Hands-on Lab: Responsive Web Design using Tailwind & Bootstrap Frameworks
动手实验室：使用 Tailwind & Bootstrap 框架进行响应式网页设计

Skills Network

Estimated Time: 30 minutes
预计时间：30 分钟

Introduction: 引言：

Responsive web design ensures that websites adapt seamlessly to various screen sizes, offering optimal usability across devices such as mobile phones, tablets, and desktops. In this lab, you will implement responsiveness using **Tailwind CSS** and **Bootstrap** frameworks.
响应式网页设计确保网站能够无缝适应各种屏幕尺寸，在手机、平板电脑和桌面等设备上提供最佳可用性。在本实验中，您将使用 Tailwind CSS 和 Bootstrap 框架实现响应式设计。

Objective: 目标：

By the end of this lab, learners will be able to:
到本实验结束时，学习者将能够：

1. Create a responsive webpage that adjusts for mobile, tablet, and desktop screens
   创建一个响应式网页，可适应手机、平板和桌面屏幕
2. Use **Tailwind CSS's responsive modifiers** to enhance responsiveness
   使用 Tailwind CSS 的响应式修饰符来增强响应性
3. Use **Bootstrap's grid system and navbar components** to add responsiveness and functionality
   使用 Bootstrap 的网格系统和导航栏组件以增加响应性和功能性
4. Understand the differences between Tailwind CSS and Bootstrap
   理解 Tailwind CSS 和 Bootstrap 之间的区别

Exercise 1: Understanding starter code
练习 1：理解起始代码

- You will be provided with a basic webpage layout consisting of the following sections:
  您将获得一个包含以下部分的网页布局基础：
  - Header 标题
  - Navbar 导航栏
  - Main Section 主要部分
  - Aside 除
  - Article 文章
  - Footer 页脚
- Each section will have distinct background colors to visualize the layout structure.
  每个部分都将有独特的背景颜色来可视化布局结构。
- Create a new HTML file and insert the following code into it:
  创建一个新的 HTML 文件，并将以下代码插入到其中：

Starter code 启动代码

```
<!DOCTYPE html>
<html lang="en">
    <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Basic Layout</title>
    <style>
        body {
        margin: 0;
        font-family: Arial, sans-serif;
        }
        header, footer {
        background-color: red;
        text-align: center;
        padding: 10px;
        }
        nav {
        background-color: blue;
```

```
            padding: 10px;
            text-align: center;
            }
            section {
            background-color: white;
            padding: 10px;
            }
            aside {
            background-color: yellow;
            padding: 10px;
            }
            article {
            background-color: lightgreen;
            padding: 10px;
            }
            .container {
            display: flex;
            flex-direction: column;
            }
            .row {
            display: flex;
            }
        </style>
        </head>
        <body>
        <header>Header</header>
        <nav>Navigation</nav>
        <div class="container">
            <div class="row">
            <aside>Aside</aside>
            <section>Main Section</section>
            <article>Article</article>
            </div>
        </div>
        <footer>Footer</footer>
        </body>
    </html>
```

Description: 描述：
This code creates a simple webpage layout with distinct background colors for visualizing the structure. The code is not fully responsive as it lacks media queries for adapting the layout to different screen sizes. The `.container` uses flexbox to stack elements vertically, but there's no flexibility in the layout to adjust for smaller screens. Additionally, the fixed padding and absence of width adjustments for the elements can cause overflow on smaller devices.
此代码创建了一个具有不同背景颜色的简单网页布局，以可视化结构。代码未完全响应，因为它缺少媒体查询以适应不同屏幕尺寸。 `.container` 使用 flexbox 垂直堆叠元素，但布局在较小屏幕上没有灵活性进行调整。此外，固定的填充和元素宽度调整的缺失可能导致在较小设备上溢出。

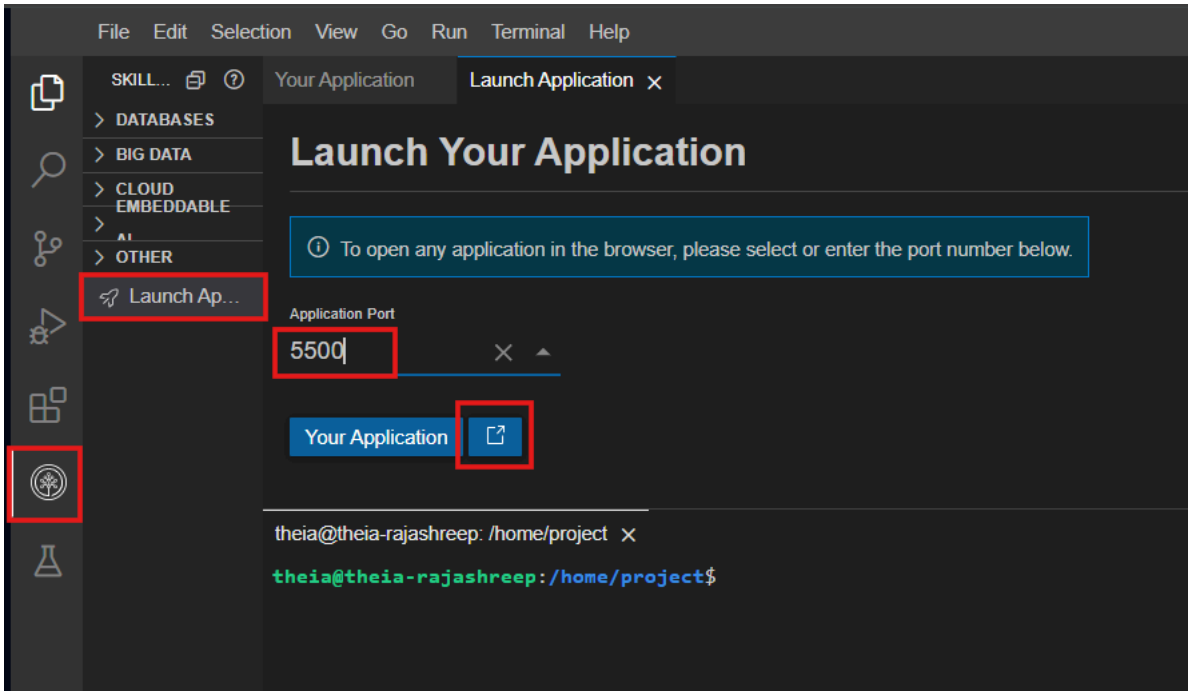
Preview and test your webpage's responsiveness
预览和测试您的网页响应性

To preview your file, you can use the built-in Live Server extension by following the instructions below.
预览您的文件，您可以通过以下说明使用内置的 Live Server 扩展。

Part 1: Launch the webpage using Live Server
第一部分：使用 Live Server 启动网页

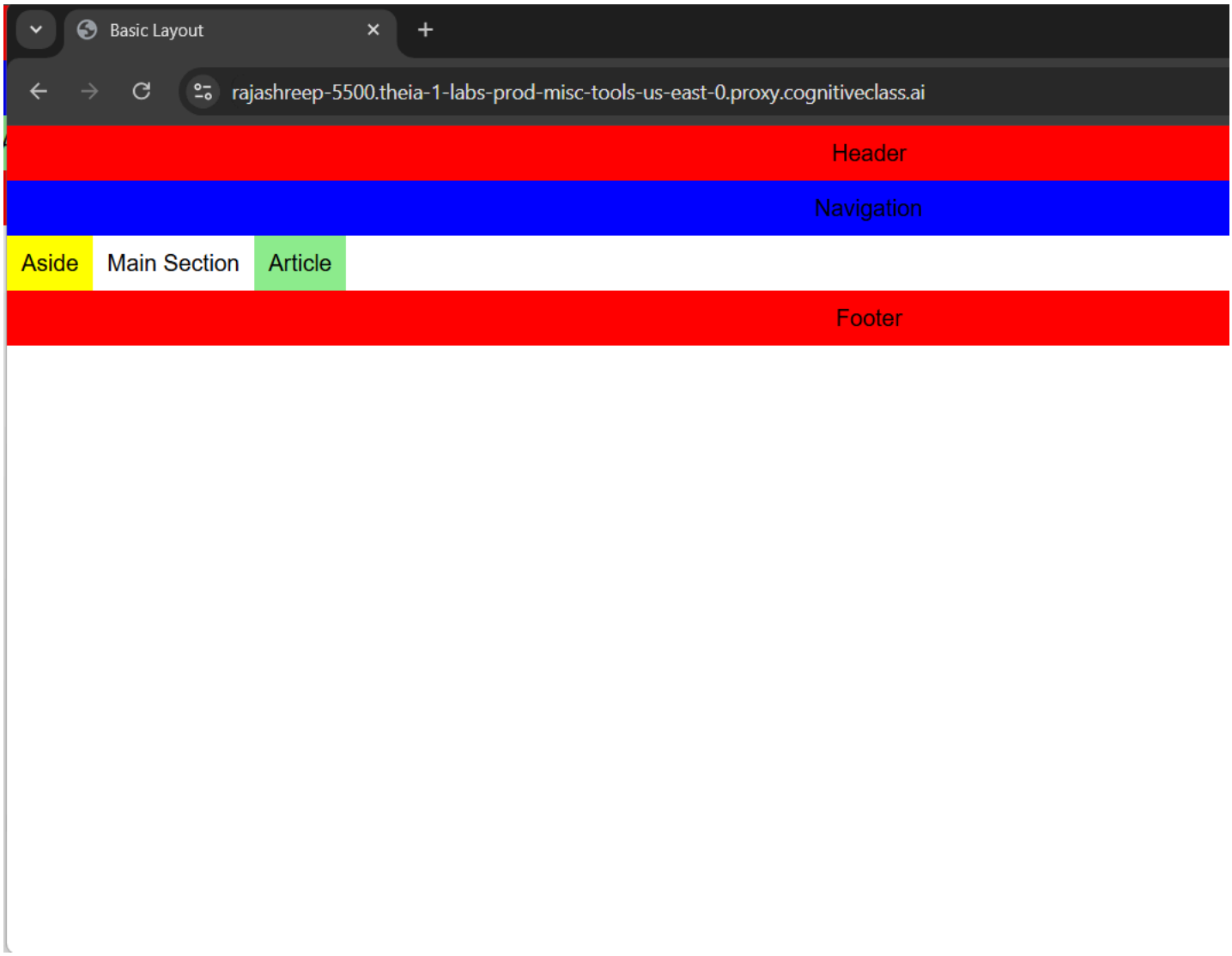
1. Open your file explorer and locate your HTML file
   打开文件资源管理器，找到您的 HTML 文件
2. Right-click the file and select **Open with Live Server**
   右键单击文件并选择使用 Live Server 打开
3. A notification will appear indicating that the server has started on port 5500
   通知将显示服务器已在 5500 端口启动
4. Click the button below the notification or use the **Skills Network Toolbox**, then navigate to **Other > Launch Application**, enter port `5500`, and click the launch button highlighted in the screenshot below. This will open the webpage in a new browser tab
   点击通知下方的按钮或使用技能网络工具箱，然后导航到其他 > 启动应用程序，输入端口 `5500` ，并点击下图中突出显示的启动按钮。这将在新浏览器标签页中打开网页。

Launch App 启动应用



Part 2: Test responsiveness using Developer Tools
第二部分：使用开发者工具测试响应性

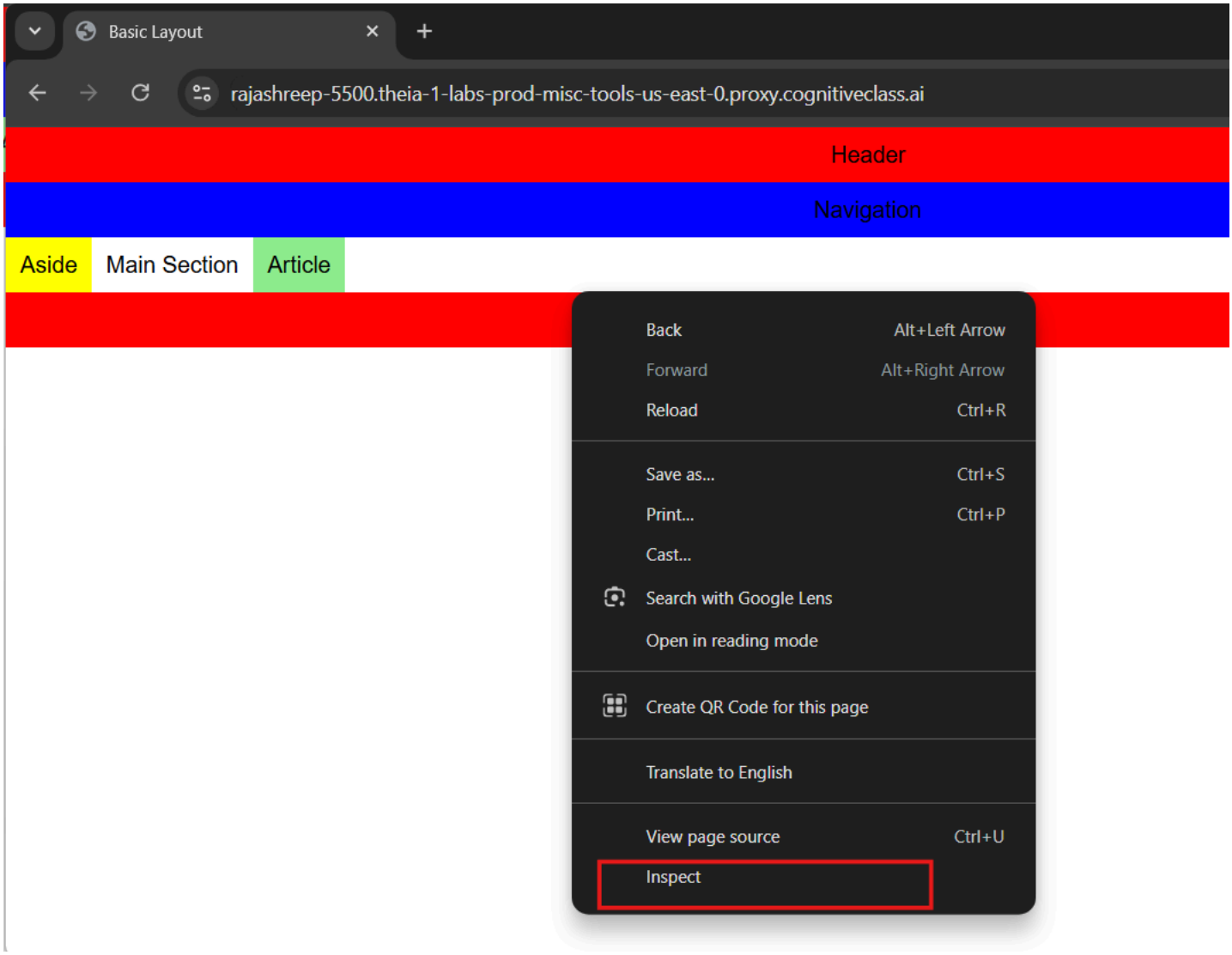
1. Make sure your webpage is opened in a new tab of the browser
   确保网页在浏览器的新标签页中打开



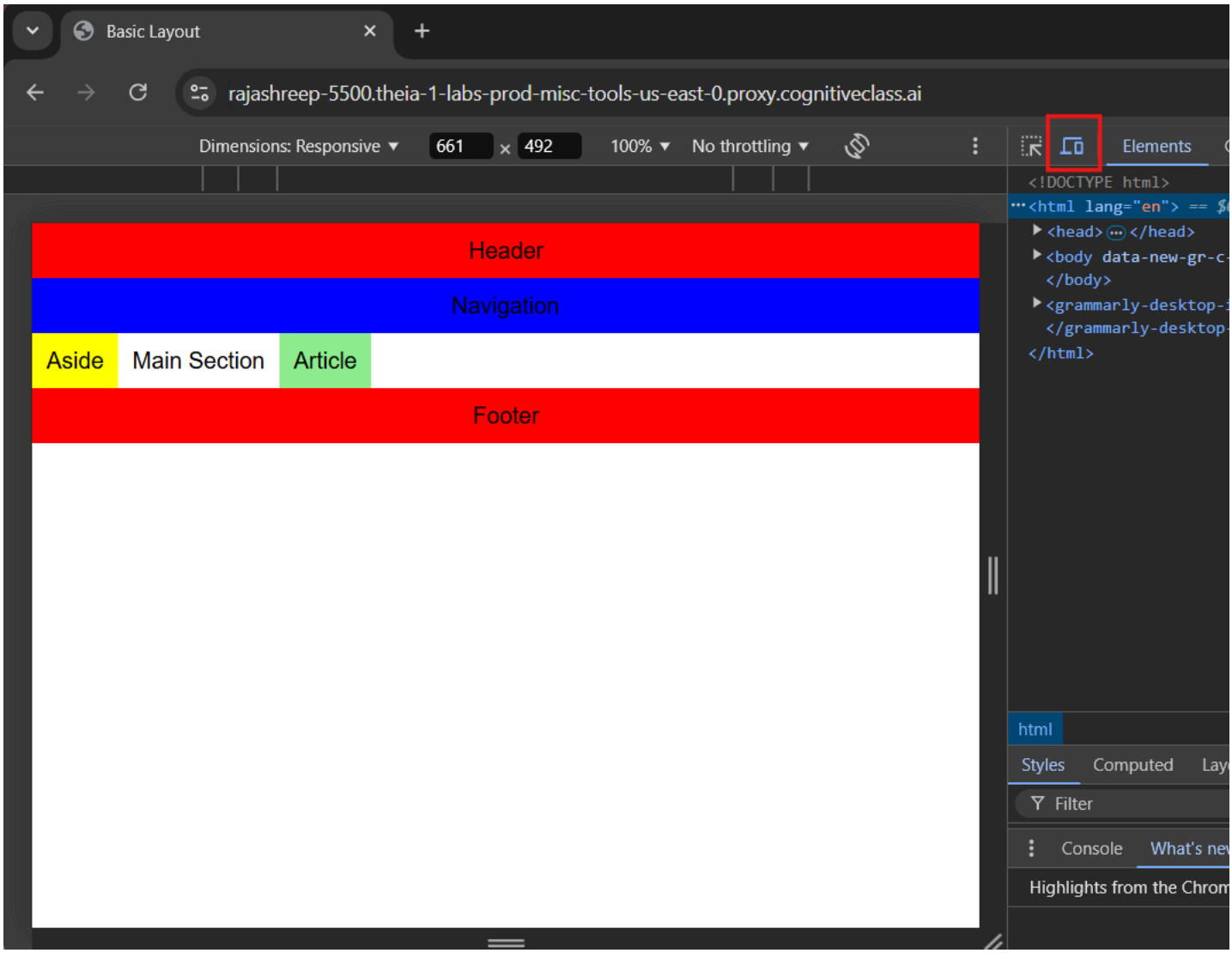
2. Right-click the webpage and select **Inspect** or **Inspect Element** (varies by browser)
   右键点击网页并选择“检查”或“检查元素”（浏览器不同而异）

Alternatively, use the shortcut:
或者，使用快捷键：

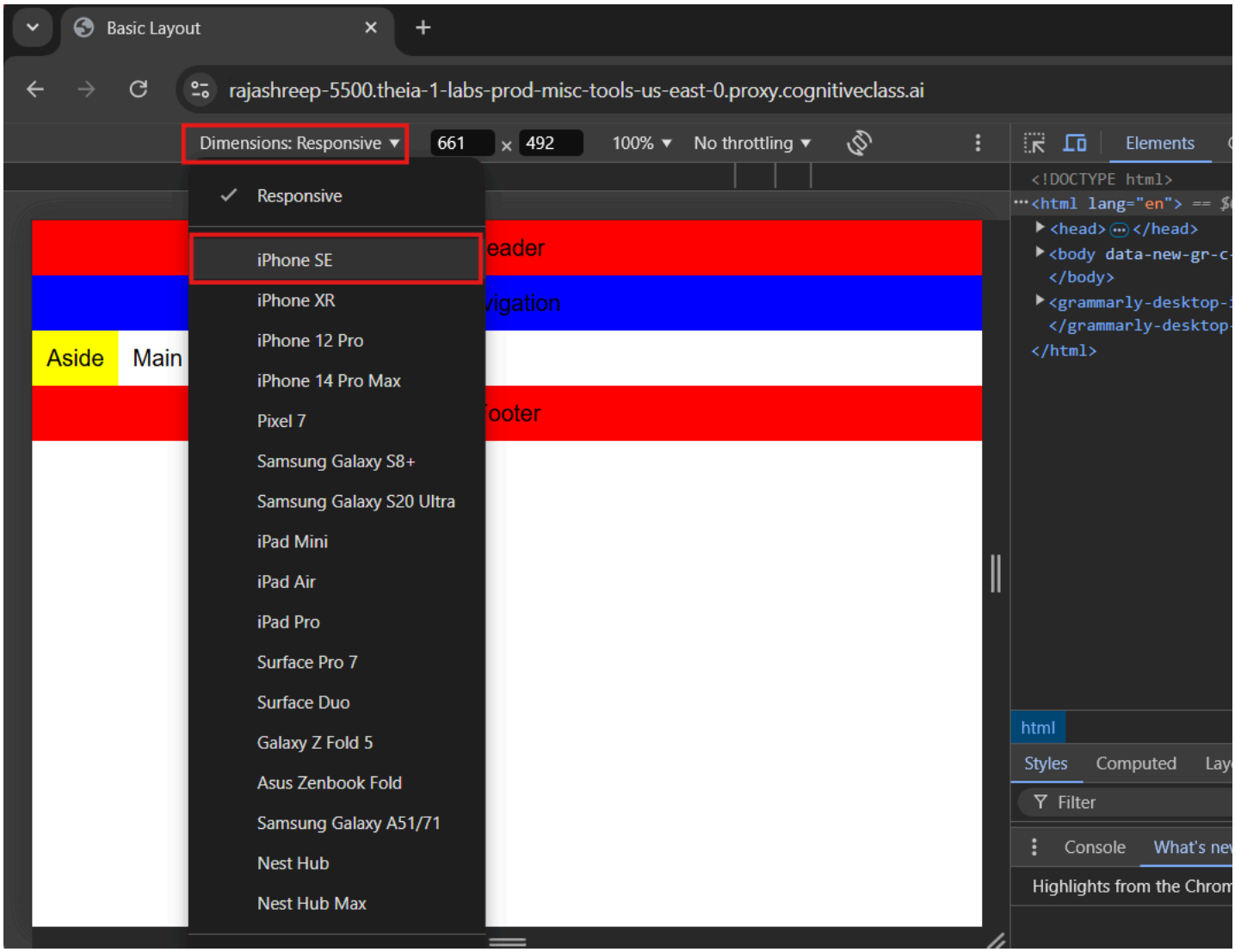
- `Ctrl + Shift + I` (Windows/Linux) `Ctrl + Shift + I` (Windows/Linux)
- `Cmd + Option + I` (Mac) `Cmd + Option + I` (Mac)



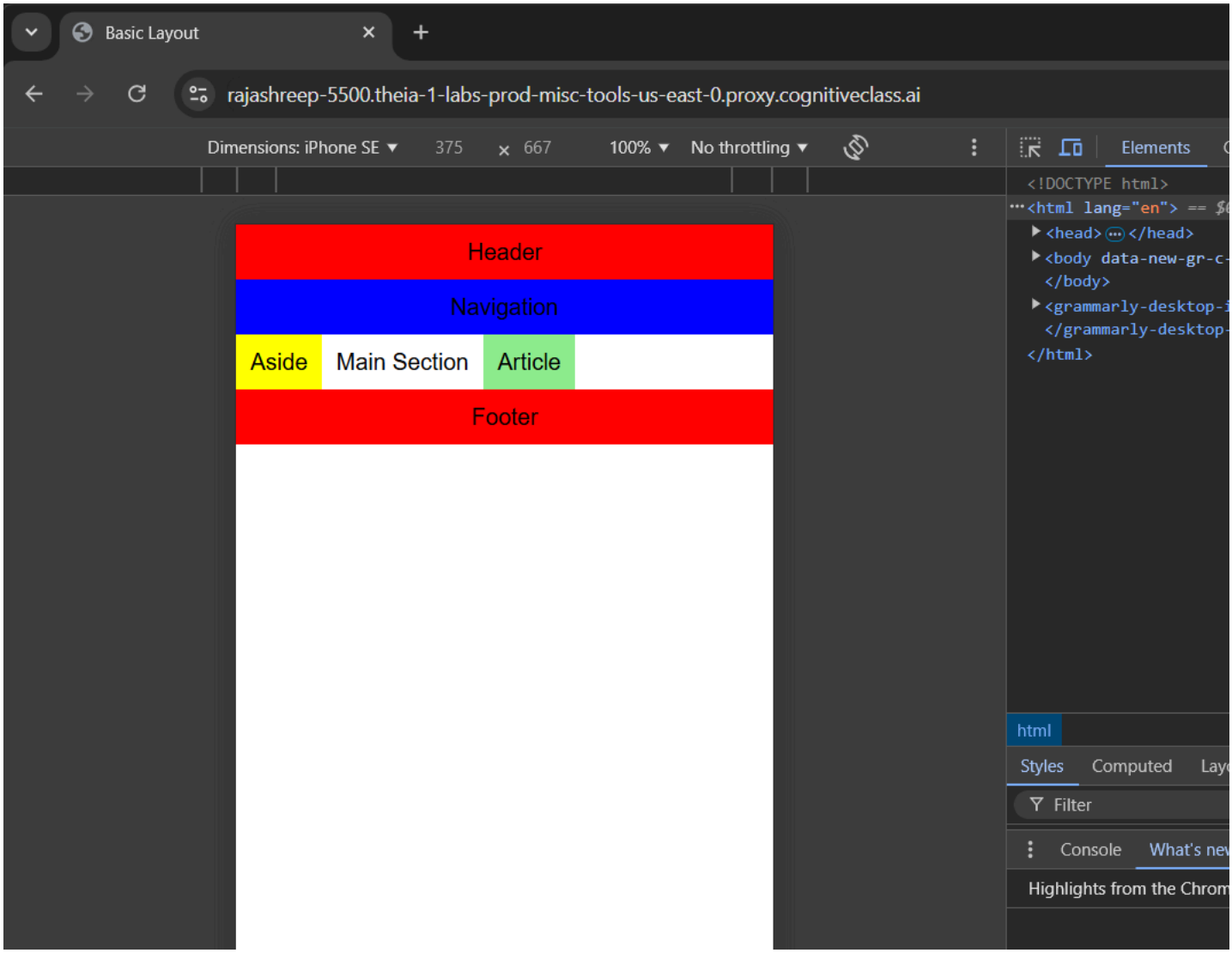
3. In the Developer Tools panel, click the **Device Toggle Toolbar** icon (phone/tablet icon) to switch to mobile view
   在开发者工具面板中，点击设备切换工具栏图标（手机/平板图标）以切换到移动视图
   - Shortcut: 快捷方式：
     - `Ctrl + Shift + M` (Windows/Linux) `Ctrl + Shift + M` (Windows/Linux)
     - `Cmd + Shift + M` (Mac) `Cmd + Shift + M` (Mac)



4. Use the dropdown in the toolbar to choose preset devices (e.g., iPhone, iPad) or set custom dimensions to simulate different screen sizes
使用工具栏中的下拉菜单选择预设设备（例如，iPhone、iPad）或设置自定义尺寸以模拟不同屏幕大小



5. Observe that the layout has not readjusted according to the device size because the page is not yet responsive
请注意，布局尚未根据设备大小进行调整，因为页面尚未响应式设计



6. To return to desktop view click the **Device Toggle Toolbar** icon again or use the shortcut (`Ctrl + Shift + M` or `Cmd + Shift + M`)
返回桌面视图，请再次点击设备切换工具栏图标或使用快捷键（`Ctrl + Shift + M` 或 `Cmd + Shift + M`）

Exercise 2: Adding responsiveness with Tailwind CSS
练习 2：使用 Tailwind CSS 添加响应性

Introduction to Tailwind CSS
引言：Tailwind CSS 简介

Tailwind CSS is a utility-first CSS framework that enables developers to design directly in HTML. It provides responsive design utilities using **breakpoints** (`sm`, `md`, `lg`, and `xl`). These breakpoints correspond to different screen sizes, allowing the layout to adapt to various devices, from small mobile screens to large desktop monitors.
Tailwind CSS 是一个以实用工具为首要的 CSS 框架，它允许开发者直接在 HTML 中进行设计。它提供了使用断点（`sm`、`md`、`lg` 和 `xl`）的响应式设计工具。这些断点对应不同的屏幕尺寸，使布局能够适应从小型移动屏幕到大型桌面显示器等各种设备。

Steps to implement Tailwind CSS
实施 Tailwind CSS 的步骤

1. Include the Tailwind CSS CDN in the `<head>` of your HTML:
在您的 HTML 中包含 Tailwind CSS CDN：`<head>`

```
<link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
```

2. Update the starter code to use Tailwind classes for responsiveness
更新启动代码以使用 Tailwind 类实现响应式

Updated code 更新代码

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Responsive Layout - Tailwind</title>
```

```
  <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">
</head>
<body class="bg-gray-100">
  <header class="bg-red-500 text-white text-center p-4">Header</header>
  <nav class="bg-red-500 text-white text-center p-4">Navigation</nav>
  <div class="flex flex-col md:flex-row">
    <aside class="bg-yellow-300 p-4 w-full md:w-1/4">Aside</aside>
    <section class="bg-white p-4 flex-grow">Main Section</section>
    <article class="bg-green-300 p-4 w-full md:w-1/4">Article</article>
  </div>
  <footer class="bg-blue-500 text-white text-center p-4">Footer</footer>
</body>
</html>
```

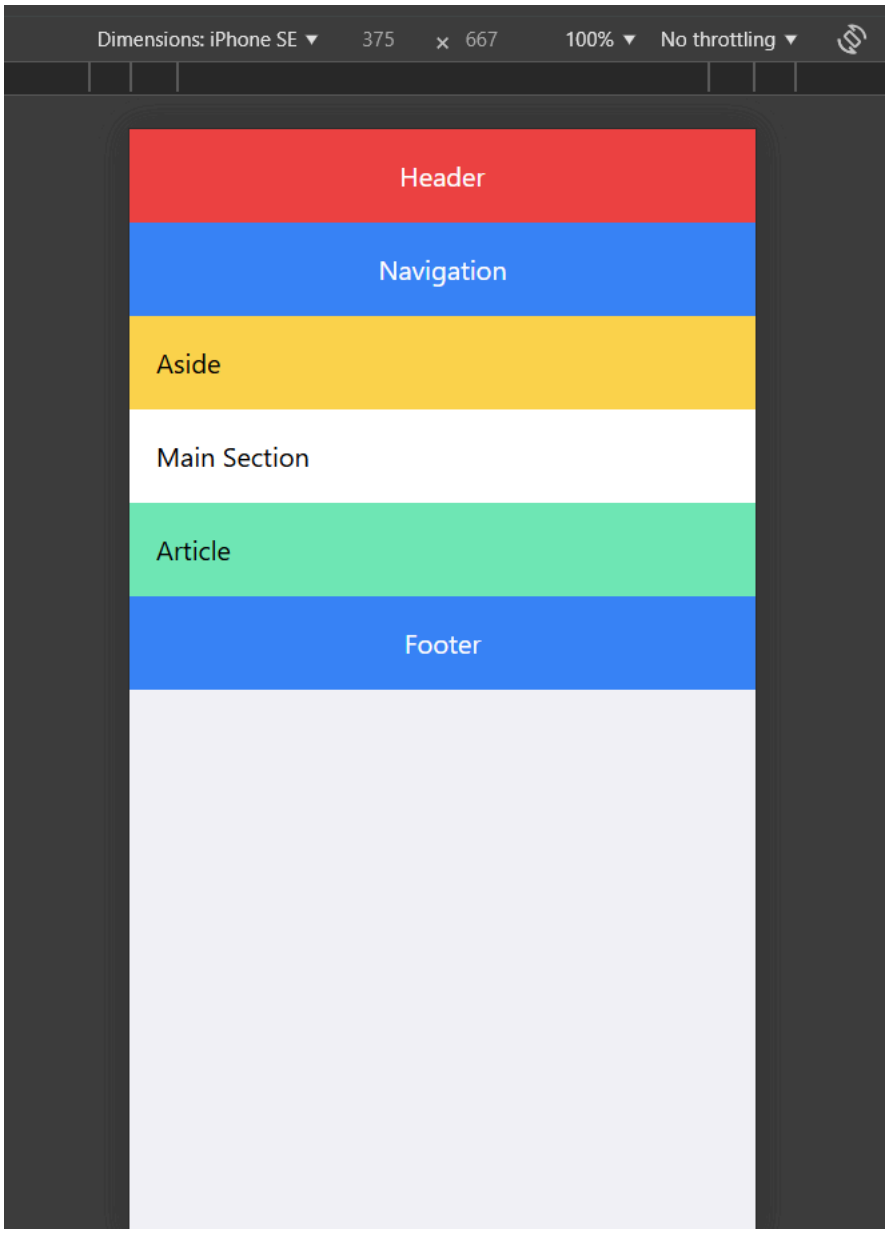
Key Tailwind classes used
关键 Tailwind 类

1. **`flex` and `flex-col` `flex` 和 `flex-col`**
   - **`flex`**: Activates flexbox layout, allowing easy alignment and distribution of child elements (the elements inside it) along a row or column
     激活 flexbox 布局，允许沿行或列轻松对齐和分配子元素（其中的元素）
   - **`flex-col`**: By default, arranges child elements vertically in a column layout, stacking them one below the other
     默认情况下，以列布局排列子元素，将它们堆叠在一列中，一个接一个
2. **`md:flex-row`**
   - **`md:`**: A responsive prefix that targets medium screens (768px and larger)
     `md:`：针对中等屏幕（768 像素及以上）的响应式前缀
   - **`flex-row`**: Changes the flex container's layout from a column to a row on medium and larger screens, arranging child elements horizontally side by side
     将 flex 容器布局从列改为中等及以上屏幕的行，子元素水平排列
3. **`w-full` and `md:w-1/4` `w-full` 和 `md:w-1/4`**
   - **`w-full`**: Sets the element's width to 100% of its parent container, making it full width on small screens
     设置元素的宽度为其父容器宽度的 100%，使其在小屏幕上全宽
   - **`md:w-1/4`**: On medium and larger screens, sets the element's width to 25% of the container, making it more compact and allowing for a horizontal layout
     在中等和较大屏幕上，将元素的宽度设置为容器宽度的 25%，使其更加紧凑，并允许水平布局
4. **`p-4`**
   - **`p-4`**: Adds padding of 1rem (16px) to all sides of the element, providing consistent spacing around the content and improving visual appeal
     为元素的所有边添加 1rem（16px）的填充，提供一致的内容间距并提升视觉效果
5. **`min-h-screen`**
   - Ensures the element's height is at least as tall as the viewport height, so the layout fills the screen even if the content inside is minimal
     确保元素的高度至少与视口高度相同，因此即使内容内部最少，布局也能填满屏幕
6. **`bg-*`**
   - **`bg-*`** classes are used to set background colors, such as `bg-gray-100`, which help distinguish different sections and improve the overall visual clarity of the layout
     类用于设置背景颜色，例如 `bg-gray-100` ，有助于区分不同部分并提高布局的整体视觉清晰度

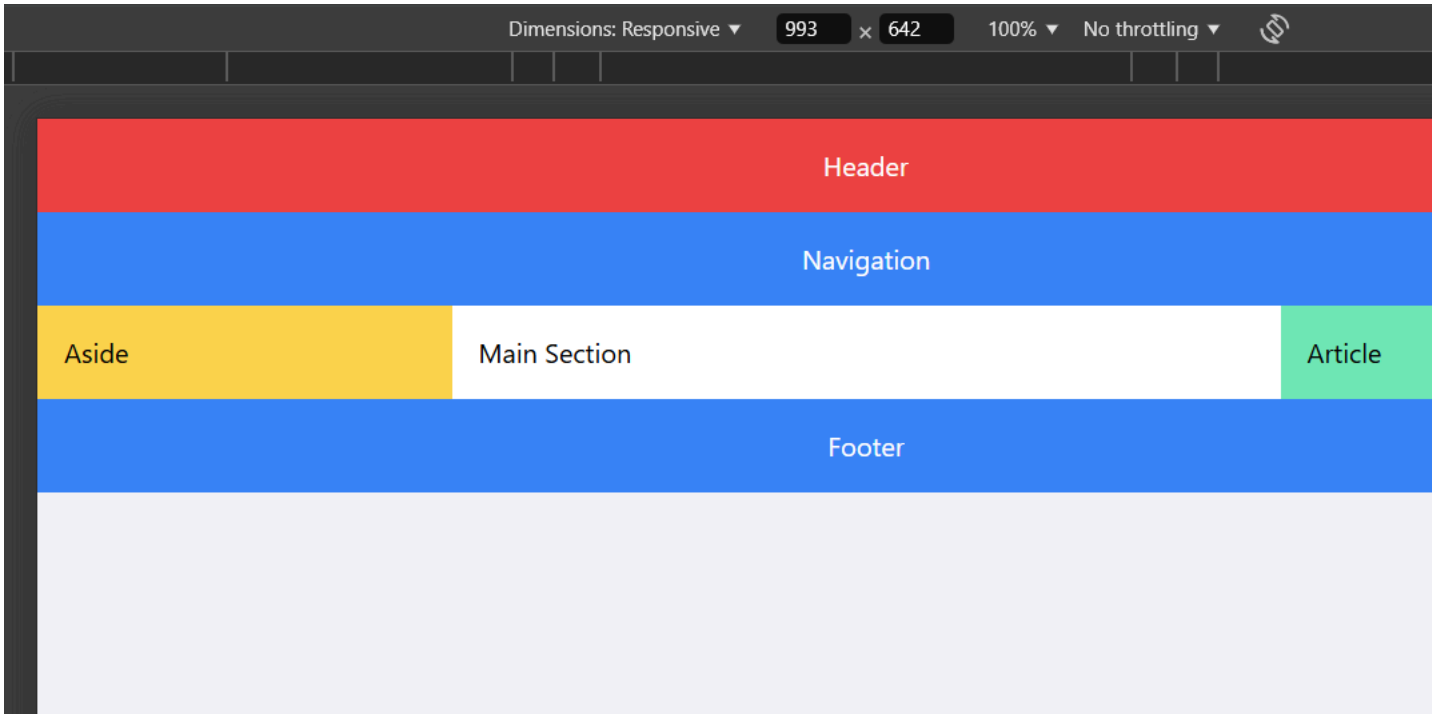
Test responsiveness: 测试响应性：

Resize the browser window manually or select a preset device (e.g., iPhone, iPad) to simulate different screen sizes. Refer to the **Preview and test your webpage's responsiveness** section.
手动调整浏览器窗口大小或选择预设设备（例如，iPhone、iPad）以模拟不同的屏幕尺寸。请参阅预览和测试网页响应性部分。

- On **mobile**, the sections stack vertically. The sections (aside, main, and article) will stack one below the other.
  在移动端，部分垂直堆叠。部分（除、主要和文章）将一个接一个地堆叠。



- On **tablet/desktop**, the layout will adjust to a more compact and horizontal design. The sections will align side by side (aside, main, and article), with the main section taking the central focus, and the navigation bar will be expanded with links shown directly.
  在平板/桌面设备上，布局将调整为更紧凑的水平设计。各部分将并排排列（侧边栏、主要内容区和文章），主要内容区占据中央焦点，导航栏将扩展并直接显示链接。



Exercise 3: Adding responsiveness with Bootstrap
练习 3：使用 Bootstrap 添加响应性

Introduction to Bootstrap
Bootstrap 简介

Bootstrap is a popular CSS framework with pre-designed components and a powerful grid system. It simplifies responsiveness with classes such as `col-*` and built-in navigation components.
Bootstrap 是一个流行的 CSS 框架，包含预设计的组件和强大的网格系统。它通过类如 `col-*` 和内置导航组件简化了响应式设计。

Steps to implement Bootstrap CSS
Bootstrap CSS 实现步骤

1. Include the Bootstrap CSS and JavaScript CDN in the `<head>` of your HTML:
   在您的 HTML 文件中 `<head>` 处包含 Bootstrap CSS 和 JavaScript CDN：

   ```
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>
   ```

2. Create a new HTML file and insert the following code into it to use Bootstrap's grid system and include a hamburger menu
   创建一个新的 HTML 文件，并将以下代码插入其中以使用 Bootstrap 的网格系统并包含汉堡菜单

Updated code 更新代码

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Responsive Layout - Bootstrap</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
  <header class="bg-danger text-white text-center p-3">Header</header>
  <nav class="bg-primary navbar navbar-expand-lg navbar-light bg-light">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">Brand</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
          <li class="nav-item"><a class="nav-link" href="#">Home</a></li>
          <li class="nav-item"><a class="nav-link" href="#">About</a></li>
          <li class="nav-item"><a class="nav-link" href="#">Contact</a></li>
        </ul>
      </div>
    </div>
  </nav>
  <div class="container my-3">
    <div class="row">
      <aside class="col-12 col-md-3 bg-warning p-3">Aside</aside>
      <section class="col-12 col-md-6 bg-white p-3">Main Section</section>
      <article class="col-12 col-md-3 bg-success p-3">Article</article>
    </div>
    <div class="row">
      <footer class="col-12 bg-primary text-white text-center p-3">Footer</footer>
    </div>
  </div>
</body>
</html>
```

Key Bootstrap classes used
键 Bootstrap 类

1. **`navbar` and `navbar-expand-lg` `navbar` 和 `navbar-expand-lg`**
   - **`navbar`**: Creates a navigation bar with built-in features such as text alignment, background color, and dropdown menu support
     创建一个具有内置功能（如文本对齐、背景颜色和下拉菜单支持）的导航栏
   - **`navbar-expand-lg`**: Makes the navigation bar expand horizontally on large screens. On smaller screens, it collapses into a hamburger menu (three horizontal lines) that reveals the menu when clicked.
     使导航栏在大屏幕上水平展开。在小屏幕上，它折叠成汉堡菜单（三条水平线），点击时显示菜单。
   - **Together**: The combination of `navbar` and `navbar-expand-lg` creates a responsive navigation bar that adapts to different screen sizes, expanding on large screens and collapsing on smaller ones.
     共同： `navbar` 和 `navbar-expand-lg` 的组合创建了一个响应式导航栏，该导航栏能够适应不同的屏幕尺寸，在大屏幕上展开，在小屏幕上折叠。

2. **`col-*` classes `col-*` 类别**
   - **`col-12`**: In Bootstrap's grid system, `col-12` makes the element take up 100% width on extra small and small screens, stacking elements vertically on mobile devices.
     在 Bootstrap 的栅格系统中， `col-12` 使元素在超小屏幕和小屏幕上占据 100%的宽度，在移动设备上垂直堆叠元素。

- **`col-md-3`**: For medium screens (768px and larger), `col-md-3` sets the element's width to 25%, allowing it to sit next to other elements in a horizontal layout.
  对于中等屏幕（768 像素及以上）， `col-md-3` 将元素的宽度设置为 25%，使其能够与其他元素在水平布局中并排放置。
- **`col-md-6`**: On medium and larger screens, `col-md-6` makes the element take up 50% width, ensuring a balanced layout.
  在中等和较大屏幕上， `col-md-6` 使元素占据 50%的宽度，确保布局平衡。
- **Together**: Combining `col-12` for small screens and `col-md-*` for medium and larger screens ensures a responsive layout that adjusts dynamically.
  同时：结合 `col-12` 用于小屏幕和 `col-md-*` 用于中等及大屏幕，确保响应式布局能够动态调整。

3. **`p-3`**

   - **`p-3`**: Adds 1rem (16px) of padding to all sides of an element, ensuring there's space inside, preventing content from touching the edges, and improving the layout's visual appeal.
     为元素的所有边添加 1rem（16 像素）的内边距，确保内部有空间，防止内容触及边缘，并提高布局的视觉吸引力。

4. **`bg-danger`, `bg-primary`, `bg-warning`, and `bg-success`**
   `bg-danger` , `bg-primary` , `bg-warning` , 以及 `bg-success`

   - **`bg-*`**: These classes apply specific background colors to elements, helping distinguish sections and enhancing visual clarity.
     这些类为元素应用特定的背景颜色，有助于区分部分并增强视觉清晰度。

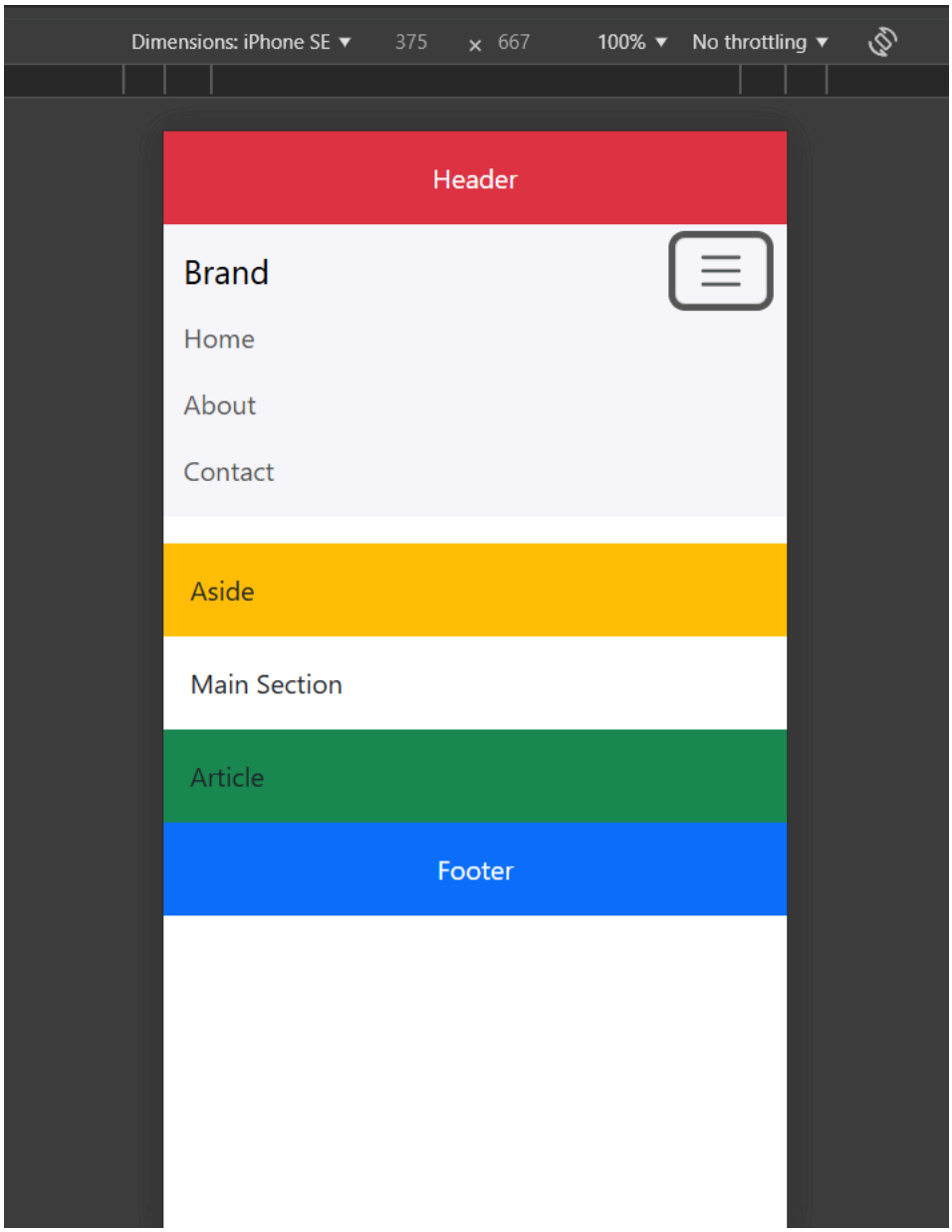
5. **`navbar-toggler` and `navbar-toggler-icon` `navbar-toggler` 和 `navbar-toggler-icon`**

   - **`navbar-toggler`**: A button used to toggle the visibility of menu items in the navigation bar on smaller screens, showing or hiding the links.
     `navbar-toggler` ： 用于在小屏幕上切换导航栏菜单项可见性的按钮，显示或隐藏链接。
   - **`navbar-toggler-icon`**: Displays the "hamburger menu" icon (three horizontal lines) inside the button, indicating the collapsible menu.
     显示按钮内的“汉堡菜单”图标（三条水平线），表示可折叠菜单。

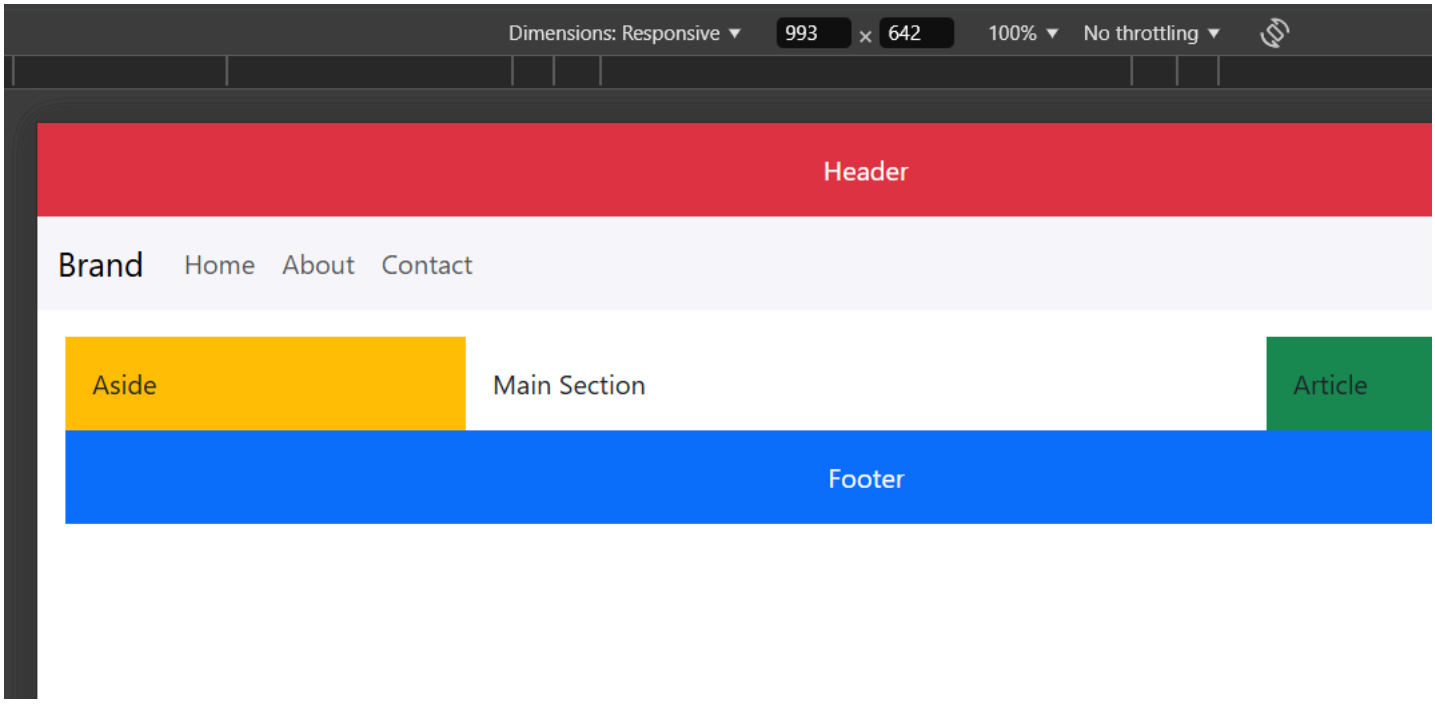
Test responsiveness: 测试响应性：

Resize the browser window manually or select a preset device (e.g., iPhone, iPad) to simulate different screen sizes. Refer to the **Preview and test your webpage's responsiveness** section.
手动调整浏览器窗口大小或选择预设设备（例如，iPhone、iPad）以模拟不同的屏幕尺寸。请参阅预览和测试网页响应性部分。

- On **mobile**, the layout should stack vertically. The navigation bar will collapse into a hamburger menu, and the sections (aside, main, and article) will stack one below the other.
  在移动端，布局应垂直堆叠。导航栏将折叠成汉堡菜单，部分（除、主、文章）将一个接一个地堆叠。



- On **tablet/desktop**, the layout will adjust to a more compact and horizontal design. The sections will align side by side (aside, main, and article), with the main section taking the central focus, and the navigation bar will be expanded with links shown directly.
  在平板/桌面设备上，布局将调整为更紧凑的水平设计。各部分将并排排列（侧边栏、主要内容区和文章），主要内容区占据中央焦点，导航栏将扩展并直接显示链接。



Exercise 4: Comparison between Tailwind CSS and Bootstrap
练习 4：Tailwind CSS 与 Bootstrap 的比较

Feature 特性	Tailwind CSS	Bootstrap 引导程序
Approach 方法	Utility-first, allows for full customization 以实用主义为核心，允许完全自定义	Predefined components and styles 预定义组件和样式
Customization 定制	High level of customization and flexibility 高定制化和灵活性	Limited flexibility, relies on pre-designed components 灵活性有限，依赖于预设计的组件
Learning curve 学习曲线	Steeper, requires understanding of CSS 更陡峭，需要理解 CSS	Easier, built-in classes and components 更简单，内置类和组件
File size 文件大小	Smaller, as unused styles can be purged 更小，因为未使用的样式可以被清除	Larger, as it includes all components by default 更大，因为它默认包含所有组件
Design control 设计控制	Full control over design and layout 完全控制设计和布局	Limited control without overriding default styles 有限控制，不覆盖默认样式
Grid system 网格系统	Flexbox-based grid system 基于 Flexbox 的网格系统	12-column grid system with predefined widths 12 列网格系统，具有预定义的宽度
Components 组件	No pre-designed components 无预设计组件	Offers a variety of pre-designed components such as buttons, modals, forms 提供各种预设计组件，如按钮、模态框、表单

Conclusion

In this lab, you created responsive layouts using **Tailwind CSS** and **Bootstrap**. You learned about the key features of both frameworks and implemented a responsive webpage that adapts to different screen sizes. With these skills, you can now build modern, mobile-friendly websites.

Author

Rajashree Patil

© IBM Corporation. All rights reserved.