# CHAPTER 2 – THE INTRODUCTION & ELEMENTS

# TOPICS TO BE COVERED

- **The Introduction**

  Markup tags tell the Web browser how to display the page.
- **HTML5 structure and anatomy**

  Used to define the meanings of the contents of an HTML page
- **HTML Text Formatting**

  Define sections of text, usually a blank line is inserted before and after the element itself.

**More info on this topic can be found in e-books provided in LMS.**
Presentation slides made by Fifah S., M. Khalid.

# TOPICS TO BE COVERED

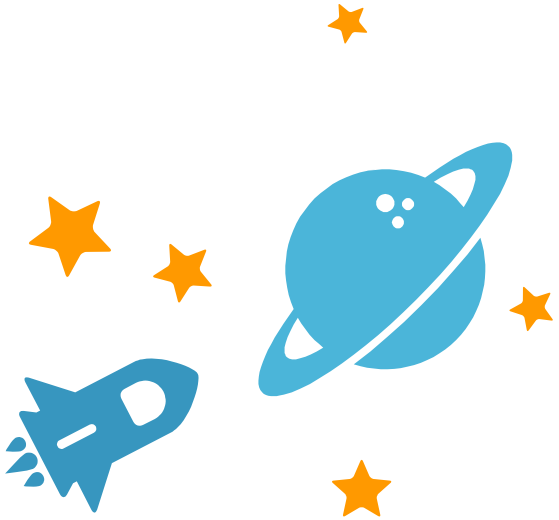
- **HTML List**

  Used to group a set of items in a webpage

- **HTML Hyperlink**

  Specifies relationship between the current document and an external resource.

**More info on this topic can be found in e-books provided in LMS.**
Presentation slides made by Fifah S., M. Khalid.



# THE INTRODUCTION

Markup tags tell the Web browser how to display the page.

# HYPERTEXT MARKUP LANGUAGE

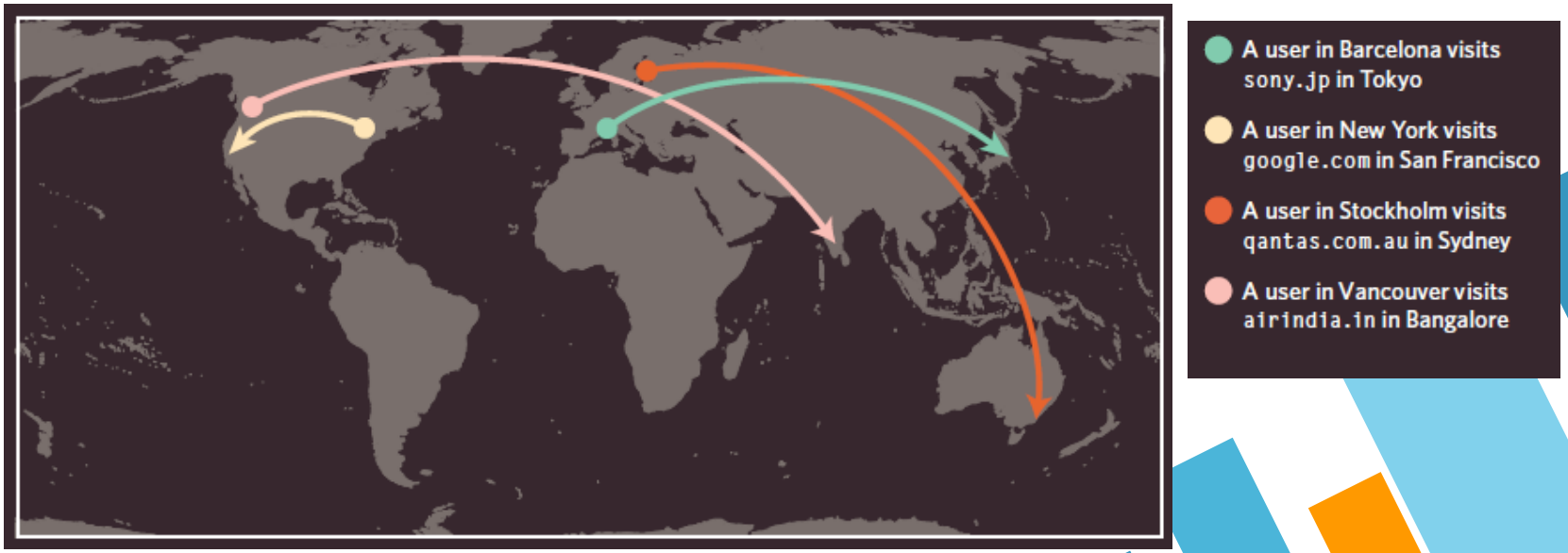
- Hypertext is a link.
- A link is a text / phrase / image in a webpage that points to another web page.
- When you click one of these links, your browser transports you to the other page immediately.
- **HTML defines the overall structure of the web page.**

# ..THE HYPERLINKS

Hyperlinks can be part of a hypermedia document to:

- Another part of the same document file.
- Another document file on the same server computer.
- Another document file on a server computer located elsewhere in the world.

- When you visited a website, the web server hosting that site could be anywhere in the world.
- To find the location of the web server, your browser will first connect to Domain Name System (DNS) server.



# ...THE MARKUP

- define the meanings of the contents of an HTML page.
- Is used to structure the contents of a web page.
- To provide a semantic web and machine readability, *contents* and *presentation* should be separated.

# ...THE LANGUAGE

- Simply imply that HTML is a programming language.
- Nothing to do with computer programming.
- Rather, HTML is a “language” that has a small collection of words that you use to specify how you want your text to appear.

# CASCADING STYLE SHEET (CSS)

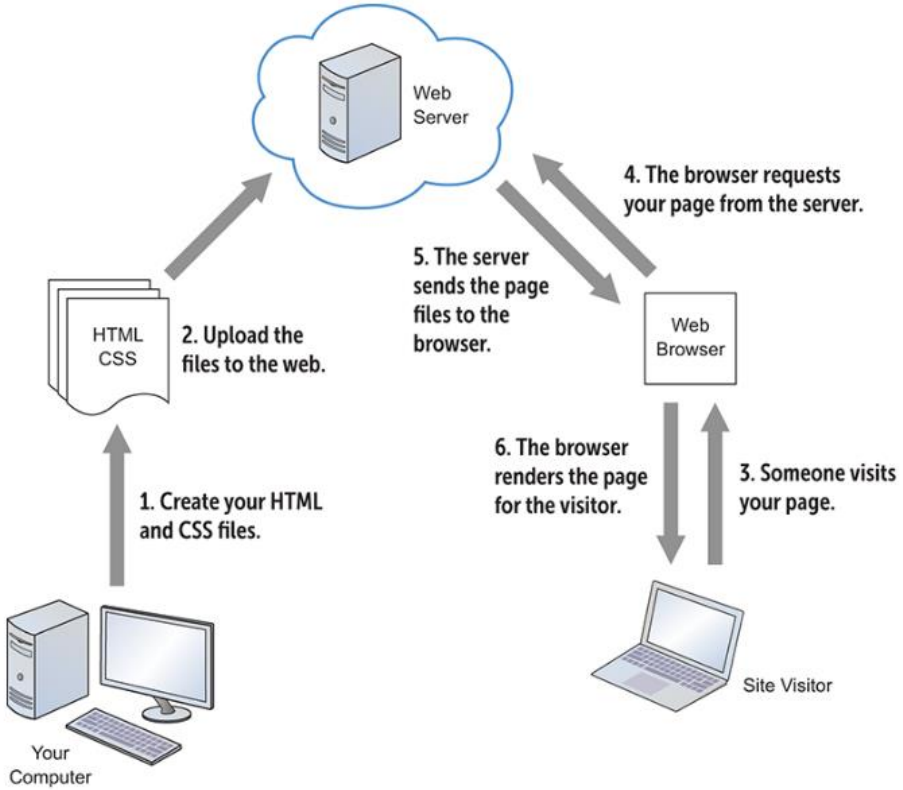
- An instruction to the browser to modify how contents is being displayed on the webpage.
- The modifications usually are formatting-related, such as changing the typeface or text color, control page layout and create animated effects.
- **CSS defines the visual presentation of the web page.**

# DON'TS WITH HTML AND CSS

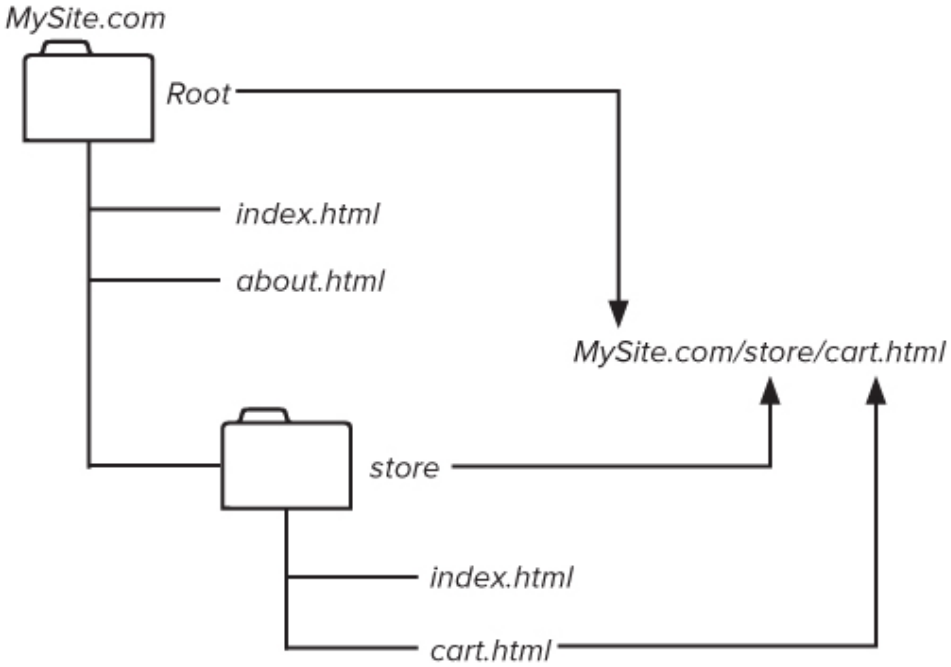
Just to name a few:

- Get data from a server database or other remote addresses.
- Process data submitted through a form
- Handle user accounts, logins and passwords.
- Add, hide, or remove web page elements on-the-fly.

# CREATING A WEBSITE USING HTML AND CSS



# CREATING A WEBSITE USING HTML AND CSS




The URL of a webpage reflects its location in the website directory structure

# NAMING CONVENTIONS

1. **Use proper suffix for your files.**

   HTML, CSS and JavaScript files must end with *.html, .css and .js* respectively. Web graphics must be labeled according to their file format *.gif, .png or .jpg*.

2. **Never use character spaces within filenames**

   It is common to use an underline character or hyphens (_) to visually separate words within filenames, such as *student_info.html*

3. **Avoid special characters.**

   such as ?, %, #, /, :, ;, * etc. Limit filenames to letters, numbers, underscores, hyphens and periods.

4. **Filenames may be case-sensitive**

   Depending on your server configuration. Consistently using all lowercase letters in filenames, although not necessary, is one way to make your filename easier to manage.

# NAMING CONVENTIONS

5. **Keep filenames short.**
   Short names keep the character count and file size of your HTML file in check. If you really really must give the file a long, multiword name, you can separate the words with hyphens, such as *a-long-document-filename.html*, to improve readability.
6. **Self-imposed conventions.**
   It is helpful to develop a consistent naming scheme for huge sites.

# CHALLENGES OF CREATING WEBSITE

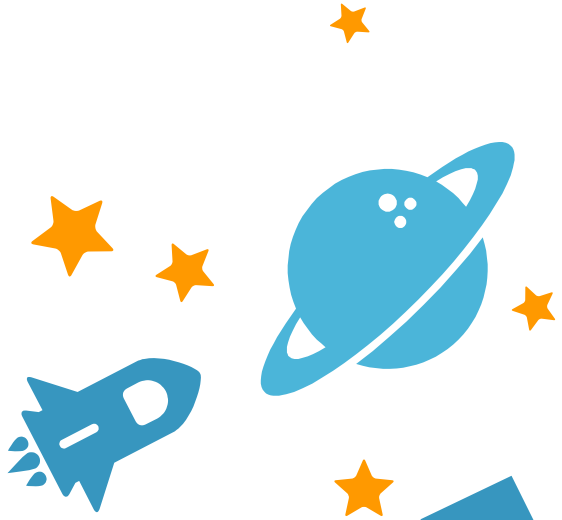



1. Addressing different screen sizes to make the layout and everything consistent throughout.
2. Considering users using a site over broadband connection and over 4G/5G network. Network speed will be differ.
3. Varying context of the web.

- Developers and designers need to resist making assumptions about network speed and context based on the screen size. It's complicated.

# HTML5 STRUCTURE & ANATOMY

Used to define the meanings of the contents of an HTML page



# HTML5 BASIC STRUCTURE

```
<!DOCTYPE html>
  <html>
    <head>
      <!--information on HTML-->
    </head>
    <body>
      <!--contents of the webpage-->
    </body>
  </html>
```

# HTML5 DOCTYPE

**<!DOCTYPE html>**

- The DOCTYPE tells the browser which *type* and *version* of document to expect, and what to do with the file. Which includes HTML5 validation, search engine, design tools and etc.

- Without a doctype, most browsers will lapse into *quirks* mode (rendering page according to older versions rules).

# Activity! Try it out

1. Download Chap2 folder from the LMS and open the file in your text editing software (Atom / Sublime / Bracket / Notepad++).

2. Save the file as index.html.

3. Put the entire document in HTML root elements.

# <html> ELEMENT

**<html lang="en">**

- Define the overall container for the rest of the HTML page.

- The **lang** attribute in the <html> element declares which language the page content is in. Though not strictly required, it should *always* be specified, as it can assist search engines and screen readers.
- Each of the world's major languages has two-character code.

  e.g. Spanish = "es", French = "fr", German = "de".

# <head> SECTION

```
<head>
    <meta charset=“utf-8”>
    <title>My First HTML5 page</title>
</head>
```

The <head> tag acts like an introduction to the page because web browsers use the header to glean various types of information about the page.

# <head> SECTION

Some of the elements that <head> usually includes:

| ATTRIBUTE | DESCRIPTION |
|---|---|
| title | Specifies the title of a web page. This is used for the title in the browser toolbar. |
| link | Used to link an HTML page with external resources, usually a stylesheet (CSS)<br>*<link rel="stylesheet" href="stylefile.css">* |
| script | Used to define scripts (e.g. Javascript)<br>*<script src="scriptfile.js"></script>* |
| meta | Used to provide metadata about an HTML page |

# <meta> TAG

```
<head>
    <meta name="Keywords" content="Fundamentals, HTML, CSS">
</head>
```

- <meta> is used to send information about the web page to search engines. The most common attributes are *name* and *content*.

- Values for name attribute:
  - **Author:** defines the name of author of the document
  - **Description:** provides description of the document that show up in search engine results.

# <meta> TAG

```
<head>
    <meta name="Keywords" content="Fundamentals, HTML, CSS">
</head>
```

- **Color-scheme:** determines whether a page supports dark mode on devices that allows it.
- **Viewport:** gives information about the initial size of the document. Used on mobile devices only.
- **Robots**: determines whether the document should be included in search engine results.
- **Charsets:** for character encoding.

# <meta> CHARSET

```
<head>
    <meta charset="utf-8">
</head>
```

- Use <meta> to set the character set for the HTML page.
- Charset encoding is the standard that tells the computer how to convert your text into sequence of bytes when stored in files – and how to convert it back again when the file is opened.
- By adding this, you can type characters such as dash (-) and copyright symbols, among others, directly in your code.

# <meta> CHARSET

- Some HTML5 character entities and their code.

| Character | Hex Code | Decimal Code | Entity Name |
|---|---|---|---|
| " | &#x0022; | &#34; | &quot; |
| & | &#x0026; | &#38; | &amp; |
| < | &#x003c; | &#60; | &lt; |
| > | &#x003e; | &#62; | &gt; |
| ¢ | &#x00a2; | &#162; | &cent; |
| £ | &#x00a3; | &#163; | &pound; |
| © | &#x00a9; | &#169; | &copy; |
| ® | &#x00ae; | &#174; | &reg; |
| ½ | &#x00bd; | &#189; | &frac12; |
| é | &#x00e9; | &#233; | &eacute; |
| — | &#x2014; | &#8212; | &mdash; |

# <body> SECTION

The <head> tag *contain the visible content of HTML web page.*

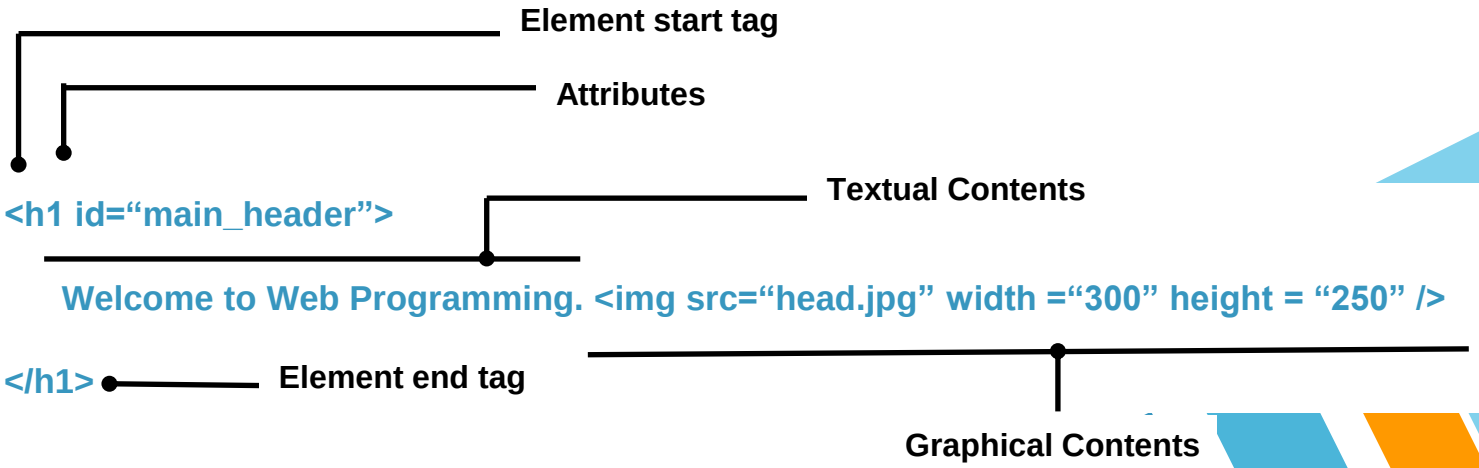
In HTML5:

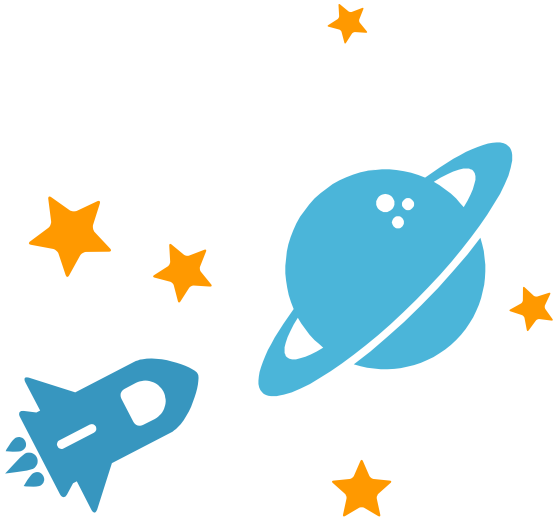
```
<html>
    <head>
    </head>

    <body>
        <h1>My Page</h1>
    </body>
</html>
```

# TAG ANATOMY

- HTML has hundreds of element tags that can be used to define a web page.
- Element tags must have a *start* and *end* tag. E.g. <h1> </h1>





# HTML TEXT FORMATTING

Define sections of text, usually a blank line is inserted before and after the element itself.

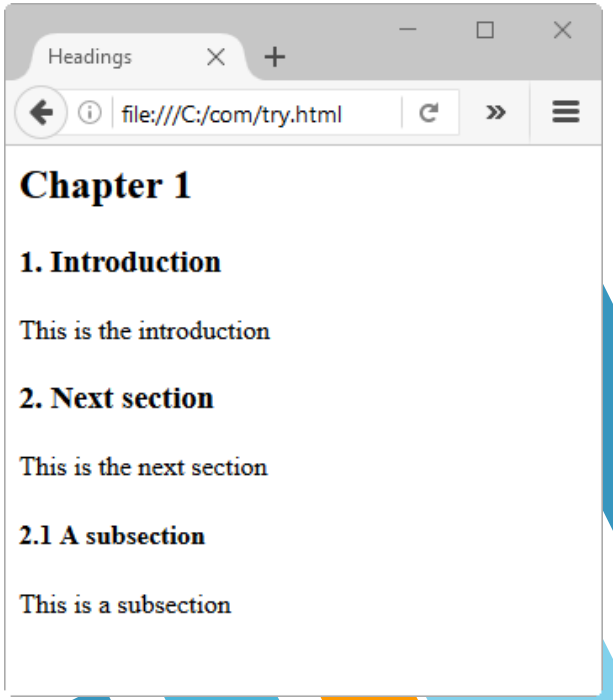
# HEADINGS – BLOCK ELEMENT

Use headings to divide document into sections.

There are **SIX (6)** types of headings in HTML:

- <h1>..</h1>
- <h2>..</h2>
- <h3>..</h3>
- <h4>..</h4>
- <h5>..</h5>
- <h6>..</h6>

Let's try!



# TEXT CONTENT TAGS – BLOCK ELEMENT

1. **Paragraph**
   - Forces a break between the enclosed text and the text surrounding it.
   - The tagged region of text may be subject to special formatting.

** Here is another paragraph **

- Centered the text in the web page.

```
<!DOCTYPE html>
<html>
 <head>
  <title>My first HTML</title>
 </head>
 <body>
 <p>here is a piece of text that has
been placed inside a paragraph</p>
<p align="center">Here is another
paragraph</p>
 </body>
</html>
```

# TEXT CONTENT TAGS – BLOCK ELEMENT

2. **Division**
   - Used to identify a section of the document that may be subject to special formatting (for example, using stylesheets).

# TEXT STYLE TAGS

Text style tags modify the text between the opening tag and the closing tag.

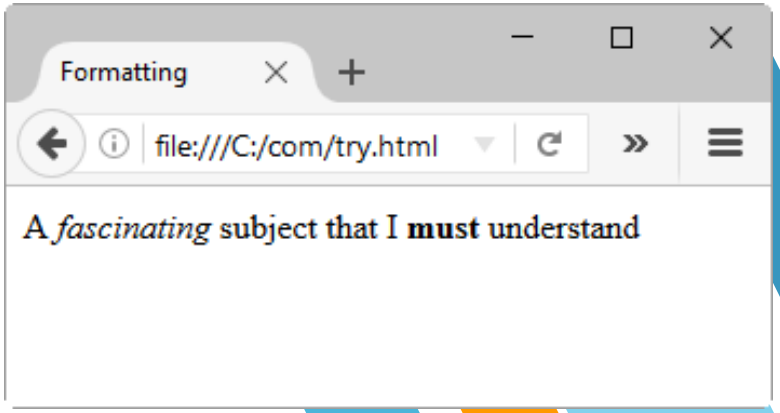
» **Hello**

| | |
|---|---|
| **** | **bold** |
| **** | *italicized* |
| **<u></u>** | <u>underlined</u> |
| **<sup></sup>** | Sample$^{\text{superscript}}$ |
| **<sub></sub>** | Sample$_{\text{subscript}}$ |
| **** | **strong** |
| **** | *emphasized* |
| **<pre></pre>** | `Preformatted text` |
| **<blockquote></blockquote>** | `Quoted text block` |
| **<del></del>** | Deleted text – ~~strike through~~ |

# INLINE DESCRIPTIVE ELEMENTS

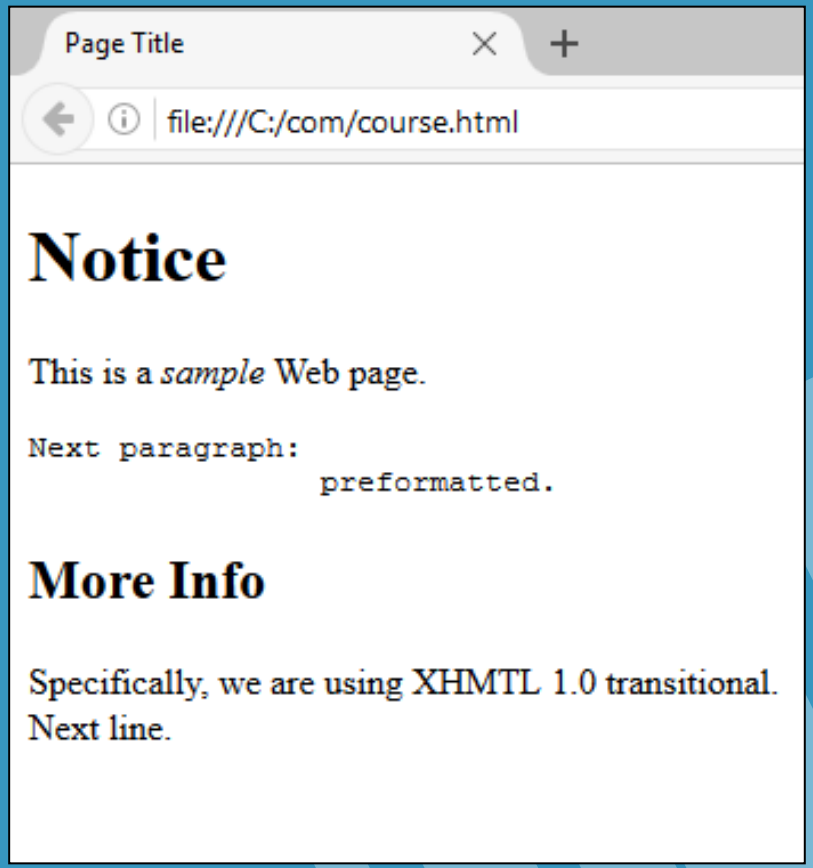
- Descriptive elements affect the appearance of text depending on how the text is described.

```
<body>
A <em>fascinating</em> subject that I <strong>must</strong> understand
</body>
```



# EXERCISE

Create a web page as follows using Text Editor.



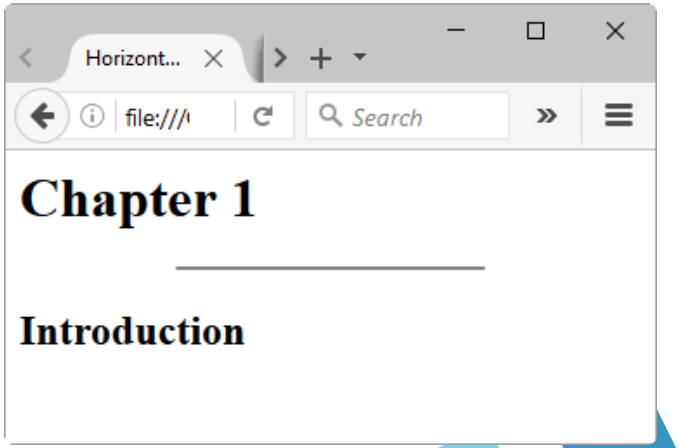
# COMMENTS

- Comments are delimited by <!-- and -->
- Comments may span multiple lines, and are not displayed by browser.

```
<body>
 <!--
  this is
  a comment
 -->
</body>
```

# HORIZONTAL LINES

- To insert a horizontal line to divide up parts of a document, we use the empty tag <hr>
- Attributes: align, size (in px), width (px/%), noshade

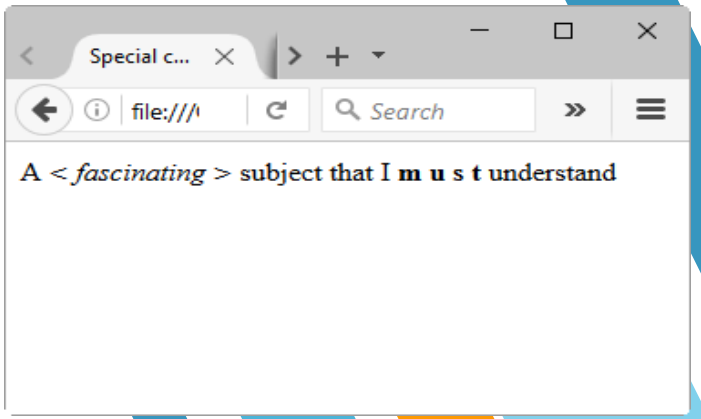



```
<body>
 <h1>Chapter 1</h1>
  <hr align="center" size="3" width="50%" noshade>
 <h2>Introduction</h2>
</body>
```

# SPECIAL CHARACTERS

- Some characters such as <, >, “ and & have special meanings.
- To prevent them being interpreted as HTML code, they must be written as follows: **< > " &**
- Multiple whitespaces are ignored in HTML. To include more than one whitespaces, use a number of ** **

```
<body>
A <em> &lt; fascinating &gt; </em>
subject that I
<strong>m u s t
</strong> understand
</body>
```

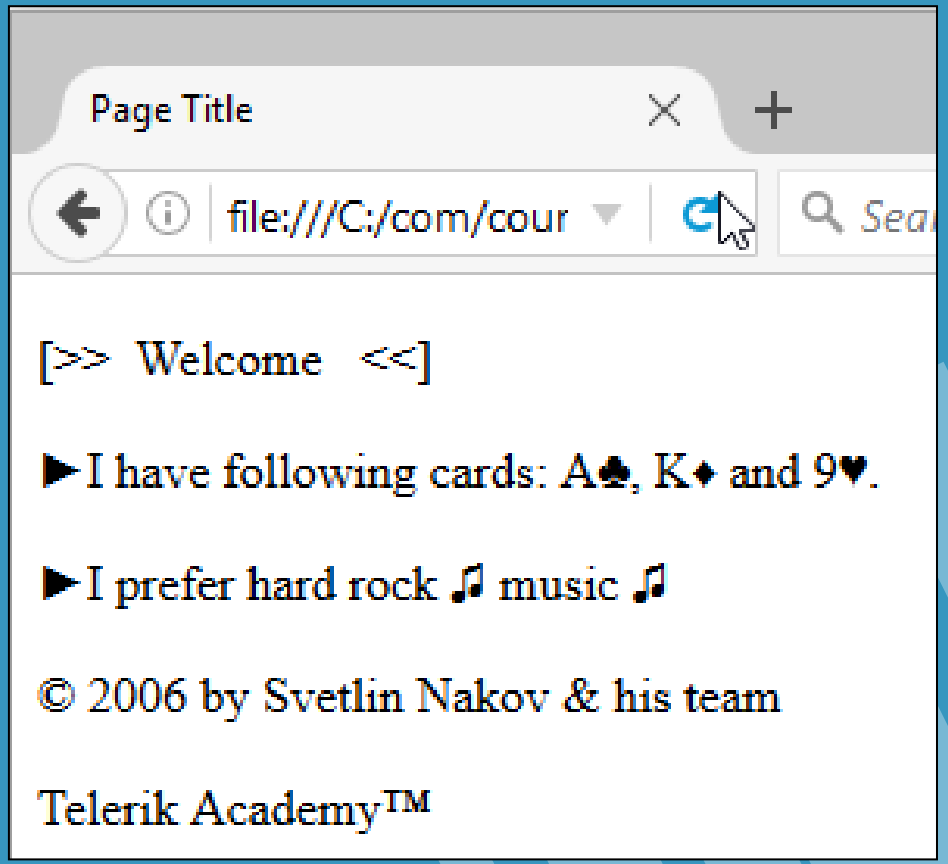


# SPECIAL CHARACTERS

| Symbol Name | HTML Entity | Symbol |
|---|---|---|
| Copyright Sign | © | © |
| Registered Trademark Sign | ® | ® |
| Trademark Sign | ™ | ™ |
| Less Than | < | < |
| Greater Than | > | > |
| Ampersand | & | & |
| Non-breaking Space |   | |
| Em Dash | — | — |
| Quotation Mark | " | " |
| Euro | € | € |
| British Pound | £ | £ |
| Japanese Yen | ¥ | ¥ |

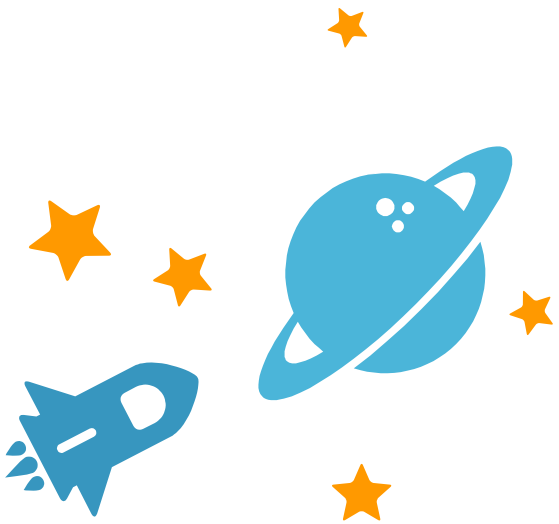
# EXERCISE

Create a web page as follows using Text Editor.



# Activity! Try it out

- The first line of text, “Black Goose Bistro” is the main heading of the page. Mark it as **<h1>.**
- There are three subheads: “The Restaurant”, “Catering” and “Location and Hours”. Mark them as **<h2>.**
- Each subhead is followed by a brief paragraph of the text, so mark it as **<p>.**
- In “Catering” section, emphasize that visitors should leave the cooking to us. Mark the text as emphasize.



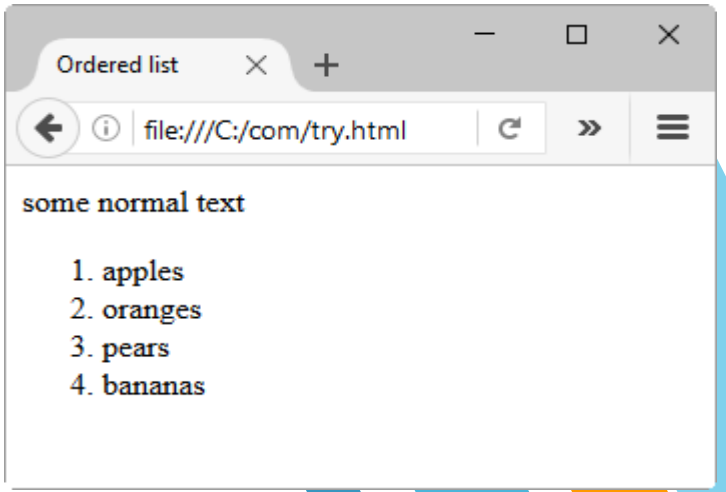
# HTML LIST

Used to group a set of items in a webpage.

# LISTS

1. **Ordered List**
   <li>…</li> for the list elements
   Each item has a number.

```
some normal text
<ol>
<li>apples</li>
<li>oranges</li>
<li>pears</li>
<li>bananas</li>
</ol>
```

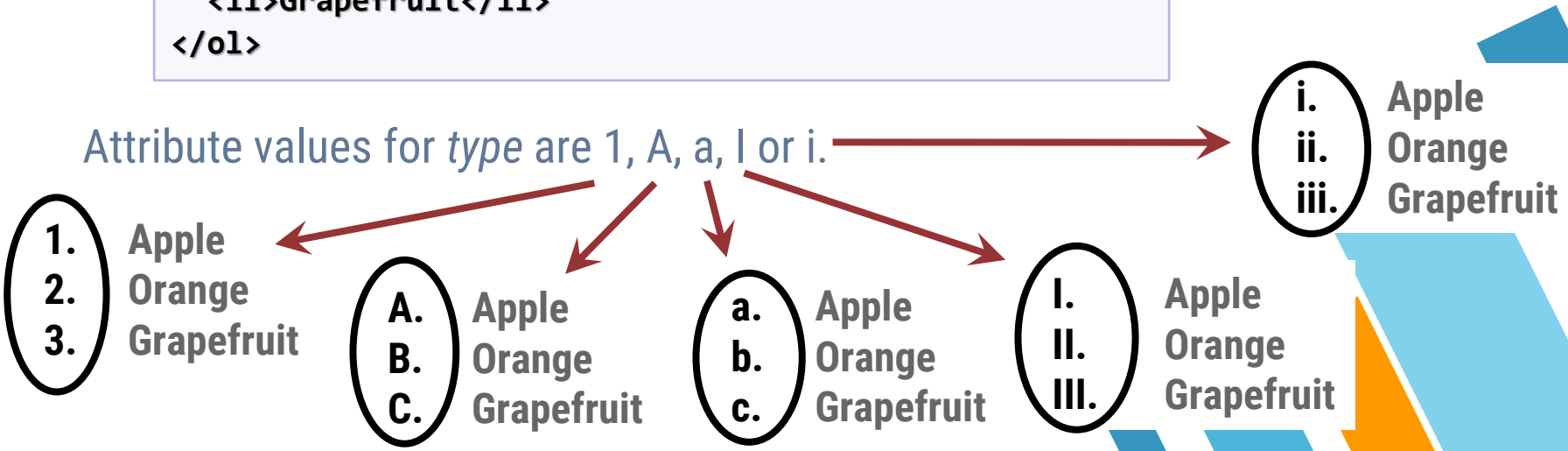


# THE <ol> TAG

Create an **O**rdered **L**ist:

```
<ol type="1">
  <li>Apple</li>
  <li>Orange</li>
  <li>Grapefruit</li>
</ol>
```

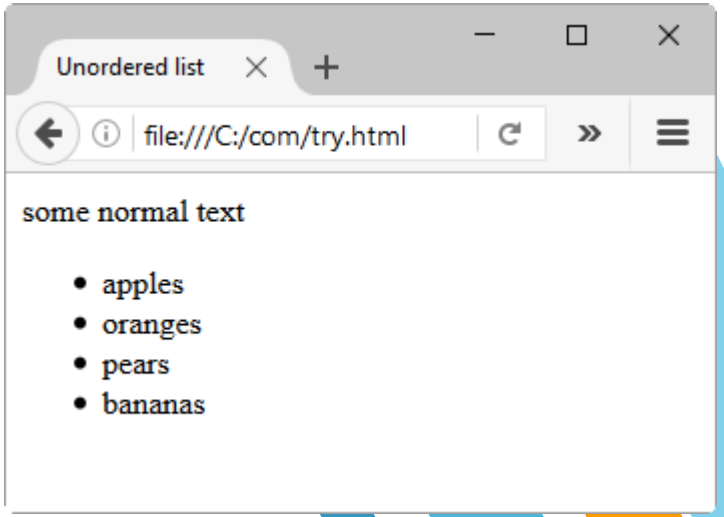
Attribute values for *type* are 1, A, a, I or i.



# LISTS

2. **Unordered List**
   <li>…</li> for the list elements
   Each item has a bullet.

```
some normal text
<ul>
<li>apples</li>
<li>oranges</li>
<li>pears</li>
<li>bananas</li>
</ul>
```



# THE <ul> TAG

Create an **U**nordered **L**ist:

```
<ul type="disk">
  <li>Apple</li>
  <li>Orange</li>
  <li>Grapefruit</li>
</ul>
```

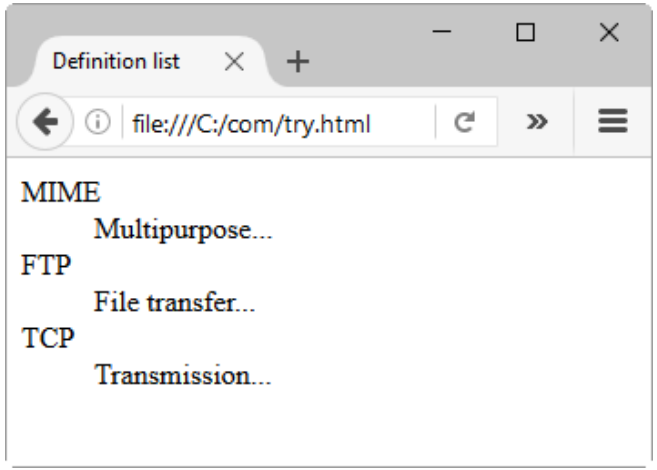
Attribute values for *type* are disc, circle or square.

# DEFINITION LISTS

<dl></dl> - The enclosing tags

<dt></dt> - The definition term

<dd></dd> The definition

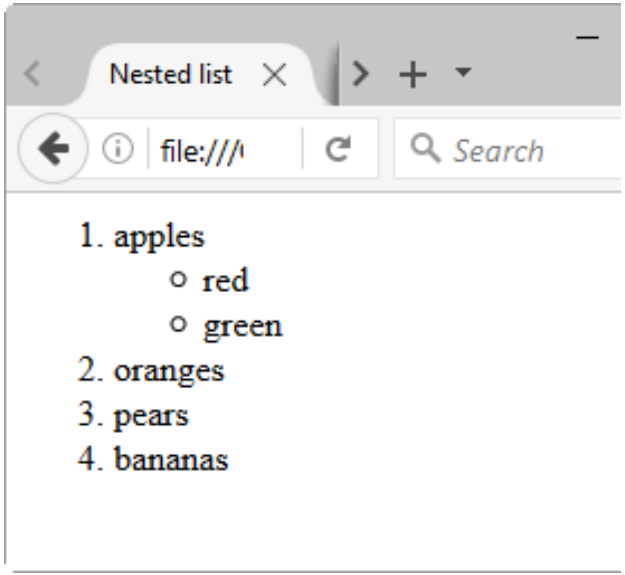



```
<dl>
 <dt>MIME</dt>
  <dd>
   Multipurpose...
  </dd>
 <dt>FTP</dt>
  <dd>
   File transfer...
  </dd>
 <dt>TCP</dt>
  <dd>
   Transmission...
  </dd>
</dl>
```

# NESTED LISTS

A list may contain another list.

The inner list is nested inside the outer list.




```
<ol>
 <li>apples</li>
  <ul>
   <li>red</li>
   <li>green</li>
  </ul>
 <li>oranges</li>
 <li>pears</li>
 <li>bananas</li>
</ol>
```

# Activity! Try it out

The owner of the Black Goose Bistro has decided to start a blog to share recipes and announcement.

You will find a raw text recipe at *recipe.txt*. It is up to you to decide which element is the best semantic match for each chunk of content.

Once you have finished, and happy with the results, save the file as **recipe.html**.

Do the same thing with *post.txt, and save it as* ***post.html*.**

# HTML HYPERLINKS

Specifies relationship between the current document and an external resource

# HYPERLINK

The link (anchor) element **<a>…</a>** provides hypertext links between

- Different documents (using a URL) within the same or different domain.
- Different parts of an individual document.

Attributes= href, target

- href – hyperlink reference
- target – “_blank” opens a new page

# LINK WITH URL

- The *href* attribute gives the URL of the target page.
- The text between the tags is highlighted – selecting it activates the link.

```
<body>
The Department of
<a href="http://www.doc.gold.ac.uk/index.html"> Computer Science</a> is a very ....
</body>
```

# RELATIVE ADDRESSING

The previous example gave the full path name, known as the **absolute address.**

- Absolute address is used to link to different domain.

**Relative address** is the partial address of the URL.

```
<a href="../pub.html">Publications</a>
<a href="../../index.html">Computer Science home</a>
```

- Relative addresses are used to link within different parts of website within the same domain.

# RELATIVE ADDRESSING

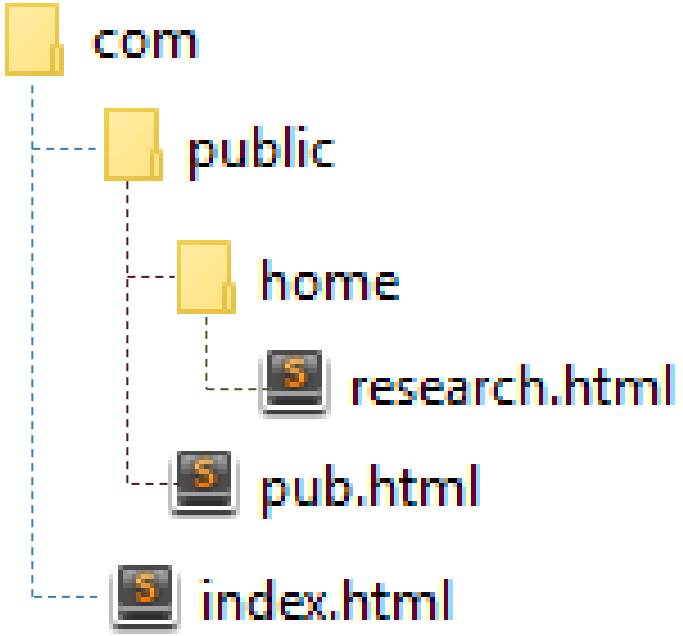
- Assume the currently viewed page is *research.html* which is located in the directory *com/public/home.*

```
<a href="research.html">Research</a>
<a href="../pub.html">Publications</a>
<a href="../../index.html">Computer Science home</a>
```

- The **double dots** in the "../pub.html" means the pub.html is located ONE level up from the current directory.
  - Sonce we are at com/public/home/, going up one level means we will be at com/public/ directory. This is where the pub.html resides.

# RELATIVE ADDRESSING

Graphical Representation



# HYPERLINKS AND SECTIONS

- Link to another location in the same document:

```
<a href="#section1">Go to Introduction</a>
...
<h2 id="section1">Introduction</h2>
```

- Link to a specific location in another document:

```
<a href="chapter3.html#section3.1.1">Go to Section 3.1.1</a>

<!-- In chapter3.html -->
...
<div id="section3.1.1">
  <h3>3.1.1. Technical Background</h3>
</div>
```

# Activity! Try it out

Now, create a blog page by downloading *blog.txt*.

1. Home, Menu, Blog and Contact should be in an unordered list.
2. Link html pages onto the navigation accordingly.
3. It is up to you to decide which element is the best semantic match for each chunk of content.
4. Save the file as **bistro_blog.html**