



INTRODUCTION TO WEB DESIGN

HTML CSS JS WORDPRESS MAMP php elementor

6 WEEK COURSE VIA ZOOM

TUESDAYS/THURSDAYS NIGHTS 7:00 pm
April 13th Through May 20th

WHAT YOU WILL LEARN

- Set up a Web server on your computer
- Learn some basic HTML/CSS Code
- Create a basic html web page
- Create a database
- Download and install WordPress
- Connect Wordpress to the database
- Create a 4 page Wordpress Web Site

Suitable for Students in Grades 9-12

Limited Seats Available

Register Via Jot Form

<https://form.jotform.com/210695408004248>

Digital Creative Arts Centre

WANT TO BE A WEB DESIGNER?

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

A Simple HTML Document

This is how a simple html document is set up:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Index</title>
  </head>
  <body>

    <h1>My First Heading</h1>
    <p>My first paragraph. </p>

  </body>
</html>
```

Example Explained

- The `<!DOCTYPE html>` declaration defines that this document is an HTML5 document
- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about the HTML page
- The `<title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The `<h1>` element defines a large heading
- The `<p>` element defines a paragraph

HTML Documents

All HTML documents must start with a document type declaration: `<!DOCTYPE html>`.

The HTML document itself begins with `<html>` and ends with `</html>`.

The visible part of the HTML document is between `<body>` and `</body>`.

Example

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Index</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph. </p>

</body>
</html>
```

HTML Headings

HTML headings are defined with the `<h1>` to `<h6>` tags.

`<h1>` defines the most important heading. `<h6>` defines the least important heading:

Example

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

HTML Paragraphs

HTML paragraphs are defined with the `<p>` tag:

Example

```
<p>This is a paragraph. </p>  
<p>This is another paragraph.</p>
```

HTML elements with no content are called empty elements.

The `
` tag defines a line break, and is an empty element without a closing tag:

Example

```
<p>This is a <br> paragraph with a line break. </p>
```

This is a
Paragraph with a line break

HTML Links

HTML links are defined with the `<a>` tag:

Example

```
<a href="https://www.dcaclondon.ca">This is a link</a>
```

The link's destination is specified in the `href` attribute.

Attributes are used to provide additional information about HTML elements.

HTML Images

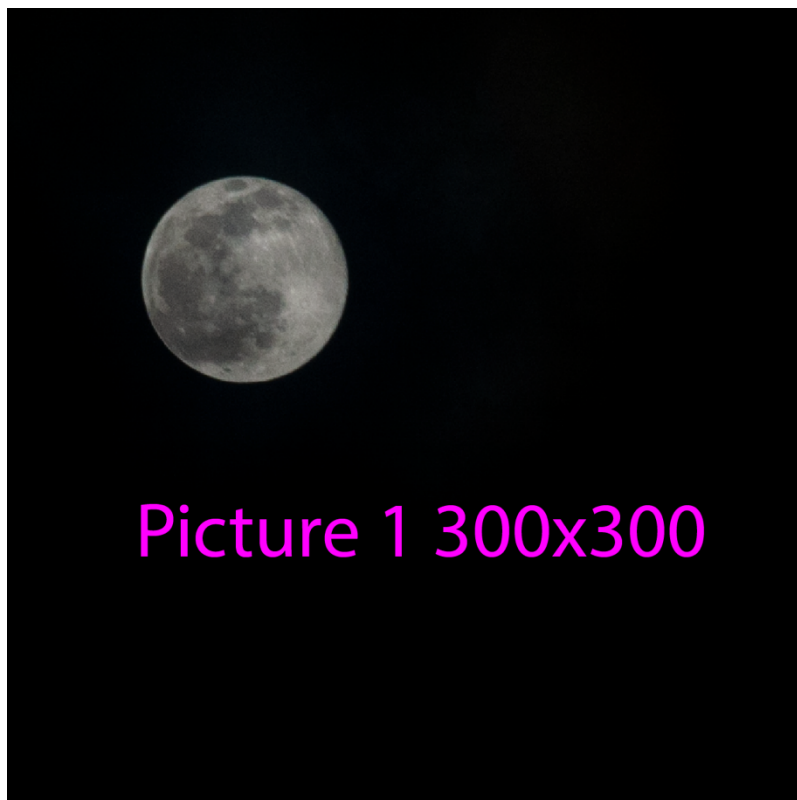
HTML images are defined with the `` tag.

The source file (`src`), alternative text (`alt`), `width`, and `height` are provided as attributes:

Example

```

```



If for some reason the image link is broken the (alt text) `MOON` will appear

PUTTING IT ALL TOGETHER

```
<!doctype html>

<html>

<head>

    <meta charset="UTF-8">

    <title>Simple Html page </title>

</head>

<body>

    <h1> </h1>

    <br>

    <p> </p>

</body>

</html>
```

CSS stands for Cascading Style Sheets.

CSS saves a lot of work. It can control the layout of multiple web pages all at once.

What is CSS?

Cascading Style Sheets (CSS) is used to format the layout of a webpage.

With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!

Tip: The word **cascading** means that a style applied to a parent element will also apply to all children elements within the parent. So, if you set the color of the body text to "blue", all headings, paragraphs, and other text elements within the body will also get the same color (unless you specify something else)!

Without CSS
Facebook would look
like this.

Jump to
[Sections of this page](#)
[Accessibility help](#)
Press alt + / to open this menu

[Facebook](#)

Email or Phone Password

[Forgotten account?](#)

Facebook helps you connect and share with the people in your life.

Create an account
It's quick and easy.
An error occurred. Please try again.
We couldn't create your account
We were not able to sign you up for Facebook.

First name

Surname

Mobile number or email address

Re-enter email address

New password

Birthday 14 ▼ Jan ▼ 1995 ▼

Gender ☒ Female ☐ Male ☐ Custom

Select your pronoun ▼

Your pronoun is visible to everyone.

Gender (optional)

By clicking Sign Up, you agree to our [Terms](#), [Data Policy](#) and [Cookie Policy](#). You may receive SMS notifications from us and can opt out at any time.

Using CSS

CSS can be added to HTML documents in 3 ways:

- **Inline** - by using the `style` attribute inside HTML elements
- **Internal** - by using a `<style>` element in the `<head>` section
- **External** - by using a `<link>` element to link to an external CSS file

The most common way to add CSS, is to keep the styles in external CSS files. However, in this tutorial we will use inline and internal styles, because this is easier to demonstrate, and easier for you to try it yourself.

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 sets the text color of the `<h1>` element to blue, and the text color of the `<p>` element to red:

Example

```
<h1 style="color:blue;">A Blue Heading </h1>
```

```
<p style="color:red;">A red paragraph.</p>
```


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 sets the text color of ALL the `<h1>` elements (on that page) to blue, and the text color of ALL the `<p>` elements to red. In addition, the page will be displayed with a "blue" background color:

Example

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Index</title>

  <style>

    body {
      background-color: purple;
      color: white;
    }

    p {
      font-size: 35px;
    }

    h1 {
      background-color: black;
      color: white;
    }

  </style>

</head>

<body>

  <h1>This is a heading</h1>
  <p>This is a paragraph. </p>
```

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

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:

Example

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Index</title>

  <link rel="stylesheet" href="styles.css">
</head>
<body>

  <h1>This is a heading</h1>
  <p>This is a paragraph. </p>

</body>
</html>
```

The external style sheet can be written in any text editor. The file must not contain any HTML code and must be saved with a `.css` extension.

Here is what the "styles.css" file looks like:

"styles.css":

```
body {  
  background-color: blue;  
}  
h1 {  
  color: blue;  
}  
p {  
  color: red;  
}
```

Tip: With an external style sheet, you can change the look of an entire web site, by changing one file!

CSS Colors, Fonts and Sizes

Here, we will demonstrate some commonly used CSS properties. You will learn more about them later.

The CSS `color` property defines the text color to be used.

The CSS `font-family` property defines the font to be used.

The CSS `font-size` property defines the text size to be used.

Example

Use of CSS color, font-family and font-size properties:

```
<!DOCTYPE html>  
<html>  
<head>  
  <style>  
    h1 {  
      color: blue;  
      font-family: Arial;  
      font-size: 20px;  
    }
```

```
p {  
  color: red;  
  font-family: Arial;  
  font-size: 14px;  
}  
  
</style>  
</head>  
<body>  
  
  <h1>This is a heading</h1>  
  <p>This is a paragraph. </p>  
  
</body>  
</html>
```

Link to External CSS

External style sheets can be referenced with a full URL or with a path relative to the current web page.

```
<link rel="stylesheet" href="styles.css">
```

A style sheet looks like this style.css and this is the code it is the same as the Internal code just on its own page ... this helps your web site run faster

```
@charset "UTF-8";  
/* CSS Document */
```

```
h1 {  
    color: blue;  
    font-family: Arial;  
    font-size: 20px;  
}
```

```
p {  
    color: red;  
    font-family: Arial;  
    font-size: 14px;  
}
```

LET'S BUILD A WEBSITE

Create a folder in the root folder that was created with MAMP.

Name the folder “MY_HTML_SITE”

Now Create and save 6 html pages and one CSS name them and save them in your “MY_HTML_SITE” folder.

index.html

page1.html

page2.html

page3.html

page4.html

page4.html

styles.css

Using Notepad to Create HTML files

You don't need fancy software to write or edit the HTML for a web page. Windows 10 Notepad is a basic text editor you can use for editing HTML; once you are comfortable writing your HTML in this simple editor, you can look into more advanced editors.

What to Know

- Use Windows 10 Notepad for editing HTML. Type *Notepad* in the Windows search bar to find and open Notepad.
- Add HTML to Notepad: Type HTML in Notepad > **File** > **Save as** > *filename.htm* > Encoding: **UTF-8** > **Save**.
- Use **.html** or **.htm** for file extension. Don't save the file with a .txt extension.

With Windows 10, Notepad became difficult for some users to find. There are several ways to open Notepad in Windows 10, but the five most often used methods are:

- Turn Notepad on in the **Start** menu. Select the **Start** button on the taskbar and then choose **Notepad**.
- Find it by searching. Type **notes** in the search box and select **Notepad** in the search results.
- Open Notepad by right clicking a blank area. Select **New** in the menu and select **Text Document**. Double-click the document.
- Press **Windows(logo)+R**, type **notepad** and then select **OK**.
- Select **Start**. Choose **All Apps** and then select **Windows Accessories**. Scroll to Notepad and select it.

How to Use Notepad With HTML

1. Open a new Notepad document.
2. Write some HTML in the document.
3. To save the file, select **File** in the Notepad menu and then **Save as**.
4. Enter the name **index.htm** and select **UTF-8** in the Encoding drop-down menu.
5. Use either .html or .htm for the extension. Do not save the file with a .txt extension.
6. Open the file in a [browser](#) by double-clicking on the file. You can also right-click and choose **Open with** to view your work.
7. To make additions or changes to the web page, return to the saved Notepad file and make the changes. Resave and then view your changes in a browser.

CSS and Javascript can also be written using Notepad. In this case, you save the file with the .css or .js extension.

How to Edit HTML With TextEdit

Write and edit HTML on a Mac

If you have a Mac, you don't need to download an HTML editor to write or edit HTML for a web page. The TextEdit program ships with all Mac computers. With it, and a knowledge of HTML, you can write and edit [HTML](#) code.

TextEdit, which works with files in a rich text format by default, must be in plain text mode to write or edit HTML.

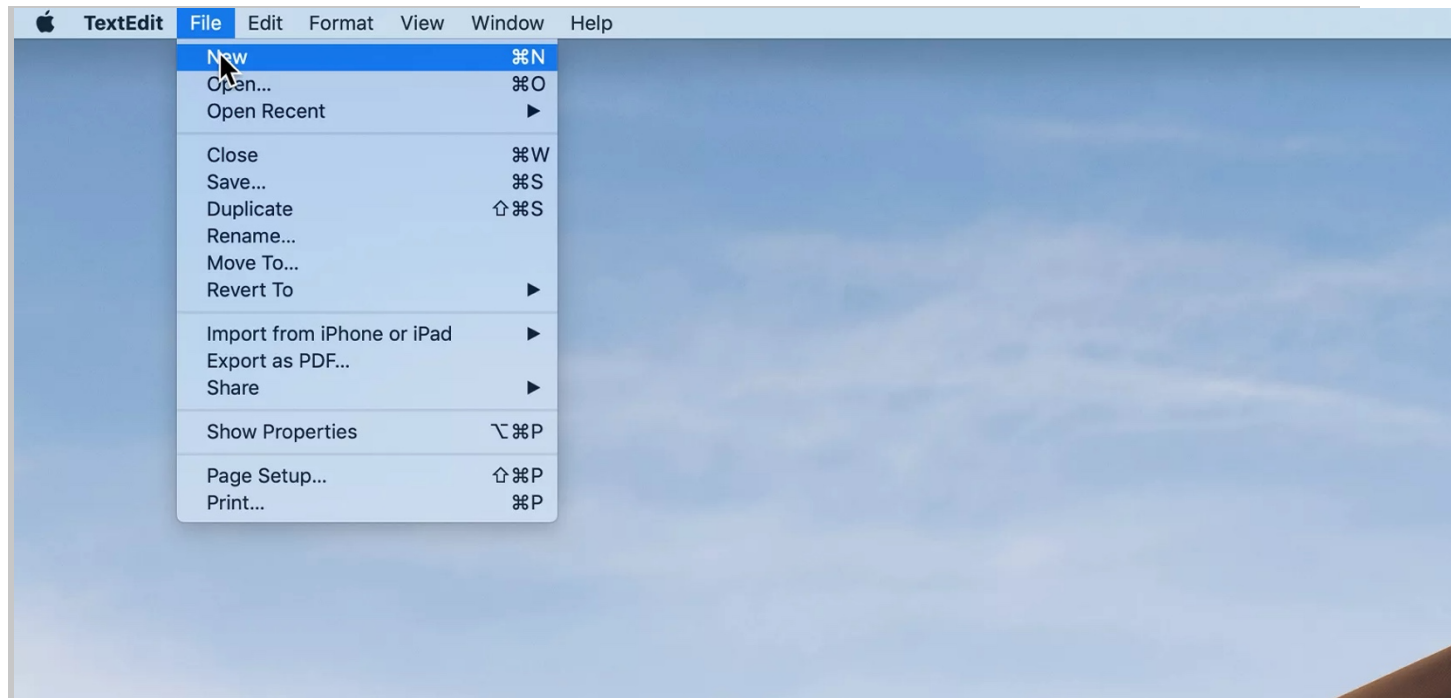
If you use TextEdit in rich text mode and save an HTML document with the [.html file extension](#) when you open that file in a [web browser](#), you see the HTML code, which isn't what you want.

To change how the HTML file displays in the browser, you change TextEdit to the plain text setting. You can do this on the fly or permanently alter the preferences if you plan to use TextEdit as your full-time code editor.

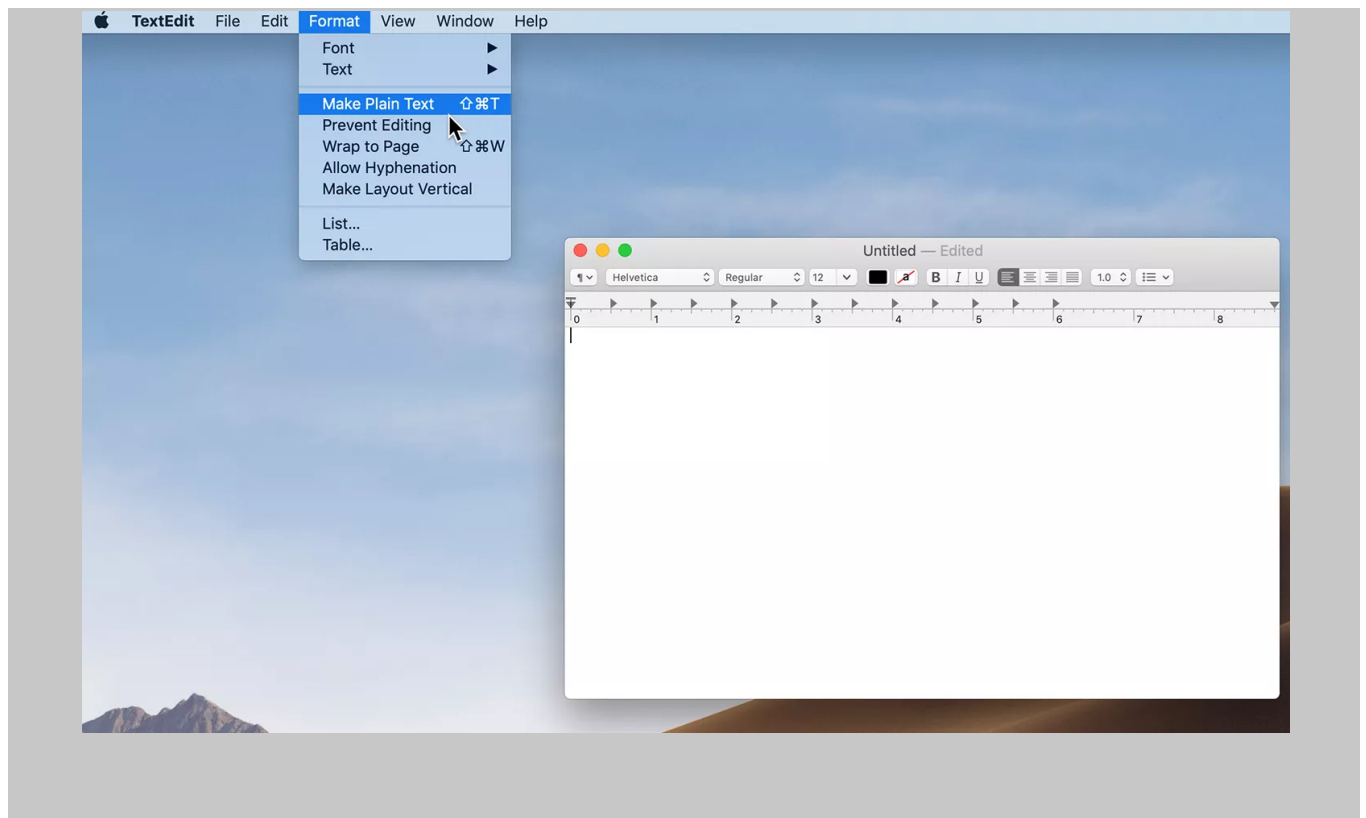
Create an HTML File in TextEdit

If you only occasionally work on [HTML files](#), you can make the change to plain text for a single document.

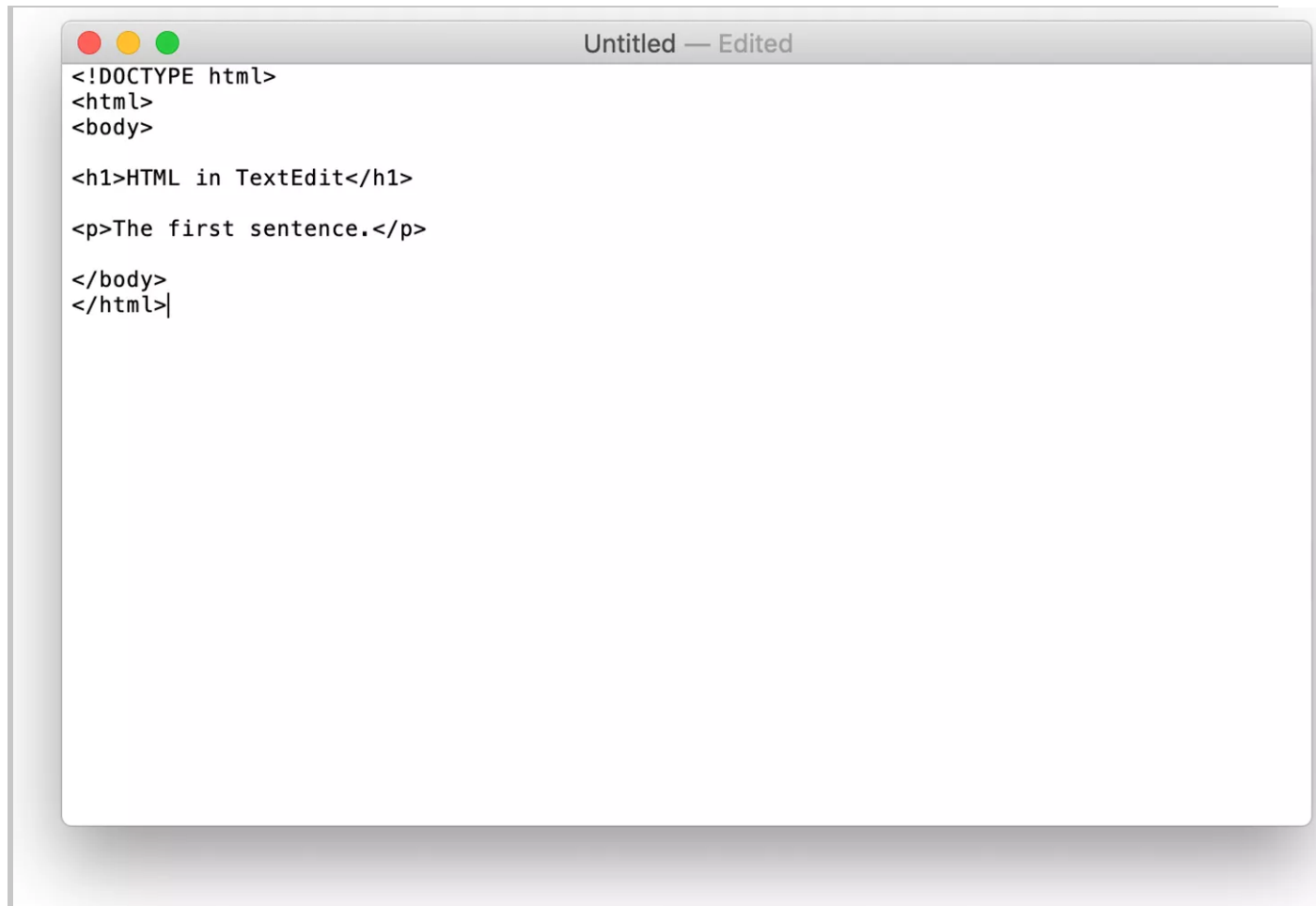
1. Open the TextEdit application on your Mac. Select **File > New** from the menu bar.



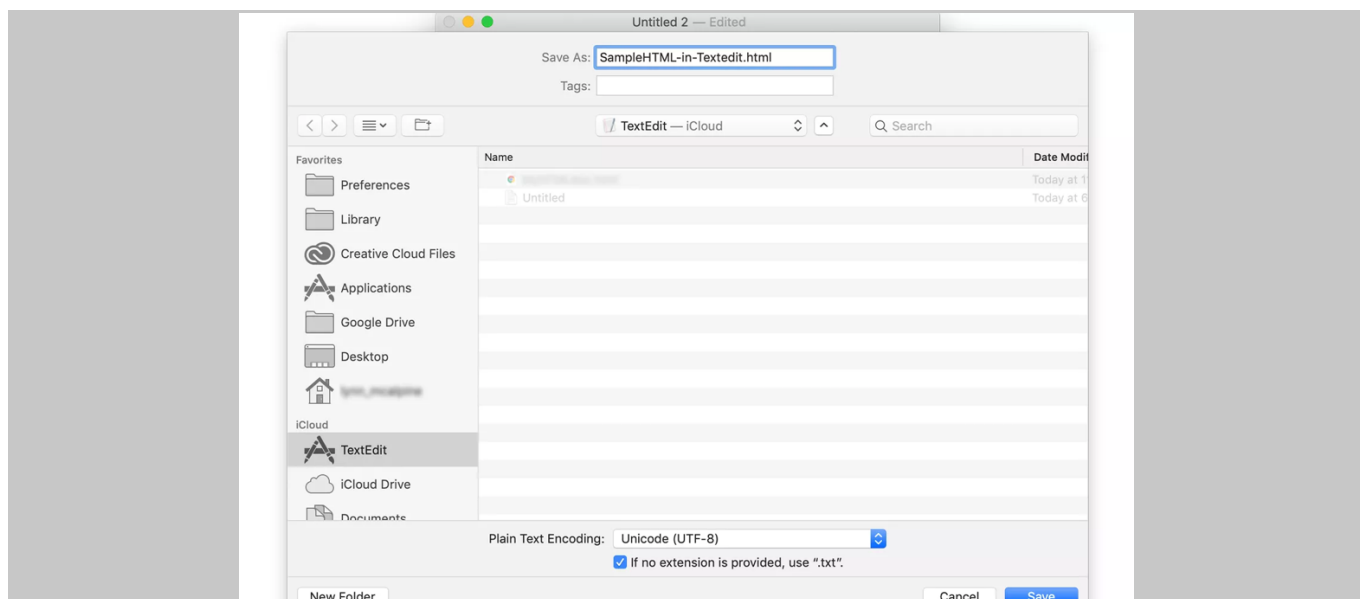
2. Select **Format** on the menu bar and click **Make Plain Text**. Confirm the plain text election in the window that opens by clicking **OK**.



1. Enter the HTML code. For example:



3. Click **File > Save**. Type a name for the file with a .html extension and choose a location to save the file.



4. Click **Save**. Confirm you want to use the **.html** extension in the screen that opens.

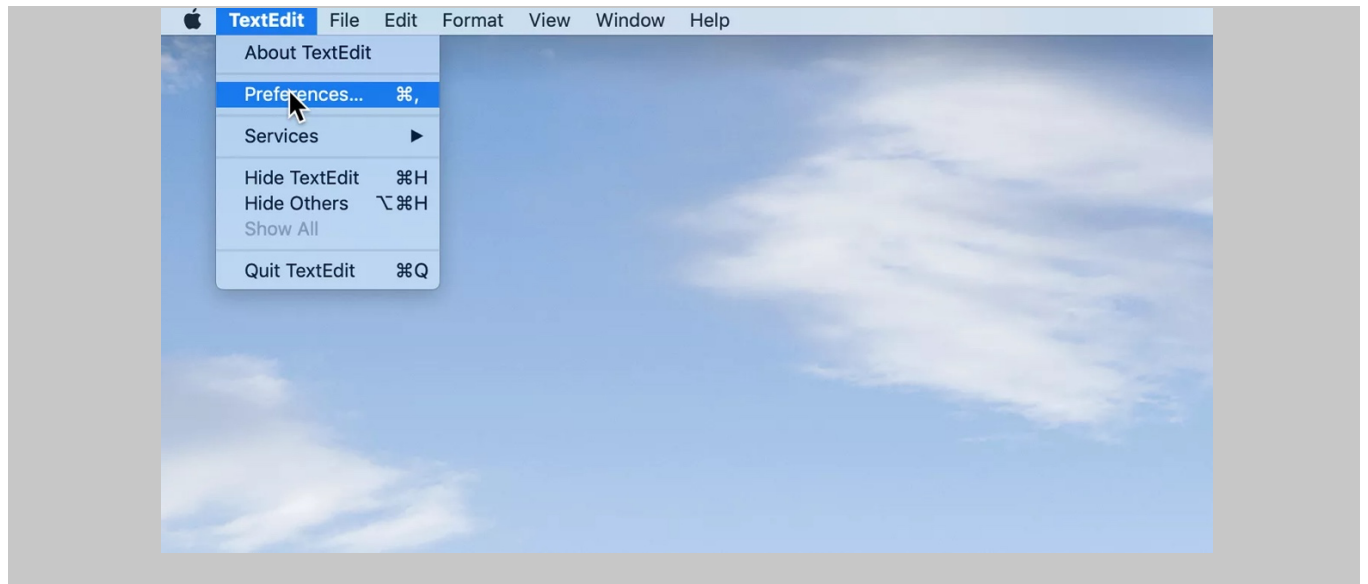
Test your work by dragging the saved file onto a browser. It should display exactly as you will see it when you publish it to the web. The example file dragged onto any browser should look like this:



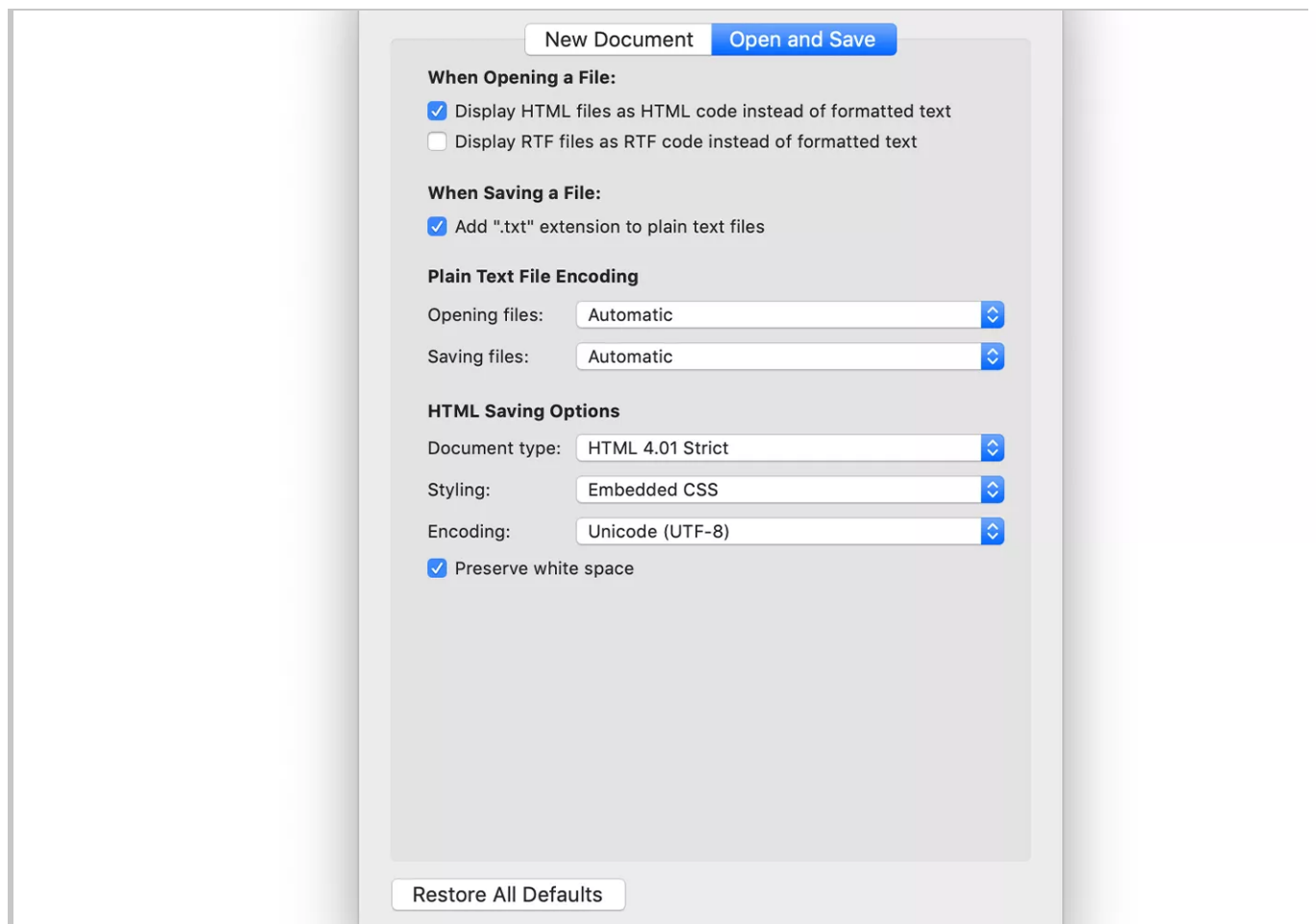
Instruct TextEdit to Open HTML as HTML

If you see any problems with your file, reopen it in TextEdit and make any necessary edits. If you open it in TextEdit and don't see the HTML, you need to make one more preference change. You only need to do this once.

5. Go to **TextEdit > Preferences.**



6. Click the **Open and Save** tab.



7. Put a check in the box next to **Display HTML files as HTML code instead of formatted text**. If you're using a version of [macOS](#) older than 10.7, this option is called **Ignore rich text commands in HTML pages**.

Changing the TextEdit Default Setting to Plain Text

If you plan to edit lots of HTML files with TextEdit, you might prefer to make the plain text format the default option. To do that, go to **TextEdit > Preferences** and open the **New Document** tab. Click the button next to **Plain text**.

PAGE 1- Open the page1.html in note pad and type the following code

```
<!doctype html>

<html>

<head>
    <meta charset="UTF-8">
    <title>Headings</title>

</head>

<body>

    <h1>This is Heading One</h1>
    <h2>This is Heading Two</h2>
    <h3>This is Heading Three</h3>

</body>
</html>
```

PAGE 2- Open the page2.html in note pad and type the following code

```
<!doctype html>

<html>

<head>

<meta charset="UTF-8">

<title>Paragraphs and Breaks</title>

</head>

<body>

    <h1>This is Heading One</h1>

    <br>

    <p>This is my FIRST paragraph</p>

    <h2>This Is Heading Two</h2>

    <br>

    <p>This is my SECOND paragraph</p>

    <h3>This is Heading Three</h3>

    <br>

    <p>This is my THIRD paragraph</p>

</body>

</html>
```

PAGE 3- Open the page3.html in note pad and type the following code

```
<!doctype html><html>

<head>
<meta charset="UTF-8">

<title>Images</title>

</head>

<body>

<h1>This is Heading One</h1>

    <br>



    <br>

    <p>This is my FIRST paragraph</p>

    <h2>This Is Heading Two</h2>

    <br>

    <img src="" alt="" width="" height="">

    <br>

    <p>This is my SECOND paragraph</p>

    <h3>This is Heading Three</h3>

    <br>

    <img src="" alt="" width="" height="">

    <br>

    <p>This is my THIRD paragraph</p>

</body>

</html>
```


PAGE 4- Open the page4.html in note pad and type the following code

```
<!doctype html>

<html>

<head>
    <meta charset="UTF-8">

    <title>Inline CSS</title>

</head>

<body>

<h1 style="color:blue;">This is Heading One</h1>

    <br>

    <img src="" alt="" width="" height="">

    <br>

    <p style="color:red;">This is my FIRST paragraph</p>

    <h2>This Is Heading Two</h2>

    <br>

    <img src="" alt="" width="" height="">

    <br>

    <p>This is my SECOND paragraph</p>

    <h3>This is Heading Three</h3>

    <br>

    <img src="" alt="" width="" height="">

    <br>

    <p>This is my THIRD paragraph</p>

</body>

</html>
```

PAGE 5- Open the page5.html in note pad and type the following code

```
<!doctype html>

<html>

<head>
<meta charset="UTF-8">

<title>Internal CSS</title>

    <style>

body {

    background-color: purple;

    color: white;

}

p {

    font-size: 35px;

}

h1 {

    background-color: black;

    color: white;

}

h2 {

    background-color: black;

    color: orange;

}
```

```
h3 {  
    background-color: black;  
    color: green;  
}
```

```
p1 {  
    font-family: "Times New Roman", serif;  
}
```

```
p2 {  
    font-family: "Arial Black", "sans-serif";  
}
```

```
p3 {  
    font-family: "Franklin Gothic Bold", " sans-serif"  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>This is Heading One</h1>
```

```
<br>
```

```
<img src="" alt="" width="" height="">
```

```
<br>
```

```
<p>This is my FIRST paragraph</p>
```

```
<h2>This Is Heading Two</h2>
```

```
<br>
```

```
<img src="" alt="" width="" height="">
```

```
<br>
```

```
<p>This is my SECOND paragraph</p>
```

```
<br>
```

```
<h3>This is Heading Three</h3>
```

```
<br>
```

```
<img src="" alt="" width="" height="">
```

```
<br>
```

```
<p>This is my THIRD paragraph</p>
```

```
</body>
```

```
</html>
```

PAGE 6- Open the page6.html in note pad and type the following code

```
<!doctype html>

<html>

<head>
<meta charset="UTF-8">
<title>External CSS</title>
<link rel="stylesheet" href="styles.css">
</head>

<body>

<h1>This is Heading One</h1>
<br>

<img src="" alt="" width="" height="">
    <br>
    <p>This is my FIRST paragraph</p>
    <h2>This Is Heading Two</h2>
    <br>
    <img src="" alt="" width="" height="">
    <br>
    <p>This is my SECOND paragraph</p>
    <h3>This is Heading Three</h3>
    <br>
    <img src="" alt="" width="" height="">
    <br>
    <p>This is my THIRD paragraph</p>
</body>

</html>
```

Styles.css- Open the Styles.css in note pad and type the following code

```
@charset "UTF-8";  
/* CSS Document */
```

```
body {  
    background-color: purple;  
    color: white;  
}
```

```
p {  
    font-size: 35px;  
}
```

```
h1 {  
    background-color: black;  
    color: white;  
}
```

```
h2 {  
    background-color: black;  
    color: orange;  
}
```

```
h3 {  
    background-color: black;  
    color: green;  
}
```

```
p1 {  
    font-family: "Times New Roman", serif;  
}
```

```
p2 {  
    font-family: "Arial Black", "sans-serif";  
}
```

```
p3 {  
    font-family: "Franklin Gothic Bold", " sans-serif"  
}
```

NOW WE CREATE AN INDEX PAGE AND A MENU

Menus are made with LISTS with LINKS inside as list items

```
<ul>  
<li><a href="page1.html">This Goes to Page one</a></li>  
</ul>
```

HTML Links

HTML links are defined with the `<a>` tag:

Example

```
<a href="https://www.dcaclondon.ca">This is a link</a>
```

The link's destination is specified in the `href` attribute.

Attributes are used to provide additional information about HTML elements.

HTML Lists

HTML lists allow web developers to group a set of related items in lists.

Example

An unordered HTML list:

- Item
- Item
- Item
- Item

An ordered HTML list:

1. First item
2. Second item
3. Third item
4. Fourth item

Unordered HTML List

An unordered list starts with the `` tag. Each list item starts with

the `` tag.

The list items will be marked with bullets (small black circles) by default:

Example

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

On the page will look like this

- Coffee
- Tea
- Milk

Ordered HTML List

An ordered list starts with the `` tag. Each list item starts with the `` tag.

The list items will be marked with numbers by default:

Example

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

On the page will look like this

1. Coffee
2. Tea
3. Milk

Let's Create a menu

MENU

index.html- Open the index.html in note pad and type the following code

```
<!doctype html>

<html>

<head>
<meta charset="UTF-8">
<title>External CSS</title>
<link rel="stylesheet" href="styles.css">
</head>

<body>

    <ul>

        <li><a href="index.html" target="_self">Index</a></li>

        <li><a href="page1.html" target="_self">Headings</a></li>

        <li><a href="page2.html" target="_self">Paragraphs and
Breaks</a></li>

        <li><a href="page3.html" target="_self">Adding Images</a></li>

        <li><a href="page4.html" target="_self">Inline CSS</a></li>

        <li><a href="page4.html" target="_self">Internal CSS</a></li>

        <li><a href="page5.html" target="_self">External CSS</a></li>

    </ul>

</body>

</html>
```

Now we need to add the below code between `<body></body>` tags and add this to each one of your html pages page1.html through page6.html

And add the following code to your styles.css

```
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    overflow: hidden;  
    background-color: #333;  
}  
  
li {  
    float: left;  
}  
  
li a {  
    display: block;  
    color: white;  
    text-align: center;  
    padding: 14px 16px;  
    text-decoration: none;  
}  
  
/* Change the link color to #111 (black) on hover */  
li a:hover {  
    background-color: red;  
}
```

This is the Schedule for the 6 Weeks

WEEK #	TUESDAY	THURSDAY
WEEK # 1	Introductions – Overview What is Web design and What tools are used? Google Chrome Developer Tools Let's Look at the Code	Difference Between HTML site and WordPress Download MAMP Install MAMP MAMP OVERVIEW Introduction to HTML
WEEK # 2	Create a basic HTML Web Page HTML Documents HTML Headings HTML Paragraphs HTML Links HTML Images	What Is CSS Use the HTML style attribute for inline styling Use the HTML <style> element to define internal CSS Use the HTML <link> element to refer to an external CSS file Use the HTML <head> element to store <style> and <link> elements Use the CSS color property for text colors Use the CSS font-family property for text fonts Use the CSS font-size property for text sizes
WEEK # 3	Use week 2 Lessons to build small HTML Site	Use PHP My admin to Create a database Modify WordPress Config File to connect to database Login to WordPress WORDPRESS OVERVIEW Change Password Delete Plugins Update WordPress Check Permalinks Change Title and Tagline
WEEK # 4	Installing Elementor Page Builder / and Hello Theme Elementor Overview Install Hello Theme Child Theme Themes and Theme Style Delete Pages Add Pages – Homepage, About, Photo Gallery Contact Us Slugs and Permalinks	Creating a Header and a footer Create a Logo Favicon Navigation Gallery Contact Us Slugs and Permalinks
WEEK # 5	Adding Elements to the pages Adding Text Photo gallery	sing Custom CSS in WordPress Formatting Text using HTML Embedding HTML Mobile Friendly SSL

WEEK # 6	BUILD your Site Teacher available for Questions	Let's Take a Look at your Sites Back up your Site ready to Migrate to a real live web server!
-----------------	--	--