

## Tutorial 03: Implementing CSS for JavaJam Coffee House website

In this exercise, you have to do the following tasks:

### Task 1: Create a folder

Create a folder called “**javajam2**” to contain your JavaJam website files. Copy **all** the files from your Tutorial 2’s folder (javajam) into the javajam2 folder. Next, copy the **new image files** (background.gif, javajamlogo.jpg, mugs.jpg, windingroad.jpg) into the javajam2 folder. You may organize files and folder structure in your project folder (e.g. create ‘images’ and ‘stylesheets’ folders to store images and css files), here are some references:

- [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/Dealing\\_with\\_files](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/Dealing_with_files)
- <https://stuyhsdesign.wordpress.com/basic-html/directory-structure/>

**NOTE:** If you have installed Git / SVN in your computer, you may consider to use Git/SVN to manage the versions of your codes. Reference for:

- Differences between Git and SVN: <https://www.geeksforgeeks.org/difference-between-git-and-svn/>
  - Install Git: <https://git-scm.com/download>
  - Git Documentation: <https://git-scm.com/doc>
- Learn Git In 15 Minutes: <https://www.youtube.com/watch?v=USjZcfj8yxE>
- GitHub with VS Code: VS Code will leverage your machine’s Git installation, you need to install Git first before you get these features. Make sure you install at least version 2.0.0. (to get started with the GitHub in VS Code, you’ll need to create a GitHub account). References:
  - <https://github.com/>
  - <https://docs.github.com/en/get-started/quickstart/hello-world>
  - <https://code.visualstudio.com/learn/students/github-pack>
  - [https://code.visualstudio.com/docs/editor/versioncontrol#\\_git-support](https://code.visualstudio.com/docs/editor/versioncontrol#_git-support)
  - <https://www.youtube.com/watch?v=jWLUhHWXMT8>
- You can also consider code hosting platforms such as GitLab and Bitbucket:
  - <https://stackshare.io/stackups/bitbucket-vs-github-vs-gitlab>

## Task 2: Configure the External Cascading Style Sheet (CSS)

You will use a text editor to create a new external cascading style sheet, save it as **javajam.css**.

### Code the **javajam.css** to configure the following:

1. Configure **global styles** for the document (use the **body element selector**) with background color #FCEBB6; text color #221811; Verdana, Arial, or any sans-serif font-family, and set a background image using background.gif.

```
body {background-color: #FCEBB6;
      color: #221811;
      font-family: Verdana, Arial, sans-serif;
      background-image: url(background.gif);
}
```

2. Configure for the **header area** with the background color #D2B48C, 150 pixels of height, background image (javajamlogo.jpg) and no repeat for background-repeat property.

```
header { background-color: #D2B48C;
          height: 150px;
          background-image:url(javajamlogo.jpg);
          background-repeat: no-repeat;
}
```

3. A wrapper is commonly used to center a layout on the page. The wrapper keeps a layout from looking too wide or too narrow depending on the device or viewport width. Configure a **wrapper** for JavaJam Website with background color #FEF6C2, 80% width, auto right margin, auto left margin, 900 pixels of minimum width, 1280 pixels of maximum width, and set box-shadow effect using 3 pixels of offset-x, 3 pixels of offset-y, 3 pixels of blur-radius and color #666666.

```
#wrapper { background-color: #FEF6C2;
           width: 80%;
           margin-right: auto;
           margin-left: auto;
           min-width: 900px;
           max-width: 1280px;
           box-shadow: 3px 3px 3px #666666;
}
```

4. Configure for the **navigation area** with center alignment (text-align: center), bold font-weight, 1.5em of font-size and 10 pixels of top padding.

```
nav {text-align: center;
     font-weight: bold;
     font-size: 1.5em;
     padding-top: 10px;
}
```

5. Configure the **navigation hyperlinks**. Use a contextual selector to add a new style rule for the anchor tags within the nav. Configure this selector with 15 pixels of bottom padding and no underline on hyperlinks.

```
nav a { text-decoration: none; }
```

6. Configure the **main area** with 2em of left padding, 2em of right padding, 2em of bottom padding and displays the element in main area as a block element (like <p>, it starts on a new line, and takes up the whole width).

```
main { padding-left: 2em;
       padding-right: 2em;
       padding-bottom: 2em;
       display: block;
}
```

7. Configure the footer area with background color #D2B48C, small font size, italic font style and 2 pixels of top border with color #221811 and solid line.

```
footer { background-color: #D2B48C;
         font-size: small;
         font-style: italic;
         text-align: center;
         padding-bottom: 10px;
         border-top: 2px #221811 solid;
}
```

8. Configure the **h1 element selector** with background color 45 pixels of top padding, 220 pixels of left padding and 3em of font-size.

```
h1 { padding-top: 45px;
     padding-left: 220px;
     font-size: 3em;
}
```

9. Configure the h4 element selector with background color #D2B48C, 1.2em of font size, 10 pixels of left padding, and 5 pixels of bottom padding.

```
h4 { background-color: #D2B48C;
      font-size: 1.2em;
      padding-left: 10px;
      padding-bottom: 5px;
    }
```

10. Configure img element selector with 10 pixels of left padding and 10 pixels of right padding.

```
img { padding-left: 10px;
      padding-right: 10px;
    }
```

11. Configure a class element named **details** with 20% left padding and 20% right padding.

This class element will be used in the music.html files to style the performance details.

```
.details { padding-left: 20%;
          padding-right: 20%;
        }
```

### Task 3: The Home Page

Launch a text editor, and open the **index.html** file. You will **modify** this file to apply styles from the javajam.css external style sheet as follows:

1. Add a `<link>` element to associate the web page with the javajam.css external style sheet file. Save and test your index.html page in a browser and you will notice that the styles configured with the body and h1 element selectors are already applied!

```
<head>
    <title>JavaJam Coffee House</title>
    <link rel="stylesheet" href="javajam.css">
</head>
```

**Note:** If you have created folders such as “stylesheets” and “images” to store css and images, then you need to include the file path as “stylesheets/javajam.css”, same goes to images “images/filename”.

2. Configure the page **body** area, add a `div` with `id` named `wrapper` to associate the web page’s body area with the wrapper layout configured in the javajam.css.

```
<body>
<div id="wrapper">

</div>
</body>
```

3. Configure the page **footer** area. **Remove** the `<small>` and `<i>` elements, because the `font-size` and `font-style` are configured in javajam.css.
4. Add a **new image (windingroad.jpg)** in the **<main>** area, **before** the “Relax at JavaJam” heading. Set the following image attributes: alternate text as “winding road through the woods”, right alignment, 400 pixels of width, and 300 pixels of height.

```

```

5. Replace the heading `<h2>Relax at JavaJam</h2>` with the new text:  
*Follow the Winding Road to JavaJam*
6. Add a **new paragraph <p>** after the heading “Follow the Winding Road to JavaJam” with the following text:  
*We're a little out of the way, but take a drive down Route 42 to JavaJam today! Indulge in our locally roasted free-trade coffee and home-made pastries. You'll feel right at home at JavaJam!*

7. After the new paragraph, add a new **heading <h3>** with the text:  
*JavaJam Coffee House features:*
8. **Save** the **index.html** file, and **test** it in a browser. Your page should look similar to the one shown in Figure 1.



**Figure 1: New JavaJam index.html**

## Task 4: The Menu Page

Launch a text editor, and open the **music.html** file. You will **modify** this file to apply styles from the **javajam.css** external style sheet as follows:

1. Add a `<link>` element to associate the web page with the **javajam.css** external style sheet file.

```
<head>
    <title>JavaJam Coffee House</title>
    <link rel="stylesheet" href="javajam.css">
</head>
```

2. Configure the page **body** area, add a `div` with `id` named `wrapper` to associate the web page's body area with the wrapper layout configured in the **javajam.css**.

```
<body>
    <div id="wrapper">

    </div>
</body>
```

3. Configure the page **footer** area. **Remove** the `<small>` and `<i>` elements, because the `font-size` and `font-style` are configured in **javajam.css**.
4. Add a **new image (mugs.jp)** in the `<main>` area, **before** the "Coffee at JavaJam" heading. Set the following image attributes: alternate text as "mugs on a wall", right alignment, 400 pixels of width, and 300 pixels of height.

```

```

5. Add a **new paragraph <p>** after the heading "Coffee at Javajam" with the following text:  
*Indulge in our locally roasted free-trade coffee and enjoy the aroma, the smooth taste, the caffeine! Join our Mug Club and get a 10% discount on each cup of coffee you purchase — ask the barista for details.*
6. **Save** the **menu.html** page, and **test** it in a browser and you will notice that the styles configured in **javajam.css** are already applied. Your page should look similar to the one shown in **Figure 2**.



**Figure 2: New JavaJam menu.html**



## Task 5: The Music Page

Launch a text editor, and open the **menu.html** file. You will **modify** this file to apply styles from the **javajam.css** external style sheet as follows:

1. Add a `<link>` element to associate the web page with the **javajam.css** external style sheet file.

```
<head>
    <title>JavaJam Coffee House</title>
    <link rel="stylesheet" href="javajam.css">
</head>
```

2. Configure the page **body** area, add a `div` with `id` named `wrapper` to associate the web page's body area with the wrapper layout configured in the **javajam.css**.

```
<body>
<div id="wrapper">

</div>
</body>
```

3. Configure the page **footer** area. **Remove** the `<small>` and `<i>` elements, because the `font-size` and `font-style` are configured in **javajam.css**.
4. **Replace** the paragraph tags `<p>` `</p>` with `<div>` `</div>` tags. Assign the new `div` section with a **class name** called **"details"** to associate the **music.html** page with the **.details** class styles configured in the **javajam.css**

```
<h4>January</h4>
<div class="details">
<a href="melanie.jpg"></a>
Melanie Morris entertains with her melodic folk style.
</div>

<h4>February</h4>
<div class="details">
<a href="greg.jpg"></a>
Tahoe Greg is back from his tour. New songs. New stories.
</div>
```

5. **Save** the **music.html** page, and **test** it in a browser and you will notice that the styles configured in **javajam.css** are already applied. Your page should look similar to the one shown in **Figure 3**.



**Figure 3: New JavaJam music.html**

## Task 6: The Jobs Page

Launch a text editor, and open the **jobs.html** file. You will **modify** this file to apply styles from the **javajam.css** external style sheet as follows:

1. Add a `<link>` element to associate the web page with the **javajam.css** external style sheet file.

```
<head>

    <title>JavaJam Coffee House</title>

    <link rel="stylesheet" href="javajam.css">

</head>
```

2. Configure the page **body** area, add a `div` with `id` named `wrapper` to associate the web page's body area with the wrapper layout configured in the **javajam.css**.

```
<body>

<div id="wrapper">

</div>

</body>
```

3. Configure the page **footer** area. **Remove** the `<small>` and `<i>` elements, because the `font-size` and `font-style` are configured in **javajam.css**.
4. **Save** the **jobs.html** page, and **test** it in a browser and you will notice that the styles configured in **javajam.css** are already applied. Your page should look similar to the one shown in **Figure 4**.



**Figure 4: New JavaJam jobs.html**