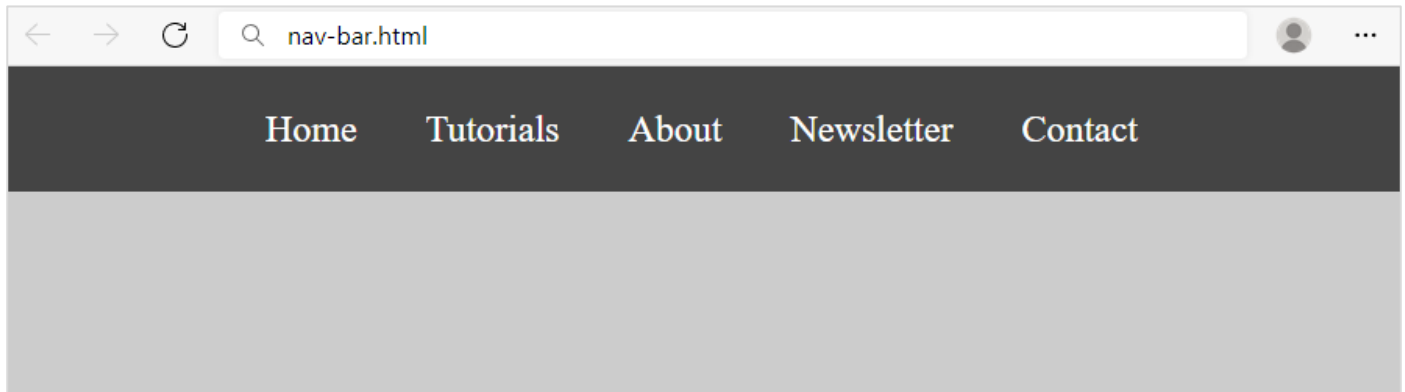


Exercise: HTML Structure

You can check your solutions here: <https://judge.softuni.org/Contests/3331/HTML-Structure>.

1. Navigation Bar

Create a Web page, holding like the following navigation bar:



Use the texts from the file **site-texts.txt**.

Create two files: **nav-bar.html** and **nav-bar.css**.

Constraints

- **<body>**
 - Margin: **0px**;
 - Padding: **0px**;
 - Background Color: **#CCCCCC**;
- **<nav>** tag as container
- **** tag for unordered list
 - Background Color: **#444**;
 - Center the text
 - Padding: **0px**;
 - Margin: **0px**;
 - **list-style:none**;
- **** tag for list item
 - Text size: **24px**;
 - Line Height: **40px**;
 - Height: **40px**;
 - Padding: **20px**;
- **<a>** tag for hyperlink
 - **text-decoration: none**;
 - Text-Color: **#ffffff**;

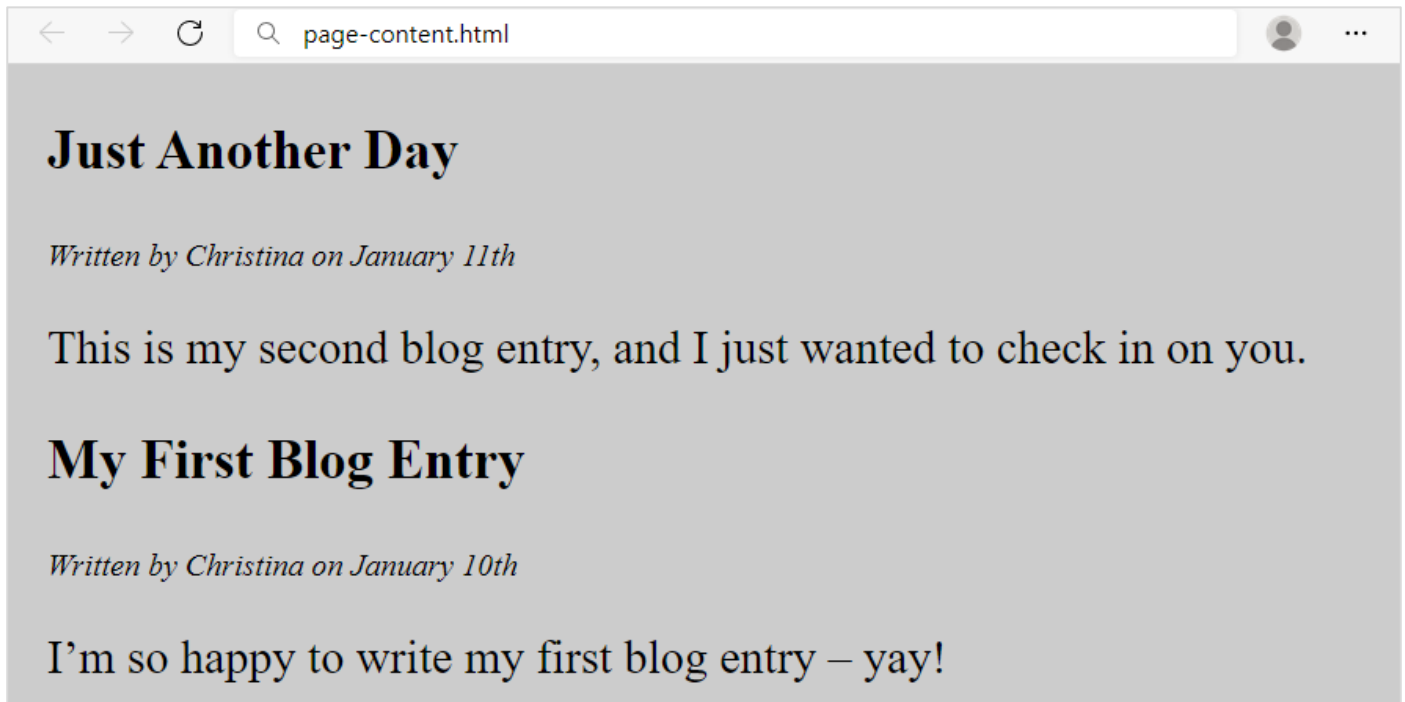
Hints

Use:

- **** with **display: inline-block**;

2. Page Content

Create a Web page like the show below, using the text from the file **site-texts.txt**.



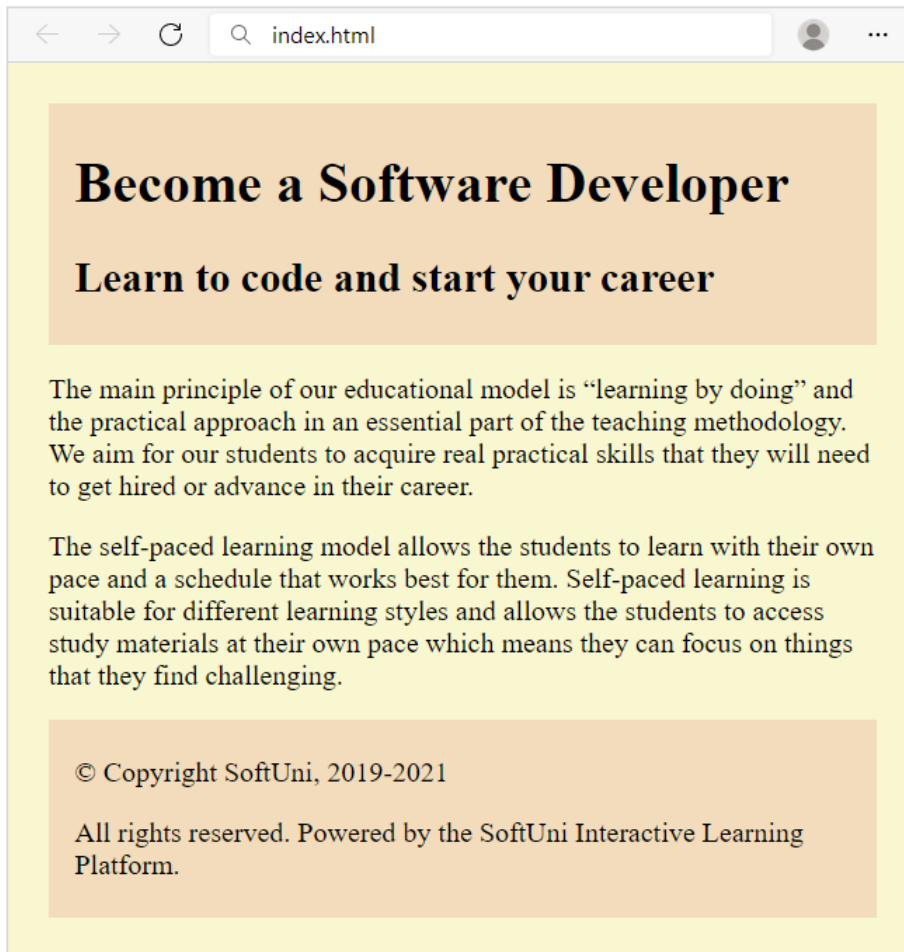
Create two files: **page-content.html** and **page-content.css**.

Constraints

- **<section>** as a container
- **<article>** tag for content
 - **<header>** with:
 - **<h1>** Text size: **28px**;
 - **<p>** Font Style: **italic**;
- Paragraph
 - Text size: **24px**;
- **<body>**
 - Margin: **0px**;
 - Padding: **0px**;
 - Background color: **#CCCCCC**;
- **<section>**
 - Margin Left: **20px**;

3. Semantic Tags

Create a web page like the following:



Use the texts from the file **site-texts.txt**.

Create files: **index.html** and **styles.css**.

Constraints

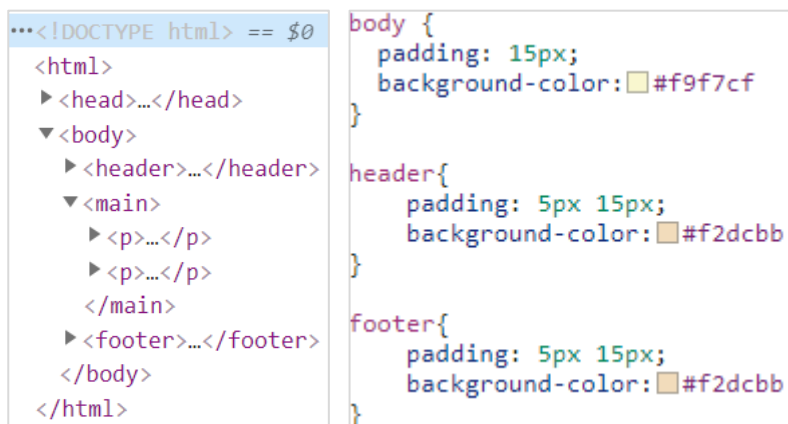
HTML:

- The title should be "**Semantic Tags**"
- Add **header** tag for the header section
 - Use **h1** tag for the heading
- Add **main** tag for the main content
 - Create two paragraphs inside (**p** tag)
- Use **footer** tag for the last section
 - Create two paragraphs inside (**p** tag)

The CSS in the example above uses:

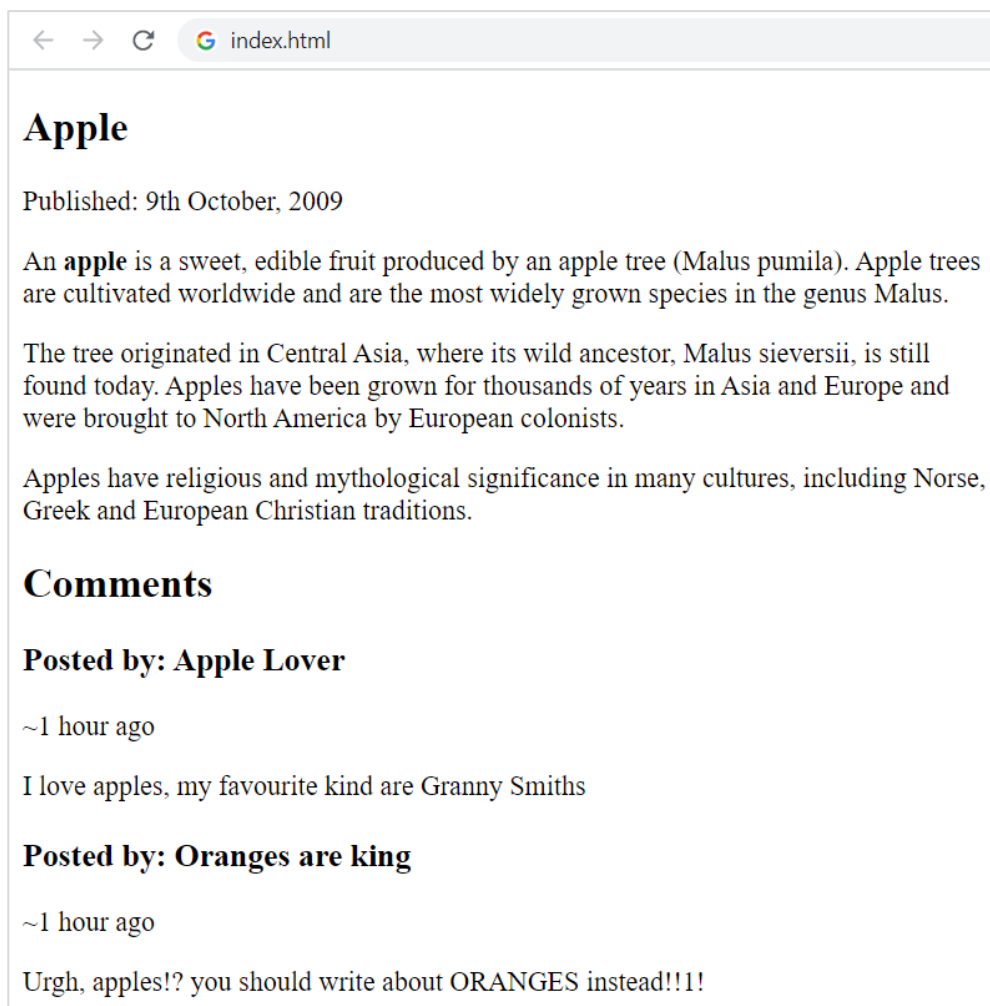
- padding: 15px, 5px 15px
- background colors: #f9f7cf, #f2dcbb

Hints

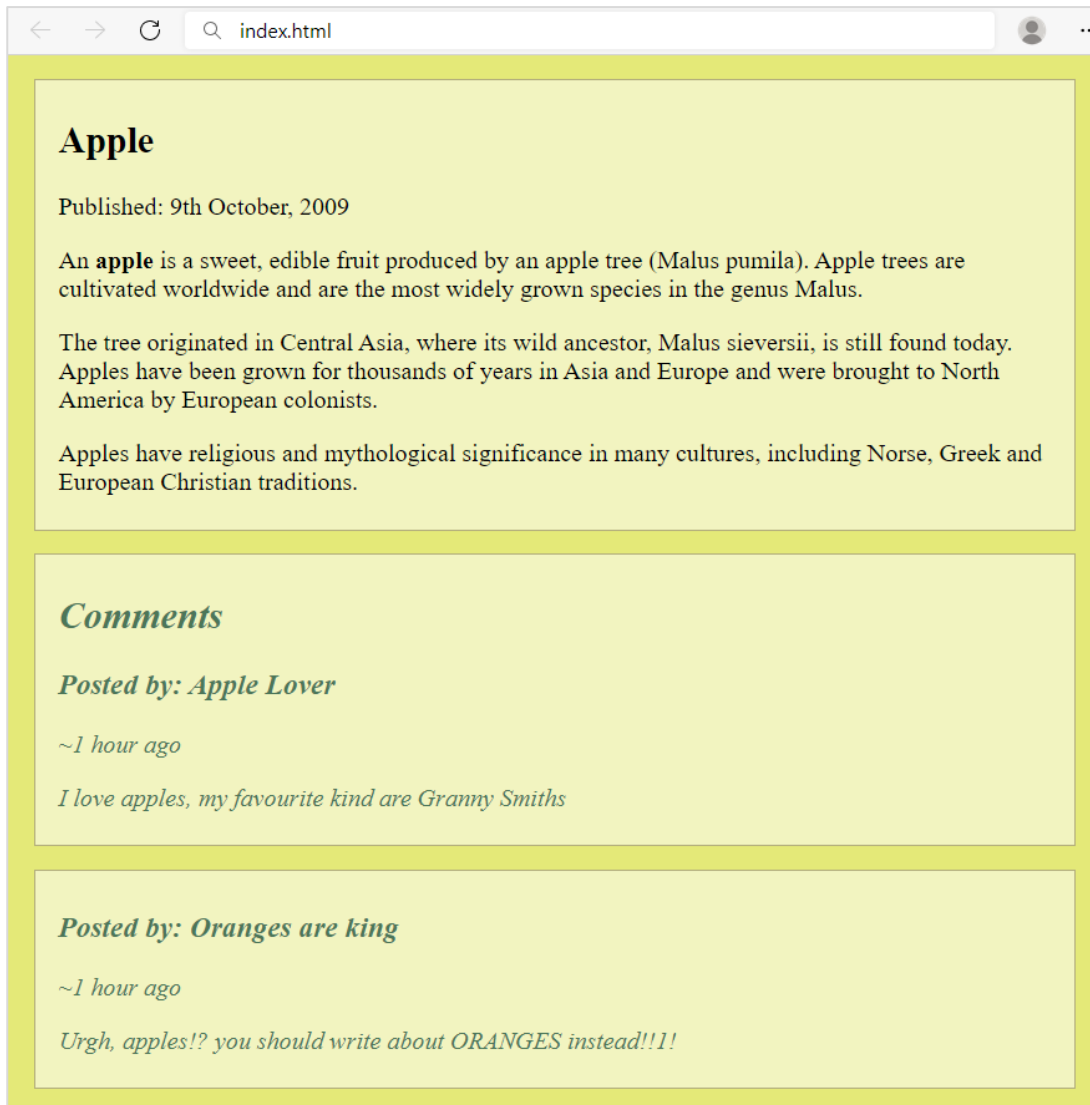


4. Semantic Article Page

Create a web page like the following:



Add CSS to make the Web page look better, like this:



Use the texts from the file **site-texts.txt**.

Create files: **index.html** and **styles.css**.

Constraints

- The title should be "Semantic Article Page"
- Use **article** tag to create an article
 - The article has header with **h1** heading and a **paragraph** for the published date inside
- Use **p** tag to create 3 paragraphs after the **header**.
 - The paragraphs contain the article content (*info for apple*)
 - Use **b** tag where is needed
- Use **section** tag to create the "Comment section"
 - Add **two articles** inside the section
 - Each **article** has a **heading (h2)** for the **title**
 - Each **article** has **two paragraphs** for the **time** and the **comment**

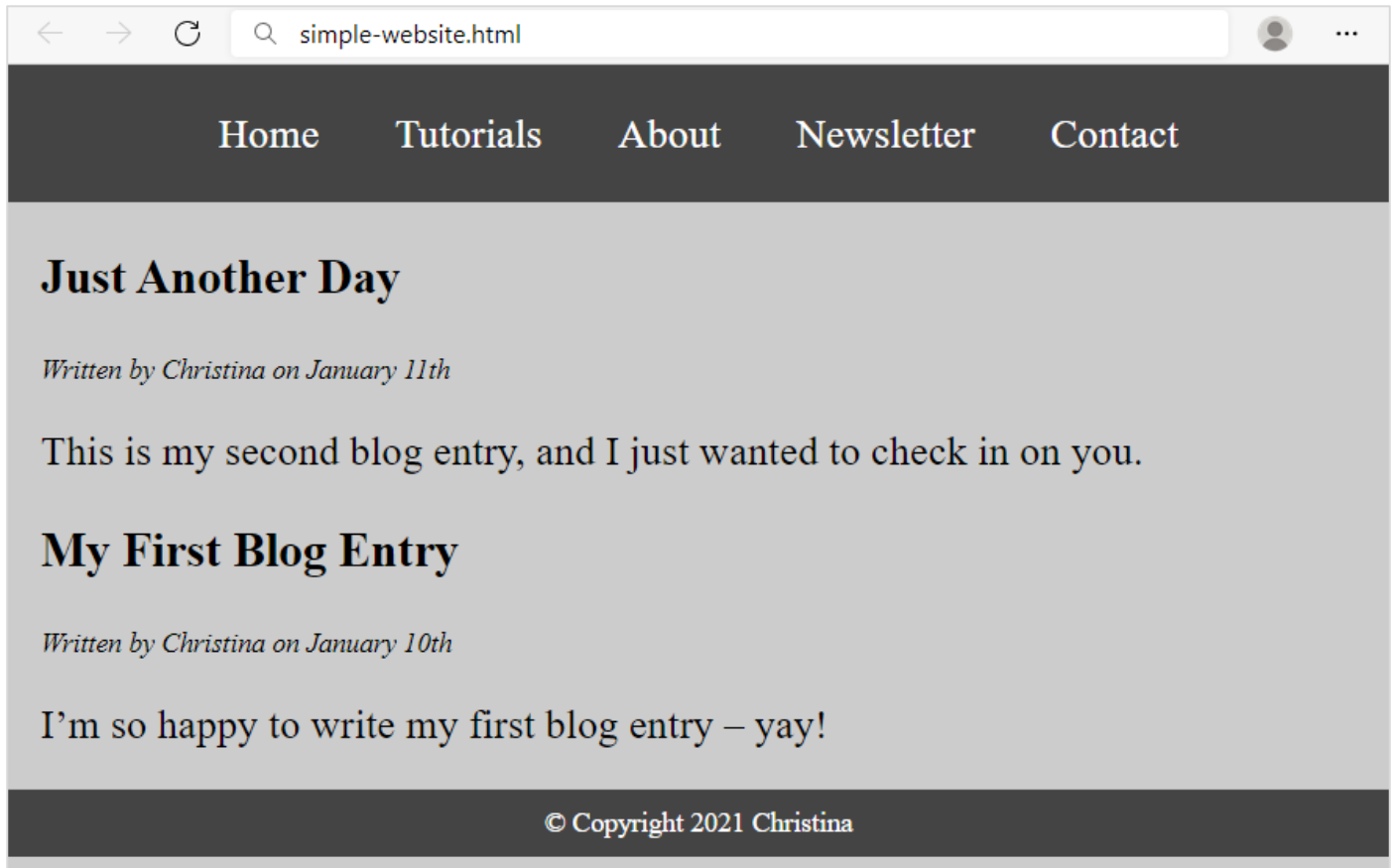
Used CSS in the example above:

- padding: 5px 15px;
- font-style: italic
- color: #5c6e91

- background colors: #f2f4c0, #e4e978, #4c7658;
- border: 1px solid #b2ad7d;
- margin: 15px 10px;

5. Simple Website

Using the navigation bar and page content from the previous two exercises, create a Web page like the following:



Use the texts from the file **site-texts.txt**.

Create two files: **simple-website.html** and **simple-website.css**.

Constraints

- Use the **HTML** and **CSS** from previous two problems
- **<footer>** tag
 - Background color: **#444**;
 - **<p>** tag for text
 - Text Color: **#fff**;
 - Center the text
 - Padding: **10px**

Hints

Use:

- **©** for copyright symbol

6. Semantic Blog Layout

Create a **blog layout page**, which holds **header**, **footer** and **main**. The **main** should hold two **sections**, each holding two **articles**. It should look like this (without any styling):



Developer News

- [HTML](#)
- [JavaScript](#)

HTML

New feature: JPEG XL image format

JPEG XL image format is a modern image format optimized for web environments. JPEG XL generally has better compression than WebP, JPEG, PNG and GIF and is designed to supersede them.

Learn more at: https://en.wikipedia.org/wiki/JPEG_XL

By The HTTP Working Group. Published January 23rd

New feature: Cookie Store API

Cookie Store API is a JS API for reading and modifying cookies. Compared to the existing `document.cookie` method, the API provides a much more modern interface, which can also be used in service workers.

Learn more at: <https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/cookies/CookieStore>

By Joshua Bell. Published December 15th

JavaScript

Billboard.js 3.0: The D3.js-Powered Chart Library

Billboard.js is a popular library supporting a whole range of chart types out of the box (bar, line, areas, donuts, radars, and various hybrids) – 3.0 brings candlestick / OHLC chart support (often used in financial contexts) and supports D3.js v6.

Learn more at: <https://github.com/naver/billboard.js>

By Jae Sung Park. Published March 24th

Nine JavaScript and TypeScript ORMs

A roundup, complete with code examples, showing off a variety of tools like Knex.js, Mongoose, TypeORM, Waterline, and Bookshelf.

Learn more at: <https://www.sitepoint.com/javascript-typescript-orms/>

By Michael Wanyoike. Published March 31st

© 2021 DeveloperNewsPortal.info

Please contact [Troy McClure](#) for questions about these articles.

Use the texts from the file **site-texts.txt**.

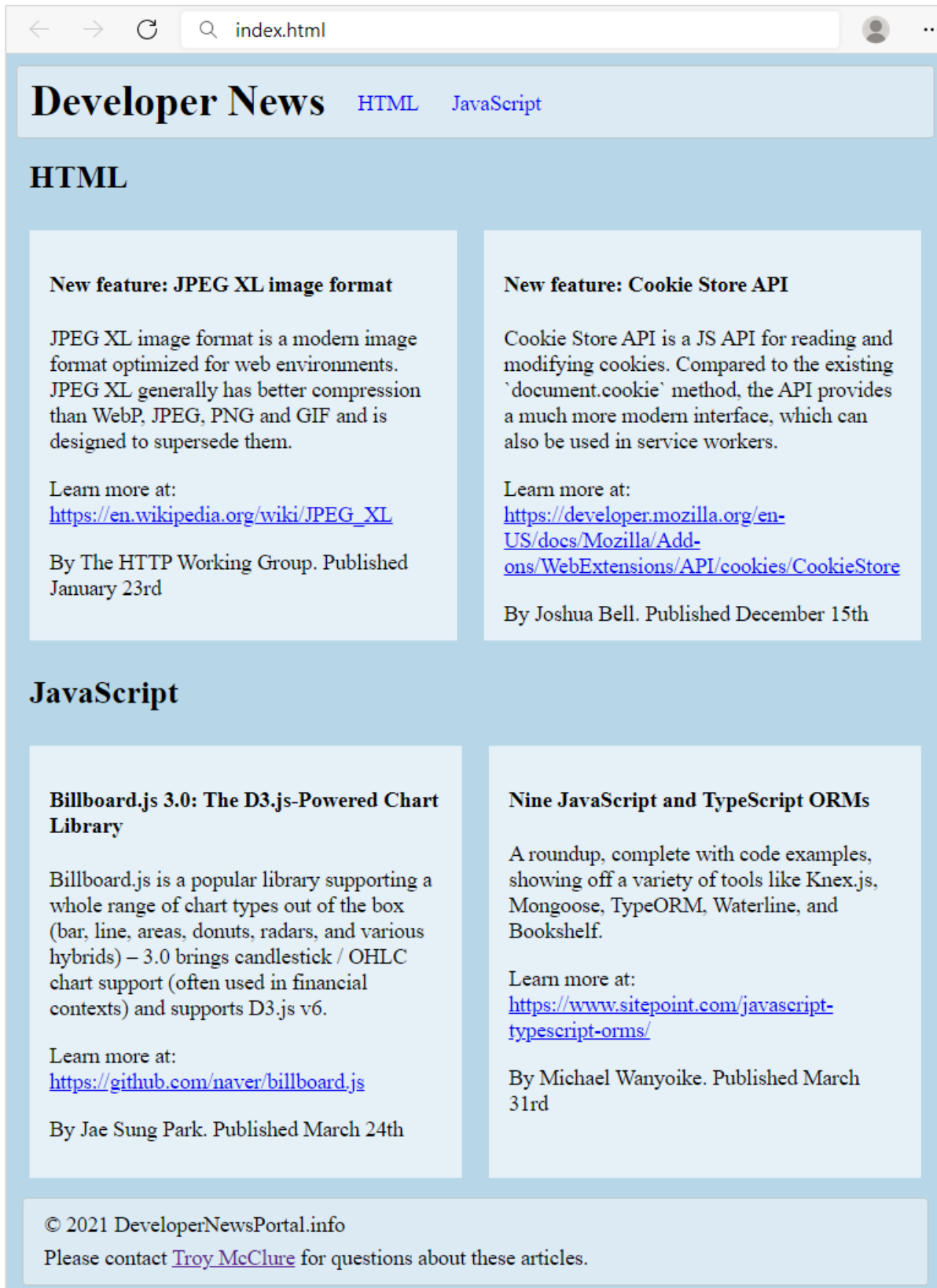
Create files: **index.html** and **styles.css**.

Constraints

- The title should be "**Semantic Blog Layout**"
- Use **header** tag for the header section
 - Use **h1** tag for the heading
 - Use **nav** tag for the navigation with an unordered list (**ul**) inside
 - Create two **list** items with **anchor (a)** tags inside
- Use **main** tag for the page main content
 - Create two **sections** in the main. Each section has:
 - A **header** with **h2** heading
 - Two **articles** with **h4** heading and three **p** tags inside
 - For the dates use **time** tag
- Use **footer** tag with **two** paragraphs Inside
 - Use **anchor** tag for the name in the last sentence

Add CSS Styling

Add **CSS styling** to make the blog look better, like at the following screenshot:



Create files: **index.html** and **styles.css**.

Use the following **CSS styles** to make your HTML look like at the screenshot:

```
header {  
    border-radius: 3px;  
    padding: 8px 10px;  
    background-color: #dceaf4;
```

```

    border: 1px solid #bfcacb;
}
header * {
    display: inline;
    vertical-align: middle;
}
nav ul {
    margin: 0;
    padding: 10px;
}
nav li {
    margin: 10px;
}
nav li a {
    text-decoration: none;
}

body{
    background-color: #b7d7e8;
}

section h2 {
    margin: 15px 10px;
}

article {
    display: table-cell;
    width: 50%;
    padding: 10px 15px;
    background-color: #e5f0f7;
    border: 10px solid #b7d7e8;
}

footer {
    margin: 5px;
    padding: 5px 10px;
    border-radius: 3px;
    background-color: #dceaf4;
    border: 1px solid #bfcacb;
}

footer p {
    margin: 5px;
}

```