

Table of Contents

[Table of Figures 3](#_Toc183277356)

[Chapter One: Introduction to Web Development 6](#_Toc183277357)

[The Internet 6](#_Toc183277358)

[Web page 6](#_Toc183277359)

[Website 6](#_Toc183277360)

[Web server 7](#_Toc183277361)

[Domain Name 7](#_Toc183277362)

[Chapter Two: HTML and CSS 8](#_Toc183277363)

[Introduction to HTML and CSS 8](#_Toc183277364)

[HTML Element 9](#_Toc183277365)

[HTML Block and Inline Elements 10](#_Toc183277366)

[Exercise 11](#_Toc183277367)

[HTML Attributes 12](#_Toc183277368)

[HTML Head Element 12](#_Toc183277369)

[HTML Title Element 12](#_Toc183277370)

[HTML Favicon 14](#_Toc183277371)

[HTML Meta Element 14](#_Toc183277372)

[HTML Style Element 15](#_Toc183277373)

[HTML Link Element 16](#_Toc183277374)

[HTML Script Element 16](#_Toc183277375)

[Exercise 16](#_Toc183277376)

[Styling HTML with CSS 17](#_Toc183277377)

[Inline CSS 17](#_Toc183277378)

[Internal CSS 17](#_Toc183277379)

[External CSS 18](#_Toc183277380)

[HTML Body Element 19](#_Toc183277381)

[Basic sections of a document 19](#_Toc183277382)

[Header Element <header> 19](#_Toc183277383)

[Navigation Bar Element <nav> 19](#_Toc183277384)

[Main Content Element <main> 19](#_Toc183277385)

[Section Element <section> 19](#_Toc183277386)

[Sidebar Element <aside> 20](#_Toc183277387)

[Footer Element <footer> 20](#_Toc183277388)

[HTML Image Element <img> 21](#_Toc183277389)

[HTML id and class attributes 21](#_Toc183277390)

[HTML List Element 23](#_Toc183277391)

[HTML Link Element 24](#_Toc183277392)

[CSS Introduction 27](#_Toc183277393)

[CSS Syntax 27](#_Toc183277394)

[CSS Background 28](#_Toc183277395)

[CSS Height and Width 30](#_Toc183277396)

[CSS Display 31](#_Toc183277397)

[CSS Flexbox 31](#_Toc183277398)

[CSS List 34](#_Toc183277399)

[CSS Colors 35](#_Toc183277400)

[CSS Margin 36](#_Toc183277401)

[CSS Padding 38](#_Toc183277402)

[CSS Border 40](#_Toc183277403)

[CSS Font 43](#_Toc183277404)

[CSS Google Font 45](#_Toc183277405)

[CSS Text 47](#_Toc183277406)

[Adding Icons with Font Awesome 50](#_Toc183277407)

[References 55](#_Toc183277408)

# Table of Figures

[Figure 1: BellBank Website 7](#_Toc183277613)

[Figure 2: Basic HTML structure 9](#_Toc183277614)

[Figure 3: Basic CSS Structure 9](#_Toc183277615)

[Figure 4: HTML Element 10](#_Toc183277616)

[Figure 5: Empty element 10](#_Toc183277617)

[Figure 6: Nesting HTML Elements 10](#_Toc183277618)

[Figure 7: HTML Attribute 13](#_Toc183277619)

[Figure 8: Head Element 13](#_Toc183277620)

[Figure 9: Title element 14](#_Toc183277621)

[Figure 10: Title element II 14](#_Toc183277622)

[Figure 11: HTML Favicon 15](#_Toc183277623)

[Figure 12: HTML Favicon II 15](#_Toc183277624)

[Figure 13: HTML Meta 15](#_Toc183277625)

[Figure 14: HTML Style 16](#_Toc183277626)

[Figure 15: HTML Link 17](#_Toc183277627)

[Figure 16: HTML Script 17](#_Toc183277628)

[Figure 17: Inline CSS 18](#_Toc183277629)

[Figure 18: Internal CSS 18](#_Toc183277630)

[Figure 19: External CSS 19](#_Toc183277631)

[Figure 20: HTML5 Semantics Standard 21](#_Toc183277632)

[Figure 21: HTML img 22](#_Toc183277633)

[Figure 22: HTML id attribute 22](#_Toc183277634)

[Figure 23: CSS id selector 23](#_Toc183277635)

[Figure 24: HTML class attribute 23](#_Toc183277636)

[Figure 25: CSS class selector 23](#_Toc183277637)

[Figure 26: HTML ordered list 24](#_Toc183277638)

[Figure 27: HTML ordered list output 24](#_Toc183277639)

[Figure 28: HTML unordered list 25](#_Toc183277640)

[Figure 29: HTML unordered list output 25](#_Toc183277641)

[Figure 30: HTML link element 26](#_Toc183277642)

[Figure 31: Header section HTML Structure 27](#_Toc183277643)

[Figure 32: CSS Syntax 28](#_Toc183277644)

[Figure 33: CSS background-color property 29](#_Toc183277645)

[Figure 34: CSS background-image property 29](#_Toc183277646)

[Figure 35: CSS background-repeat property 30](#_Toc183277647)

[Figure 36: CSS background-position property 30](#_Toc183277648)

[Figure 37: CSS background-attachment property 30](#_Toc183277649)

[Figure 38: CSS background shorthand property 31](#_Toc183277650)

[Figure 39: CSS height and width property 32](#_Toc183277651)

[Figure 40: CSS display property 32](#_Toc183277652)

[Figure 41: Header Section 33](#_Toc183277653)

[Figure 42: adding id attribute 33](#_Toc183277654)

[Figure 43: CSS flex display property 34](#_Toc183277655)

[Figure 44: Header section after add flex display property 34](#_Toc183277656)

[Figure 45: CSS justify-content and align-items property 35](#_Toc183277657)

[Figure 46: Header Section Display 35](#_Toc183277658)

[Figure 47: CSS list-style-type property 35](#_Toc183277659)

[Figure 48: List-style-type output 36](#_Toc183277660)

[Figure 49: CSS flex-direction property 36](#_Toc183277661)

[Figure 50: CSS flex-decoration property output 36](#_Toc183277662)

[Figure 51: CSS color property 36](#_Toc183277663)

[Figure 52: CSS color property output 36](#_Toc183277664)

[Figure 53: CSS text-decoration property 37](#_Toc183277665)

[Figure 54: CSS text-decoration property output 37](#_Toc183277666)

[Figure 55: CSS margin property 38](#_Toc183277667)

[Figure 56: CSS margin property output 38](#_Toc183277668)

[Figure 57: CSS margin property II 38](#_Toc183277669)

[Figure 58: CSS margin property III 39](#_Toc183277670)

[Figure 59: CSS margin property IV 39](#_Toc183277671)

[Figure 60: headerSection much better! 39](#_Toc183277672)

[Figure 61: CSS padding property 40](#_Toc183277673)

[Figure 62: CSS padding property II 40](#_Toc183277674)

[Figure 63: CSS padding property III 41](#_Toc183277675)

[Figure 64: headerSection much better! II 41](#_Toc183277676)

[Figure 65: CSS border-style property 42](#_Toc183277677)

[Figure 66: CSS border-width property 42](#_Toc183277678)

[Figure 67: CSS border-color property 43](#_Toc183277679)

[Figure 68: CSS border property 43](#_Toc183277680)

[Figure 69: CSS border-bottom property 43](#_Toc183277681)

[Figure 70: headerSection much better! III 44](#_Toc183277682)

[Figure 71: CSS font-family property 44](#_Toc183277683)

[Figure 72: CSS font-style property 45](#_Toc183277684)

[Figure 73: CSS font-size property 45](#_Toc183277685)

[Figure 74: CSS font-weight property 45](#_Toc183277686)

[Figure 75: CSS font-variant property 46](#_Toc183277687)

[Figure 76: CSS Google Font 46](#_Toc183277688)

[Figure 77: CSS Google Font II 46](#_Toc183277689)

[Figure 78: CSS Google Font III 47](#_Toc183277690)

[Figure 79: CSS Google Font IV 47](#_Toc183277691)

[Figure 80: CSS Google Font V 47](#_Toc183277692)

[Figure 81: CSS Google Font VI 48](#_Toc183277693)

[Figure 82: headerSection much better! IV 48](#_Toc183277694)

[Figure 83: CSS Text property 49](#_Toc183277695)

[Figure 84: main element 50](#_Toc183277696)

[Figure 85: section element 50](#_Toc183277697)

[Figure 86: working on slogan div 50](#_Toc183277698)

[Figure 87: Slogan div output 51](#_Toc183277699)

[Figure 88: Download Link div content 51](#_Toc183277700)

[Figure 89: adding fontawesome icons 51](#_Toc183277701)

[Figure 90: embedding fontawesome link 52](#_Toc183277702)

[Figure 91: fontawesome website 52](#_Toc183277703)

[Figure 92: searching on fontawesome 52](#_Toc183277704)

[Figure 93: searching on fontawesome II 53](#_Toc183277705)

[Figure 94: Adding the star icons using fontawesome 53](#_Toc183277706)

[Figure 95: Adding the star icons using fontawesome II 54](#_Toc183277707)

[Figure 96: Slogan and download link section 54](#_Toc183277708)

[Figure 97: adding id attribute to the section element 54](#_Toc183277709)

[Figure 98: CSS display property (grid) 55](#_Toc183277710)

[Figure 99: CSS display property (grid) II 55](#_Toc183277711)

[Figure 100: Styling Slogan div 55](#_Toc183277712)

[Figure 101: Slogan stylesheet 56](#_Toc183277713)

[Figure 102: Slogan div output 56](#_Toc183277714)

# 

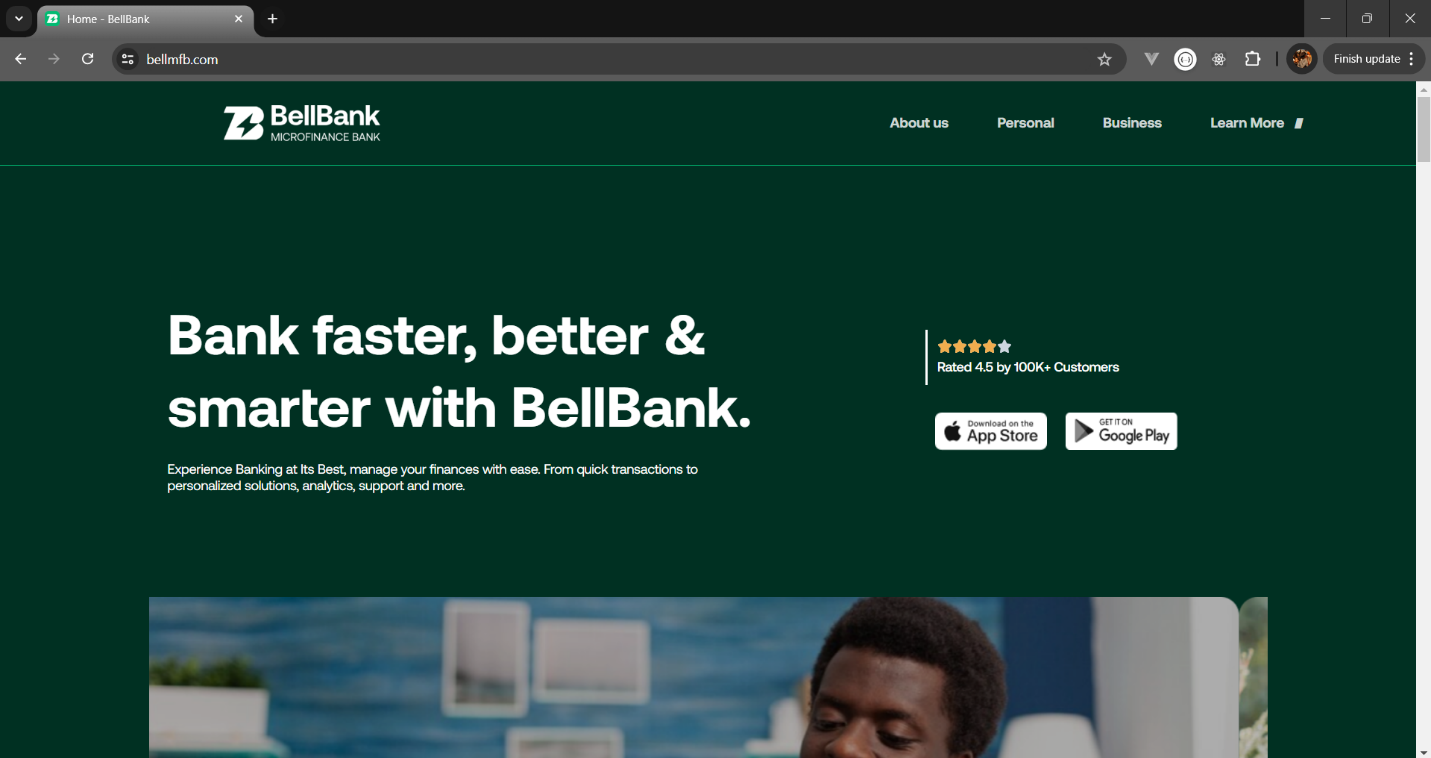
# Chapter One: Introduction to Web Development

## The Internet

The *Internet* is the backbone of the Web, the technical infrastructure that makes the Web possible. The Internet is a large network of computers which communicate all together. The various technologies that support the Internet have evolved over time, but the way it works hasn't changed that much: Internet is a way to connect computers all together and ensure that, whatever happens, they find a way to stay connected.

## [Web page](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/Pages_sites_servers_and_search_engines#web_page_2)

A *web page* is a simple document displayable by a browser. Web Pages are what make up a website. For example, on www.bellmfb.com, you can find the home page, about us page, personal page, business page, etc. Such documents are written in the HTML language. All web pages available on the web are reachable through a unique address. To access a page, just type its address (domain name) in your browser address search bar as shown below:



www.bellmfb.com

Figure 1: BellBank Website

## [Website](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/Pages_sites_servers_and_search_engines#website_2)

A *Website* is a collection of related web pages located under a single domain name. When you combine the home page, about us page, personal page, business page, etc. and store them under a single domain name www.bellmfb.com. A website is a collection of linked web pages that share a unique domain name. Each web page of a given website provides links, most of the time in the form of clickable portions of text, that allow the user to move from one page of the website to another. To access a website, type its domain name in your browser search bar, and the browser will display the website’s main web page, or homepage as shown in the figure 1 above.

## [Web server](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/Pages_sites_servers_and_search_engines#web_server_2)

A *web server* is a computer hosting one or more *websites*. “Hosting” means that all the *web pages* and their supporting files are available on that computer. The *web server* will send any *web page* from the *website* it is hosting to any user’s browser when the user sends a request.

## Domain Name

*Domain names* are a key part of the Internet infrastructure. They provide a human-readable address for any *web server* available on the Internet. Any Internet-connected computer can be reached through a public IP Address, either an IPv4 address (e.g. *104.21.35.168*) or an IPv6 address (*2606:4700:3037::6815:23a8*). Computers can handle such addresses easily, but people have a hard time finding out who is running the server or what service the website offers. IP addresses are hard to remember and might change over time. To solve all those problems, we use human-readable addresses called *domain names* for example [www.bellmfb.com](http://www.bellmfb.com).

**Footnote**: *You cannot “buy a domain name”, instead, you pay for the right to use a domain name for one or more years. You can renew your right, and your renewal has priority over other people’s applications. But you never own the domain name.*

# Chapter Two: HTML and CSS

## Introduction to HTML and CSS

HTML stands for Hyper Text Markup Language. HTML is a *markup language* that defines the structure of your content. HTML consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way.



Figure 2: Basic HTML structure

CSS stands for Cascading Style Sheets. CSS is used to styles web content. CSS describes how HTML elements are to be displayed on the screen, paper or in other media.

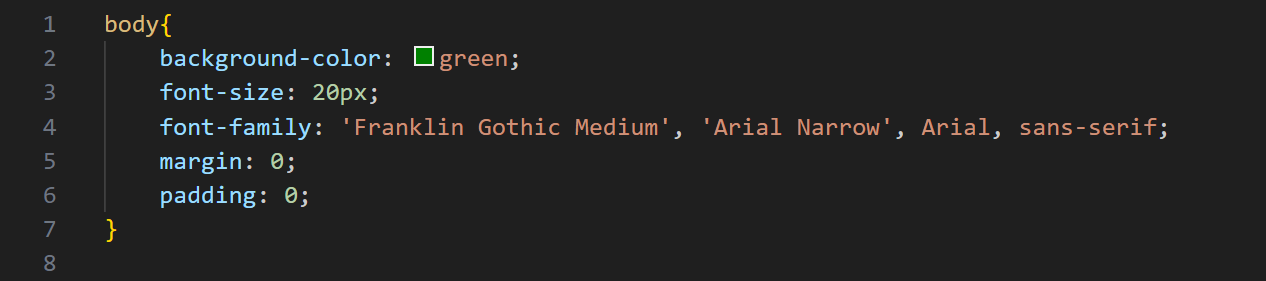


Figure 3: Basic CSS Structure

**Footnote**: *You save an HTML file with the extension of either .html or .htm but use .html as most developers use that. You save a CSS file with the extension of .css*

## HTML Element

An HTML element is an individual component of an HTML document. It represents semantics, or meaning. For example, the title element represents the title of the document as shown above.

An HTML element is defined by a start tag, some content, and an end tag.

Start tag

End tag

Contents

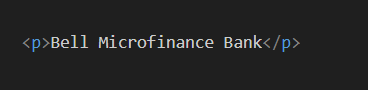


Figure 4: HTML Element

**Note:** Not all HTML element require the start tag and the end tag or close. They have no content and called **empty elements** example like *img, br, hr, etc.*



Figure 5: Empty element

HTML elements can be nested (this means that elements can contain other elements). All HTML documents consist of nested HTML elements. In fig. below, the *title* element is nested inside the *head* element and the *head* element is nested inside the *html* element.



Figure 6: Nesting HTML Elements

### HTML Block and Inline Elements

Every HTML element has a default display value, depending on what type of element it is.

A **block-level** element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can). The *<div>* element is the widely used block level element. Below are examples of block level elements in HTML:

<address> <article> <aside> <blockquote> <canvas> <dd> <div> <dl> <dt> <fieldset> <figcaption> <figure> <footer> <form> <h1>-<h6> <header> <hr> <li> <main> <nav> <noscript> <ol> <p> <pre> <section> <table> <tfoot> <ul> <video

An **inline element** does not start on a new line and only takes up as much width as necessary. The *<span>* element is the widely used inline level element. Below are examples of inline level elements in HTML:

<a> <abbr> <acronym> <b> <bdo> <big> <br> <button> <cite> <code> <dfn> <em> <i> <img> <input> <kbd> <label> <map> <object> <output> <q> <samp> <script> <select> <small> <span> <strong> <sub> <sup> <textarea> <time> <tt> <var>

### Exercise

1. HTML elements are used to style the website.
2. True
3. False
4. You don’t have to nest your HTML element.
5. Agree
6. Neutral
7. Disagree
8. You can nest the *title* element within the *body* element.
9. Yes
10. No
11. HTML elements can have attributes.
12. True
13. False
14. Choice the correct syntax and most appropriate syntax
15. *<img src=”myphoto.png” alt=”my-photo”></img>*
16. *<img src=”myphoto.png”></img>*
17. *<img src=”myphoto.png”>*
18. *<img src=”myphoto.png” alt=”my-photo”></img>*

**Footnote**:

1. *HTML elements are not case sensitive <P> means the same as <p>. The HTML standard does not require lowercase tags, but W3C recommends lowercase in HTML.*
2. *Never skip the end tag of an HTML element. Some HTML elements will display correctly, even if you forget the end tag. However, never rely on this! Unexpected results and errors may occur.*

## HTML Attributes

All HTML elements can have attributes. Attributes provide additional information about elements. Attributes are always specified in the start tag.



Figure 7: HTML Attribute

The *<img>* tag is used to embed an image in an HTML page. The *src* attribute specifies the path to the image to be displayed. We will learn more about images in HTML later, smile!

The *alt* attribute for the *<img>* tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to slow connection, or an error in the *src* attribute, or if the user uses a screen reader.

## HTML Head Element

The *<head>* element is a container for metadata (data about data) and is placed between the *<html>* tag and the *<body>* tag.

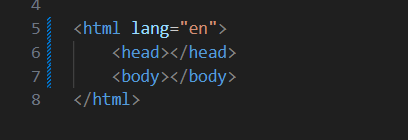


Figure 8: Head Element

HTML metadata is data about the HTML document. Metadata is not displayed. Metadata typically define the document title, character set, styles, scripts, and other meta information.

### HTML Title Element

The *<title>* element defines the title of the document. The title must be text-only, and it is shown in the browser’s title bar or in the page’s tab as shown in the figure below. The *<title>* element is required in HTML documents. The content of a page title is very important for search engine optimization (SEO). The page title is used by search engine algorithms to decide the order when listing pages in search results.

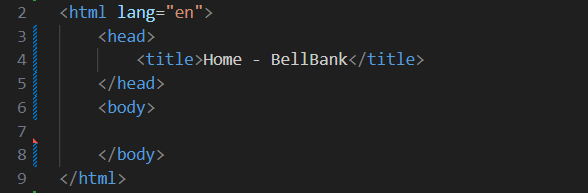


Figure 9: Title element

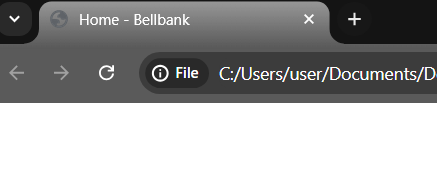


Figure 10: Title element II

### HTML Favicon

A favicon is a small image displayed next to the page title in the browser tab. You can use any image you like as your favicon.

To add a favicon to your page, use the *<link>* element with HTML attributes of *rel, type* and *href* as shown below.

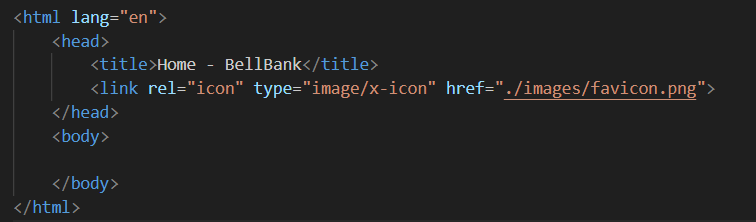


Figure 11: HTML Favicon

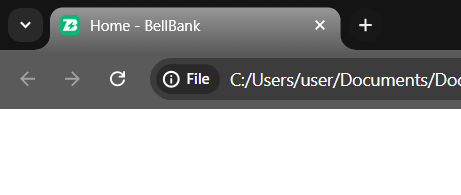


Figure 12: HTML Favicon II

### HTML Meta Element

The *<meta>* element is typically used to specify the character set, viewport settings, author, keywords and description.

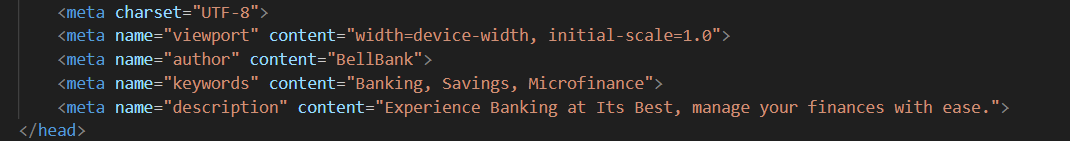


Figure 13: HTML Meta

The metadata will not be displayed on the page, but is used by browsers (how to display content or reload page), by search engines (keywords), and other web services.

### HTML Style Element

The *<style>* element is used to define style information for a single HTML page or in internal css style. You can style a web page either using in-line CSS, internal CSS and external CSS. We will learn more about these methods of styling later.

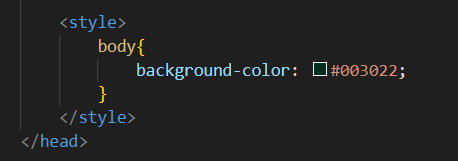


Figure 14: HTML Style

**Footnote**:

*When you add the <style> element, you first write the CSS property you want to apply to the element then a colon then the CSS value then you terminate it by using a semi-colon. We will learn more about CSS later in this book*.

### HTML Link Element

The *<link>* element defines the relationship between the current document and an external resource. The*<link>* tag is most often used to link to external style sheets.

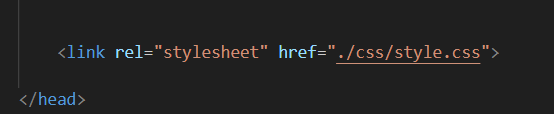


Figure 15: HTML Link

### HTML Script Element

The *<script>* element is used to JavaScript to an HTML web page. We will learn deep about JavaScript later in this book.

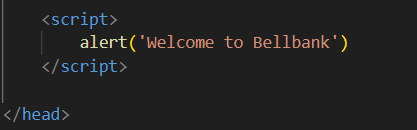


Figure 16: HTML Script

### Exercise

1. The *<base>* element is an HTML element.
2. True
3. False
4. You can add only one CSS property to an HTML element.
5. Yes
6. No
7. Search Engine Optimization (SEO) is important when creating a web page.
8. Agree
9. Disagree

### Styling HTML with CSS

We have started learning about HTML and have covered some of its elements. Now, let’s begin learning about CSS and how to add it to an HTML document. CSS can be added to an HTML document in three ways: inline CSS, internal CSS, and external CSS.

### Inline CSS

An inline CSS is used to apply a unique style to a single HTML element. An inline CSS uses the *style* attribute of an HTML element. The following example below sets the font-size of the element to 24px, and text-color to white:

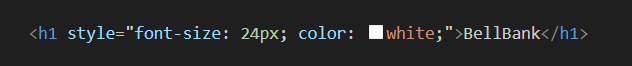


Figure 17: Inline CSS

### Internal CSS

An internal CSS is used to define a style for a single HTML page. An internal CSS is defined in the *<head>* section of an HTML page, within a *<style>* element. The following example below sets the font-size of ALL the <p> elements (on that page) to 12px, and the font-family of **ALL** the *<p>* elements to Arial, if Arial is not installed on the system; it will set the font-family to Helvetica, if Helvetica not found then it will be set to sans-serif.

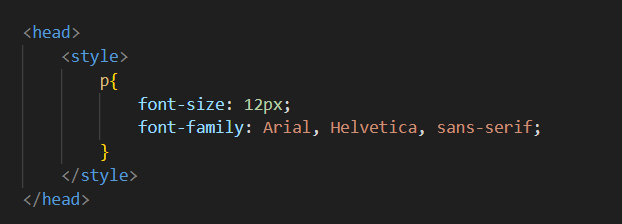


Figure 18: Internal CSS

### External CSS

An external style sheet is used to define the style for many HTML pages. To use an external style sheet, add a link to it in the *<head>* section of each HTML page.

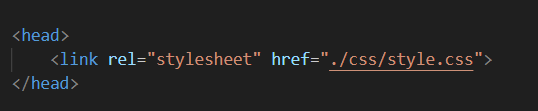


Figure 19: External CSS

The external style sheet can be written in any text editor like Visual Studio Code (VSCode), Sublime, notepad++, etc. The file must not contain any HTML code, and must be saved with a .css extension.

**Footnote**:

**Summary**

* When you want to style a particular HTML element on a page; the Inline CSS is the best approach to use.
* When you want to style a particular Web Page on your Website; the Internal CSS is the best approach to use.
* When you want to style the entire website; the External CSS is the best approach to use.
* With an external style sheet, you can change the look of an entire web site, by changing one file.

**Recommendation**

* When you want to style an HTML element, a web page or the entire website; the best approach is to use the External Styling, because it makes your code much easier to maintain and it is the standard most developers used.

## HTML Body Element

The *<body>* tag defines the document’s body. The *<body>* element contains all the contents of an HTML document, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

### Basic sections of a document

Webpages can and will look pretty different from one another, but they all tend to share similar standard components, unless the page is displaying a fullscreen video or game, is part of some kind of art project, or is just badly structured:

### **H**eader **Element <header>**

The *<header>* element in HTML represents a container for introductory or navigational content, usually located at the top of a webpage or section. This element is often used for branding, primary navigation, and other elements like the site logo, search bar, or introductory text. This usually stays the same from one webpage to another.

### [**N**avigation **B**ar **Element <nav>**](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure#navigation_bar)

The *<nav>* element in HTML is used to define a block of navigation links on a webpage. It helps to semantically organize the links that guide users to different sections of a site or related web pages, making it easier for both users and search engines to understand the structure and navigation options of a website. Having inconsistent navigation on your website will just lead to confused, frustrated users and poor user experience.

### Main Content Element <main>

The *<main>* element in HTML is used to designate the primary content of a webpage, containing information that is directly related to the main purpose of the page. It’s designed to hold the content unique to that page, excluding repetitive elements like headers, footers and navigation bars that are common across multiple pages. This is the one part of the website that definitely will vary from page to page.

### Section Element <section>

The *<section>* element in HTML is used to define a distinct area or block within a webpage that groups together related content. It provides a way to structure content semantically, making it easier for both users and search engines to understand the context of each part of a webpage.

### Sidebar Element <aside>

The *<aside>* element in HTML often contains supplemental information, such as sidebars, quotes, ads, or other related content, but isn’t crucial to the main subject of the page. Some peripheral info, links, quotes, ads, etc.

### Footer Element <footer>

The *<footer>* element in HTML is used to define the footer section of a webpage or a section within it. This section generally contains metadata or related information, like copyright notices, contact details, links to privacy policies, or author information. It’s typically located at the bottom of the page or section and serves as a concluding element. The footer is also sometimes used for SEO purposes, by providing links for quick access to popular content.

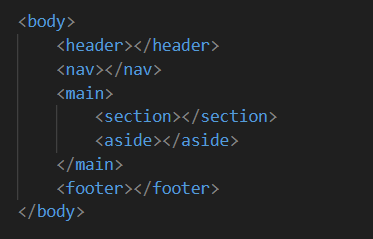
****

Figure 20: HTML5 Semantics Standard

**Footnote**:

There can only be one <body> element in an HTML document.

### HTML Image Element <img>

High resolution images improve the design and appearance of your website. They are a lot of free websites on the internet that you can download high resolution images for free. One of my favorite websites is **pixabay.com**.

In HTML, images are defined with the *<img>* tag. The *<img>* tag is empty, it contains attributes only, and does not have a closing tag.

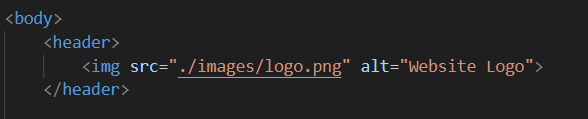


Figure 21: HTML img

The *src* attribute specifies the URL (web address) of the image as in figure above. The *alt* attribute provides an alternate text for an image, if the user for some reason cannot view it (because of slow connection, an error in the src attribute, or if the user uses a screen reader). The value of the alt attribute should describe the image.

We have learned how to add CSS to an HTML document. Now, let’s start learning about CSS properties. In the meantime, let’s learn about the *id* and *class* attributes.

### HTML id and class attributes

The HTML *id* attribute is used to specify a unique id for an HTML element (the value must be unique within the HTML document).

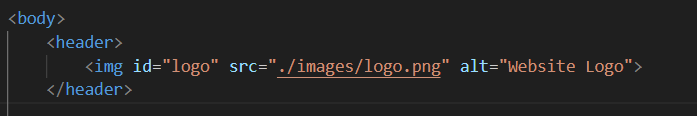


Figure 22: HTML id attribute

The *id* attribute is used in CSS or JavaScript to perform certain tasks for the element with the specific *id* value.

In CSS, to select an element with a specific id, write a hash (#) character, followed by the id value of the element as shown below:

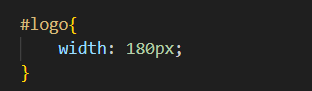


Figure 23: CSS id selector

The HTML *class* attribute is used to define equal styles for elements with the same class name. All HTML elements with the same class attribute will get the same style.

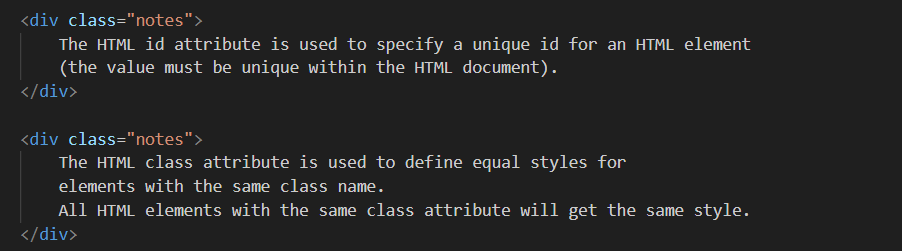


Figure 24: HTML class attribute

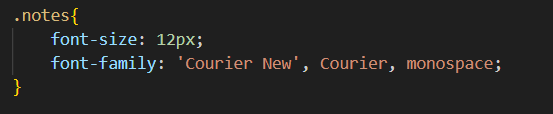


Figure 25: CSS class selector

**Footnote**:

* The image can be of any image type example portable network graphic (PNG) etc.
* A screen reader is a software program that reads the HTML code, converts the text, and allows the user to "listen" to the content. Screen readers are useful for people who are visually impaired or learning disabled.
* An HTML element can have more than one class.

### HTML List Element

In HTML, we have mainly two (2) types of lists. Ordered list and unordered list. HTML lists allow web authors to group a set of related items in lists.

An **ordered** list starts with the *<ol>* tag. Each list item starts with the *<li>* tag. The list items will be marked with numbers by default as shown below:

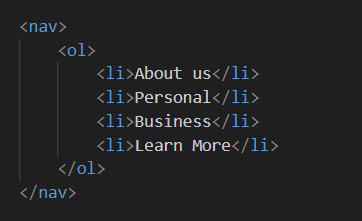


Figure 26: HTML ordered list

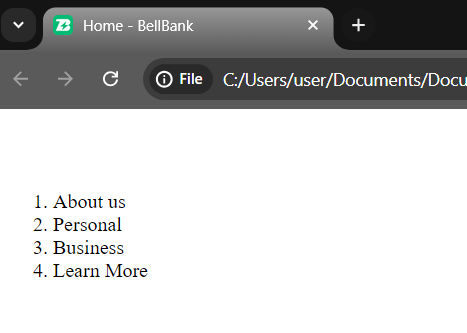


Figure 27: HTML ordered list output

An **unordered** list starts with the *<ul>* tag. Each list item starts with the *<li>* tag. The list items

will be marked with bullets (small black circles) by default as shown below:

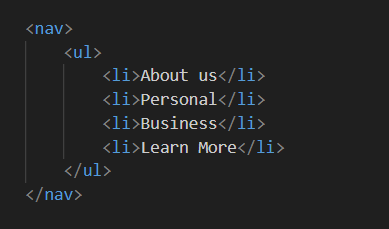


Figure 28: HTML unordered list

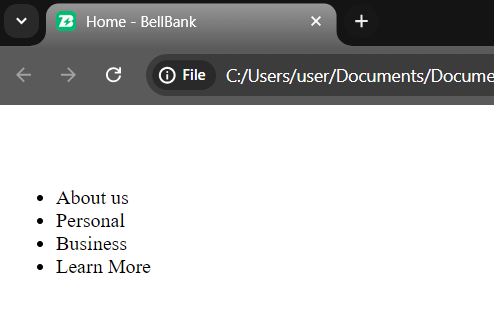


Figure 29: HTML unordered list output

### HTML Link Element

Links are found in nearly all web pages. Links allow users to click their way from page to page. HTML links are hyperlinks.

You can click on a link and jump to another document or to even another website.

When you move the mouse over a link, the mouse arrow will turn into a little hand. The HTML *<a>* tag defines a hyperlink as shown below:



Figure 30: HTML link element

The *href* attribute is the most important attribute of the <a> element is the href attribute, which indicates the link’s destination. The link text is the part that will be visible to the reader. Clicking on the link text, will send the reader to the specified URL address as shown above (HTML links)

The *target* attribute specifies where to open the linked document. The target attribute can have one of the following values:

* \_self - Default. Opens the document in the same window/tab as it was clicked
* \_blank - Opens the document in a new window or tab
* \_parent - Opens the document in the parent frame
* \_top - Opens the document in the full body of the window

**Footnote**:

* You can use a text, an Image or a Button as the link.
* You can link to a document like a Portal Document Format (PDF)

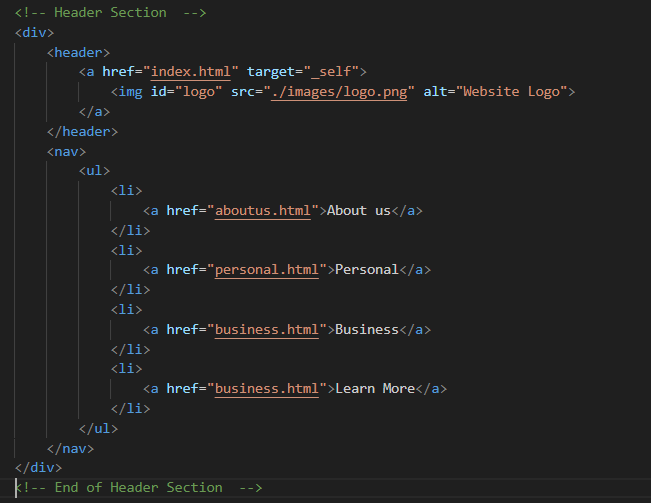


Figure 31: Header section HTML Structure

## CSS Introduction

CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media.

CSS saves a lot of work. It can control the layout of multiple web pages all at once. CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

With CSS, you can design styles for different screen size like tablet, iPad, iPhone, Android, desktop, laptop etc. When the website is visited using a laptop, the display varies when visited on an iPhone or a tablet. It is called responsive web design and we will learn about it later in this book, Smile!

The style definitions are normally saved in external .css files.

### CSS Syntax

A CSS rule-set consists of a selector and a declaration block as shown below:

Declaration



CSS Value

CSS Property

Selector

Figure 32: CSS Syntax

The selector points to the HTML element you want to style. The declaration block contains one or more declarations separated by semicolons. Each declaration includes a CSS property name and a value, separated by a colon. Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

### CSS Background

The CSS background properties are used to add background effects for elements.

The *background-color* property specifies the background color of an element. A color is most often specified by:

* a valid color name - like “black”
* a HEX value - like “#003022”
* an RGB value - like “rgb(0,0,0)”

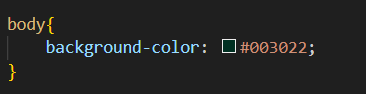


Figure 33: CSS background-color property

Any of the three (3) methods you decide to choose will work just fine. Whether using a valid color name like “black” or using the hexadecimal value like “#003022” or using RGB value like “rgb (0, 0, 0)”

**Note**: RGB stands for Red, Green and Blue.

The *background-image* property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element. The background image for a page can be set as shown below:

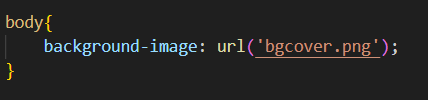


Figure 34: CSS background-image property

By default, the *background-image* property repeats an image both horizontally and vertically. You can alter this behavior. If you want the image to repeat horizontally only, there is a property called *background-repeat* with value of repeat-x. If you want the image to repeat vertically, change the value to repeat-y and if you don’t want the image to repeat, use the value of no-repeat as shown below:

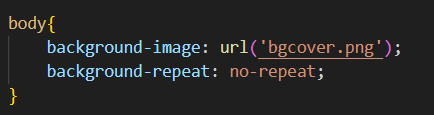


Figure 35: CSS background-repeat property

In some scenarios, the background image might disturb the text on the website or make them not visible. You can alter the position of the image using the *background-position* property. You can get it a value of “right top”, “right bottom”, “left top” or “right top” depending on the position you want to the image to appear as shown below:

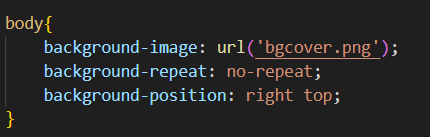


Figure 36: CSS background-position property

To specify that the background image should be fixed (will not scroll with the rest of the page), use the *background-attachment* property as shown below:



Figure 37: CSS background-attachment property

To shorten the code, it is also possible to specify all the background properties in one single property. This is called a shorthand property. The shorthand property for background is background as shown below:

background-position

background-color

background-color



background-repeat

background-attachment

Figure 38: CSS background shorthand property

### CSS Height and Width

The *height* and *width* properties are used to set the height and width of an element.

The *height* and *width* properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.

The *height* and *width* properties may have the following values:

* auto - This is default. The browser calculates the height and width
* length - Defines the height/width in px, cm, etc.
* % - Defines the height/width in percent of the containing block
* initial - Sets the height/width to its default value
* inherit - The height/width will be inherited from its parent value

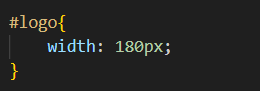


Figure 39: CSS height and width property

### CSS Display

The display property is the most important CSS property for controlling layout. The display property specifies if/how an element is displayed.

Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline. Hiding an element can be done by setting the display property to none. The element will be hidden, and the page will be displayed as if the element is not there:

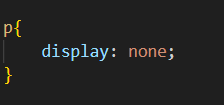


Figure 40: CSS display property

### CSS Flexbox

The *flex* property sets the flexible length on flexible items. The flex container becomes flexible by setting the display property to flex.

The flex container properties are:

* flex-direction
* flex-wrap
* flex-flow
* justify-content
* align-items
* align-content

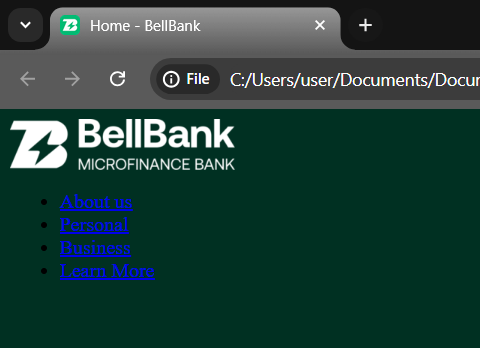


Figure 41: Header Section

Let’s style the header section and use Flexbox to make the logo and navigation appear on the same row. We will start by adding an *id* attribute to the *div* element that contained the *header* and *nav* element as shown below:



Figure 42: adding id attribute

In our style.css file, we will use the *id* selector to style the *div* as shown below:

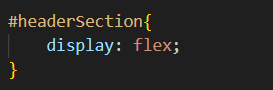


Figure 43: CSS flex display property

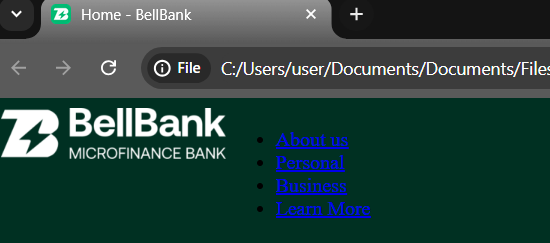


Figure 44: Header section after add flex display property

The flex container can have the following properties, depending on the design requirements:

* flex-direction
* flex-wrap
* flex-flow
* justify-content
* align-items
* align-content

These properties can have different values. For example, the align-items property can take values such as *center*, *flex-start*, *flex-end*, *baseline*, or *stretch*. Similarly, the justify-content property can have values like *center*, *flex-start*, *flex-end*, *space-around*, or *space-between*.

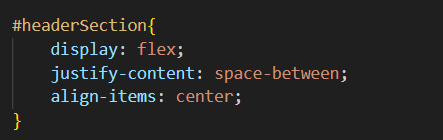


Figure 45: CSS justify-content and align-items property

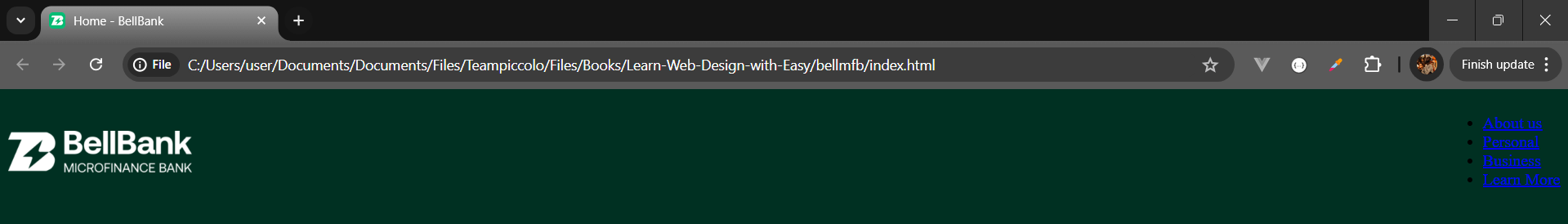


Figure 46: Header Section Display

### CSS List

We have already learned in HTML that there are two main types of lists: unordered lists (*<ul>*) - the list items are marked with bullets and ordered lists (*<ol>*) - the list items are marked with numbers or letters.

The CSS list properties allow you to:

* Set different list item markers for ordered lists
* Set different list item markers for unordered lists
* Set an image as the list item marker
* Add background colors to lists and list items

The list-style-type property specifies the type of list item marker. This property has some values like: *circle*, *decimal*, *none*, *square*, *upper-roman*, *lower-alpha*, or *none* etc.

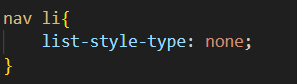


Figure 47: CSS list-style-type property

The list-style-type property with the value none will remove the default style of HTML list items.

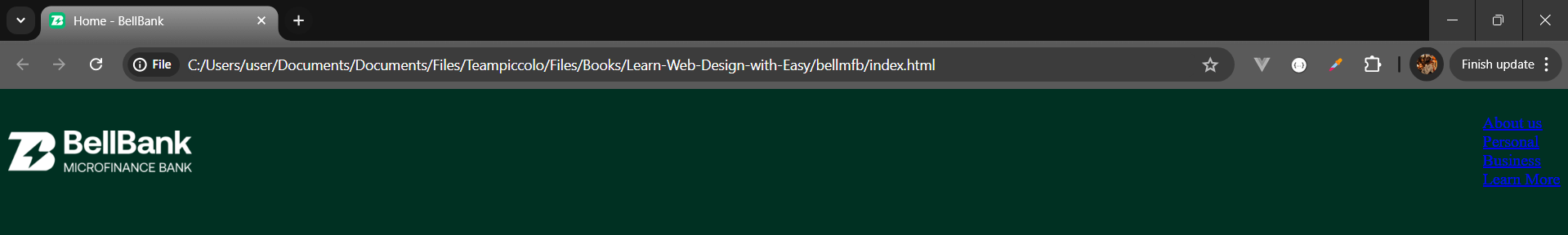


Figure 48: List-style-type output

Let’s work on the navigation menu using what we have learned from HTML and CSS.

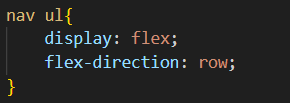


Figure 49: CSS flex-direction property

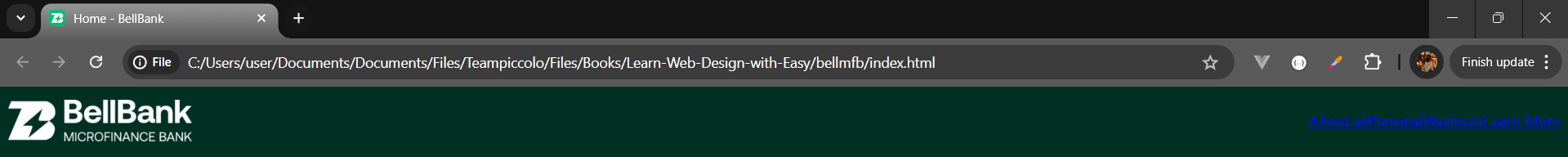


Figure 50: CSS flex-decoration property output

### CSS Colors

Colors are specified using predefined color names, or RGB, HEX, HSL, RGBA, HSLA values. Let’s change the font color of the navigation menu using the CSS color property as shown below:

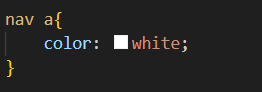


Figure 51: CSS color property

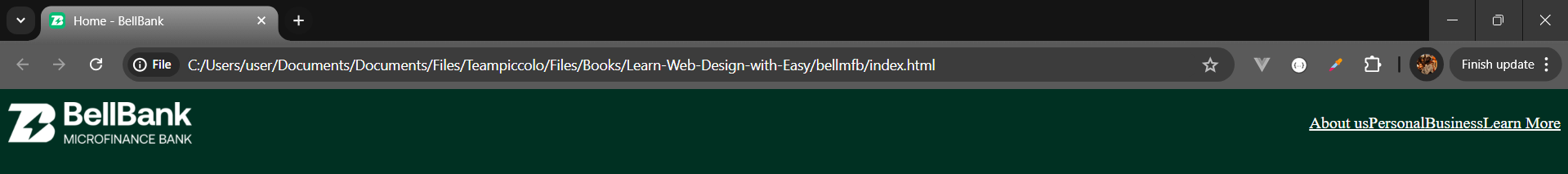


Figure 52: CSS color property output

To remove the underline in the navigation menu, the *text-decoration* property is used. The text-decoration property is mostly used to remove underlines from links.

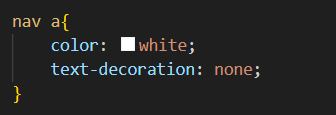


Figure 53: CSS text-decoration property

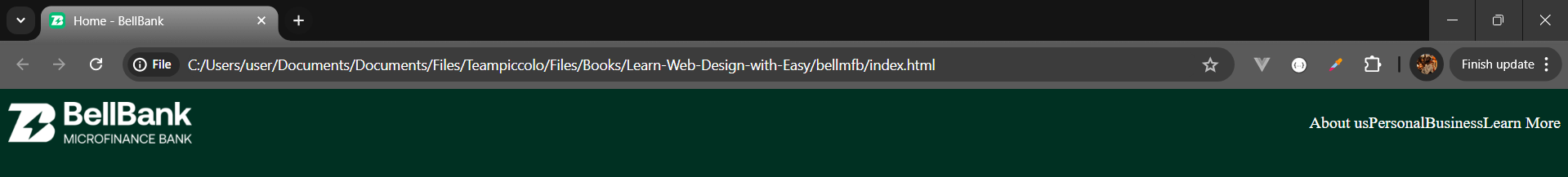


Figure 54: CSS text-decoration property output

### CSS Margin

The CSS margin properties are used to create space around elements, outside of any defined borders.

With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

CSS has properties for specifying the margin for each side of an element: margin-top, margin-right, margin-bottom, margin-left.

All the margin properties can have the following values:

* auto - the browser calculates the margin
* length - specifies a margin in px, pt, cm, etc.
* % - specifies a margin in % of the width of the containing element
* inherit - specifies that the margin should be inherited from the parent element

**Note**: Negative values are allowed.

Let’s add some margin to our navigation menu using the CSS margin property, as shown below:

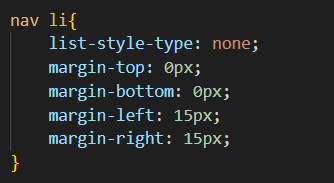


Figure 55: CSS margin property

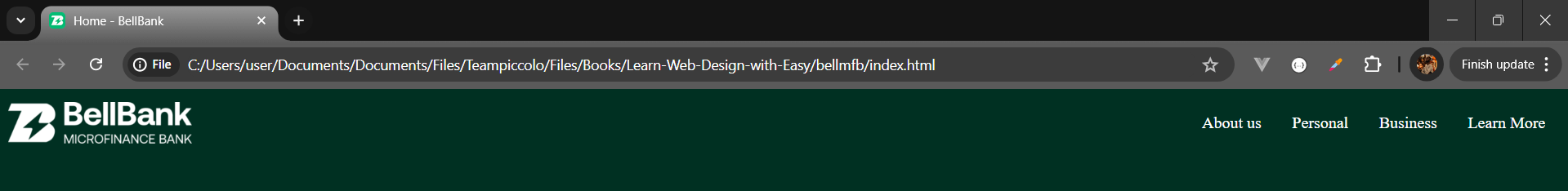
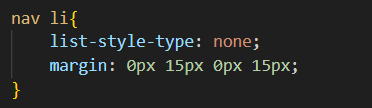


Figure 56: CSS margin property output

You can shorten the margin code above; it is possible to specify all the margin properties in one property using the margin property as shown below:



margin-top

margin-right

margin-bottom

margin-left

Figure 57: CSS margin property II

If the margin property has three values: margin: 0px 15px 0px; top-margin is 0px, right and left margins are 15px, bottom-margin is 0px.

If the margin property has two values: margin: 0px 10px; top and bottom margins are 0px, right and left margins are 15px.

If the margin property has one value: margin: 10px; all four margins are 10px.

You can set the margin property to auto to horizontally center the element within its container. The element will then take up the specified width, and the remaining space will be split equally between the left and right margins.

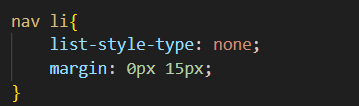


Figure 58: CSS margin property III

Let’s add some margin to the *headerSection* to make the header look better!

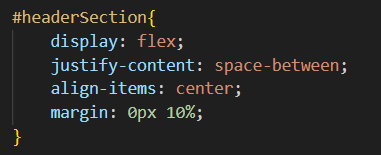


Figure 59: CSS margin property IV

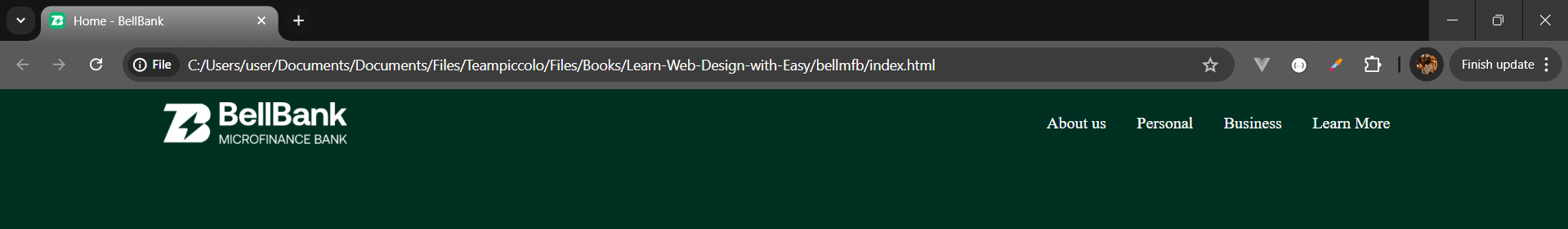


Figure 60: headerSection much better!

### CSS Padding

The CSS padding properties are used to generate space around an element’s content, inside of any defined borders.

With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

CSS has properties for specifying the padding for each side of an element: padding-top, padding-right, padding-bottom, padding-left.

All the padding properties can have the following values:

* length - specifies a padding in px, pt, cm, etc.
* % - specifies a padding in % of the width of the containing element
* inherit - specifies that the padding should be inherited from the parent element

Note: Negative values are not allowed.

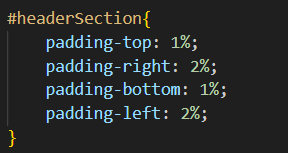
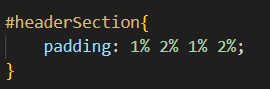


Figure 61: CSS padding property

You can shorten the code above; it is possible to specify all the padding properties in one property using the padding property as shown below:



padding-top

padding-right

padding-bottom

padding-left

Figure 62: CSS padding property II

If the padding property has four values: padding: 1% 2% 1% 2%; top-padding is 1%, right-padding is 2%, bottom-padding is 1%, left-padding is 2%.

If the padding property has three values: padding: 1% 2% 1%; top-padding is 1%, right and left paddings are 2%, bottom padding is 1%.

If the padding property has two values: padding: 1% 2%; top and bottom paddings are 1%, right and left paddings are 2%.

If the padding property has one value: padding: 2%; all four paddings are 2%.

Let’s update the *headerSection* CSS code and add the padding property to the *headerSection*.

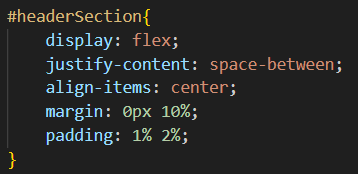


Figure 63: CSS padding property III

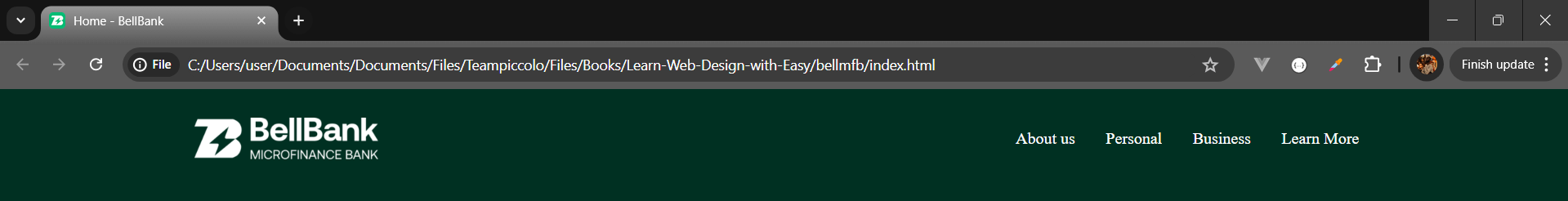


Figure 64: headerSection much better! II

### CSS Border

The CSS border properties allow you to specify the style, width, and color of an element’s border.

The *border-style* property specifies what kind of border to display. The following values are allowed:

* dotted - Defines a dotted border.
* dashed - Defines a dashed border.
* solid - Defines a solid border.
* double - Defines a double border
* groove - Defines a 3D grooved border. The effect depends on the border-color value.
* ridge - Defines a 3D ridged border. The effect depends on the border-color value.
* inset - Defines a 3D inset border. The effect depends on the border-color value.
* outset - Defines a 3D outset border. The effect depends on the border-color value.
* none - Defines no border.
* hidden - Defines a hidden border.

The *border-style* property can have from one to four values (for the top border, right border, bottom border, and the left border) example border-top, border-bottom, border-left, border-right.

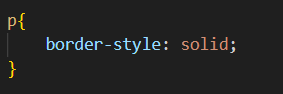


Figure 65: CSS border-style property

**NOTE**: None of the other CSS border properties we will learn below will have any effect unless the border-style property is set. Make sure to set it first before using any other border-related properties.

The *border-width* property specifies the width of the four borders. The width can be set as a specific size (in px, pt, cm, em, etc.) or by using one of the three pre-defined values: thin, medium, or thick. The border-width property can have from one to four values (for the top border, right border, bottom border, and the left border).

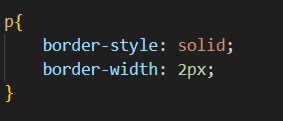


Figure 66: CSS border-width property

The border-color property is used to set the color of the four borders. The color can be set by:

* name - specify a color name, like “black”
* Hex - specify a hex value, like “#000000”
* RGB - specify a RGB value, like “rgb(0,0,0)”
* transparent - the border-color property can have from one to four values (for the top border, right border, bottom border, and the left border). If border-color is not set, it inherits the color of the element.

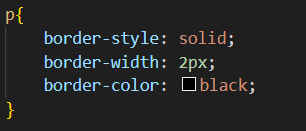
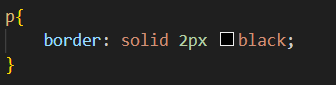


Figure 67: CSS border-color property

You can shorten the code above on a single line using the border property as shown below:



border-style

border-width

border-color

Figure 68: CSS border property

Let’s add a border-bottom property to the *headerSection*. We will use the CSS border property to achieve this, as shown below:

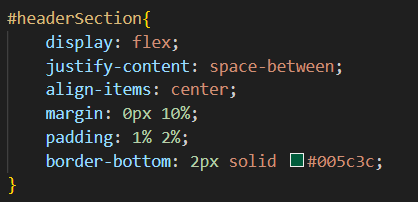


Figure 69: CSS border-bottom property

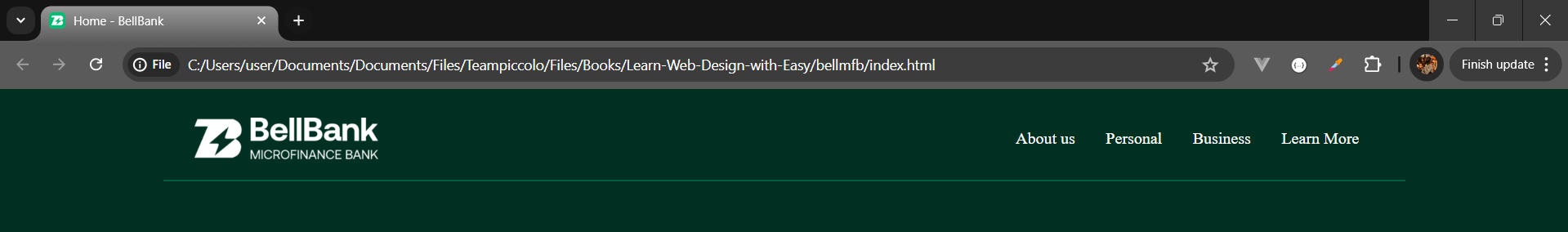


Figure 70: headerSection much better! III

### CSS Font

Choosing the right font has a huge impact on how the readers experience a website.

The right font can create a strong identity for your brand.

Using a font that is easy to read is important. The font adds value to your text. It is also important to choose the correct color and text size for the font.

The CSS font properties define the font family, boldness, size, and the style of a text. With CSS, you can alter the font family of your web page for make your website much attractive and user friendly.

The font family of a text is set with the *font-family* property. The font-family property should hold several font names as a “fallback” system. If the browser does not support the first font, it tries the next font, and so on.

**Note**: If the name of a font family is more than one word, it must be in quotation marks, like: "Times New Roman”.

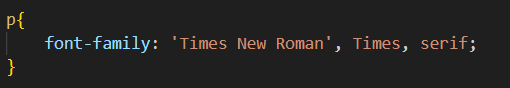


Figure 71: CSS font-family property

The *font-style* property is mostly used to specify italic text. This property has three values:

* normal - The text is shown normally,
* italic - The text is shown in italics,
* oblique - The text is “leaning” (oblique is very similar to italic, but less supported).

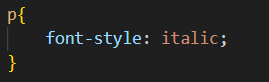


Figure 72: CSS font-style property

The *font-size* property sets the size of the text. Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.

**Note**: Always use the proper HTML tags, like *<h1>* - *<h6>* for headings and *<p>* for paragraphs. You can use cm, em, px or % as the unit of measurement.

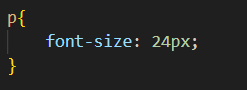


Figure 73: CSS font-size property

**Note**: Many developers use em instead of pixels. The em size unit is recommended by the W3C.

The *font-weight* property specifies the weight of a font: This property has some values:

* bold,
* bolder and
* normal.

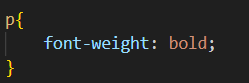


Figure 74: CSS font-weight property

The *font-variant* property specifies whether or not a text should be displayed in a small-caps font. In a small-caps font, all lowercase letters are converted to uppercase letters. However, the converted uppercase letters appear in a smaller font size than the original uppercase letters in the text.

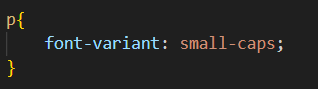


Figure 75: CSS font-variant property

### CSS Google Font

If you do not want to use any of the standard fonts in HTML, you can use Google Fonts. Google Fonts are free to use, and have more than 1000 fonts to choose from.

Let’s try to use “DM Sans” font type on our website.

1. Go to <https://fonts.google.com/>
2. Use the search box and type DM Sans as shown below:

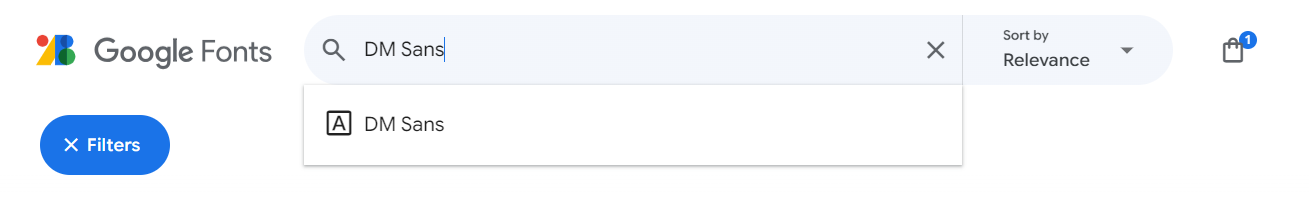


Figure 76: CSS Google Font

1. Click on “Get Font” button as shown below:

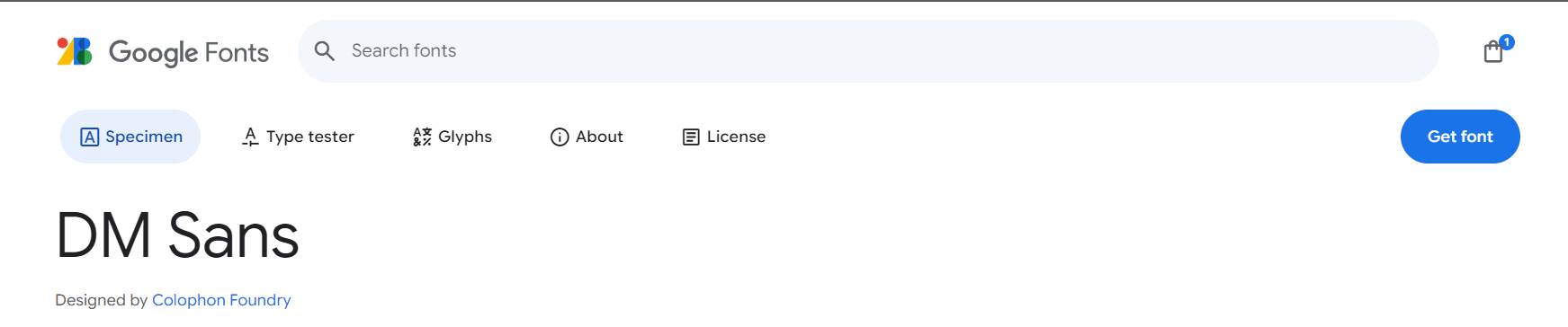


Figure 77: CSS Google Font II

1. Click on “Get embed Code” button as shown below:

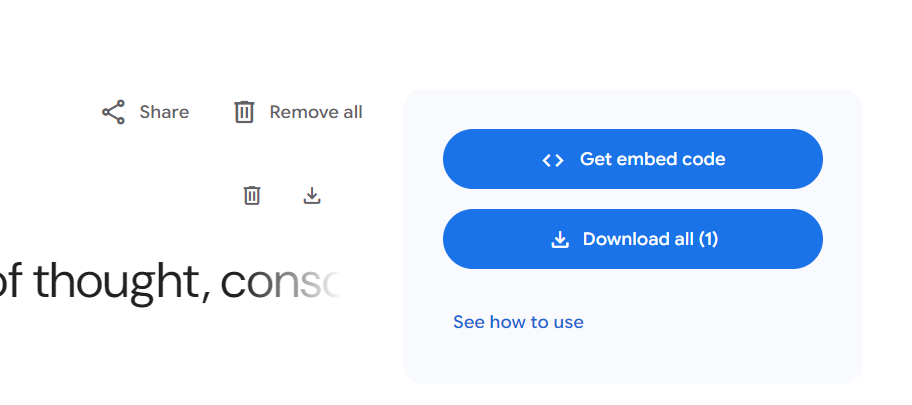


Figure 78: CSS Google Font III

1. Click on “Copy Code” button as shown below:

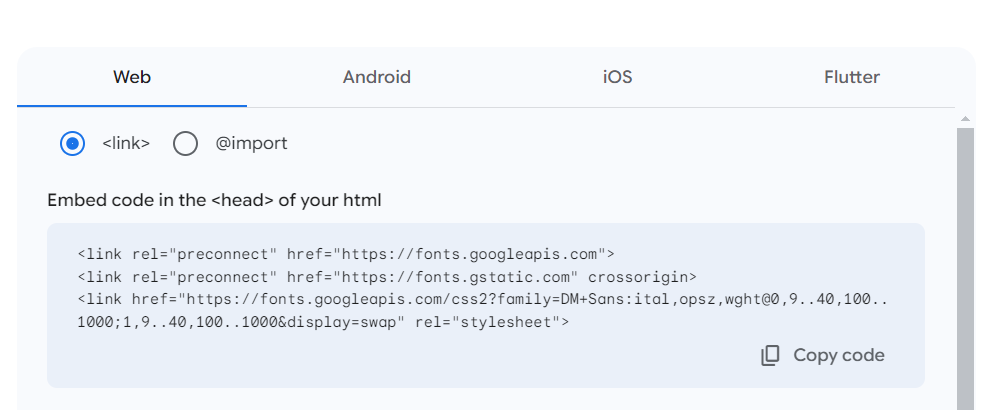


Figure 79: CSS Google Font IV

1. Paste the code inside the *head* element on the index.html file as shown below:

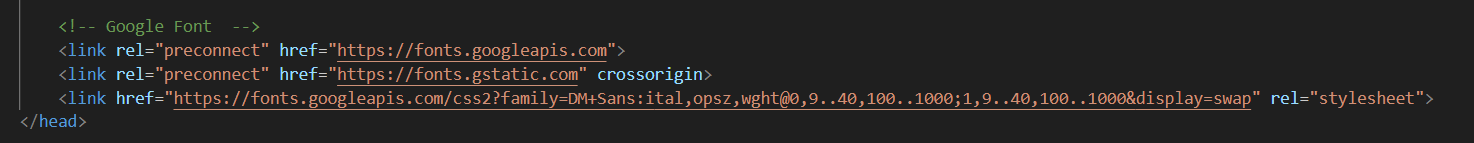


Figure 80: CSS Google Font V

1. Add the font-family property to the body element in the style.css file, as shown below:

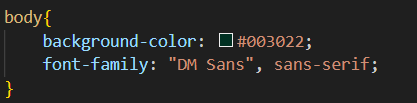


Figure 81: CSS Google Font VI

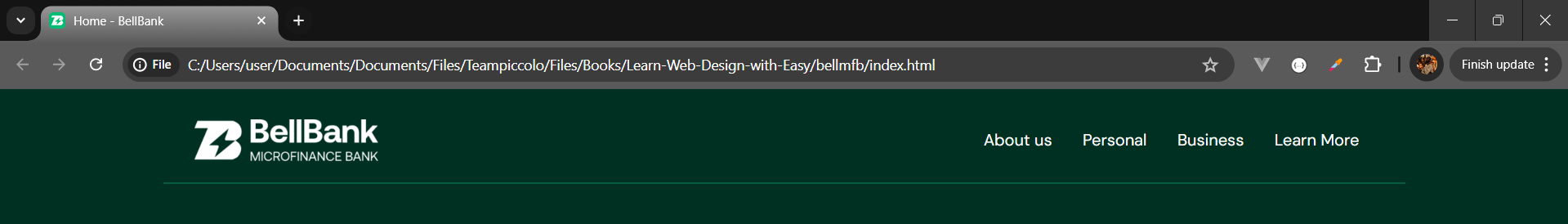


Figure 82: headerSection much better! IV

**Note**: sans-serif is a fallback font-family. If DM Sans is not available, the system will default to sans-serif.

### CSS Text

In this section, we will learn about some CSS properties used for text formatting. CSS properties like *color*, *text-align*, *text-decoration*, *text-transform,* *text-indent*, *letter-spacing*, *line-height*, *direction* and *word-spacing*.

The *color* property is used to set the color of the text. With CSS, a color is most often specified by: a color name - like “red”, a HEX value - like “#ff0000”, an RGB value - like “rgb(255,0,0)”.

**Note**:

* The default text color for a page is defined in the body selector.
* For W3C compliant CSS: If you define the color property, you must also define the background-color property.

The *text-align* property is used to set the horizontal alignment of a text. A text can be left or right aligned, centered, or justified. The value of text-align property includes: *left*, *right*, *center* and *justify*. If you have used a word processing software like Microsoft word, the text-align property works exactly as in Microsoft word.

The *text-decoration* property is used to set or remove decorations from text. The value text-decoration: none; is often used to remove underlines from links. The value of text-decoration property includes: *underline*, *line-through*, *over-line*.

**Note**: It is not recommended to underline text that is not a link, as this often confuses the reader.

The *text-transform* property is used to specify uppercase and lowercase letters in a text. It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word. The value of text-transform property includes: *capitalize*, *lowercase*, *none*, *uppercase*, *initial*, *inherit*, *unset*.

The *text-indent* property is used to specify the indentation of the first line of a text. You use the unit of measurement like px, cm, em, etc. when specifying the value of the property: text-indent: 30px;

The *letter-spacing* property is used to specify the space between the characters in a text. You use the unit of measurement like px, cm, em, etc. when specifying the value of the property: letter-spacing: 5px;

The *line-height* property is used to specify the space between lines. Using the line-height, you don’t specify the unit of measurement like cm, px or em: line-height: 1.5;

The *direction* property is used to change the text direction of an element. Some of the values you can use on the direction property includes: rtl and ltr.

The *word-spacing* property is used to specify the space between the words in a text. You use the unit of measurement like px, cm, em, etc. when specifying the value of the property: word-spacing: 3px;

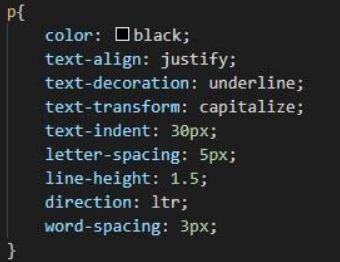


Figure 83: CSS Text property

Let’s continue with the website design. Having completed the header section, we will now start working on the main content of the website.

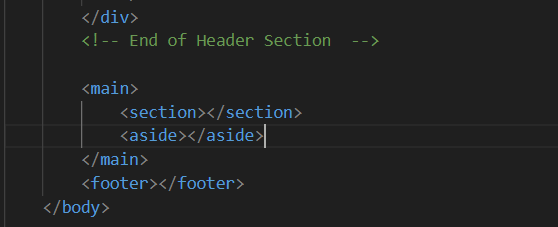


Figure 84: main element

Create two divs element inside the *section* element within the *main* element as shown below:

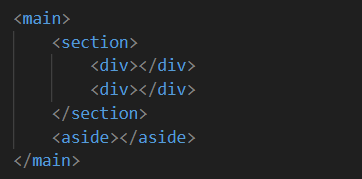


Figure 85: section element

Inside the first, create two divs and write the content as shown below:

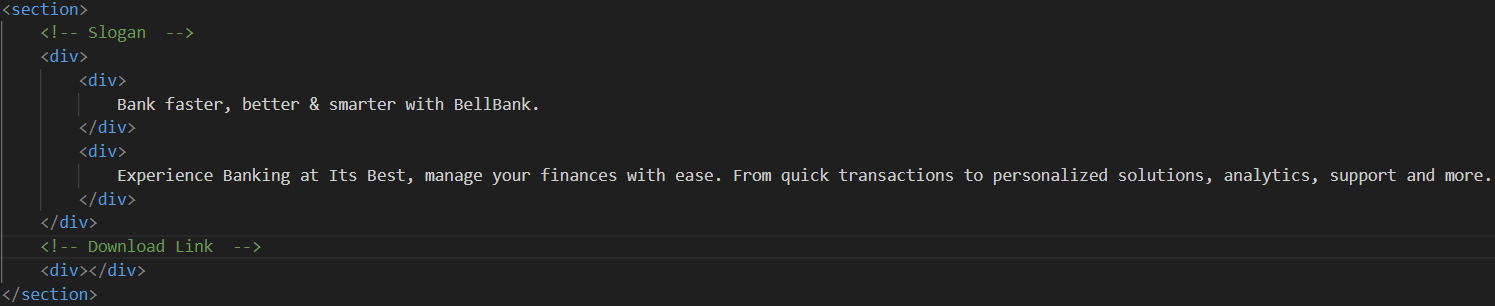


Figure 86: working on slogan div

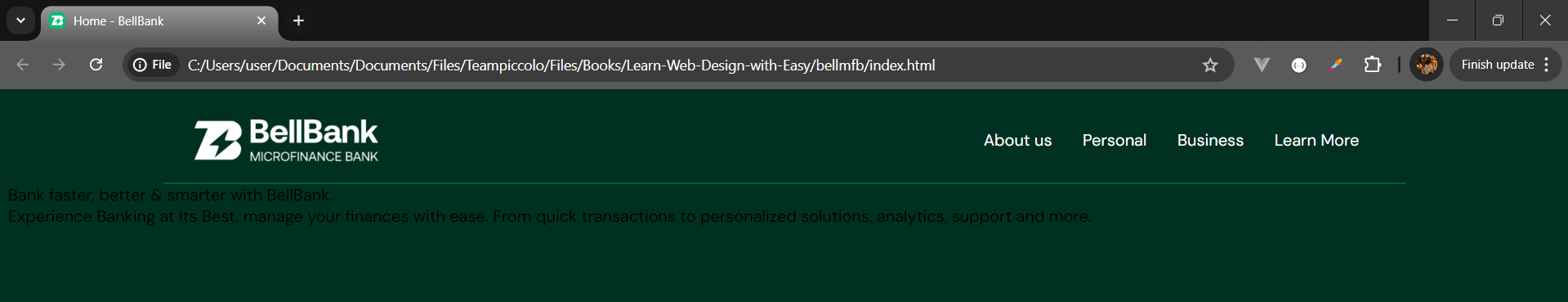


Figure 87: Slogan div output

Let’s add the download-link div content. Below is the code snippet:

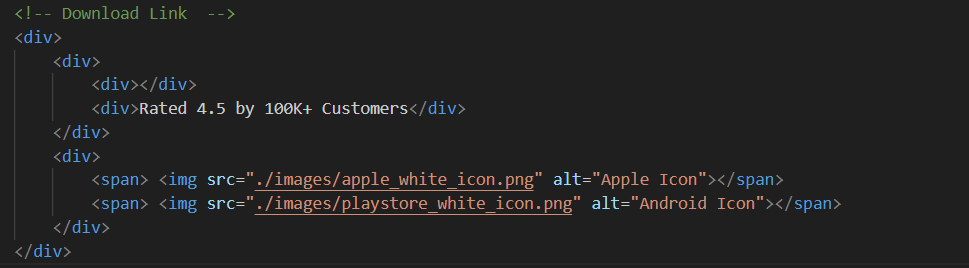


Figure 88: Download Link div content

### Adding Icons with Font Awesome

You can add icons to your website using various method like using bootstrap icons, flaticons, fontawesome, etc.

To embed fontawesome, follow the steps below:

1. Go to google.com and search “cdn font awesome” as shown below:

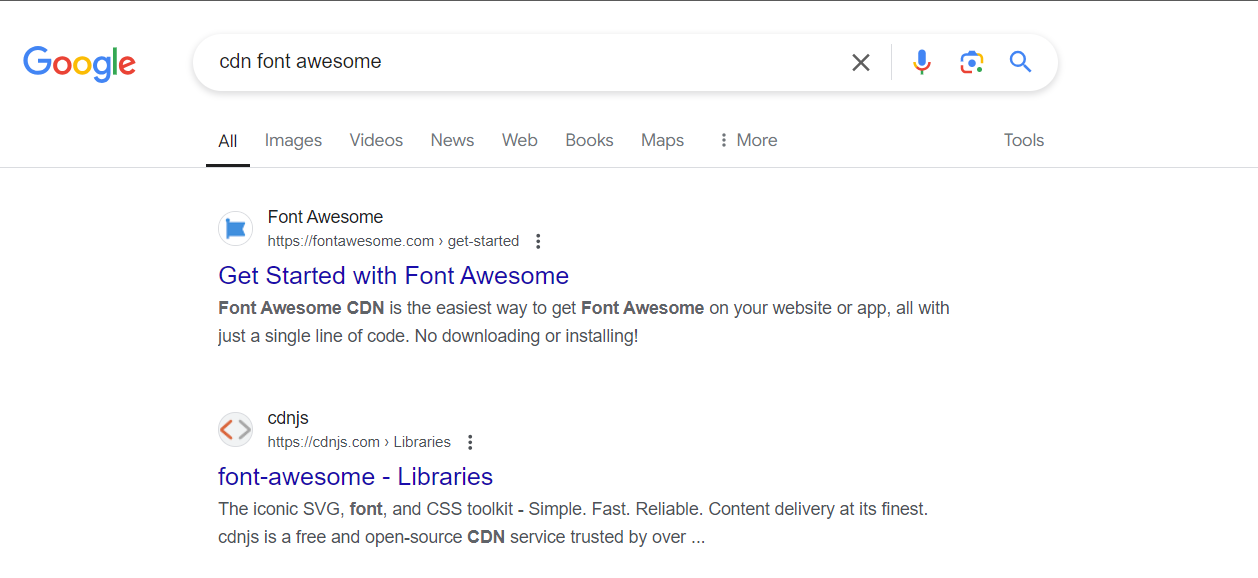
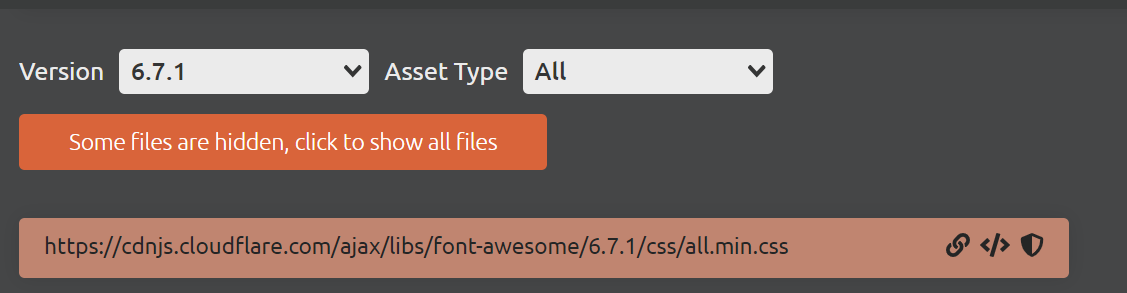


Figure 89: adding fontawesome icons

1. Click on the “**cdnjs”** as shown below:



Click on the tag

1. In the index.html file, within the *<head>* element, paste the code shown below:

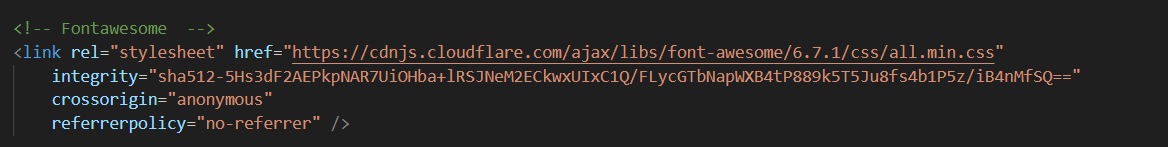


Figure 90: embedding fontawesome link

1. Go to [www.fontawesome.com](http://www.fontawesome.com/)

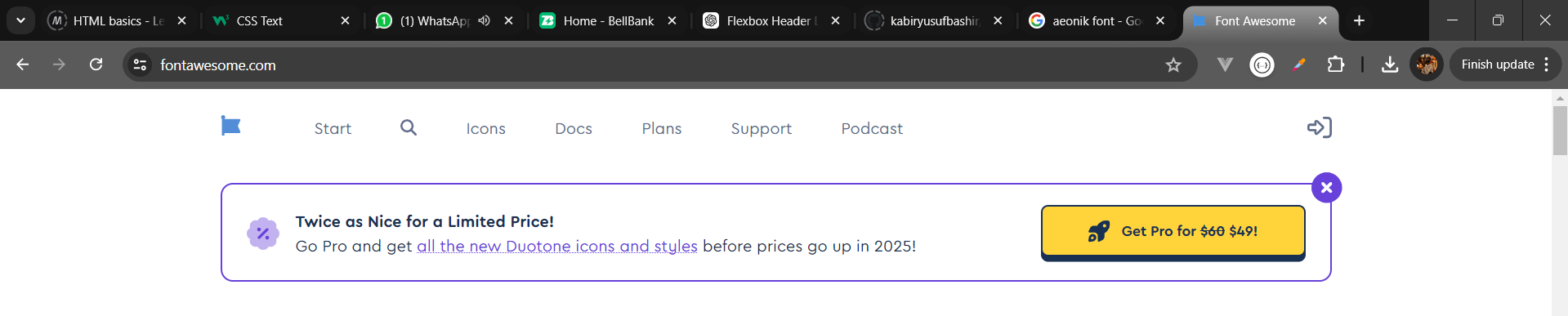


Figure 91: fontawesome website

1. Click on the search button and search for “**star**” as shown below:

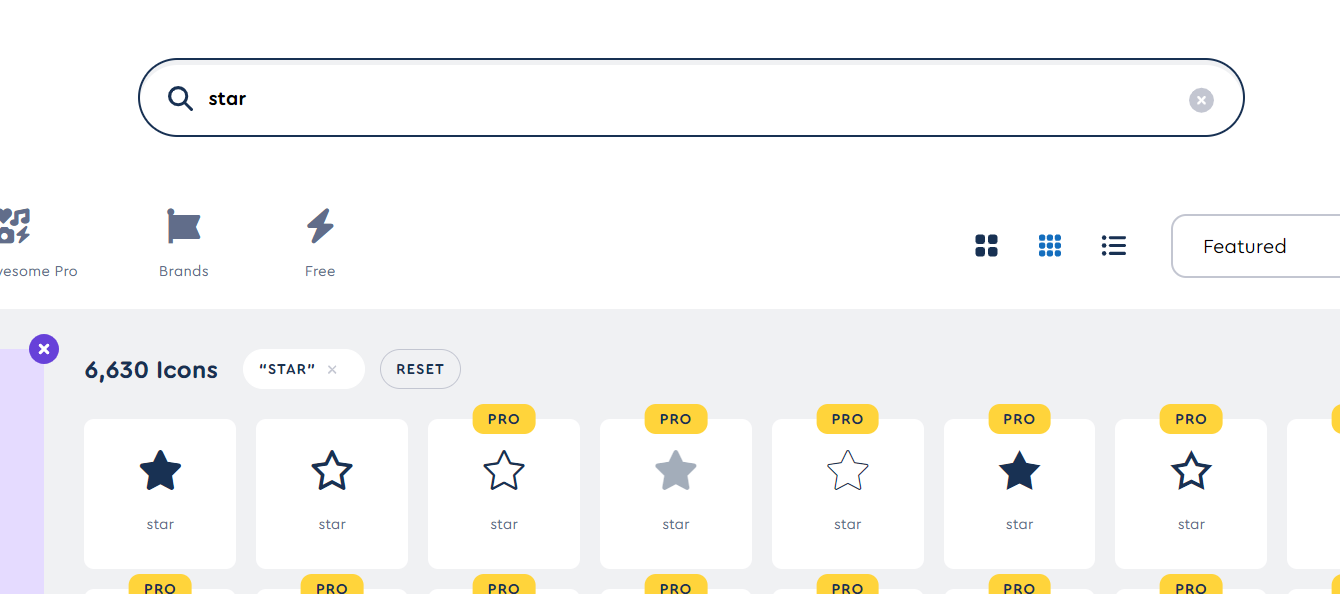


Figure 92: searching on fontawesome

1. Click on the star icon. We will use the first item from the search results.

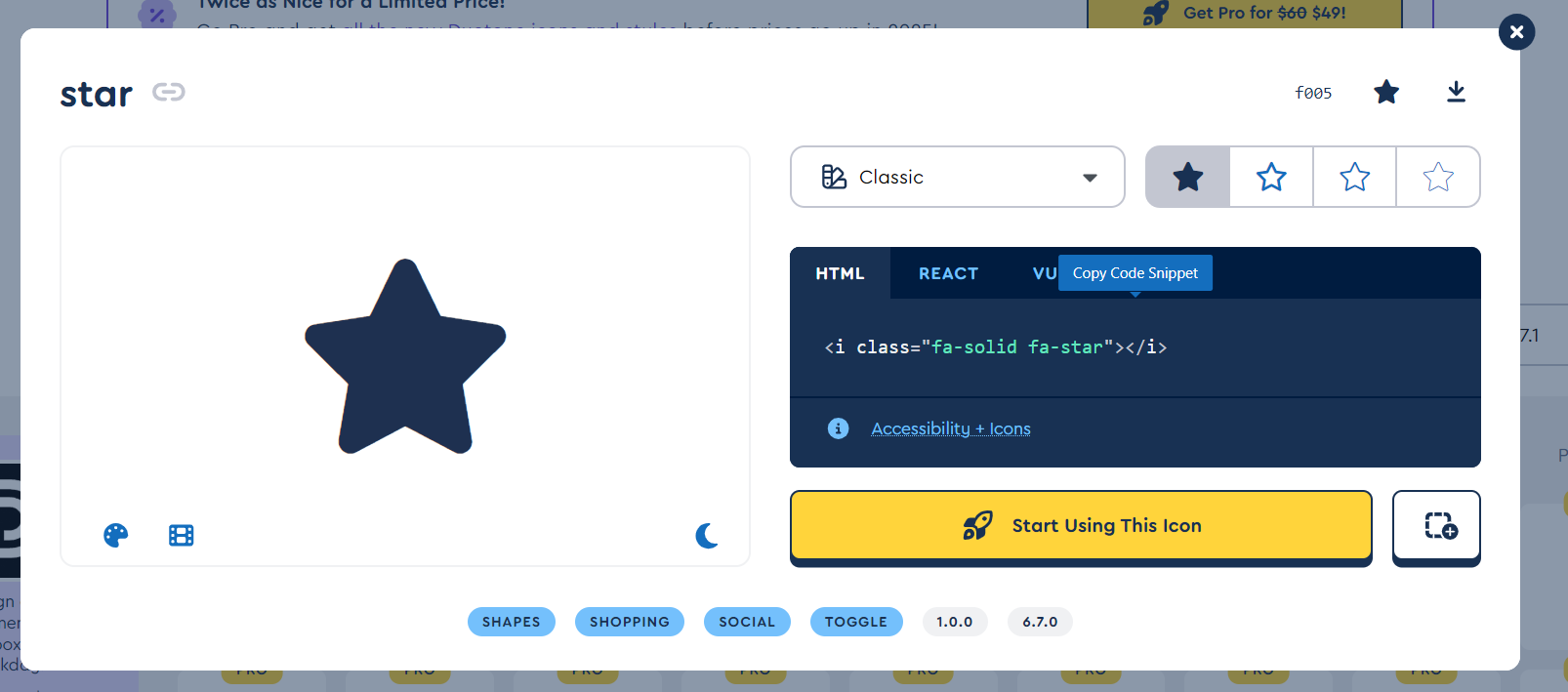


Figure 93: searching on fontawesome II

1. Copy the code snippet and paste it into the index.html file, within the download link section, as shown below:

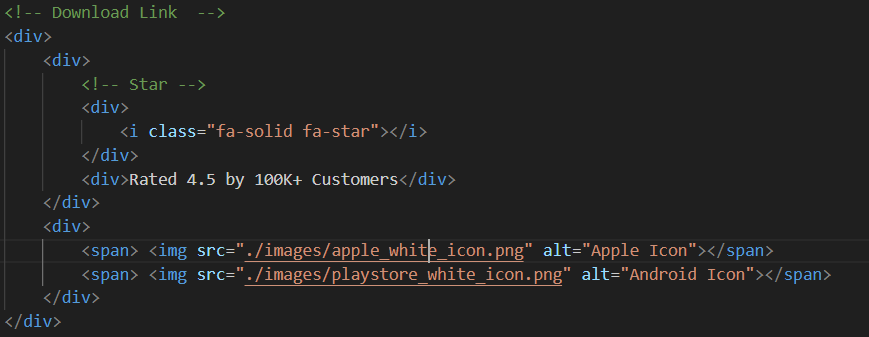


Figure 94: Adding the star icons using fontawesome

1. We will be needing five (5) stars, duplicate the star icon as shown below:

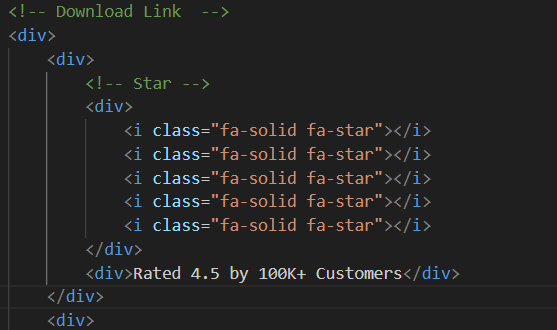


Figure 95: Adding the star icons using fontawesome II

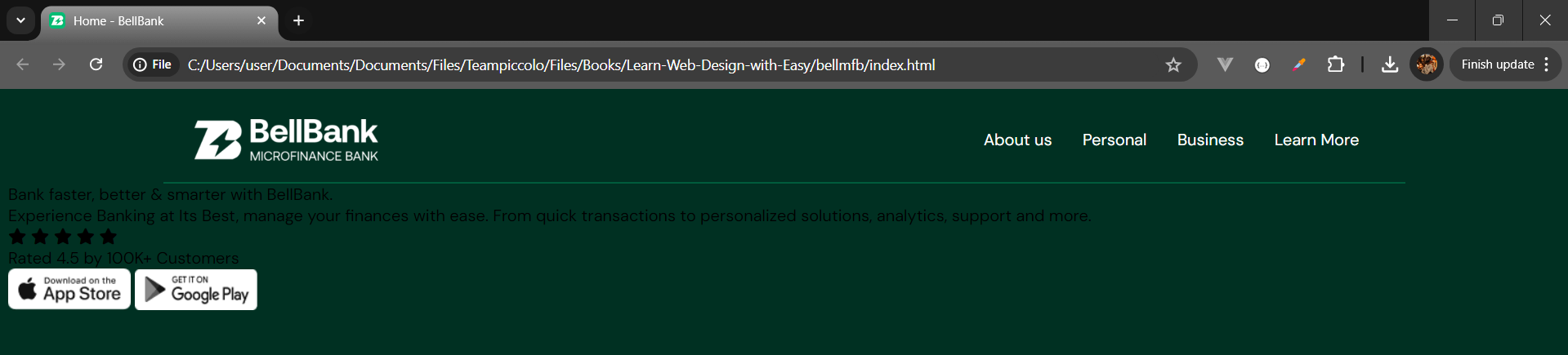


Figure 96: Slogan and download link section

The section structure is complete, but the styling needs improvement. Let’s apply what we’ve learned in CSS to enhance the website’s user interface and user experience (UI/UX).

1. Add an id attribute to the *<section>* element which contains the slogan and download link divs as shown below:

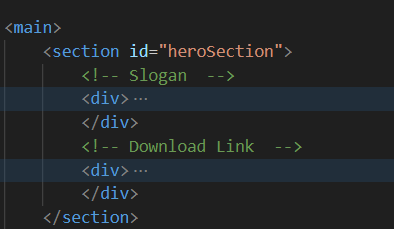


Figure 97: adding id attribute to the section element

1. In the style.css file, use the ID selector to style the hero section, as shown below:

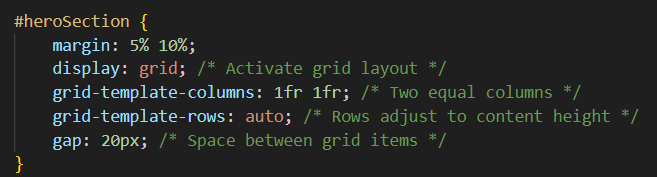


Figure 98: CSS display property (grid)

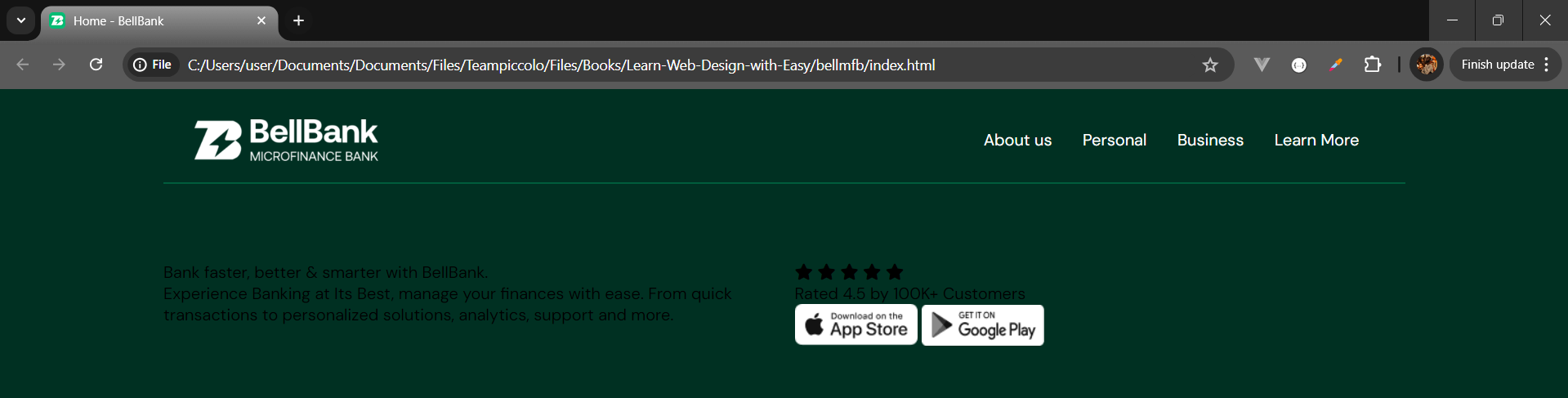


Figure 99: CSS display property (grid) II

**Read:** you can learn more about CSS grid on this link [www.w3schools.com/css/css\_grid.asp](http://www.w3schools.com/css/css_grid.asp)

1. Add an ID attribute to the slogan div as shown below:

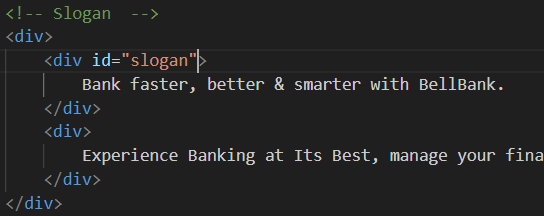


Figure 100: Styling Slogan div

1. Add the following CSS properties to the div as shown below:

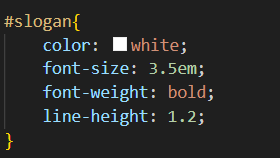


Figure 101: Slogan stylesheet

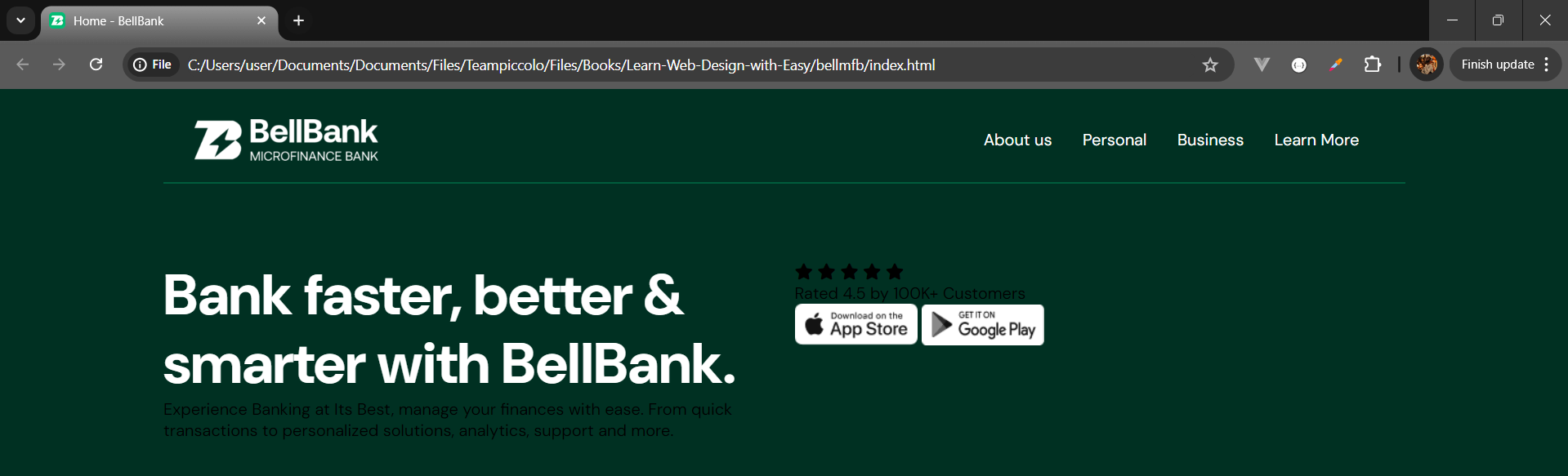


Figure 102: Slogan div output

# References

1. <https://developer.mozilla.org/enUS/docs/Learn/Common_questions/Web_mechanics/How_does_the_Internet_work>
2. <https://developer.mozilla.org/enUS/docs/Learn/Common_questions/Web_mechanics/Pages_sites_servers_and_search_engines>
3. <https://developer.mozilla.org/enUS/docs/Learn/Common_questions/Web_mechanics/What_is_a_domain_name>
4. <https://developer.mozilla.org/enUS/docs/Learn/Getting_started_with_the_web/HTML_basics>
5. <https://developer.mozilla.org/enUS/docs/Learn/Getting_started_with_the_web/CSS_basics>
6. <https://www.w3schools.com/html/html_elements.asp>
7. <https://www.w3schools.com/html/html_head.asp>
8. <https://www.w3schools.com/html/html_attributes.asp>
9. <https://www.w3schools.com/html/html_favicon.asp>
10. <https://www.w3schools.com/tags/tag_body.asp>
11. <https://developer.mozilla.org/enUS/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure>
12. <https://www.w3schools.com/css/css_background.asp>
13. <https://www.w3schools.com/css/css_dimension.asp>
14. <https://www.w3schools.com/css/css_display_visibility.asp>
15. <https://www.w3schools.com/css/css3_flexbox_container.asp>
16. <https://www.w3schools.com/css/css_link.asp>
17. <https://www.w3schools.com/css/css_list.asp>
18. <https://www.w3schools.com/css/css_selectors.asp>
19. <https://www.w3schools.com/css/css_colors.asp>
20. <https://www.w3schools.com/css/css_margin.asp>
21. <https://www.w3schools.com/css/css_padding.asp>
22. <https://www.w3schools.com/css/css_border.asp>
23. <https://www.w3schools.com/css/css_font.asp>
24. <https://www.w3schools.com/css/css_font_google.asp>
25. <https://www.w3schools.com/css/css_text.asp>
26. <https://www.w3schools.com/css/css_grid.asp>