HTML

Hypertext markup language

OGTECH NETWORK'S LTD

If you can picture it, we can capture it…….

www.ogtechnetworks.com

Like us on facebook

https://facebook.com/ogtechnetwork

Follow us on Instagram

https://instagram.com/ogtechnetworks

About the Tutorial

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages.

HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

Audience

This tutorial is designed for the aspiring Web Designers and Developers with a need to understand the HTML in enough detail along with its simple overview, and practical examples. This tutorial will give you enough ingredients to start with HTML from where you can take yourself at higher level of expertise.

1. HTML - OVERVIEW

HTML stands for Hypertext Markup Language, and it is the most widely used language to write Web Pages.

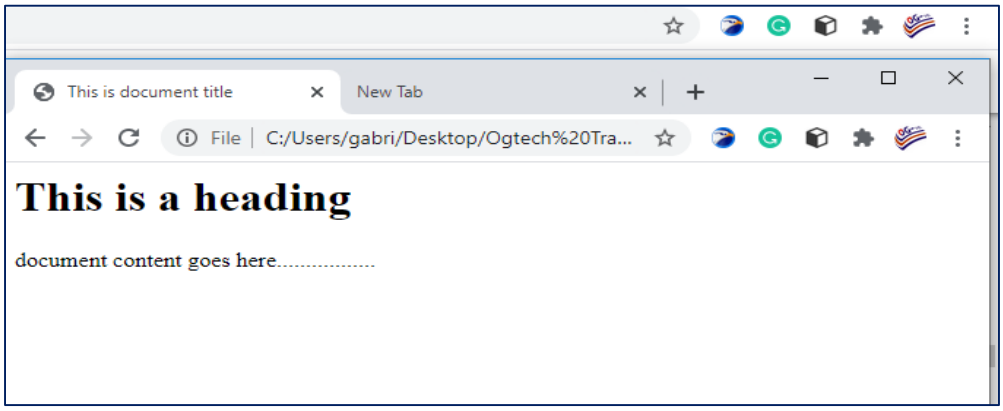
- Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
- As its name suggests, HTML is a Markup Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers. Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

Basic HTML Document

In its simplest form, following is an example of an HTML document:

```
<html>
  <head>
    <title>This is document title</title>
  </head>

  <body>
    <h1>This is a heading</h1>
    <p>document content goes here.....</p>
  </body>
</html>
```



HTML Tags

As told earlier, HTML is a markup language and makes use of various tags to format the content. These tags are enclosed within angle braces **<Tag Name>**. Except few tags, most of the tags have their corresponding closing tags. For example, **<htmL>** has its closing tag **</htmL>** and **<body>** tag has its closing tag **</body>** tag etc. Above example of HTML document uses the following tags:

Tag	Description
<!DOCTYPE…>	This tag defines the document type and HTML version.
<html>	This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>....</head> and document body which is represented by <body>……</body> tags.
<head>	This tag represents the document's header which can keep other HTML tags like <title>, <link>, etc
<title>	The <title> tag is used inside the <head> tag to mention the document title.
<body>	This tag represents the document's body which keeps other HTML tags like
<h1>	This tag represents the heading.
<p>	This tag represents a paragraph.

To learn HTML, you will need to study various tags and understand how they behave, while formatting a textual document. Learning HTML is simple as users have to learn the usage of different tags in order to format the text or images to make a beautiful webpage. World Wide Web Consortium (W3C) recommends to use lowercase tags starting from HTML 4.

The <!DOCTYPE> Declaration

The declaration tag is used by the web browser to understand the version of the HTML used in the document. Current version of HTML is 5 and it makes use of the following declaration:

```
<!DOCTYPE html>
```

There are many other declaration types which can be used in HTML document depending on what version of HTML is being used. We will see more details on this while discussing **<!DOCTYPE…>** tag along with other HTML tags.

2. HTML – BASIC TAGS

Heading Tags

Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements **<h1>, <h2>, <h3>, <h4>, <h5> and <h6>.** While displaying any heading, browser adds one line before and one line after that heading.

Example

```
<!DOCTYPE html>
<html>

<head>
    <title>Heading Example</title>
</head>

<body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <h3>This is heading 3</h3>
    <h4>This is heading 4</h4>
    <h5>This is heading 5</h5>
    <h6>This is heading 6</h6>
</body>

</html>
```

This will produce the following result:

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

Paragraph Tag

The ** ** tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening ** ** and a closing ** **tag as shown below in the example:

Example

```
<!DOCTYPE html>
<html>

<head>
    <title>Paragraph Example</title>
</head>

<body>
    <p>Here is a first paragraph of text.</p>
    <p>Here is a second paragraph of text.</p>
    <p>Here is a third paragraph of text.</p>
</body>

</html>
```

This will produce the following result:

Here is a first paragraph of text.

Here is a second paragraph of text.

Here is a third paragraph of text.

Line Break Tag

Whenever you use the **
** element, anything following it starts from the next line. This tag is an example of an **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

The <br /> tag has a space between the characters **br** and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use <br> it is not valid in XHTML.

EXAMPLES

```
<!DOCTYPE html>
<html>

    <head>
        <title>Line Break Example</title>
    </head>

    <body>
        <p>Hello <br />
            You delivered your assignment on time.<br />
            Thanks <br />
            DE-PROF
        </p>
    </body>

</html>
```

This will produce the following result:

Hello
You delivered your assignment on time.
Thanks
DE-PROF

Horizontal Lines

Horizontal lines are used to visually break-up sections of a document. The **<hr>** tag creates a line from the current position in the document to the right margin and breaks the line accordingly.
For example, you may want to give a line between two paragraphs as in the given example below:

Example

```
<!DOCTYPE html>
<html>
    <head>
        <title>Horizontal Line Example</title>
    </head>

    <body>
        <p>This is paragraph one and should be on top</p>
        <hr />
        <p>This is paragraph two and should be at bottom</p>
    </body>
</html>
```

This will produce the following result:

This is paragraph one and should be on top

This is paragraph two and should be at bottom

Again **<hr />** tag is an example of the **empty** element, where you do not need opening and closing tags, as there is nothing to go in between them.

The <**hr** /> element has a space between the characters **hr** and the forward slash. If you omit this space, older browsers will have trouble rendering the horizontal line, while if you miss the forward slash character and just use **<hr>** it is not valid in XHTML

Nonbreaking Spaces

Suppose you want to use the phrase "12 Angry Men." Here, you would not want a browser to split the "12, Angry" and "Men" across two lines:

```
An example of this technique appears in the movie "12 Angry Men."
```

In cases, where you do not want the client browser to break text, you should use a nonbreaking space entity ** ** instead of a normal space. For example, when coding the "12 Angry Men" in a paragraph, you should use something similar to the following code:

Example

```
<!DOCTYPE html>
<html>

    <head>
        <title>Nonbreaking Spaces Example</title>
    </head>

    <body>
        <p>An example of this technique appears in the movie "12 Angry&nbs
p;Men."</p>
    </body>

</html>
```

3. HTML – Elements

An **HTML element** is defined by a starting tag. If the element contains other content, it ends with a closing tag, where the element name is preceded by a forward slash as shown below with few tags:

Start Tag	Content	End Tag
** **	This is paragraph content	
<h1>	This is heading content	</h1>
** **	This is division content	
** **		

So here ** ... ** is an HTML element, **<h1>...</h1>** is another HTML element. There are some HTML elements which don't need to be closed, such as **<img..../>, <hr** /> and **<br** /> elements. These are known as **void elements**.

HTML documents consists of a tree of these elements and they specify how HTML documents should be built, and what kind of content should be placed in what part of an HTML document.

HTML Tag vs. Element

An HTML element is defined by a starting tag. If the element contains other content, it ends with a closing tag.

For example, ** ** is starting tag of a paragraph and ** ** is closing tag of the same paragraph but ** This is paragraph ** is a paragraph element.

Nested HTML Elements

It is very much allowed to keep one HTML element inside another HTML element:

Example

```
<!DOCTYPE html>
<html>
```

```
    <head>
        <title>Nested Elements Example</title>
    </head>

    <body>
        <h1>This is <i>italic</i> heading</h1>
        <p>This is <u>underlined</u> paragraph</p>
    </body>

</html>
```

This will display the following result:

This is *italic* heading
This is underlined paragraph

4. HTML – Attributes

We have seen few HTML tags and their usage like heading tags **<h1>, <h2>** paragraph tag **<p>** and other tags. We used them so far in their simplest form, but most of the HTML tags can also have attributes, which are extra bits of information.

An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag. All attributes are made up of two parts: a **name** and a **value**:

- The **name** is the property you want to set. For example, the paragraph **<p>** element in the example carries an attribute whose name is **align**, which you can use to indicate the alignment of paragraph on the page.

- The **value** is what you want the value of the property to be set and always put within quotations. The below example shows three possible values of align attribute: **left**, **center** and **right**.

Attribute names and attribute values are case-insensitive. However, the World Wide Web Consortium (W3C) recommends lowercase attributes/attribute values in their HTML 4 recommendation.

Example

```
<!DOCTYPE html>
<html>
    <head>
        <title>Align Attribute Example</title>
    </head>

    <body>
        <p align="left">This is left aligned</p>
        <p align="center">This is center aligned</p>
        <p align="right">This is right aligned</p>
    </body>
</html>
```

This will display the following result:

This is left aligned

This is center aligned

This is right aligned

Core Attributes

The four core attributes that can be used on the majority of HTML elements (although not all) are:

- Id
- Title
- Class
- Style

The Id Attribute

The **id** attribute of an HTML tag can be used to uniquely identify any element within an HTML page. There are two primary reasons that you might want to use an id attribute on an element:

- If an element carries an id attribute as a unique identifier, it is possible to identify just that element and its content.

- If you have two elements of the same name within a Web page (or style sheet), you can use the id attribute to distinguish between elements that have the same name.

We will discuss style sheet in separate tutorial. For now, let's use the id attribute to distinguish between two paragraph elements as shown below.

Example

```
<p id="html">This para explains what is HTML</p>

<p id="css">This para explains what is Cascading Style Sheet</p>
```

The title Attribute

The **title** attribute gives a suggested title for the element. They syntax for the **title** attribute is similar as explained for **id** attribute:

The behavior of this attribute will depend upon the element that carries it, although it is often displayed as a tooltip when cursor comes over the element or while the element is loading.

Example

```
<!DOCTYPE html>

<html>

<head>
```

```
<title>The title Attribute Example</title>
</head>
<body>
<h3 title="Hello HTML!">Titled Heading Tag Example</h3>
</body>
</html>
```

This will produce the following result:

Now try to bring your cursor over "Titled Heading Tag Example" and you will see that whatever title you used in your code is coming out as a tooltip of the cursor.

The class Attribute

The **class** attribute is used to associate an element with a style sheet, and specifies the class of element. You will learn more about the u se of the class attribute when you will learn Cascading Style Sheet (CSS). So for now you can avoid it.

The value of the attribute may also be a space-separated list of class names. For example:

```
class="className1 className2 className3"
```

The style Attribute

The style attribute allows you to specify Cascading Style Sheet (CSS) rules within the element**.**

```
<!DOCTYPE html>
<html>
     <head>
           <title>The style Attribute</title>
     </head>
     <body>
            <p style="font-family:arial; color:#FF0000;">Some text...</p>
     </body>
</html>
```

At this point of time, we are not learning CSS, so just let's proceed without bothering much about CSS. Here, you need to understand what are HTML attributes and how they can be used while formatting content.

Internationalization Attributes

There are three internationalization attributes, which are available for most (although not all) XHTML elements.

- dir
- lang
- xml:lag

The dir Attribute

The dir attribute allows you to indicate to the browser about the direction in which the text should flow. The dir attribute can take one of two values, as you can see in the table that follows:

Value	Meaning
ltr	Left to right (the default value)
rtl	Right to left (for languages such as Hebrew or Arabic that are read right to left)

Example

```
<!DOCTYPE html>
<html dir="rtl">
     <head>
          <title>Display Directions</title>
     </head>
     <body>
          This is how IE 5 renders right-to-left directed text.
     </body>
</html>
```

This will produce the following result:

```
                    This is how IE 5 renders right-to-left directed text.
```

When *dir* attribute is used within the <html> tag, it determines how text will be presented within the entire document. When used within another tag, it controls the text's direction for just the content of that tag.

The lang Attribute

The **lang** attribute allows you to indicate the main language used in a document, but this attribute was kept in HTML only for backwards compatibility with earlier versions of HTML. This attribute has been replaced by the **xml:lang** attribute in new XHTML documents.

The values of the ***lang*** attribute are ISO-639 standard two-character language codes. Check HTML Language Codes: ISO 639 for a complete list of language codes.

Example

```
<!DOCTYPE html>
<html lang="en">
     <head>
          <title>English Language Page</title>
     </head>
     <body>
          This page is using English Language
     </body>
</html>
```

The xml:lang Attribute

The *xml:lang* attribute is the XHTML replacement for the *lang* attribute. The value of the *xml:lang* attribute should be an ISO-639 country code as mentioned in previous section.

Generic Attributes

Here's a table of some other attributes that are readily usable with many of the HTML tags.

Attribute	Options	Function
align	right, left, center	Horizontally aligns tags
valign	top, middle, bottom	Vertically aligns tags within an HTML element.
bgcolor	numeric, hexidecimal, RGB values	Places a background color behind an element
background	URL	Places a background image behind an element
id	User Defined	Names an element for use with Cascading Style Sheets.
class	User Defined	Classifies an element for use with Cascading Style Sheets.
width	Numeric Value	Specifies the width of tables, images, or table cells.
height	Numeric Value	Specifies the height of tables, images, or table cells.
title	User Defined	"Pop-up" title of the elements.

We will see related examples as we will proceed to study other HTML tags. For a complete list of HTML Tags and related attributes please check reference to HTML Tags List.

5. HTML – Formating

If you use a word processor, you must be familiar with the ability to make text bold, italicized, or underlined; these are just three of the ten options available to indicate how text can appear in HTML and XHTML.

Bold Text

Anything that appears within **...** element, is displayed in bold as shown below:

Example

```
<!DOCTYPE html>
<html>
     <head>
          <title>Bold Text Example</title>
     </head>
     <body>
          <p>The following word uses a <strong>bold</strong> typeface.</p>
     </body>
</html>
```

This will produce the following result:

```
The following word uses a bold typeface.
```

Italic Text

Anything that appears within **...** element is displayed in italicized as shown below:

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Italic Text Example</title>
</head>

<body>
<p>The following word uses a <em>italicized</em> typeface </p>
</body>

</html>
```

This will produce the following result:

```
The following word uses an italicized typeface.
```

Underlined Text

Anything that appears within **<u>...</u>** element, is displayed with underline as shown below:

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Underlined Text Example</title>
</head>
<body>
<p>The following word uses a <u>underlined</u> typeface.</p>
</body>
</html>
```

This will produce the following result:

The following word uses an underlined typeface.

Strike Text

Anything that appears within **<strike>...</strike>** element is displayed with strikethrough, which is a thin line through the text as shown below:

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Strike Text Example</title>
</head>
<body>
<p>The following word uses a <strike>strikethrough</strike> typeface.</p>
</body>
</html>
```

This will produce the following result:

The following word uses a ~~strikethrough~~ typeface.

Monospaced Font

The content of a **<tt>...</tt>** element is written in monospaced font. Most of the fonts are known as variable-width fonts because different letters are of different widths (for example, the letter 'm' is wider than the letter 'i'). In a monospaced font, however, each letter has the same width.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Monospaced Font Example</title>
</head>
<body>
<p>The following word uses a <tt>monospaced</tt> typeface.</p>
```

```
</body>
</html>
```

This will produce the following result:

```
The following word uses a monospaced typeface.
```

Superscript Text

The content of a ^{...} element is written in superscript; the font size used is the same size as the characters surrounding it but is displayed half a character's height above the other characters.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Superscript Text Example</title>
</head>
<body>
<p>The following word uses a <sup>superscript</sup> typeface.</p>
</body>
</html>
```

This will produce the following result:

```
The following word uses a superscript typeface.
```

Subscript Text

The content of a **_{...}** element is written in subscript; the font size used is the same as the characters surrounding it, but is displayed half a character's height beneath the other characters.

Example

```
<!DOCTYPE html>
```

```
<html>
     <head>
           <title>Subscript Text Example</title>
     </head>
     <body>
           <p>The following word uses a <sub>subscript</sub> typeface.</p>
     </body>
</html>
```

This will produce the following result:

The following word uses a subscript typeface.

36

Inserted Text

Anything that appears within **<ins>...</ins>** element is displayed as inserted text.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Inserted Text Example</title>
</head>
<body>
<p>I want to drink <del>cola</del> <ins>wine</ins></p>
</body>
</html>
```

This will produce the following result:

I want to drink ~~cola~~ wine

Deleted Text

Anything that appears within **~~...~~** element, is displayed as deleted text.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Deleted Text Example</title>
</head>
<body>
<p>I want to drink <del>cola</del> <ins>wine</ins></p>
</body>
</html>
```

This will produce the following result:

I want to drink ~~cola~~ wine

Larger Text

The content of the **<big>...</big>** element is displayed one font size larger than the rest of the text surrounding it as shown below:

Example

```
<!DOCTYPE html>
<html>
    <head>
        <title>Larger Text Example</title>
    </head>
    <body>
        <p>The following word uses a <big>big</big> typeface.</p>
    </body>
```

```
</html>
```

This will produce the following result:

```
The following word uses a big typeface.
```

Smaller Text

The content of the **<small>...</small>** element is displayed one font size smaller than the rest of the text surrounding it as shown below:

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Smaller Text Example</title>
</head>
<body>
<p>The following word uses a <small>small</small> typeface.</p>
</body>
```

38

```
</html>
```

This will produce the following result:

```
The following word uses a small typeface.
```

Grouping Content

The ** ** and **** elements allow you to group together several elements to create sections or subsections of a page. For example, you might want to put all of the footnotes on a page within a element to indicate that all of the elements within that element relate to the footnotes. You might then attach a style to this element so that they appear using a special set of style rules.

Example

```
<!DOCTYPE html>
<html>
     <head>
           <title>Div Tag Example</title>
     </head>
     <body>
        <div id="menu" align="middle" >
           <a href="/index.htm">HOME</a> |
           <a href="/about/contact_us.htm">CONTACT</a> |
           <a href="/about/index.htm">ABOUT</a>
        </div>

        <div id="content" align="left" bgcolor="white">
            <h5>Content Articles</h5>
            <p>Actual content goes here.....</p>
        </div>
     </body>
</html>
```

This will produce the following result:

HOME | CONTACT | ABOUT

CONTENT ARTICLES

Actual content goes here.....

The element, on the other hand, can be used to group inline elements only. So, if you have a part of a sentence or paragraph which you want to group together, you could use the element as follows

Example

```
<!DOCTYPE html>

<html>

     <head>

           <title>Span Tag Example</title>

     </head>

     <body>

           <p>This is the example of <span style="color:green">span
tag</span> and the <span style="color:red">div tag</span> alongwith CSS</p>

     </body>

</html>
```

This will produce the following result:

This is the example of span tag and the div tag along with CSS

These tags are commonly used with CSS to allow you to attach a style to a section of a page.

6. HTML – Phrase Tags

The phrase tags have been desicolgned for specific purposes, though they are displayed in a similar way as other basic tags like **<b>, <i>, <pre>,** and **<tt>,** you have seen in previous chapter. This chapter will take you through all the important phrase tags, so let's start seeing them one by one.

Marked Text

Anything that appears with-in **<mark>...</mark>** element, is displayed as marked with yellow ink.

Example

```
<!DOCTYPE html>

<html>

     <head>

           <title>Marked Text Example</title>
     </head>

     <body>

           <p>The following word has been <mark>marked</mark> with
yellow</p>

     </body>

</html>
```

This will produce the following result:

The following word has been marked with yellow.

Quoting Text

When you want to quote a passage from another source, you should put it in between**<blockquote>...</blockquote>** tags.

Text inside a **<blockquote>** element is usually indented from the left and right edges of the surrounding text, and sometimes uses an italicized font.

Example

```
<!DOCTYPE html>
<html>
     <head>
          <title>Blockquote Example</title>
     </head>
     <body>
         <p>The following description of XHTML is taken from the
W3C Web site:</p>

         <blockquote>XHTML 1.0 is the W3C's first Recommendation for XHTML,
following on from earlier work on HTML 4.01, HTML 4.0, HTML 3.2 and HTML
2.0.</blockquote>
     </body>
</html>
```

This will produce the following result:

The following description of XHTML is taken from the W3C Web site:

> XHTML 1.0 is the W3C's first Recommendation for XHTML, following on from earlier work on HTML 4.01, HTML 4.0, HTML 3.2 and HTML 2.0.

Computer Code

Any programming code to appear on a Web page should be placed inside **<code>...</code>** tags. Usually the content of the <code> element is presented in a monospaced font, just like the code in most programming books.

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Computer Code Example</title>
</head>
<body>
<p>Regular text. <code>This is code.</code> Regular text.</p>
</body>
</html>
```

This will produce the following result:

Regular text. `This is code`. Regular text.

Address Text

The **<address>...</address>** element is used to contain any address.

Example

```
<!DOCTYPE html>
<html>
     <head>
           <title>Address Example</title>
     </head>
     <body>
          <address>1st Avenue, House No 22, Gwarimpa -  Abuja</address>
     </body>
</html>
```

This will produce the following result:

1st Avenue, House No 22, Gwarimpa - Abuja

7. HTML – Meta Tags

HTML lets you specify metadata - additional important information about a document in a variety of ways. The META elements can be used to include name/value pairs describing properties of the HTML document, such as author, expiry date, a list of keywords, document author etc.

The **<meta>** tag is used to provide such additional information. This tag is an empty element and so does not have a closing tag but it carries information within its attributes.

You can include one or more meta tags in your document based on what information you want to keep in your document but in general, meta tags do not impact physical appearance of the document so from appearance point of view, it does not matter if you include them or not.

Adding Meta Tags to Your Documents

You can add metadata to your web pages by placing <meta> tags inside the header of the document which is represented by **<head>** and **</head>** tags. A meta tag can have following attributes in addition to core attributes:

Attribute	Description
Name	Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc.
content	Specifies the property's value.
scheme	Specifies a scheme to interpret the property's value (as declared in the content attribute).
httpequiv	Used for http response message headers. For example, http-equiv can be used to refresh the page or to set a cookie. Values include content-type, expires, refresh and set-cookie.

Specifying Keywords

You can use <meta> tag to specify important keywords related to the document and later these keywords are used by the search engines while indexing your webpage for searching purpose.

Example

Following is an example, where we are adding HTML, Meta Tags, Metadata as important keywords about the document.

```
<!DOCTYPE html>
<html>
     <head>
           <title>Meta Tags Example</title>
           <meta name="keywords" content="HTML, Meta Tags, Metadata" />
     </head>
     <body>
           <p>Hello HTML5!</p>
     </body>
</html>
```

This will produce the following result:

```
Hello HTML5!
```

Document Description

You can use <meta> tag to give a short description about the document. This again can be used by various search engines while indexing your webpage for searching purpose.

Example

```
<!DOCTYPE html>
<html>
     <head>
           <title>Meta Tags Example</title>
           <meta name="keywords" content="HTML, Meta Tags, Metadata" />
           <meta name="description" content="Learning about Meta Tags." />
     </head>
     <body>
           <p>Hello HTML5!</p>
     </body>
</html>
```

Document Revision Date

You can use <meta> tag to give information about when last time the document was updated. This information can be used by various web browsers while refreshing your webpage.

Example

```
<!DOCTYPE html>
<html>
     <head>
           <title>Meta Tags Example</title>
           <meta name="keywords" content="HTML, Meta Tags, Metadata" />
           <meta name="description" content="Learning about Meta Tags." />
           <meta name="revised" content="Ogtech Networks Ltd, 5/9/2020" />
     </head>
     <body>
           <p>Hello HTML5!</p>
     </body>
</html>
```

Document Refreshing

A <meta> tag can be used to specify a duration after which your web page will keep refreshing automatically.

Example

If you want your page keep refreshing after every 5 seconds then use the following syntax.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Meta Tags Example</title>
        <meta name="keywords" content="HTML, Meta Tags, Metadata" />
        <meta name="description" content="Learning about Meta Tags." />
        <meta name="revised" content="Ogtech Networks Ltd, 5/9/2020" />
        <meta http-equiv="refresh" content="5" />
    </head>
    <body>
        <p>Hello HTML5!</p>
    </body>
</html>
```

Page Redirection

You can use <meta> tag to redirect your page to any other webpage. You can also specify a duration if you want to redirect the page after a certain number of seconds.

Example

Following is an example of redirecting current page to another page after 5 seconds. If you want to redirect page immediately then do not specify *content* attribute.

```
<!DOCTYPE html>
<html>
     <head>
           <title>Meta Tags Example</title>
           <meta name="keywords" content="HTML, Meta Tags, Metadata" />
           <meta name="description" content="Learning about Meta Tags." />
           <meta name="revised" content="Ogtech Networks Ltd, 5/9/2020" />
           <meta http-equiv="refresh" content="5;
url=http://www.tutorialspoint.com" />
     </head>
     <body>
           <p>Hello HTML5!</p>
     </body>
</html>
```

Setting Cookies

Cookies are data, stored in small text files on your computer and it is exchanged between web browser and web server to keep track of various information based on your web application need.

You can use <meta> tag to store cookies on client side and later this information can be used by the Web Server to track a site visitor.

Example

Following is an example of redirecting current page to another page after 5 seconds. If you want to redirect page immediately then do not specify *content* attribute.

```
<!DOCTYPE html>
<html>
     <head>
           <title>Meta Tags Example</title>
           <meta name="keywords" content="HTML, Meta Tags, Metadata" />
           <meta name="description" content="Learning about Meta Tags." />
           <meta name="revised" content="Ogtech Networks Ltd, 5/9/2020" />
```

Cascading Style Sheet

About the Tutorial

CSS is used to control the style of a web document in a simple and easy way. CSS stands for Cascading Style Sheets. This tutorial covers both the versions CSS1 and CSS2 and gives a complete understanding of CSS, starting from its basics to advanced concepts.

Audience

This tutorial will help both students as well as professionals who want to make their websites or personal blogs more attractive.

1. CSS – OVERVIEW

What is CSS?

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, as well as a variety of other effects.

CSS is easy to learn and understand but it provides a powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantages of CSS

- **CSS saves time** - You can write CSS once and then reuse the same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many web pages as you want.
- **Pages load faster** - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.
- **Easy maintenance** - To make a global change, simply change the style, and all the elements in all the web pages will be updated automatically.
- **Superior styles to HTML** - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility** - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cellphones or for printing.

Global web standards – Now HTML attributes are being deprecated and it is being recommended to use CSS. So it's a good idea to start using CSS in all the HTML pages to make them compatible with future browsers.

2. CSS – SYNTAX

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts:

- Selector: A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
- Property: A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be *color*, *border*, etc.
- Value: Values are assigned to properties. For example, *color* property can have the value either *red* or *#F1F1F1* etc.

You can put CSS Style Rule Syntax as follows:

```
selector { property: value }
```

Example: You can define a table border as follows:

```
table{

     border :1px solid #C00;

}
```

Here table is a selector and border is a property and the given value *1px solid #C00* is the value of that property.

You can define selectors in various simple ways based on your comfort. Let me put these selectors one by one.

The Type Selectors

This is the same selector we have seen above. Again, one more example to give a color to all level 1 headings:

```
h1 {

    color:#36CFFF;

}
```

The Universal Selectors

Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type:

```
* {

   color: #000000;

}
```

This rule renders the content of every element in our document in black.

The Descendant Selectors

Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, the style rule will apply to element only when it lies inside the <ul> tag.

```
ul em {

   color: #000000;

}
```

The Class Selectors

You can define style rules based on the class attribute of the elements. All the elements having that class will be formatted according to the defined rule.

```
.black {

   color: #000000;

}
```

This rule renders the content in black for every element with class attribute set to *black* in our document. You can make it a bit more particular. For example:

```
h1.black {
color: #000000;

}
```

This rule renders the content in black for only <h1> elements with class attribute set to *black*.

You can apply more than one class selectors to a given element. Consider the following example:

```
<p class="center bold">

This para will be styled by the classes center and bold.
</p>
```

The ID Selectors

You can define style rules based on the *id* attribute of the elements. All the elements having that *id* will be formatted according to the defined rule.

```
#black {

   color: #000000;

}
```

This rule renders the content in black for every element with *id* attribute set to *black* in our document. You can make it a bit more particular. For example:

```
h1#black {

color: #000000;

}
```

This rule renders the content in black for only <h1> elements with *id* attribute set to *black*.

The true power of *id* selectors is when they are used as the foundation for descendant selectors. For example:

```
#black h2 {

color: #000000;

}
```

In this example, all level 2 headings will be displayed in black color when those headings will lie within tags having *id* attribute set to *black*.

The Child Selectors

You have seen the descendant selectors. There is one more type of selector, which is very similar to descendants but have different functionality. Consider the following example:

```
body > p {
color: #000000;

}
```

This rule will render all the paragraphs in black if they are a direct child of the <body> element. Other paragraphs put inside other elements like or <td> would not have any effect of this rule.

The Attribute Selectors

You can also apply styles to HTML elements with particular attributes. The style rule below will match all the input elements having a type attribute with a value of *text*:

```
input[type="text"]{
color: #000000;

}
```

The advantage to this method is that the <input type="submit" /> element is unaffected, and the color applied only to the desired text fields.

There are following rules applied to attribute selector.

- p[lang] - Selects all paragraph elements with a *lang* attribute.
- p[lang="fr"] - Selects all paragraph elements whose *lang* attribute has a value of exactly "fr".
- p[lang~="fr"] - Selects all paragraph elements whose *lang* attribute contains the word "fr".
- p[lang|="en"] - Selects all paragraph elements whose *lang* attribute contains values that are exactly "en", or begin with "en-".

Multiple Style Rules

You may need to define multiple style rules for a single element. You can define these rules to combine multiple properties and corresponding values into a single block as defined in the following example:

```
h1 { color: #36C; font-
weight: normal; letter-
spacing: .4em; margin-
bottom: 1em; text-
transform: lowercase;

}
```

Here all the property and value pairs are separated by a semicolon (;). You can keep them in a single line or multiple lines. For better readability, we keep them in separate lines.

For a while, don't bother about the properties mentioned in the above block. These properties will be explained in the coming chapters and you can find the complete detail about properties in CSS References.

Grouping Selectors

You can apply a style to many selectors if you like. Just separate the selectors with a comma, as given in the following example:

```
h1, h2, h3 { color:
#36C; font-weight:
normal; letter-spacing:
.4em; margin-bottom:
1em; text-transform:
lowercase;

}
```

This define style rule will be applicable to h1, h2 and h3 element as well. The order of the list is irrelevant. All the elements in the selector will have the corresponding declarations applied to them.

You can combine the various *class* selectors together as shown below:

```
#content, #footer, #supplement {
position: absolute; left: 510px;
width: 200px; }
```

3. CSS – INCLUSION

There are four ways to associate styles with your HTML document. Most commonly used methods are inline CSS and External CSS.

Embedded CSS - The <style> Element

You can put your CSS rules into an HTML document using the <style> element. This tag is placed inside the <head>...</head> tags. Rules defined using this syntax will be applied to all the elements available in the document. Here is the generic syntax:

```
<head>
    <style type="text/css" media="...">
          Style Rules
          ............
    </style>
</head>
```

Attributes

Attributes associated with <style> elements are:

Attribute	Value	Description
type	text/css	Specifies the style sheet language as a content-type (MIME type). This is a required attribute.
media	screen tty tv projection handheld print braille	Specifies the device, the document will be displayed on. Default value is *all*. This is an optional attribute.

Example

Following is an example of embed CSS based on the above syntax:

```
<head>
<style type="text/css"
media="all"> h1{ color: #36C;

}
</style>
</head>
```

Inline CSS - The *style* Attribute

You can use *style* attribute of any HTML element to define style rules. These rules will be applied to that element only. Here is the generic syntax:

```
<element style="...style rules....">
```

Attributes

Attribute	Value	Description
style	style rules	The value of *style* attribute is a combination of style declarations separated by semicolon (;).

Example

Following is the example of inline CSS based on the above syntax:

```
<h1 style ="color:#36C;"> This is inline CSS </h1>
```

It will produce the following result:

This is inline CSS

External CSS - The <link> Element

The <link> element can be used to include an external stylesheet file in your HTML document.

An external style sheet is a separate text file with .css extension. You define all the Style rules within this text file and then you can include this file in any HTML document using <link> element.

Here is the generic syntax of including external CSS file:

```
<head>
    <link type="text/css" href="..." media="..." />
</head>
```

Attributes

Attributes associated with <style> elements are:

Attribute	Value	Description
type	text/css	Specifies the style sheet language as a content-type (MIME type). This attribute is required.
href	URL	Specifies the style sheet file having Style rules. This attribute is a required.
media	screen tty tv projection handheld print braille	Specifies the device the document will be displayed on. Default value is *all*. This is an optional attribute.
	aural all	

Example

Consider a simple style sheet file with a name *mystyle.css* having the following rules:

```
h1, h2, h3 {

color: #36C;

font-weight: normal;
letter-spacing: .4em;
margin-bottom: 1em;
text-transform:
lowercase;

}
```

Now you can include this file *mystyle.css* in any HTML document as follows:

```
<head>
     <link type="text/css" href="mystyle.css" media="all" />
</head>
```

Imported CSS - @import Rule

@import is used to import an external stylesheet in a manner similar to the <link> element. Here is the generic syntax of @import rule.

```
<head>
     <@import "URL";
</head>
```

Here URL is the URL of the style sheet file having style rules. You can use another syntax as well:

```
<head>
     <@import url("URL");
</head>
```

Example

Following is the example showing you how to import a style sheet file into an HTML document:

```
<head>
    @import "mystyle.css";
</head>
```

CSS Rules Overriding

We have discussed four ways to include style sheet rules in an HTML document. Here is the rule to override any Style Sheet Rule.

- Any inline style sheet takes the highest priority. So, it will override any rule defined in <style>...</style> tags or the rules defined in any external style sheet file.
- Any rule defined in <style>...</style> tags will override the rules defined in any external style sheet file.
- Any rule defined in the external style sheet file takes the lowest priority, and the rules defined in this file will be applied only when the above two rules are not applicable.

CSS Comments

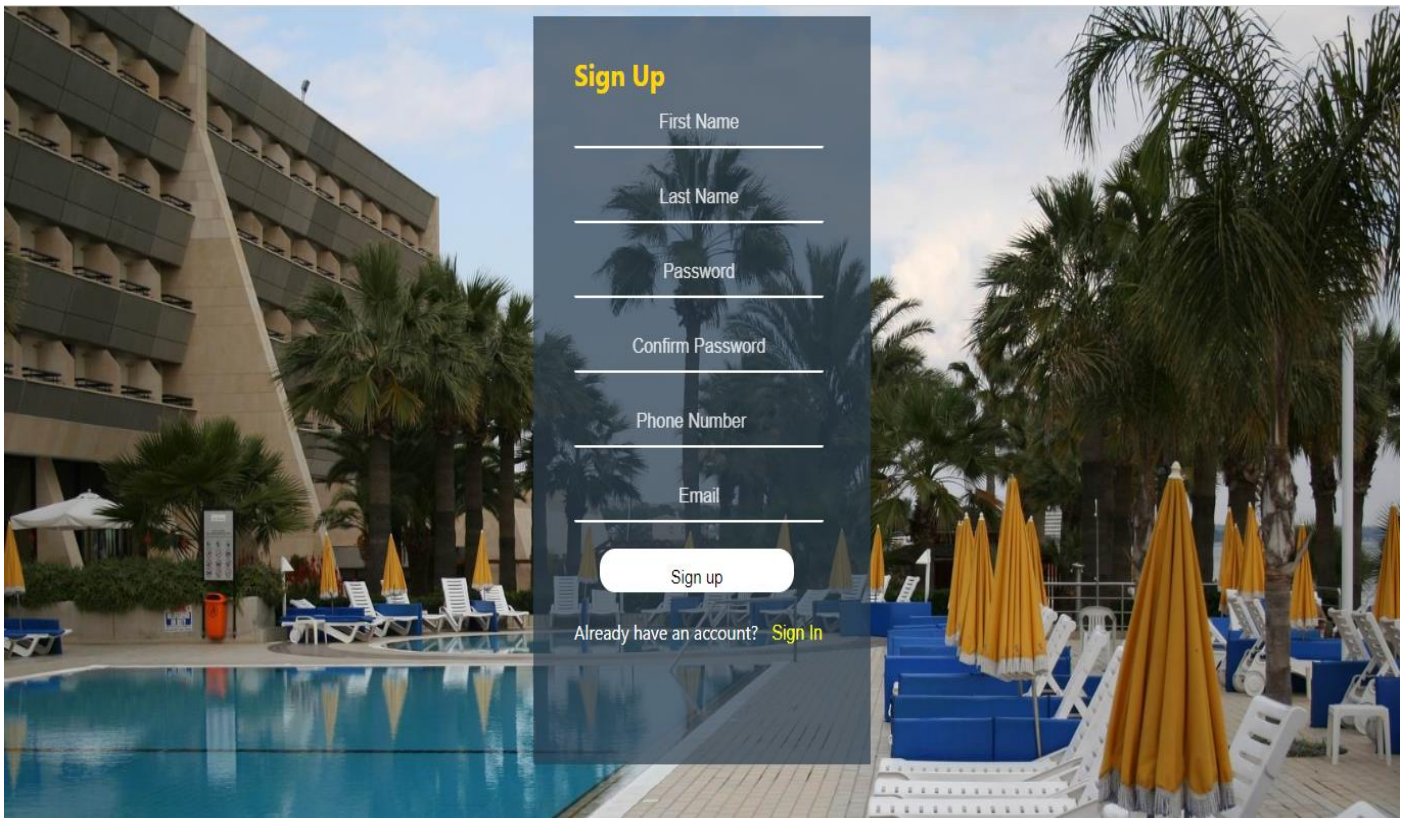
Many times, you may need to put additional comments in your style sheet blocks. So, it is very easy to comment any part in the style sheet. You can simply put your comments inside /*.....this is a comment in style sheet.....*/.

You can use /* ....*/ to comment multi-line blocks in similar way you do in C and C++ programming languages. **Example**

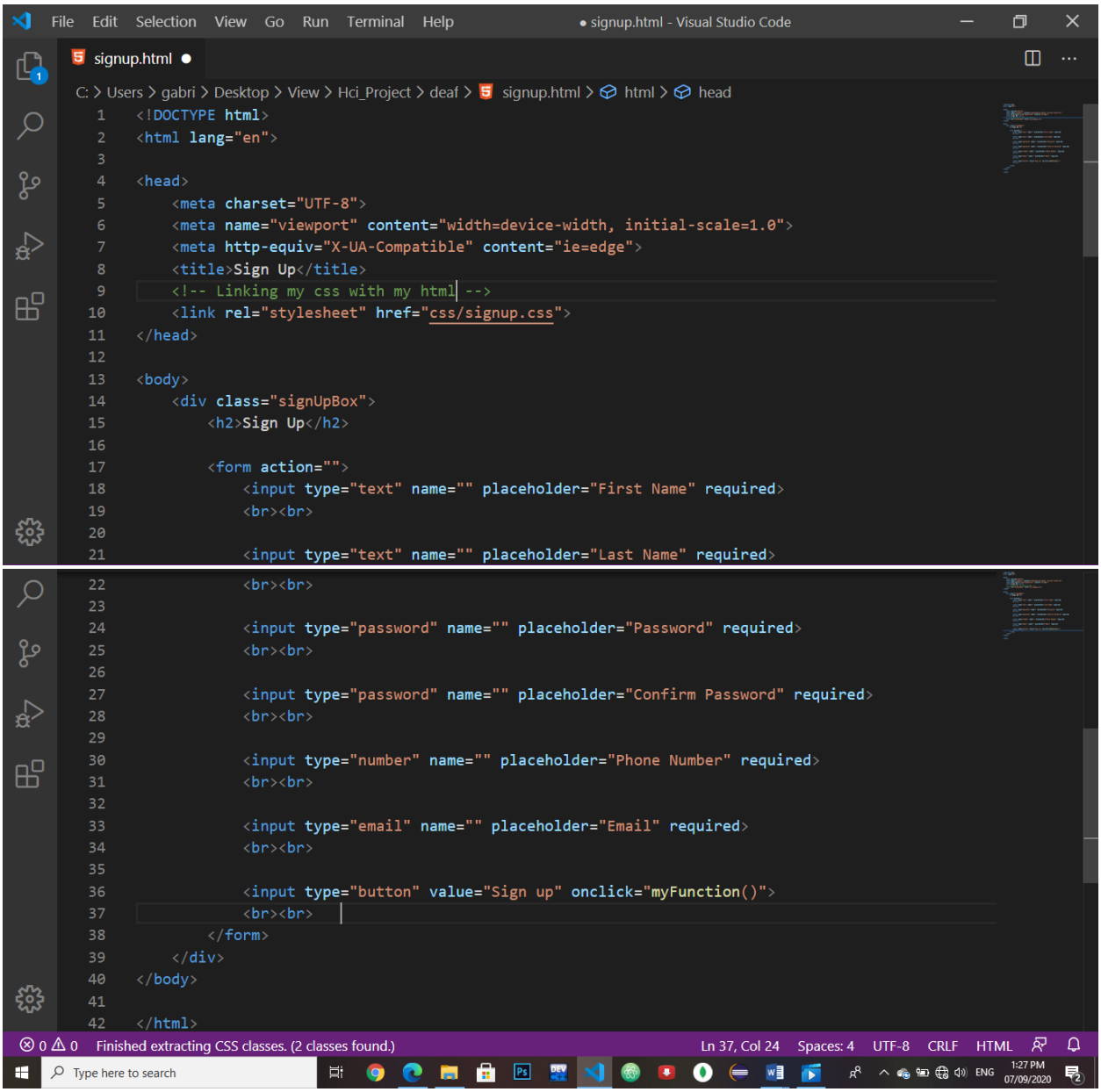
```
/* This is an external style sheet file */
h1, h2, h3 { color: #36C; font-weight:
normal; letter-spacing: .4em; margin-
bottom: 1em; text-transform: lowercase;

}
/* end of style rules. */
```

4. CSS – **PROJECTS**

FIRST PROJECT – PROJECT OVERVIEW

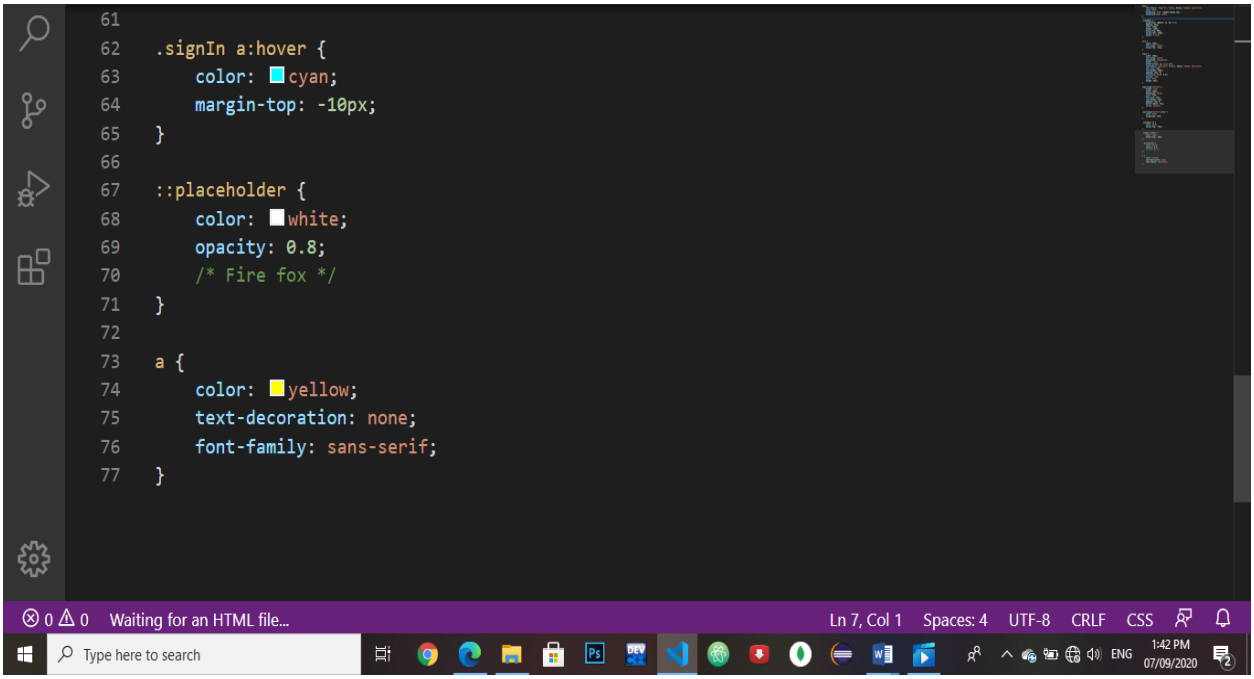


The html source codes



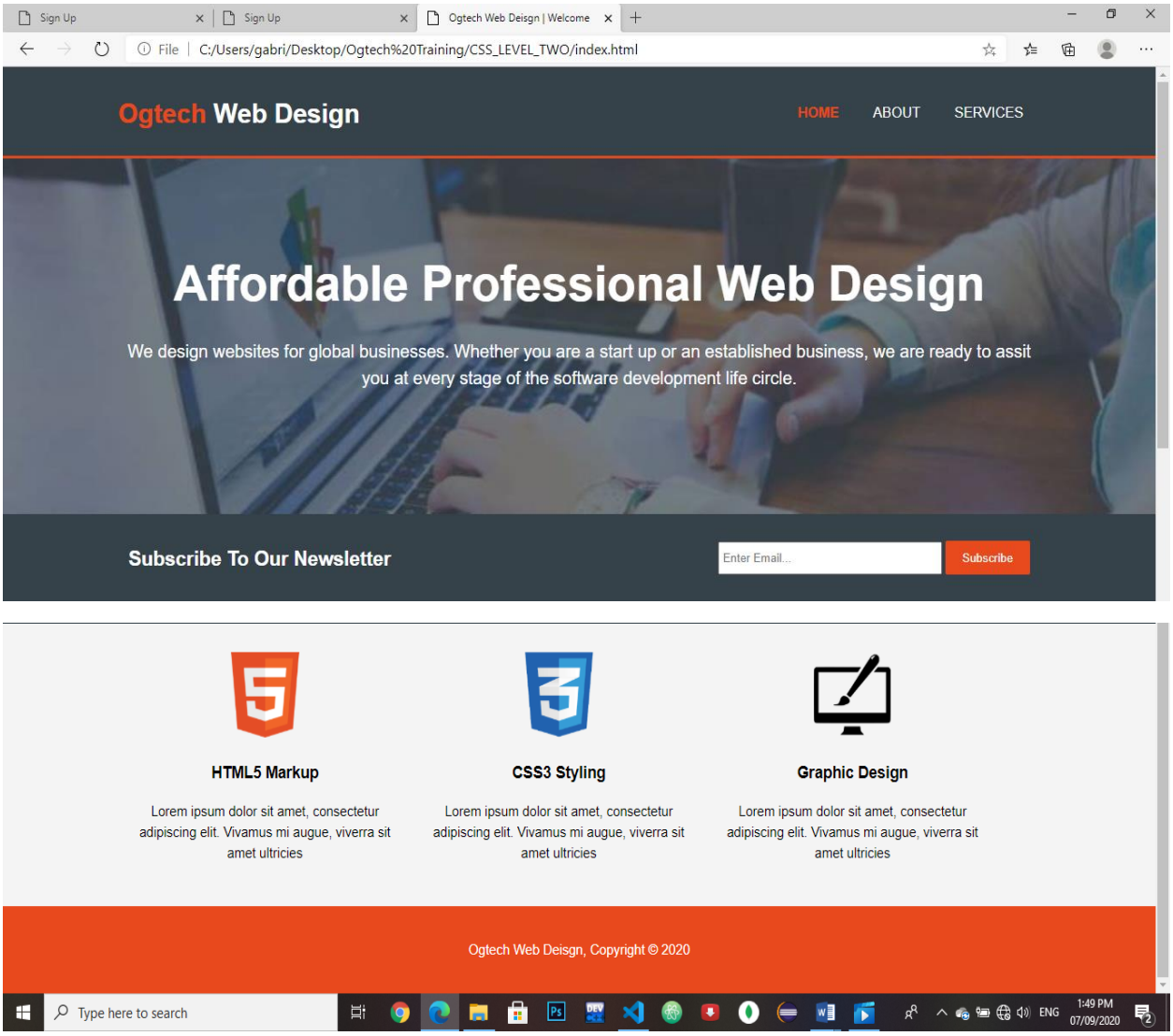
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Sign Up</title>
    <!-- Linking my css with my html -->
    <link rel="stylesheet" href="css/signup.css">
</head>

<body>
    <div class="signUpBox">
        <h2>Sign Up</h2>

        <form action="">
            <input type="text" name="" placeholder="First Name" required>
            <br><br>

            <input type="text" name="" placeholder="Last Name" required>
            <br><br>

            <input type="password" name="" placeholder="Password" required>
            <br><br>

            <input type="password" name="" placeholder="Confirm Password" required>
            <br><br>

            <input type="number" name="" placeholder="Phone Number" required>
            <br><br>

            <input type="email" name="" placeholder="Email" required>
            <br><br>

            <input type="button" value="Sign up" onclick="myFunction()">
            <br><br>
        </form>
    </div>
</body>

</html>
```

The Css source codes

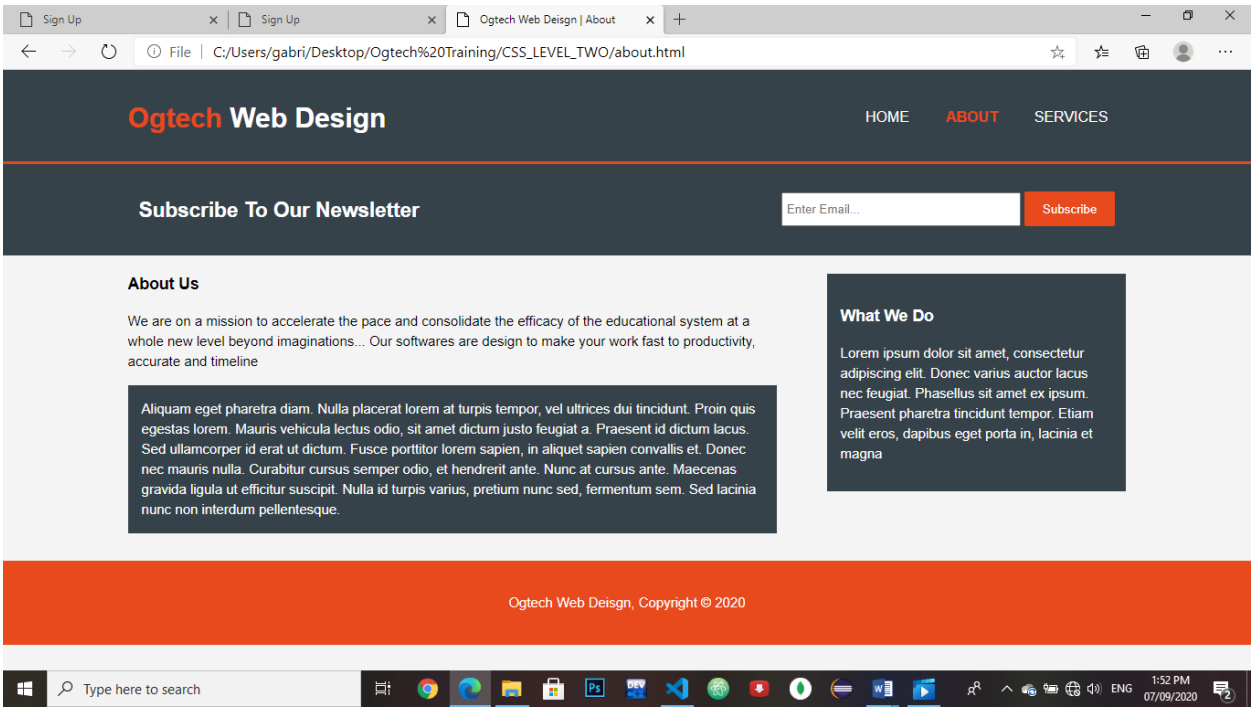

61
62 .signIn a:hover {
63 color: cyan;
64 margin-top: -10px;
65 }
66
67 ::placeholder {
68 color: white;
69 opacity: 0.8;
70 /* Fire fox */
71 }
72
73 a {
74 color: yellow;
75 text-decoration: none;
76 font-family: sans-serif;
77 }
0 0 Waiting for an HTML file...
Ln 7, Col 1 Spaces: 4 UTF-8 CRLF CSS
Type here to search
ENG
1:42 PM
07/09/2020

SECOND PROJECT – 3 PAGE WEBSITE

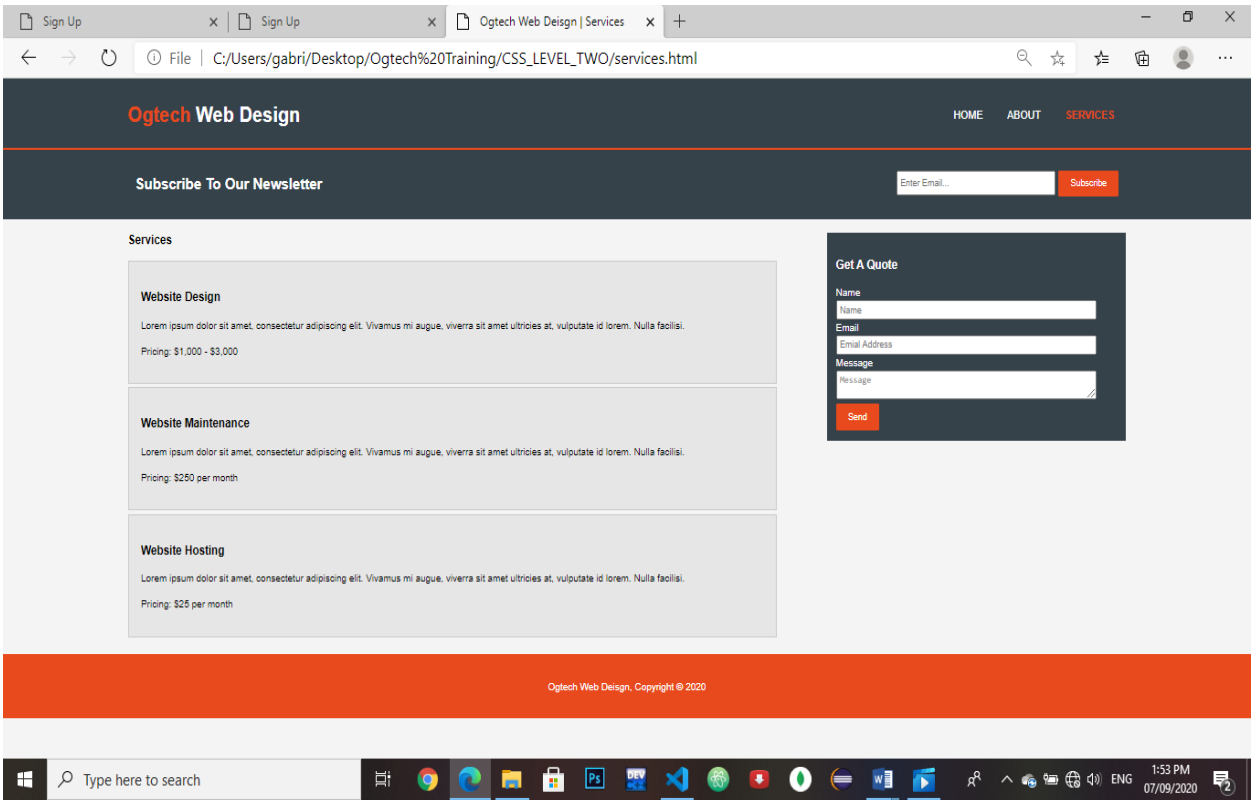
PROJECT OVERVIEW – HOME PAGE



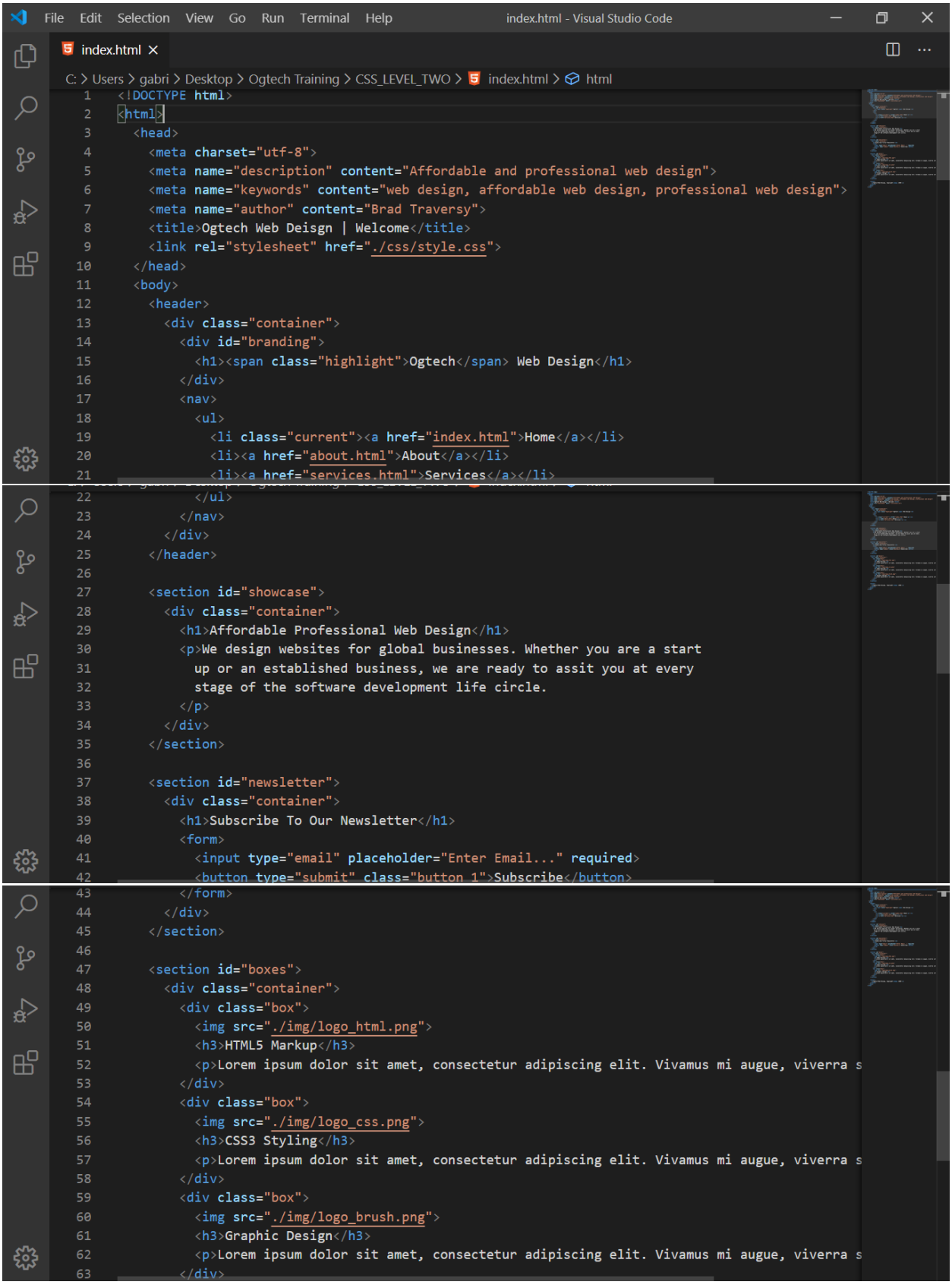
ABOUT PAGE



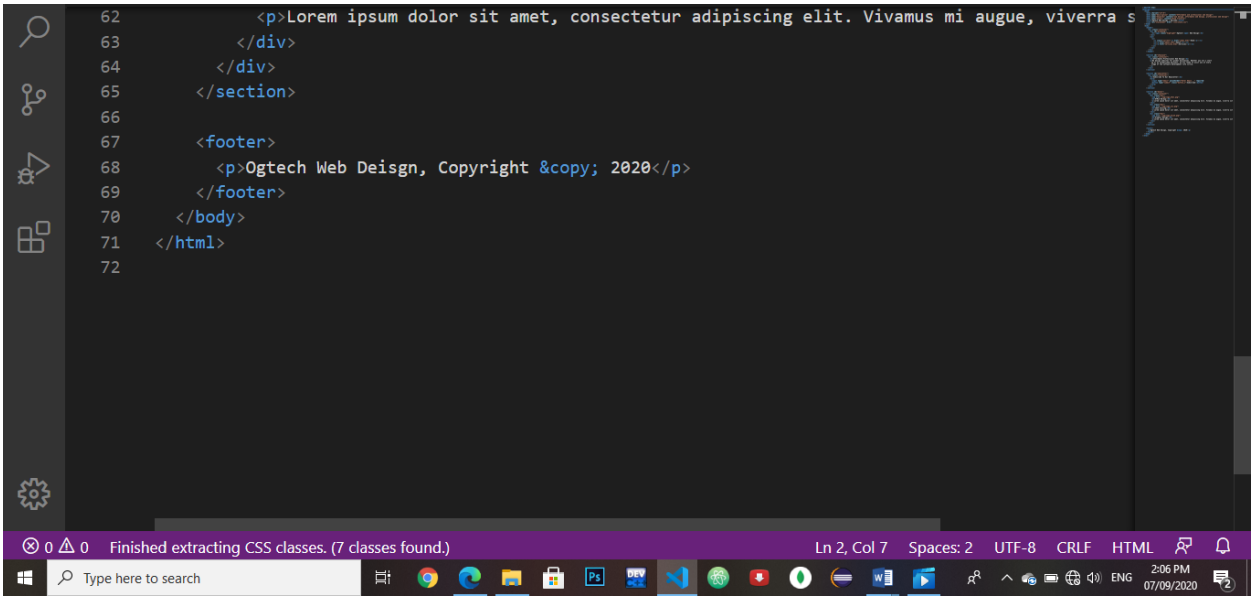
SERVICES PAGE



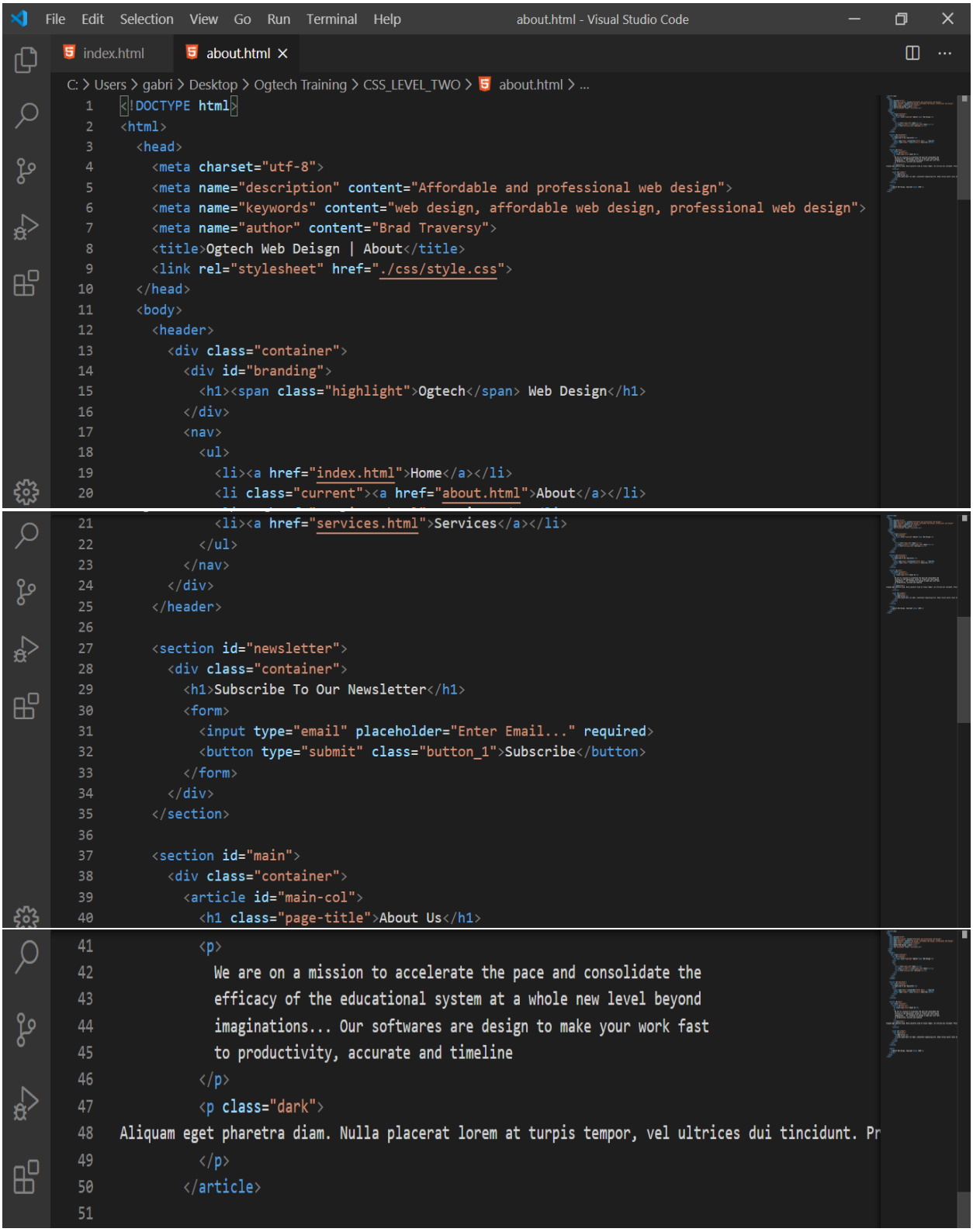
Html source codes for the home page

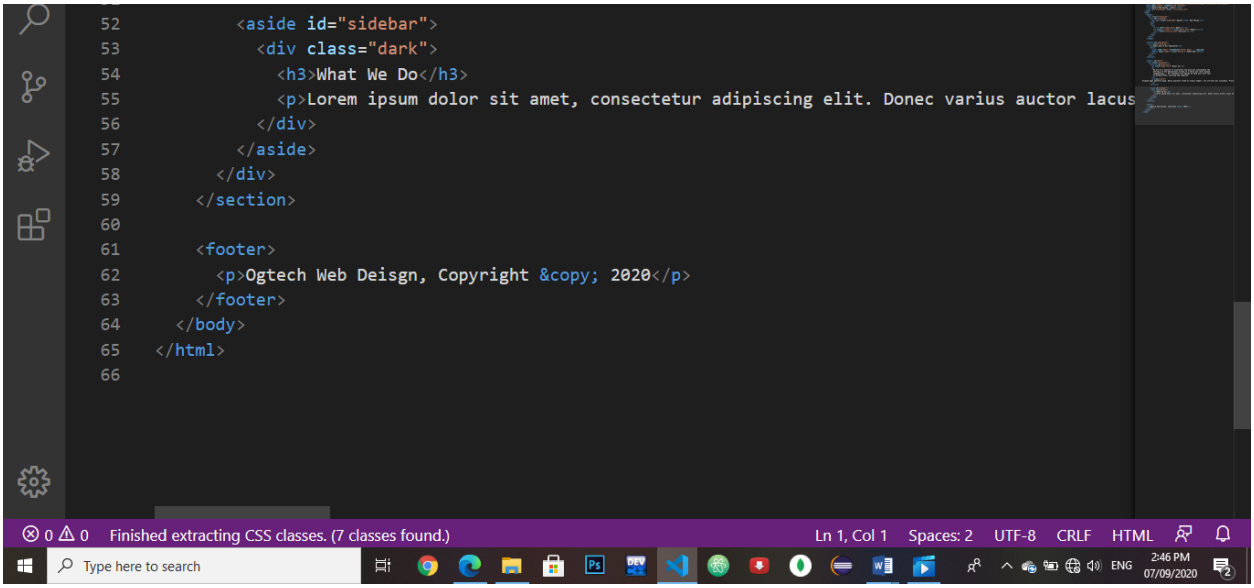



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="description" content="Affordable and professional web design">
    <meta name="keywords" content="web design, affordable web design, professional web design">
    <meta name="author" content="Brad Traversy">
    <title>Ogtech Web Deisgn | Welcome</title>
    <link rel="stylesheet" href="./css/style.css">
  </head>
  <body>
    <header>
      <div class="container">
        <div id="branding">
          <h1><span class="highlight">Ogtech</span> Web Design</h1>
        </div>
        <nav>
          <ul>
            <li class="current"><a href="index.html">Home</a></li>
            <li><a href="about.html">About</a></li>
            <li><a href="services.html">Services</a></li>
          </ul>
        </nav>
      </div>
    </header>

    <section id="showcase">
      <div class="container">
        <h1>Affordable Professional Web Design</h1>
        <p>We design websites for global businesses. Whether you are a start
          up or an established business, we are ready to assit you at every
          stage of the software development life circle.
        </p>
      </div>
    </section>

    <section id="newsletter">
      <div class="container">
        <h1>Subscribe To Our Newsletter</h1>
        <form>
          <input type="email" placeholder="Enter Email..." required>
          <button type="submit" class="button_1">Subscribe</button>
        </form>
      </div>
    </section>

    <section id="boxes">
      <div class="container">
        <div class="box">
          <img src="./img/logo_html.png">
          <h3>HTML5 Markup</h3>
          <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi augue, viverra s
        </div>
        <div class="box">
          <img src="./img/logo_css.png">
          <h3>CSS3 Styling</h3>
          <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi augue, viverra s
        </div>
        <div class="box">
          <img src="./img/logo_brush.png">
          <h3>Graphic Design</h3>
          <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi augue, viverra s
        </div>
```

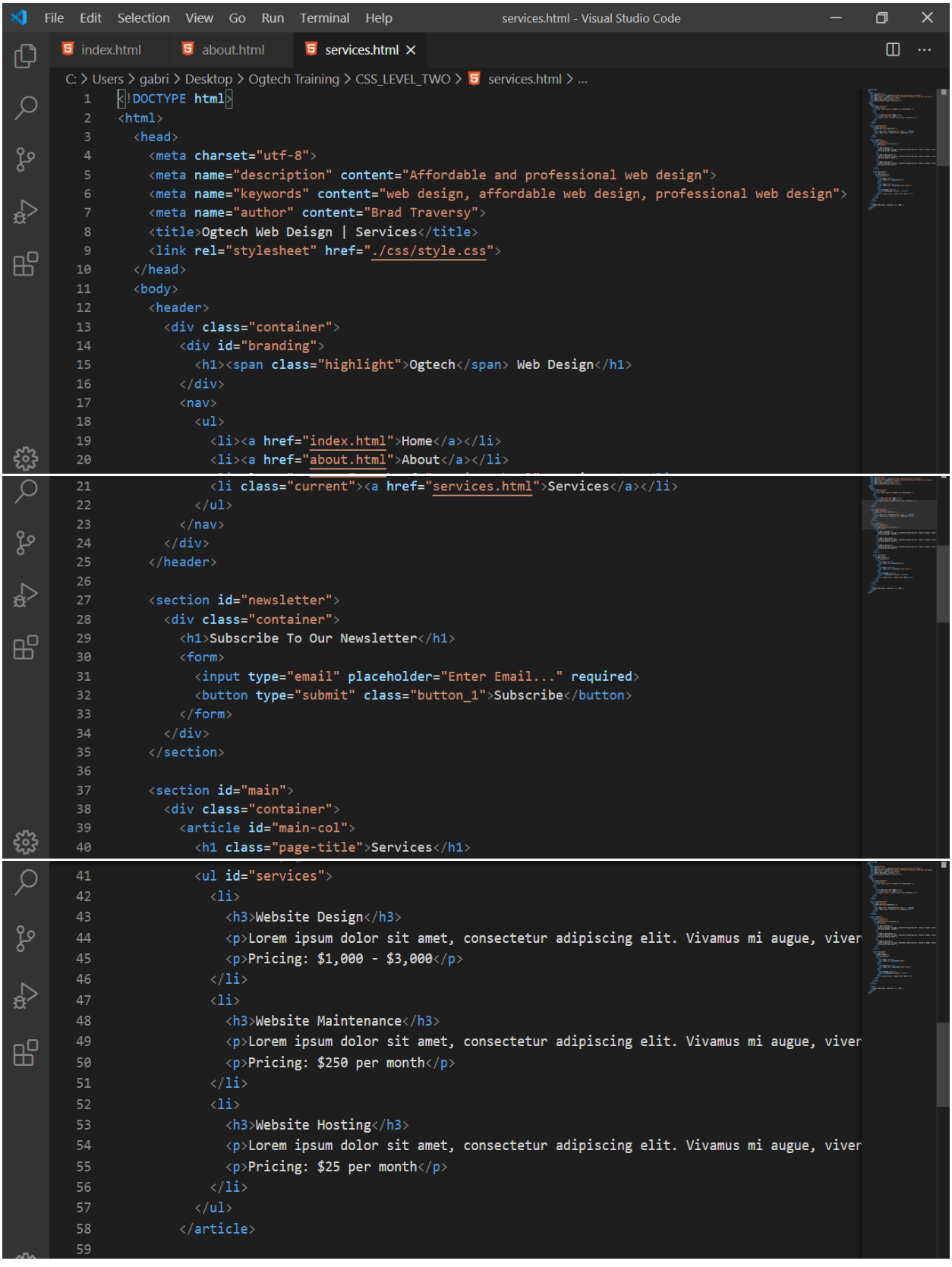

62 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus mi augue, viverra s
63
64
65 </section>
66
67 <footer>
68 Ogtech Web Deisgn, Copyright © 2020
69 </footer>
70 </body>
71 </html>
72
0 0 Finished extracting CSS classes. (7 classes found.)
Ln 2, Col 7 Spaces: 2 UTF-8 CRLF HTML
Type here to search
ENG
2:06 PM
07/09/2020

Html source codes for the about page




52 <aside id="sidebar">
53
54 <h3>What We Do</h3>
55 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec varius auctor lacus
56
57 </aside>
58
59 </section>
60
61 <footer>
62 Ogtech Web Deisgn, Copyright © 2020
63 </footer>
64 </body>
65 </html>
66
0 0 Finished extracting CSS classes. (7 classes found.)
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF HTML
Type here to search
ENG
2:46 PM
07/09/2020

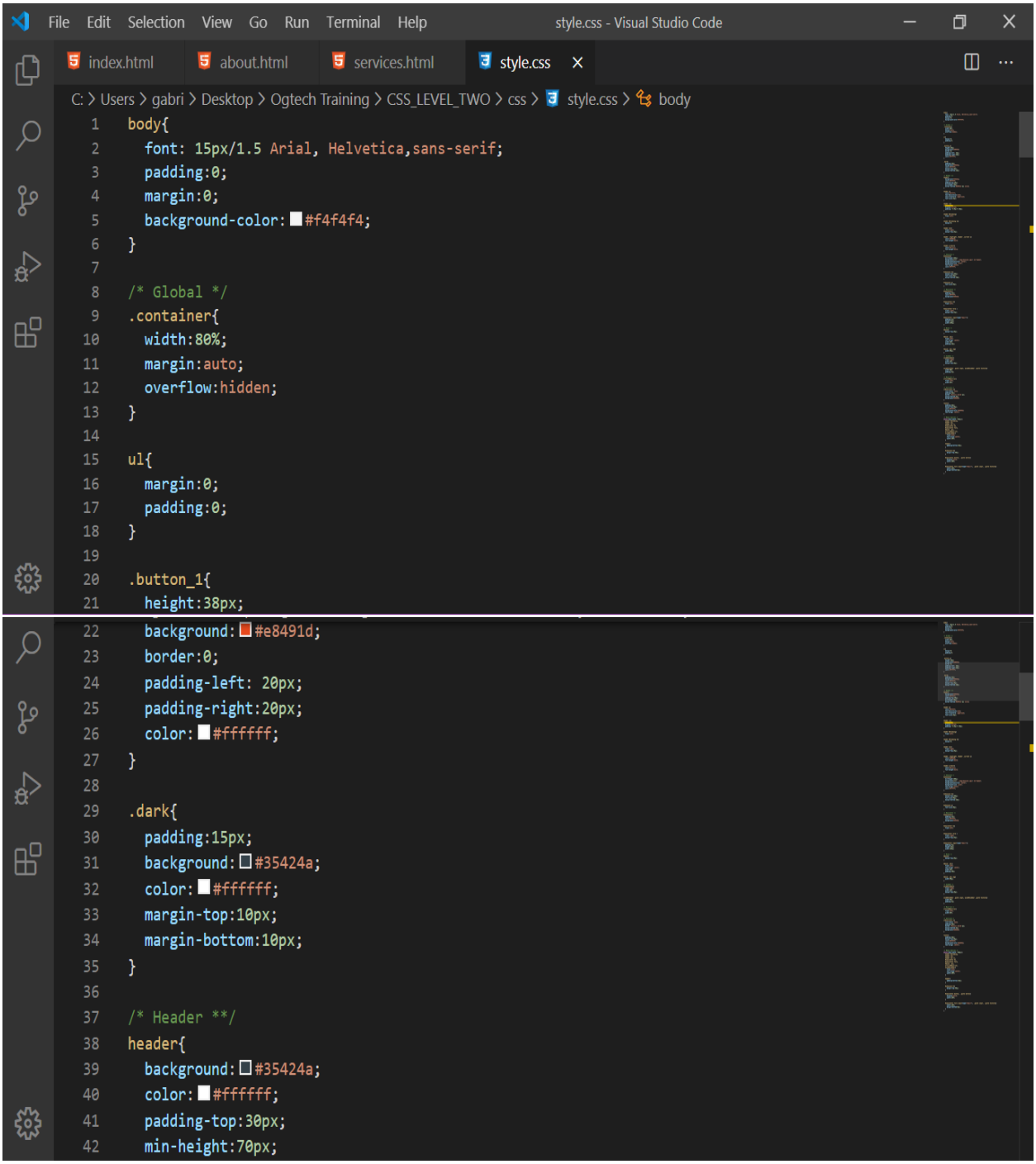
Html source codes for the service page

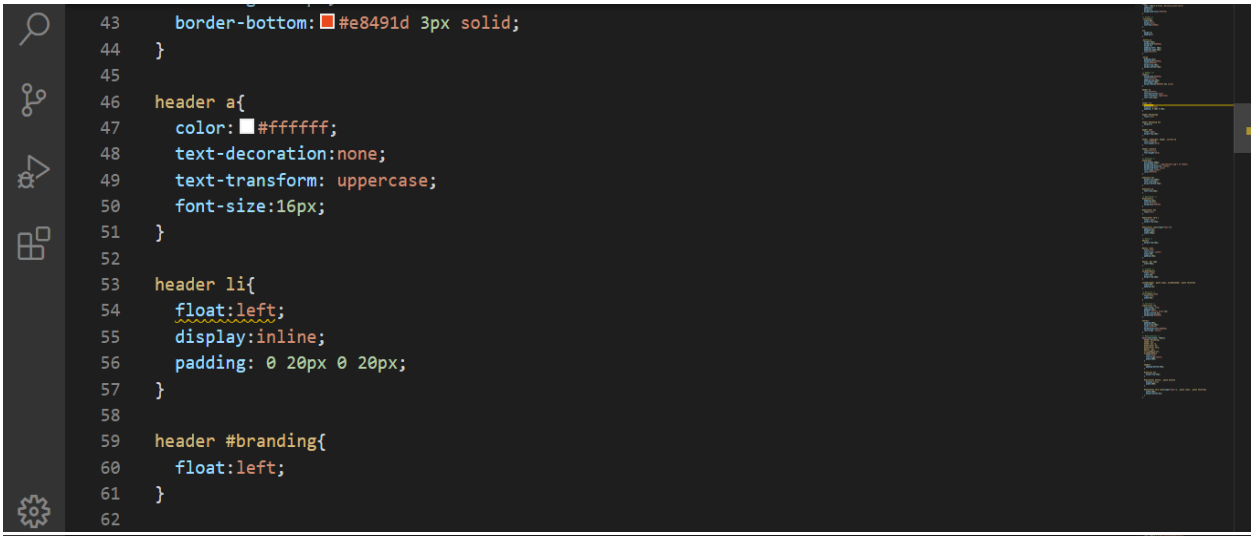


```html
        <aside id="sidebar">
          <div class="dark">
            <h3>Get A Quote</h3>
            <form class="quote">
              <div>
                <label>Name</label><br>
                <input type="text" placeholder="Name">
              </div>
              <div>
                <label>Email</label><br>
                <input type="email" placeholder="Emial Address">
              </div>
              <div>
                <label>Message</label><br>
                <textarea placeholder="Message"></textarea>
              </div>
              <button class="button_1" type="submit">Send</button>
          </form>
          </div>
        </aside>
      </div>
    </section>

    <footer>
      <p>Ogtech Web Deisgn, Copyright &copy; 2020</p>
    </footer>
  </body>
</html>
```

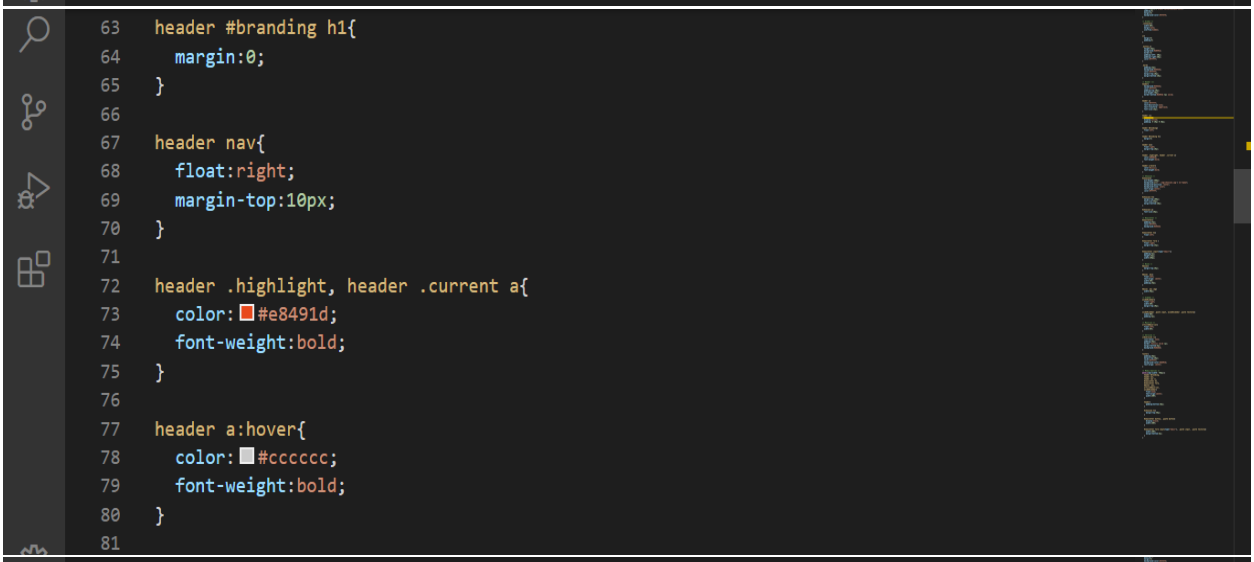
CSS source codes

We used one external **css** file for all of our html documents. – That's one advantage of using external css file.





```
  border-bottom: #e8491d 3px solid;
}

header a{
  color: #ffffff;
  text-decoration:none;
  text-transform: uppercase;
  font-size:16px;
}

header li{
  float:left;
  display:inline;
  padding: 0 20px 0 20px;
}

header #branding{
  float:left;
}

```



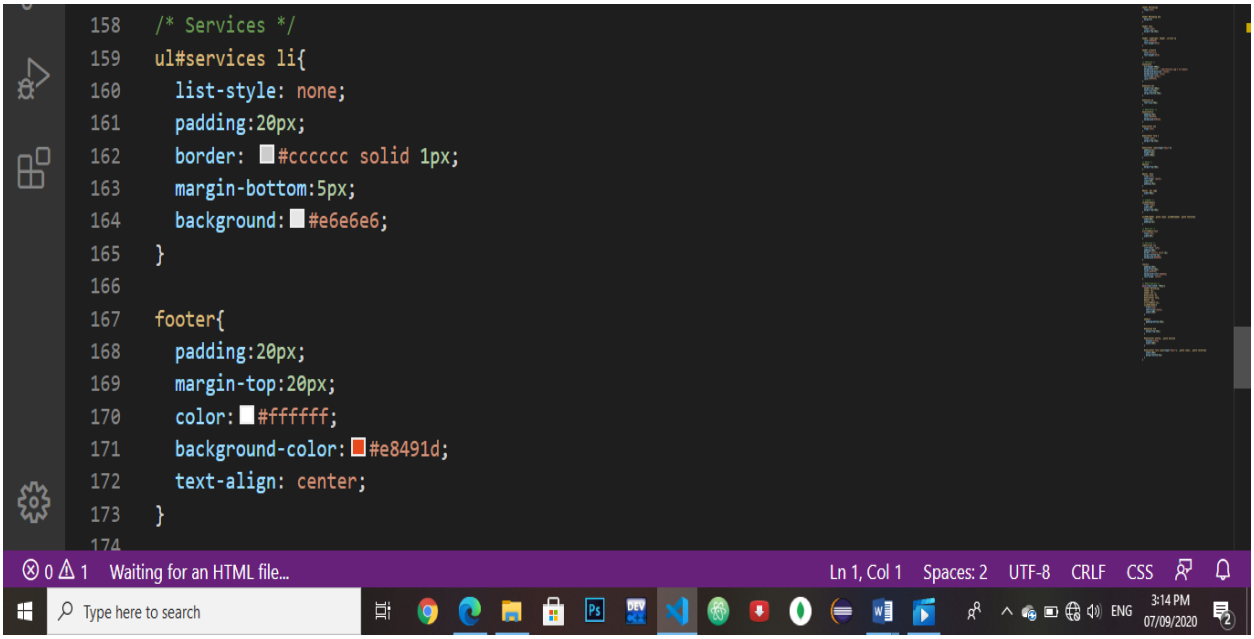
```
header #branding h1{
  margin:0;
}

header nav{
  float:right;
  margin-top:10px;
}

header .highlight, header .current a{
  color: #e8491d;
  font-weight:bold;
}

header a:hover{
  color: #cccccc;
  font-weight:bold;
}

```

```
/* Showcase */
#showcase{
  min-height:400px;
  background:url('../img/showcase.jpg') no-repeat;
  background-position: center;
  background-size: cover;
  text-align:center;
  color: #ffffff;
}

#showcase h1{
  margin-top:100px;
  font-size:55px;
  margin-bottom:10px;
}

#showcase p{
  font-size:20px;
}

```

```css
/* Newsletter */
#newsletter{
  padding:15px;
  color: #ffffff;
  background: #35424a
}

#newsletter h1{
  float:left;
}

#newsletter form {
  float:right;
  margin-top:15px;
}

#newsletter input[type="email"]{
  padding:4px;
  height:25px;
  width:250px;
}
```

```css

/* Boxes */
#boxes{
  margin-top:20px;
}

#boxes .box{
  float:left;
  text-align: center;
  width:30%;
  padding:10px;
}

#boxes .box img{
  width:90px;
}

```

```css
/* Sidebar */
aside#sidebar{
  float:right;
  width:30%;
  margin-top:10px;
}

aside#sidebar .quote input, aside#sidebar .quote textarea{
  width:90%;
  padding:5px;
}

/* Main-col */
article#main-col{
  float:left;
  width:65%;
}

```



Ogtech Network's Ltd

www.ogtechnetworks.com

Compiled

By

De-Prof

C.T.O of Ogtech Network's Ltd

www.deprof.com

Whatsapp : 08168297330

Call : 08091301280