are enabled via specific display values:
  - Flexbox
  - CSS Grid
- These methods alter how child elements are laid out inside their parents.

Flexbox

- Flexbox is a one-dimensional layout method for arranging items in rows or columns.
- Items flex (expand) to fill additional space or shrink to fit into smaller spaces.

Flexbox Layout Example



Department of Computer Science

COSI 152A

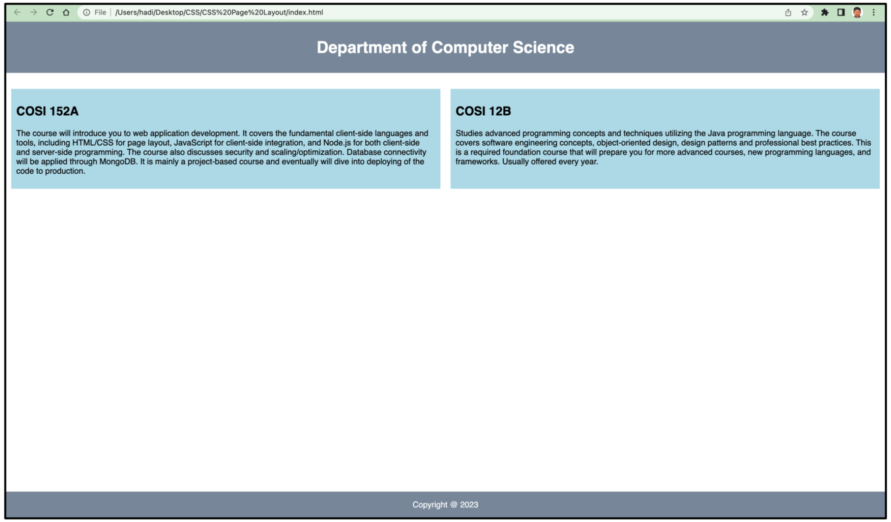
The course will introduce you to web application development. It covers the fundamental client-side languages and tools, including HTML/CSS for page layout, JavaScript for client-side integration, and Node.js for both client-side and server-side programming. The course also discusses security and scaling/optimization. Database connectivity will be applied through MongoDB. It is mainly a project-based course and eventually will dive into deploying of the code to production.

COSI 12B

Studies advanced programming concepts and techniques utilizing the Java programming language. The course covers software engineering concepts, object-oriented design, design patterns and professional best practices. This is a required foundation course that will prepare you for more advanced courses, new programming languages, and frameworks. Usually offered every year.

Copyright @ 2023

Without Flexbox



Department of Computer Science

COSI 152A

The course will introduce you to web application development. It covers the fundamental client-side languages and tools, including HTML/CSS for page layout, JavaScript for client-side integration, and Node.js for both client-side and server-side programming. The course also discusses security and scaling/optimization. Database connectivity will be applied through MongoDB. It is mainly a project-based course and eventually will dive into deploying of the code to production.

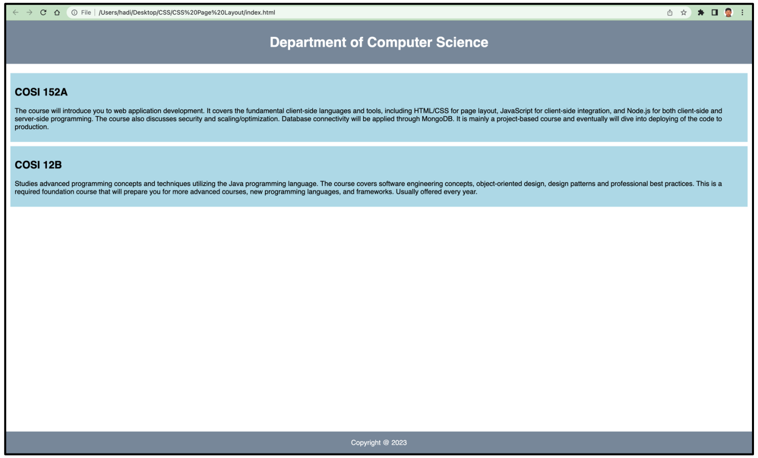
COSI 12B

Studies advanced programming concepts and techniques utilizing the Java programming language. The course covers software engineering concepts, object-oriented design, design patterns and professional best practices. This is a required foundation course that will prepare you for more advanced courses, new programming languages, and frameworks. Usually offered every year.

Copyright @ 2023

With Flexbox

Flexbox Layout Example



Department of Computer Science

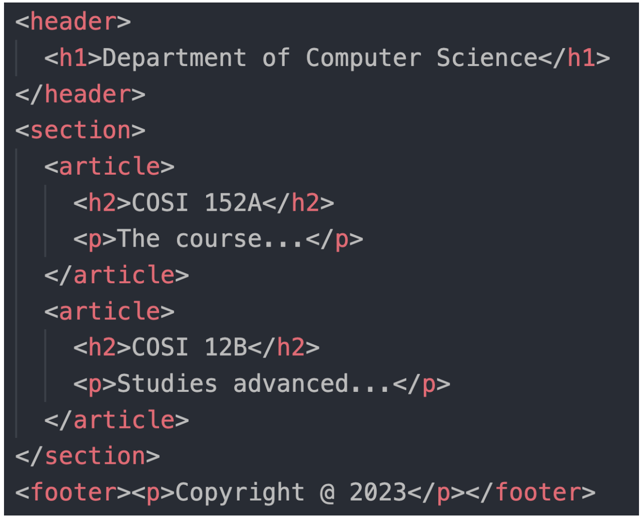
COSI 152A

The course will introduce you to web application development. It covers the fundamental client-side languages and tools, including HTML/CSS for page layout, JavaScript for client-side integration, and Node.js for both client-side and server-side programming. The course also discusses security and scaling/optimization. Database connectivity will be applied through MongoDB. It is mainly a project-based course and eventually will dive into deploying of the code to production.

COSI 12B

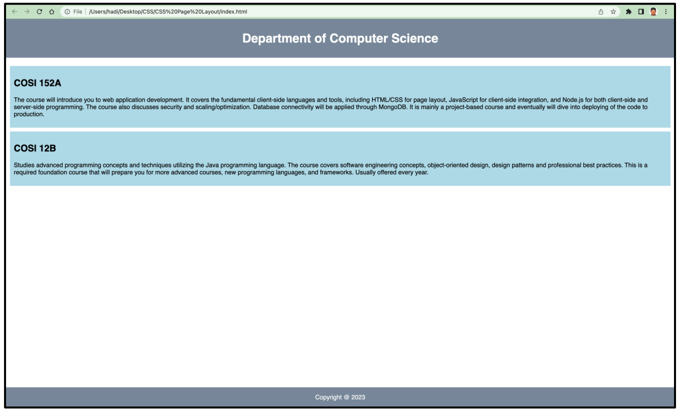
Studies advanced programming concepts and techniques utilizing the Java programming language. The course covers software engineering concepts, object-oriented design, design patterns and professional best practices. This is a required foundation course that will prepare you for more advanced courses, new programming languages, and frameworks. Usually offered every year.

Copyright @ 2023



```html
<header>
  <h1>Department of Computer Science</h1>
</header>
<section>
  <article>
    <h2>COSI 152A</h2>
    <p>The course...</p>
  </article>
  <article>
    <h2>COSI 12B</h2>
    <p>Studies advanced...</p>
  </article>
</section>
<footer><p>Copyright @ 2023</p></footer>
```

Flexbox Layout Example



Department of Computer Science

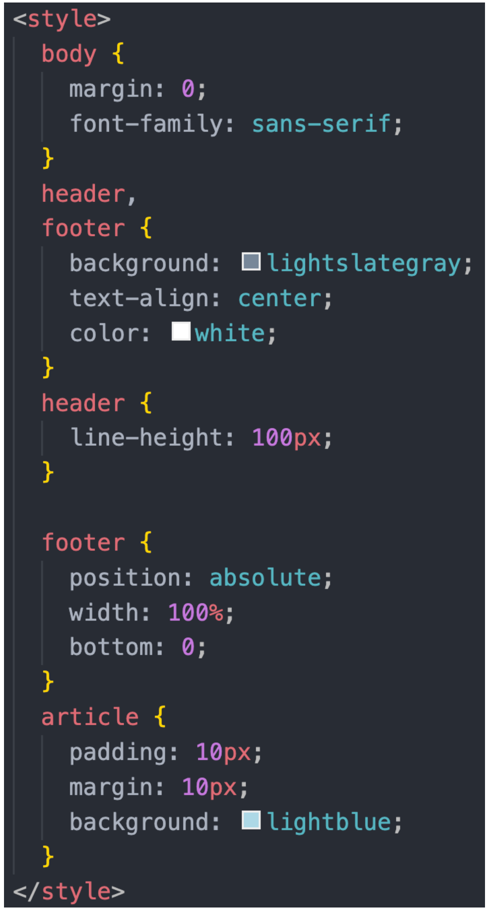
COSI 152A

The course will introduce you to web application development. It covers the fundamental client-side languages and tools, including HTML/CSS for page layout, JavaScript for client-side integration, and Node.js for both client-side and server-side programming. The course also discusses security and scaling/optimization. Database connectivity will be applied through MongoDB. It is mainly a project-based course and eventually will dive into deploying of the code to production.

COSI 12B

Studies advanced programming concepts and techniques utilizing the Java programming language. The course covers software engineering concepts, object-oriented design, design patterns and professional best practices. This is a required foundation course that will prepare you for more advanced courses, new programming languages, and frameworks. Usually offered every year.

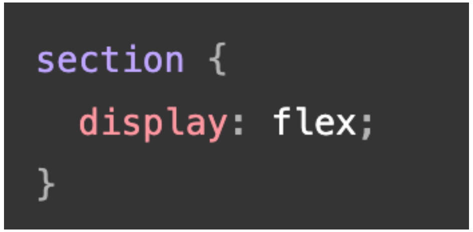
Copyright @ 2023



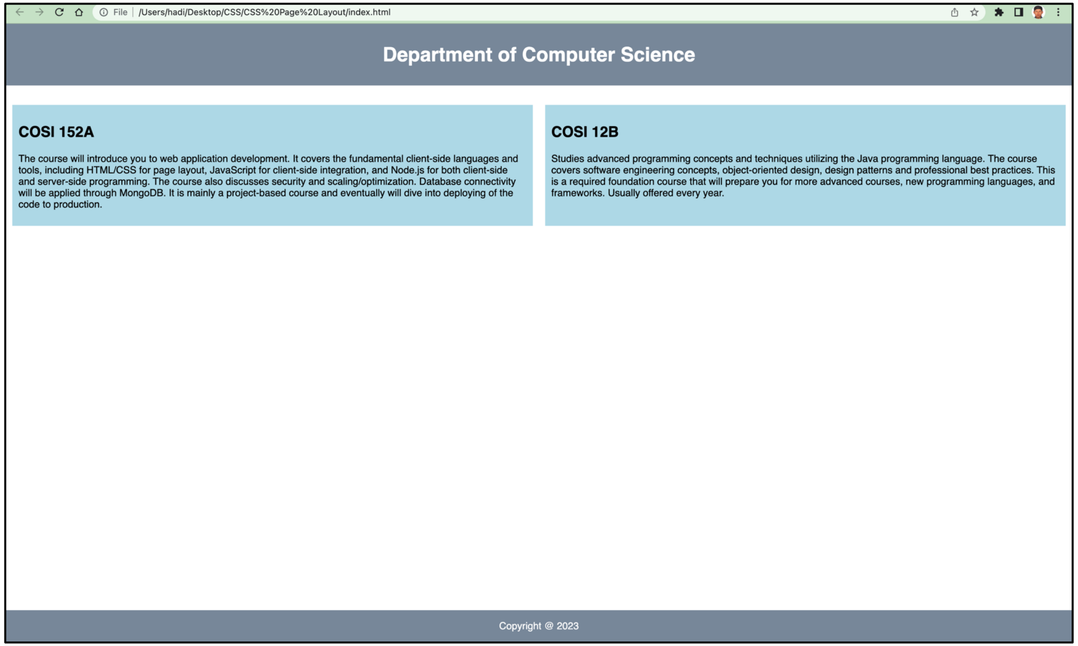
```
<style>
  body {
    margin: 0;
    font-family: sans-serif;
  }
  header,
  footer {
    background: lightslategray;
    text-align: center;
    color: white;
  }
  header {
    line-height: 100px;
  }

  footer {
    position: absolute;
    width: 100%;
    bottom: 0;
  }
  article {
    padding: 10px;
    margin: 10px;
    background: lightblue;
  }
</style>
```

Decide the Flex Container

- Set a special value of display on the parent element of the elements (flex items) you want to make it flex box .



```
section {
  display: flex;
}
```



Department of Computer Science

COSI 152A

The course will introduce you to web application development. It covers the fundamental client-side languages and tools, including HTML/CSS for page layout, JavaScript for client-side integration, and Node.js for both client-side and server-side programming. The course also discusses security and scaling/optimization. Database connectivity will be applied through MongoDB. It is mainly a project-based course and eventually will dive into deploying of the code to production.

COSI 12B

Studies advanced programming concepts and techniques utilizing the Java programming language. The course covers software engineering concepts, object-oriented design, design patterns and professional best practices. This is a required foundation course that will prepare you for more advanced courses, new programming languages, and frameworks. Usually offered every year.

Copyright @ 2023

Flex Direction

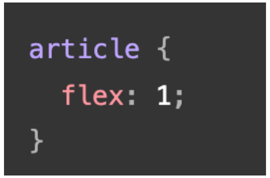
- flex-direction property specifies which direction the flexbox children are laid out in.
  - by default, this is set to row

```
section {
  display: flex;
  flex-direction: column;
}
```

Flexible Sizing of Flex Items

- The flex property is a shorthand property for:
  - flex-grow
  - flex-shrink
  - flex-basis

class	description
flex-grow	A number specifying how much the item will grow relative to the rest of the flexible items
flex-shrink	A number specifying how much the item will shrink relative to the rest of the flexible items
flex-basis	The length of the item. Legal values: "auto", "inherit", or a number followed by "%", "px", "em" or any other length unit



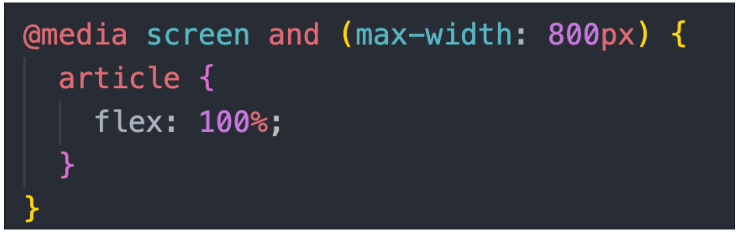
```
article {
  flex: 1;
}
```

Responsive Layout with Flexbox

- Media queries in CSS3 look at the capability of the device and can be used to check things such as width and height of the viewport.
- To have a responsive page layout for different devices, screen media type is used in media query to check the width of the target screen.

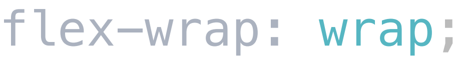
Responsive Layout with Flexbox

- Responsive layout makes a one column layout (100%) instead of a two-column layout for screens larger than 800px.



```
@media screen and (max-width: 800px) {
  article {
    flex: 100%;
  }
}
```

- Wrapping should be allowed in flexbox container



```
flex-wrap: wrap;
```

Thank You!