Temasek Junior College

2023 JC2 H2 Computing

**Web Applications 1 - HTML Basics**

Objectives

- State that web pages are written using the HyperText Markup Language (HTML) and the Cascading Style Sheets (CSS) language.

- Use the "View Source" feature of a web browser to examine the HTML source code of any web page.

- Identify the tags, attributes, character references and comments that are present in a given static HTML document.

- Explain how HTML only describes the structure of a web page (in terms of elements) and not its presentation.

- Distinguish between tags for normal elements that require an end tag and tags for void elements that do not have an end tag.

- Use a variety of tags with relevant attributes to create web pages that meet a given set of requirements without producing any syntax errors.

---

1   Anatomy of a HTML Document

Webpages are written using **Hyper-Text Markup Language (HTML)**, a computer language specially used for creating the structure of webpages.

1.1   Viewing HTML Source Code on Google Chrome

On Google Chrome, go to https://www.example.com.
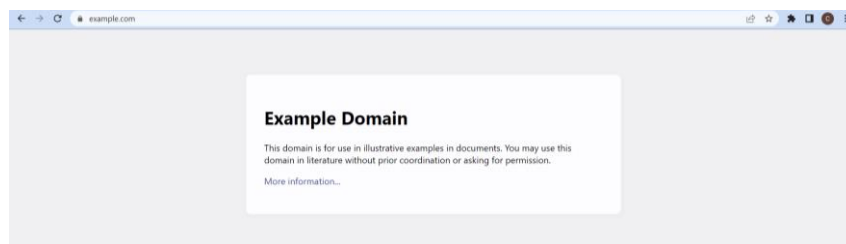
To view the HTML source of the page in Google Chrome, Press `CTRL` + `U` within the webpage.

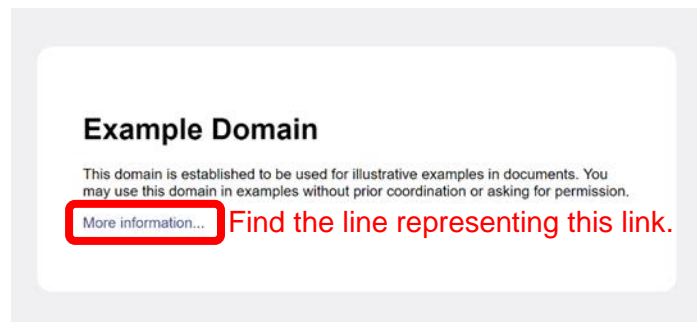Alternatively, right-click within the webpage and select `'View page source'`.

Note:
1) For examination laptops, the right-click has been disabled. You will need to know how to use the shortcut keys.

2) The HTML source obtained in this manner is similar to but may not be identical to the HTML script that was actually written at the backend.

```
Line wrap ☐
1  <!doctype html>
2  <html>
3  <head>
4      <title>Example Domain</title>
5
6      <meta charset="utf-8" />
7      <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
8      <meta name="viewport" content="width=device-width, initial-scale=1" />
9      <style type="text/css">
10     body {
11         background-color: #f0f0f2;
12         margin: 0;
13         padding: 0;
14         font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", "Helvetica Neue", Helvetica, Arial, sans-serif;
15
16     }
17     div {
18         width: 600px;
19         margin: 5em auto;
20         padding: 2em;
21         background-color: #fdfdff;
22         border-radius: 0.5em;
23         box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
24     }
25     a:link, a:visited {
26         color: #38488f;
27         text-decoration: none;
28     }
29     @media (max-width: 700px) {
30         div {
31             margin: 0 auto;
32             width: auto;
33         }
34     }
35     </style>
36 </head>
37
38 <body>
39 <div>
40     <h1>Example Domain</h1>
41     <p>This domain is for use in illustrative examples in documents. You may use this
42     domain in literature without prior coordination or asking for permission.</p>
43     <p><a href="https://www.iana.org/domains/example">More information...</a></p>
44 </div>
45 </body>
46 </html>
47
```

Exercise 1

On the page that displays the HTML source code (zoom in on the screenshot provided above), identify the line that represents the **"More information..."** link.

Answer

1.2 The Tag Structure

Further examine the HTML source code in **Section 1.1**.

Observe that the contents of the web page are surrounded by different text enclosed in a pair of angled-brackets (< >) e.g.

<h1>Example Domain</h1>

The text surrounded by angled-brackets, together with the angled-brackets, are called **tags,** which are special processing instructions meant for the web browser.

HTML tags are used to create the structure of a web page by organizing its contents into a "tree" of **elements**.

For instance, the HTML snippet below can be illustrated using a tree of elements.

```
<body>
    <h1>Mood Tracker</h1>
    <p>
        Today's Mood:
        <img src="happy.jpg" alt="happy">
    </p>
</body>
```


Designing webpages using HTML involves the use of many different tags.

1.3 Start Tags, End Tags and Tag Attributes

HTML tags can be further classified into **start tags** and **end tags**.

In general, each start tag corresponds to a single element.

Start tags may also have one or more **attributes** e.g.

and

Exercise 2

How do you tell the difference between start tags and end tags?

Answer

1.4   Normal Elements vs. Void Elements

Some start tags have a corresponding end tag while others do not.

Start tags with a corresponding end tag (e.g. `<body>`…`</body>`, `<h1>`…`</h1>` and `<p>`…`</p>`) correspond to **normal elements** that may contain a combination of text contents and other elements.

Start tags that do not have a corresponding end tag (such as `<img>` and `<input>`) correspond to **void elements** that must not contain any other content except that required by the tag.

Exercise 3

```
<html>
<head><title>Welcome Page</title></head>
<body>
    <p>Welcome to our department.</p>
    <img src="welcome.png" alt="Welcome">
</body>
</html>
```

(a)   In the HTML document above, identify all the normal elements.

Answer

(b)   Identify all the void elements.

Answer

1.5   Escape Codes / Character References

Special characters such as the angled-brackets < > have special meaning in HTML and would cause syntax errors if they are used without escaping them.

In HTML, escape codes are **character references** that start with an ampersand symbol (`&`) and end with a semi-colon (`;`).

Some common character references are as follows:

Character	Ampersand	Lesser Than	Greater Than	Quotation
	&	<	>	"
HTML Escape Code / Character Reference	`&`	`<`	`>`	`"`

Exercise 4

```
<p>I was sent to A&E in an ambulance.</p>
```

(a)   Identify the error in the above HTML snippet.

Answer

(b)   Rewrite the snippet to rectify the error so it works as intended on all browsers.

Answer

1.6   Comments

HTML documents may include comments readable by humans but ignored by the web browser.

Comments in HTML must start with `<!--` and end with `-->`.

Within the `<!-- … -->` structure, the contents of the comment must NOT contain two consecutive hyphen characters `--`.

Exercise 5

Which of the following HTML snippets is/are comments with no syntax errors?

(a)   `<!-- The markup language of the web -- HTML. -->`

(b)   `<!-- The markup language of the web - HTML. -->`

(c)   `<!- The markup language of the web - HTML. ->`

Answer

Note:
1) Proper commenting for HTML scripting is part of the examination requirements for proper and sufficient commenting.

2   Constructing a Sample Webpage

HTML documents are basically plain text files except that they use a `.html` extension.

In our curriculum, we shall use Notepad++ which has syntax highlighting functionality to create our HTML files.

Before looking at HTML in detail, let us first create a simple webpage to get a quick feel of its use.

Exercise 6

1.   Open Notepad++.

2.   Create a new file and save it in your working directory as `example.html`.

3.   Enter the following HTML.

```html
<!doctype html>
<html>
    <head>
        <title>My Web Page</title>
    </head>
    <body>
        <h1>Example</h1>
        <p>Hello, World!</p>
    </body>
</html>
```

4.   Save the file again to update it with the changes.

5.   Double-click `example.html` in your working directory to open it in a web browser.

Upon successfully completing the exercise, you should obtain the following:
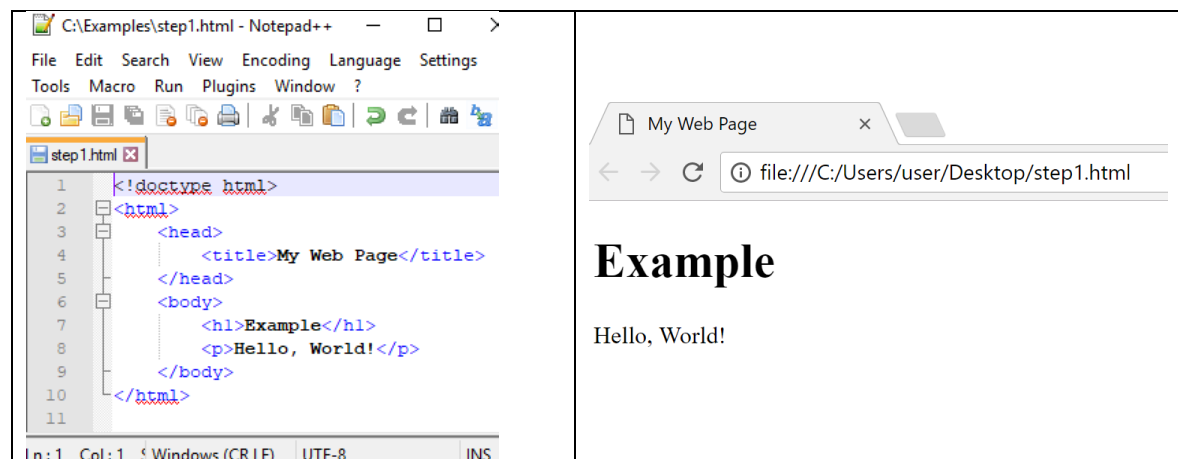
3 Basic HTML Tags

This section will introduce you to five groups of basic HTML tags:
- Required tags
- Structural tags
- List tags
- Text and media tags
- Table tags

Besides the above groups of tags, there are also
- Metadata tags (discussed together with CSS scripting)
- Forms tags (discussed separately as part of form creation)

3.1 Required Tags

All webpages written in HTML have a number of required elements.

These elements are written using **required tags**, which ensure that
- all HTML script of a webpage are contained within a single HTML element.
- metadata (data that describes other data) is placed within a head element, separate from the main content written within the body element.

(A) Document Type Declaration

Every HTML document must begin with a declaration that the document type is a HTML document.

This communicates to the web browser the version of HTML being used is HTML5, which is important as the web browser likely supports multiple versions of HTML.

The script for the declaration is as follows:

```
<!doctype html>
```

Note that `<!doctype html>` is technically not a tag.

(B) Creating a Single HTML Element

After declaring the document type, the remainder of the document must be enclosed between a `<html>` start tag and a `</html>` end tag.

This will tell the browser that all of the script within the tags is in HTML.

In addition, this creates a single HTML element that serves as the **root element** of the document's structure.

The script for creating a single HTML element is as follows:

```
<!doctype html>
<html>
     ...document goes here...
</html>
```

(C)   Head and Body Elements

Within the **`<html>`** … **`</html>`** tags are the **`<head>`** … **`</head>`** tags for the head element and **`<body>`** … **`</body>`** tags for the body element.

The head element contains metadata, which is usually the title of the webpage and any other information (excluding the main content) pertaining to the webpage.

The title is displayed on the browser tab itself while other metadata are usually not directly displayed anywhere in the browser. Additional steps are required to be taken in the browser to access these metadata.

The body element follows immediately after the head element and it contains the main content of the webpage. Everything within the body element will be displayed in the browser window.

The script for creating the head and body elements is as follows:

```
<!doctype html>
<html>
    <head>
        ...metadata goes here...
    </head>
    <body>
        ...main content goes here...
    </body>
</html>
```

(D)   Title Element

Within the **`<head>`** … **`</head>`** tags are the **`<title>`** … **`</title>`** tags for the title element. The title of the webpage that appears in the browser tab shall be written within the **`<title>`** … **`</title>`** tags.

This title will also be the title of the webpage displayed in search engine results and the default title being provided when the webpage is bookmarked.

The script for creating the title element is as follows:

```
<!doctype html>
<html>
    <head>
        <title>My Webpage</title>
    </head>
    <body>
        ...main content goes here...
    </body>
</html>
```

My Web Page

file:///C:/Users/user/De
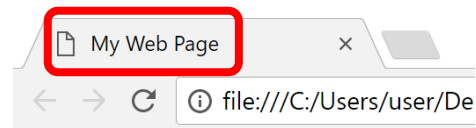
While the **`<title>`** … **`</title>`** tags are the only metadata tags required within the **`<head>`** … **`</head>`** tags, other optional metadata tags can also be included where appropriate.

Exercise 7

For each of the tag below, determine whether it is a tag for a normal or a void element.

Tag	Description	Element Type
`<html>`	Root element	Normal / Void
`<head>`	Metadata	Normal / Void
`<title>`	Document title	Normal / Void
`<body>`	Main content	Normal / Void

Answer

3.2 Structural Tags

Structural tags are used within the **<body>** … **</body>** tags to organize the main content that goes into the body element. These tags facilitate ease of reading and navigation within a webpage when used appropriately to divide long pieces of content into logical parts such as headings and paragraphs.

Only structural tags for headings, paragraphs, line breaks and horizontal rules will be discussed. The <div> and <span> structural tags shall be discussed together with CSS.

(A) Headings

There are 6 levels of headings in HTML. They are written using the following tags: **<h1>…</h1>**, **<h2>…</h2>**, **<h3>…</h3>**, **<h4>…</h4>**, **<h5>…</h5>** and **<h6>…</h6>**.

An example of a script for writing headers for the content in the body element is as follows:

```
<!doctype html>
<html>
    <head>
        <title>HTML Basics</title>
    </head>
    <body>
        <h1>Introduction</h1>
        <h1>What is HTML?</h1>
        <h2>Purpose</h2>
        <h2>Syntax</h2>
    </body>
</html>
```

(B) Paragraphs

Content in the body element can be organized into paragraphs using the **<p>…</p>** tags.

An example of the script for organizing content in the body element into paragraphs is as follows:

```
<!doctype html>
<html>
    <head>
        <title>HTML Basics</title>
    </head>
    <body>
        <h1>Introduction</h1>
        <p>This short article will explain what HTML is...</p>
        <p>It will start by going through the purpose and...</p>
        <h1>What is HTML?</h1>
        <h2>Purpose</h2>
        <p>The purpose of HTML is to describe the...</p>
        <h2>Syntax</h2>
        <p>Tags in HTML are always surrounded by angled...</p>
    </body>
</html>
```

## Exercise 8

1.   Open Notepad++.

2.   Create a new file and save it in your working directory as **headers-paras.html**.

3.   Enter the following HTML.

```
<!doctype html>
<html>
    <head>
        <title>HTML Basics</title>
    </head>
    <body>
        <h1>Introduction</h1>
        <p>This short article will explain what HTML is...</p>
        <p>It will start by going through the purpose and...</p>
        <h1>What is HTML?</h1>
        <h2>Purpose</h2>
        <p>The purpose of HTML is to describe the...</p>
        <h2>Syntax</h2>
        <p>Tags in HTML are always surrounded by angled...</p>
    </body>
</html>
```

4.   Save the file again to update it with the changes.

5.   Double-click **headers-paras.html** in your working directory to open it in a web browser.

Upon successfully completing the exercise, you should obtain the following.
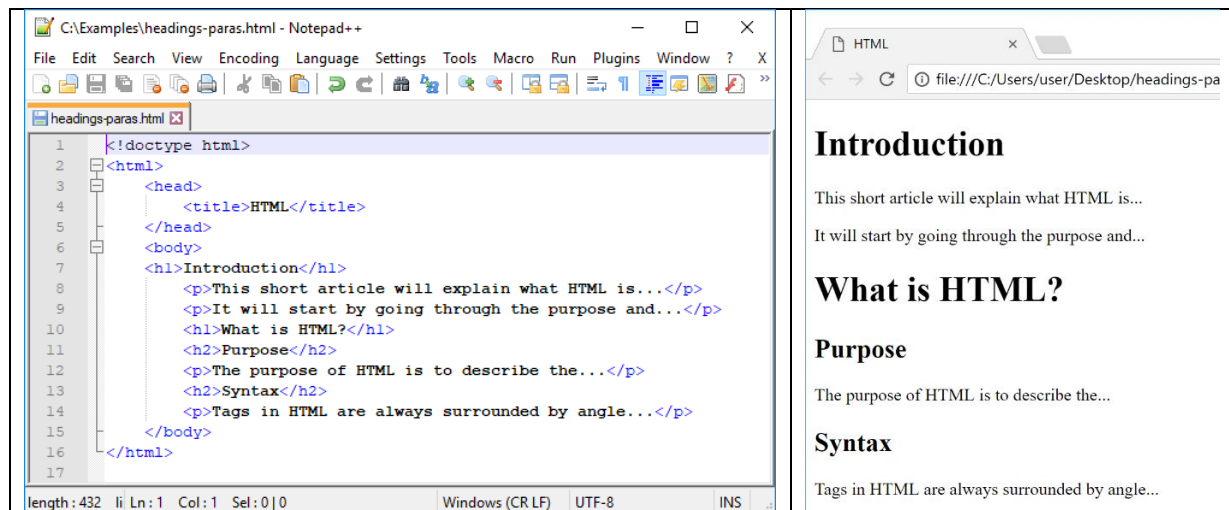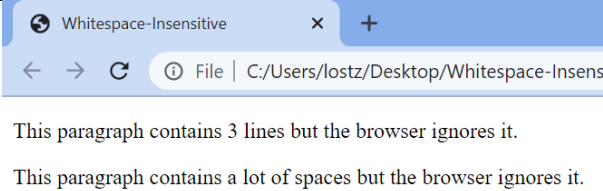
(C) Line Breaks

HTML is whitespace insensitive. This means that additional lines and spaces are ignored. An example is shown below.

```
<!doctype html>
<html>
    <head>
        <title>Whitespace-Insensitive</title>
    </head>
    <body>
        <p>This paragraph
        contains 3 lines
        but the browser ignores it.</p>

        <p>This paragraph
        contains          a lot of spaces
        but the          browser          ignores it.</p>
    </body>
</html>
```

Whitespace-Insensitive   ×   +

← → C   ⓘ File | C:/Users/lostz/Desktop/Whitespace-Insens

This paragraph contains 3 lines but the browser ignores it.
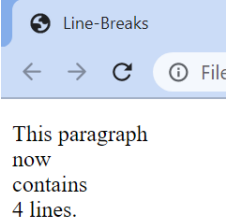
This paragraph contains a lot of spaces but the browser ignores it.

Observe that the two paragraphs in the body element are both displayed as a single line with one space between each word regardless the number of lines and amount of whitespace used.

To display content in a specific number of lines, the line break tag **
** needs to be used. An example of how the **
** tag can be used is shown below.

```
<!doctype html>
<html>
  <head>
    <title>Line-Breaks</title>
  </head>
  <body>
    <p>This paragraph <br> now <br> contains <br> 4 lines.</p>
  </body>
</html>
```

Line-Breaks

← → C   ⓘ File

This paragraph
now
contains
4 lines.

Observe that the paragraph is now displayed as 4 lines by inserting 3 **
** tags.

Note that **
** tags DO NOT have an end tag i.e. the **</br>** tag DOES NOT exist.

(D) **Horizontal Rule**

Content in the same webpage can be separated into sections using the horizontal rule tag **<hr>**. When used, a horizontal line will be inserted across the page to separate two sections of the content. An example of how the **<hr>** tag can be used is shown below.

```
<!doctype html>
<html>
  <head>
    <title>Horizontal-Rule</title>
  </head>
  <body>
    <p>This is Paragraph 1.</p>
    <p>This is Paragraph 2.</p>
    <hr><p>The horizontal rule tag separates Paragraph 3.</p>
  </body>
</html>
```
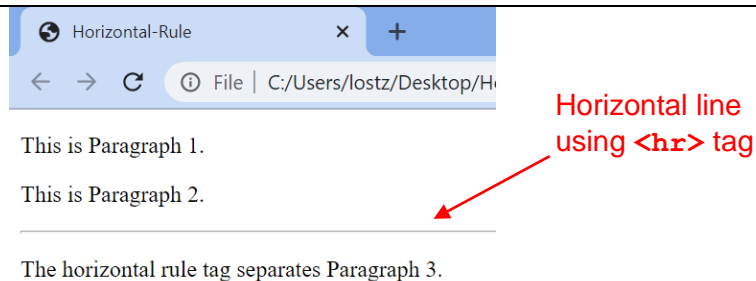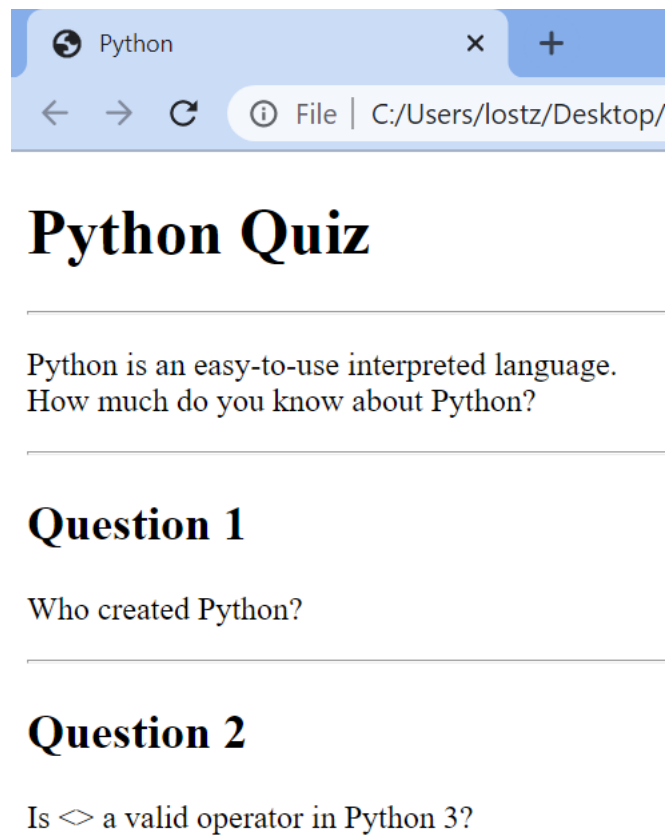

Exercise 9
For each of the tag below, determine whether it is a tag for a normal or a void element.

Tag	Description	Element Type
<h1> to <h6>	Headers	Normal / Void
<p>	Paragraph	Normal / Void
 	Line break	Normal / Void
<hr>	Horizontal rule	Normal / Void

Answer

Exercise 10

Use only the `<html>`, `<head>`, `<title>`, `<body>`, `<h1>`, `<h2>`, `<p>`, `<br>` and `<hr>` tags to create the following web page and save it as `quiz.html`.

Python Quiz

Python is an easy-to-use interpreted language.
How much do you know about Python?

Question 1

Who created Python?

Question 2

Is <> a valid operator in Python 3?

Answer

3.3   List Tags

List tags are used within the `<body>` and `</body>` tags to organize content into lists.

Two types of lists can be created: unordered lists and ordered lists.

(A)   Unordered Lists

An unordered list will appear as a bulleted list in the webpage. It can be created using the `<ul>`…`</ul>` tags.

Each item in the list is then entered using the `<li>`…`</li>` tags.

An example of how an unordered (bulleted) list can be created is shown below.

```
<!doctype html>
<html>
    <head>
        <title>Unordered-Lists</title>
    </head>
    <body>
        <h1>Creating an Unordered List</h1>
        <ul>
            <li>Item 1</li>
            <li>Item 2</li>
            <li>Item 3</li>
        </ul>
    </body>
</html>
```
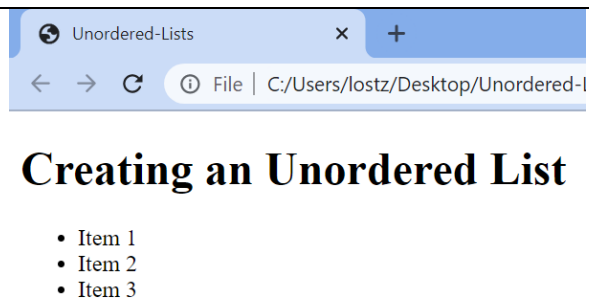
(B) <u>**Ordered Lists**</u>

An ordered list will appear as a numbered list in the webpage. It can be created using the `<ol>`…`</ol>` tags.

Each item in the list is then entered using the `<li>`…`</li>` tags.

An example of how an ordered (numbered) list can be created is shown below.

```
<!doctype html>
<html>
    <head>
        <title>Ordered-Lists</title>
    </head>
    <body>
        <h1>Creating an Ordered List</h1>
        <ol>
            <li>Item 1</li>
            <li>Item 2</li>
            <li>Item 3</li>
        </ol>
    </body>
</html>
```


Creating an Ordered List

1. Item 1
2. Item 2
3. Item 3
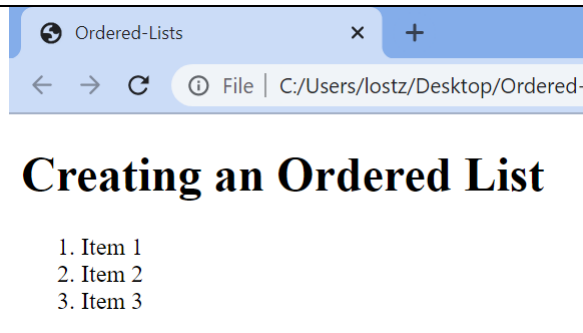
3.4 Text and Media Tags

(A) Basic Text Formatting

In general, the overall appearance of webpages should be controlled using Cascading Style Sheets (CSS).

Nevertheless, basic text formatting to bold type and italics type can be done using the **...** and **<i>…</i>** tags respectively. An example of text formatting using these tags is shown below.

```
<!doctype html>
<html>
    <head>
        <title>Text Formatting</title>
    </head>
    <body>
        <p>Text can be in <b>bold</b><p>
        <p>Text can be in <i>italics</i></p>
        <p>Text can be in both <b><i>bold and italics</i></b><p>
    </body>
</html>
```
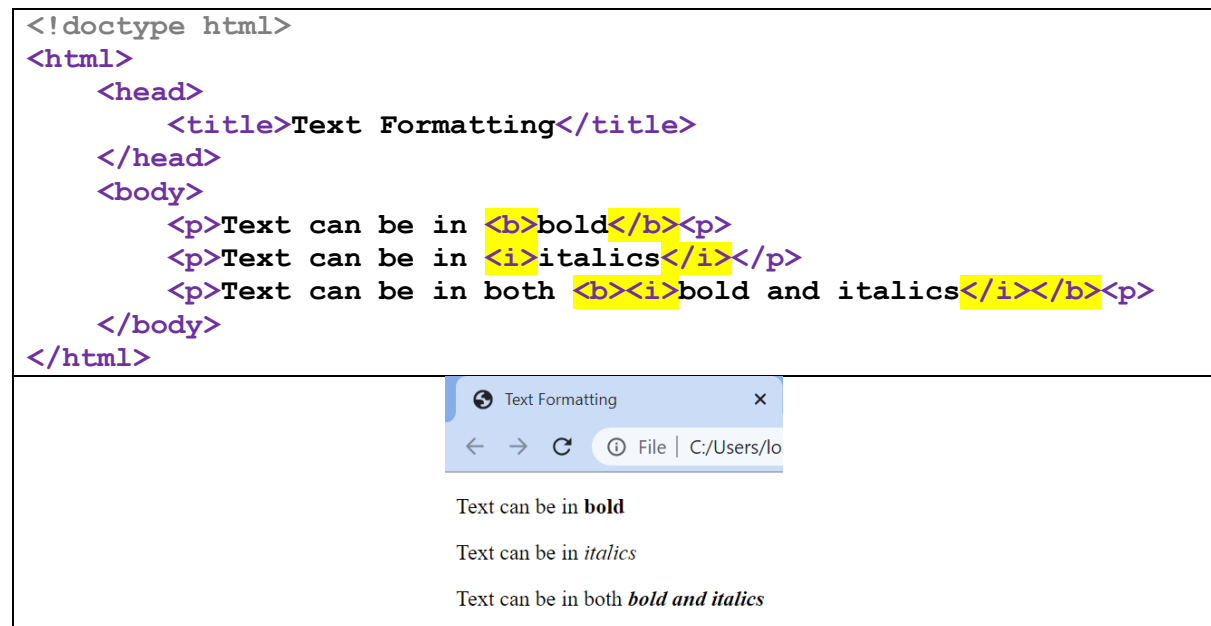

(B) Links

Before discussing how links can be created, we need to first understand what an **absolute URL** and a **relative URL** are.

> ***Absolute URL***
> A URL that contains the scheme component (often http:// or https://)
> e.g. **http://www.example.com** and **https://www.example.com**.
>
> ***Relative URL***
> An address without the scheme component
> e.g. **www.example.com**. and **quiz/question1.html**.

A webpage may contain links to
* other webpages
* files in the local drive located in the folder where the webpage is stored in
* files in the local drive located in the sub-folders of the folder where the webpage is stored in

Such links can be created using the anchor tags **<a>…** with a hypertext reference attribute **href** set to the URL where a specified text display will link to.

Linking to another webpage

To create a link to another webpage, the `href` attribute must be <u>set to an absolute URL</u>.

For example, the following line in HTML will create a link to another webpage with URL `https://www.example.com`. The link will be displayed as the text `Example`.

```
<a href="https://www.example.com">Example</a>
```

```
<!doctype html>
<html>
    <head>
        <title>Text Formatting</title>
    </head>
    <body>
        <p><a href="https://www.example.com">Example</a></p>
    </body>
</html>
```


On clicking the link
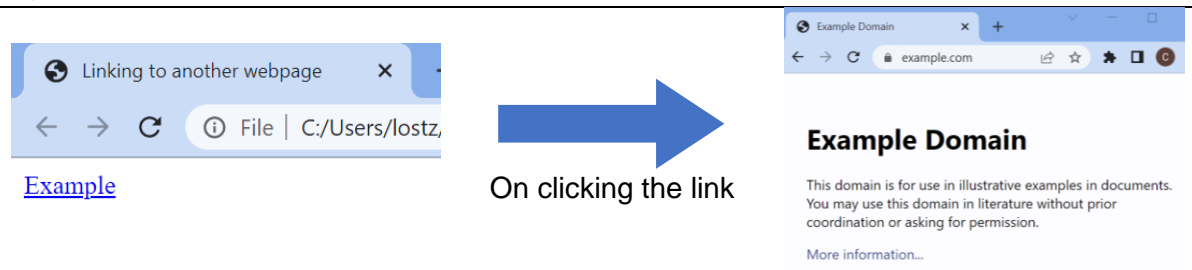
Exercise 11

Which of the following lines in HTML creates a link to TJC's homepage using the text TJC?

(A)   `<a href="TJC">www.temasekjc.moe.edu.sg</a>`
(B)   `<a href="TJC">https://www.temasekjc.moe.edu.sg</a>`
(C)   `<a href="www.temasekjc.moe.edu.sg">TJC</a>`
(D)   `<a href="https://www.temasekjc.moe.edu.sg">TJC</a>`

<u>Answer</u>

Linking to another file

To create a link to a file in the local drive, the `href` attribute should be <u>set to a relative URL</u>. The web browser will interpret the link as a path to the file.

For example, the following line in HTML creates a link to the `question1.html` file stored in the `quiz` sub-folder of the folder where the webpage is stored.

```
<a href="quiz/question1.html">View Question 1</a>
```

When an absolute URL is erroneously written as a relative URL, the web browser will interpret the relative URL as a filename instead.

For example, the following line in HTML will cause the web browser to interpret `www.example.com` as the name of a file stored in the same folder as that of the webpage.

<div align="center">

`<a href="www.example.com">Example</a>`

</div>

Notes on the use of relative URLs
- When using relative URLs to create a link as a path to a file, the forward slash (`/`) should be used. This is as opposed to the backward slash (`\`) which the Windows OS uses in specifying a path to a file.

- In addition, a "double full-stop" (`..`) can be used to refer to the parent folder i.e. the folder which is one level above the folder where the webpage is stored in.

Exercise 12
The table below shows the URL of the current webpage and the relative URL used to create a link. Write down the corresponding resulting URL.

Current URL	Relative URL	Resulting URL
`http://www.sg/en`	`example`	
`http://www.sg/en/`	`example`	
`http://www.sg/en/sg`	`example/`	
`http://www.sg/en/sg/`	`example/`	
`http://www.sg/en/sg`	`../example`	
`http://www.sg/en/sg/`	`../example`	
`http://www.sg/en/sg/`	`/example`	

Answer

Current URL	Relative URL	Resulting URL
`http://www.sg/en` Webpage "stored" in "folder" http://www.sg	`example`	
`http://www.sg/en/` Webpage "stored" in "folder" http://www.sg/en	`example`	
`http://www.sg/en/sg` Webpage "stored" in "folder" http://www.sg/en	`example/`	
`http://www.sg/en/sg/` Webpage "stored" in "folder" http://www.sg/en/sg	`example/`	
`http://www.sg/en/sg` Webpage "stored" in "folder" http://www.sg/en	`../example` One level above i.e. http://www.sg	
`http://www.sg/en/sg/` Webpage "stored" in "folder" http://www.sg/en/sg	`../example` One level above i.e. http://www.sg/en	
`http://www.sg/en/sg/` Webpage "stored" in "folder" http://www.sg/en/sg	`/example` Main level i.e. http://www.sg	

Observe that the path of the "folder" where a webpage is "stored" ends at the last forward slash used.

Exercise 13

Create three HTML documents using the following directory structure and contents.
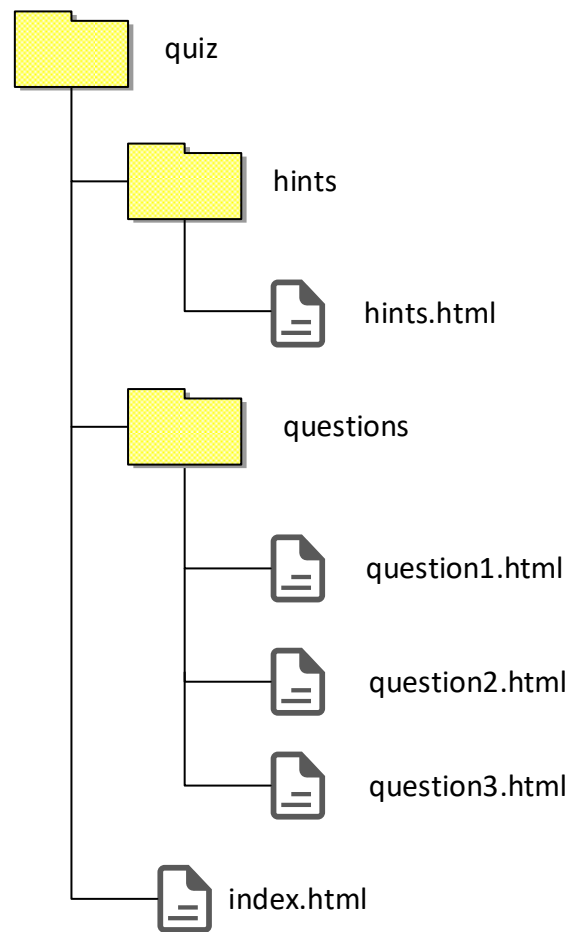
Open home.html and verify that the two links to first.html and second.html work as expected.
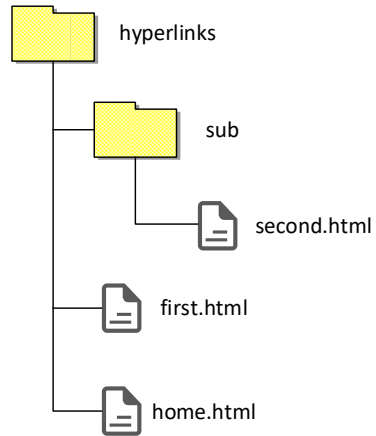
Answer

Exercise 14
Study the following directory structure.

(a) Write down the HTML line that creates a link from `question1.html` to `index.html`.

(b) Write down the HTML line that creates a link from `question 2.html` to `hints.html`.

Answer

(C) Images

An image can be displayed in a webpage using the image tag `<img>` with the source `src` and alternative `alt` attributes.

The `src` attribute needs to be set to the URL of the image (either absolute URL or relative URL, depending on the source of the image). For maximum compatibility, the image file should either be a **GIF** (`.gif`) or **JPEG** (`.jpg`) or **PNG** (`.png`) file.

To support users who cannot or do not wish to view the image, the `alt` attribute is used to provide a text description that can be used as a replacement for the image.

For example, the following line in HTML will display the image `example.png` which is located in the `images` sub-folder of the folder where the webpage is located in the local drive. If the image cannot be displayed or if the user chooses not to view the image, the text "`Example of an image`" will be displayed.

```
<img src="images/example.png" alt="Example of an image">
```

Exercises 15, 16 and **17** provide examples of how to display an image
- on its own
- within a paragraph
- as a hyperlink to another webpage

Before proceeding further, perform the following steps:
- Create a folder `display_images` in your working directory e.g. Desktop.
- Find an image in the **GIF** (**.gif**) or **JPEG** (**.jpg**) or **PNG** (**.png**) format and save it to the `display_images` folder using the name `sample_image`.

Displaying an image on its own

Exercise 15
Create a webpage using the following HTML script. Save it as `images_own.html` in the `display_images` folder.

Change the `.png` extension of the image file in the script to the appropriate extension if necessary.

```
<!doctype html>
<html>
    <head>
        <title>Images-Own</title>
    </head>
    <body>
        <h1>Images can appear...</h1>
        <h2>...on their own!</h2>
        <img src="sample_image.png" alt="SAMPLE">
    </body>
</html>
```

Displaying an image within a paragraph

Exercise 16

Create a webpage using the following HTML script. Save it as **images_para.html** in the **display_images** folder.

Change the **.png** extension of the image file in the script to the appropriate extension if necessary.

```
<!doctype html>
<html>
    <head>
        <title>Images-Para</title>
    </head>
    <body>
        <h1>Images can appear...</h1>
        <h2>...in the middle of a paragraph!</h2>
        <p>
            Sample
            <img src="sample_image.png" alt="SAMPLE">
            Image
        </p>
    </body>
</html>
```

Displaying an image as a hyperlink to another webpage

Exercise 17

Create a webpage using the following HTML script. Save it as **images_link.html** in the **display_images** folder.

Change the **.png** extension of the image file in the script to the appropriate extension if necessary.

```
<!doctype html>
<html>
    <head>
        <title>Images-Link</title>
    </head>
    <body>
        <h1>Images can appear...</h1>
        <h2>...in a hyperlink!</h2>
        <p>
            Click:
            <a href="https://www.example.com">
                <img src="sample_image.png" alt="www.example.com">
            </a>
        </p>
    </body>
</html>
```

When **Exercises 15**, **16** and **17** are done correctly, the resulting webpages will look similar to the following:

Images can appear...

...on their own!

Images can appear...

...in the middle of a paragraph!

Sample

Image

Images can appear...

...in a hyperlink!

Click:

On clicking image

Example Domain

This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission.
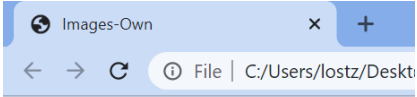
More information...

Now rename the image `sample_image` as `not_displayed_image`.

Reload the output pages in Exercises **15**, **16** and **17**.
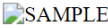
If done correctly, you should expect the following output pages where the value of the `alt` attribute is displayed.
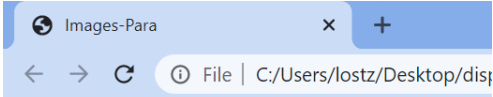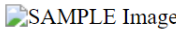
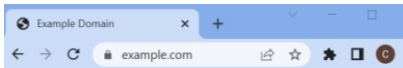| Expected output for **Exercise 15**	Expected output for **Exercise 16**

Expected output for **Exercise 17**

On clicking text

Exercise 18
For each of the tag below, determine whether it is a tag for a normal or a void element.

Tag	Description	Element Type
`<b>`	Bold	Normal / Void
`<i>`	Italics	Normal / Void
`<a>`	Hyperlink	Normal / Void
`<img>`	Image	Normal / Void

Answer

3.5   Table Tags

So far, we have seen how a HTML script is written in accordance to the document's flow, arranged from top-to-bottom, left-to-right.

The same top-to-bottom, left-to-right ordering can be used to create a table using the table tags **<table>…</table>**, **<tr>…</tr>**, **<th>…</th>** and **<td>...<td>**.

Note:
- **tr** stands for table row,
- **th** stands for table header and
- **td** stands for table data.

Consider the following HTML script for creating a table:

```
<!doctype html>
<html>
    <head>
        <title>Create-Table</title>
    </head>
    <body>
        <table>
            <tr>
                <th>Name</th>
                <th>Gender</th>
                <th>Scores</th>
            </tr>
            <tr>
                <td>Bala</td>
                <td>Male</td>
                <td>23</td>
            </tr>
            <tr>
                <td>Siti</td>
                <td>Female</td>
                <td>25</td>
            </tr>
            <tr>
                <td>Thomas</td>
                <td>Male</td>
                <td>22</td>
            </tr>
        </table>
    <body>
</html>
```

Create-Table	×	+
← → C ⓘ File │ C:/Users/lostz/Desktop/Create-Table.html		

Name	**Gender**	**Scores**
Bala	Male	23
Siti	Female	25
Thomas	Male	22

From the above script, we can see that the start tag **<table>** indicates the start of one table while the **</table>** end tag indicates the end of the table.

Within the **<table>** and **</table>** tags, the entire table will be constructed row by row from top-to-bottom using the **<tr>** start tag to indicate the start of each row and **</tr>** end tag to indicate the end of each row.
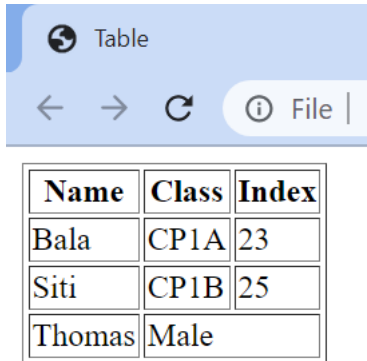
The first row being created is usually the header row. Its information is created column by column from left-to-right using the `<th>` start tag to indicate the start of each column header cell and the `</th>` end tag to indicate the end of each column header cell.

For the data in the remaining rows, each row is also created column by column from left-to-right using the `<td>` start tag to indicate the start of each data cell and the `</td>` end tag to indicate the end of each data cell.

Observe that the table generated does not have borders. This can be addressed by including the value of the `border` attribute in the `<table>` start tag. In addition, the width of a cell can also be change by including the value of the column span `colspan` attribute in either the `<th>` or `<td>` start tags.

An example is shown below.

```
<!doctype html>
<html>
    <head>
        <title>Table</title>
    </head>
    <body>
        <table border=1px>
            <tr>
                <th>Name</th>
                <th>Class</th>
                <th>Index</th>
            </tr>
            <tr>
                <td>Bala</td>
                <td>CP1A</td>
                <td>23</td>
            </tr>
            <tr>
                <td>Siti</td>
                <td>CP1B</td>
                <td>25</td>
            </tr>
            <tr>
                <td>Thomas</td>
                <td colspan=2>Male</td>
            </tr>
        </table>
    <body>
</html>
```

In the above script, the table start tag `<table border=1px>` has the `border` attribute included and specified to a thickness of one pixel `1px`. Hence the output table now has a border.

In addition, the table data start tag `<td colspan=2>` of the cell for the gender of Thomas (last row) has the column span attribute `colspan` set to `2`. This means that the cell will span a width of 2 cells instead of the default width of a single cell.

More about table formatting will be discussed in the use of CSS.

Exercise 19

Create the following table and save it as `risk.html`.

	Low Probability	**High Probability**
Low Impact	Low Risk	Medium Risk
High Impact	Medium Risk	High Risk

Answer

Exercise 20

For each of the tag below, determine whether it is a tag for a normal or a void element.

Tag	Description	Element Type
`<table>`	Table	Normal / Void
`<tr>`	Table Row	Normal / Void
`<th>`	Table Header Cell	Normal / Void
`<td>`	Table Data Cell	Normal / Void

Answer