

CS1102

Lecture 5

Web Development: HTML, CSS and Javascript

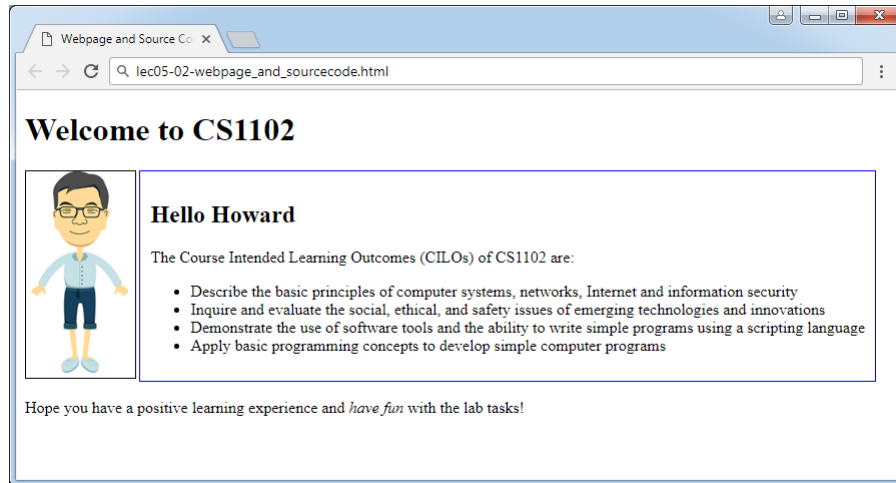


Semester A, 2017-2018
Department of Computer Science
City University of Hong Kong



Webpage and Source Code

- You can view a webpage on a browser that displays information with text, images, audio, video.
- If you look up the source code of the webpage (e.g., the file index.html), you will find that it contains text marked up with HTML tags often with CSS and Javascript



http://www.cs.cityu.edu.hk/~howard/Teaching/CS1102-Example/lec05-02-webpage_and_sourcecode.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta name="author" content="Howard Leung" />
6   <meta name="description" content="CS1102 Lecture 05 - Webpage and Source Code">
7   <meta name="keywords" content="CS1102 Lecture, Webpage and Source Code" />
8   <title>Webpage and Source Code</title>
9   <style>
10    #main_container {
11      width: 850px;
12    }
13    #left_box {
14      width: 150px;
15    }
16    #right_box {
17      border-style: solid;
18      border-width: thin;
19      border-color: blue;
20      float: right;
21      padding: 10px;
22    }
23    #fun {
24      font-style: italic;
25    }
26  </style>
27  <script>
28    var name;
29    function askName(){
30      if (name = prompt("What is your name?")){if null}
31        document.getElementById("quest").innerHTML = name;
32      }
33    </script>
34  </head>
35  <body onload="askName();">
36    <!-- Page content begins here -->
37    <div id="main_container">
38      <h1>Welcome to CS1102</h1>
39      <div id="right_box">
40        <h2>Hello <span id="quest"></span></h2>
41        <p>The Course Intended Learning Outcomes (CLOs) of CS1102 are:
42        <ul>
43          <li>Describe the basic principles of computer systems, networks, Internet and information security</li>
44          <li>Inquire and evaluate the social, ethical, and safety issues of emerging technologies and innovations</li>
45          <li>Demonstrate the use of software tools and the ability to write simple programs using a scripting language</li>
46          <li>Apply basic programming concepts to develop simple computer programs</li>
47        </ul>
48        </p>
49      </div>
50      <div id="left_box">
51        
52      </div>
53      <div id="favorite_fruit">
54        <p>Hope you have a positive learning experience and <span id="fun">have fun</span> with the lab tasks!</p>
55      </div>
56    </div>
57    <!-- Page content ends here -->
58  </body>
59 </html>
```

CS1

CSS

Javascript

HTML tags

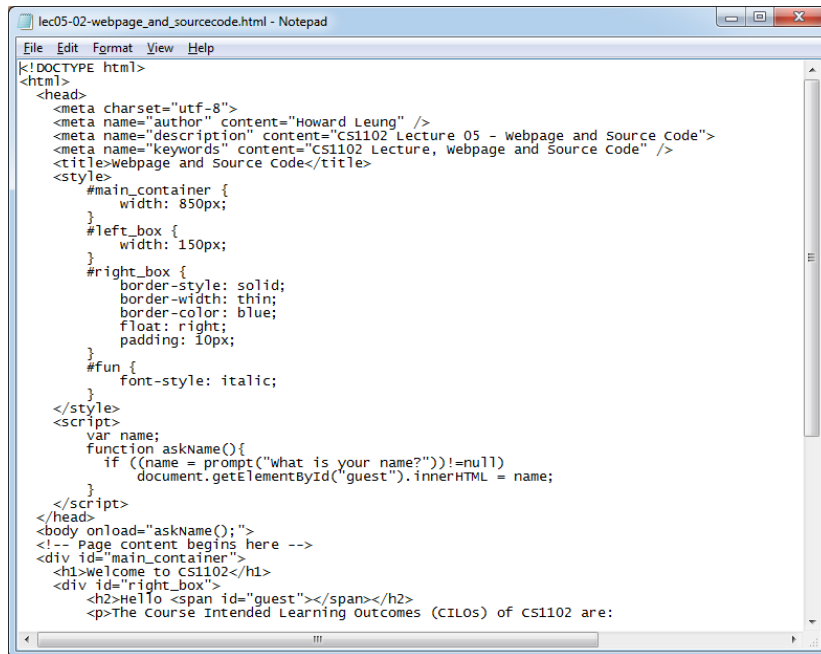
<!DOCTYPE html> gives instruction to the web browser about the version of HTML used for that page. In this case, it is HTML5. https://www.w3schools.com/tags/tag_doctype.asp

HTML, CSS and Javascript

- Often you will find 3 components in a webpage source code:
 1. HTML
 - HyperText Markup Language (HTML) uses a set of codes called tags to describe the structure of a webpage
 2. CSS
 - Cascading Style Sheets (CSS) describes how the HTML elements should be displayed by specifying the fonts, colors, layout and placement of these HTML elements
 3. Javascript
 - Javascript is a programming language that can provide instructions for a browser to dynamically generate content for a website or enhance the website interactivity

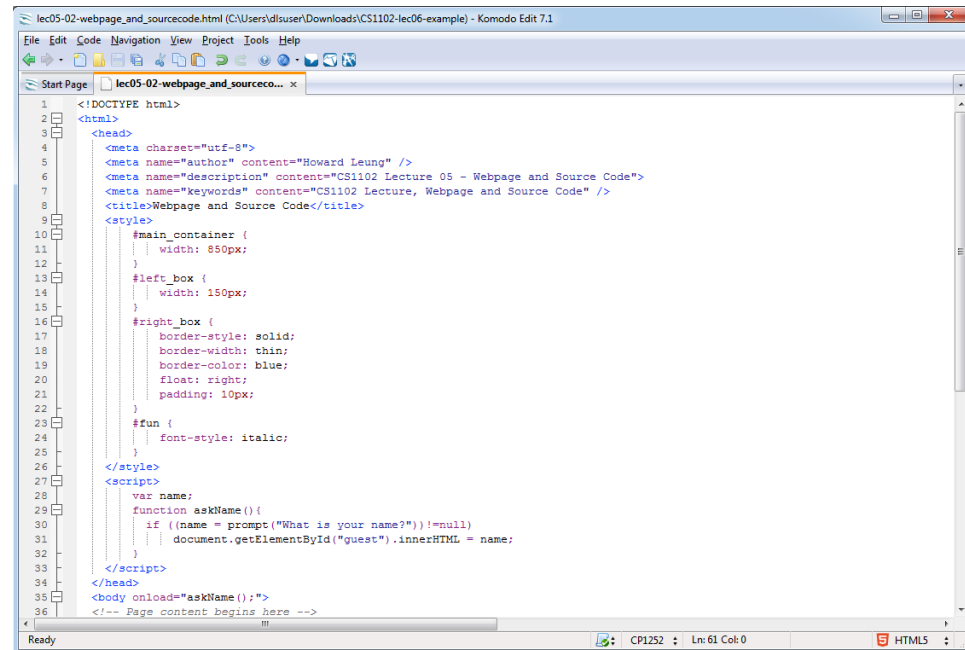
Text Editor

- A webpage source code is stored in text format so a simple editor such as Notepad can be used to edit it
- However, it is better to use a code editor such as [Komodo Edit](#) because it provides additional support for writing code, e.g., syntax coloring, syntax error checking, etc.



```
lec05-02-webpage_and_sourcecode.html - Notepad
File Edit Format View Help
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="author" content="Howard Leung" />
<meta name="description" content="CS1102 Lecture 05 - webpage and source code">
<meta name="keywords" content="CS1102 Lecture, Webpage and Source Code" />
<title>Webpage and Source code</title>
<style>
#main_container {
width: 850px;
}
#left_box {
width: 150px;
}
#right_box {
border-style: solid;
border-width: thin;
border-color: blue;
float: right;
padding: 10px;
}
#fun {
font-style: italic;
}
</style>
<script>
var name;
function askName(){
if ((name = prompt("What is your name?"))!=null)
document.getElementById("guest").innerHTML = name;
}
</script>
</head>
<body onload="askName();" >
<!-- Page content begins here -->
<div id="main_container">
<h1>welcome to CS1102</h1>
<div id="right_box">
<h2>Hello <span id="guest"></span></h2>
<p>The course intended Learning outcomes (CILOs) of CS1102 are:
```

Notepad



```
lec05-02-webpage_and_sourcecode.html (C:\Users\duser\Downloads\CS1102-lec06-example) - Komodo Edit 7.1
File Edit Code Navigation View Project Tools Help
Start Page lec05-02-webpage_and_sourceco... x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <meta name="author" content="Howard Leung" />
6 <meta name="description" content="CS1102 Lecture 05 - Webpage and Source Code">
7 <meta name="keywords" content="CS1102 Lecture, Webpage and Source Code" />
8 <title>Webpage and Source Code</title>
9 <style>
10 #main_container {
11 | width: 850px;
12 | }
13 #left_box {
14 | width: 150px;
15 | }
16 #right_box {
17 | border-style: solid;
18 | border-width: thin;
19 | border-color: blue;
20 | float: right;
21 | padding: 10px;
22 | }
23 #fun {
24 | font-style: italic;
25 | }
26 </style>
27 <script>
28 | var name;
29 | function askName(){
30 | | if ((name = prompt("What is your name?"))!=null)
31 | | | document.getElementById("guest").innerHTML = name;
32 | | }
33 </script>
34 </head>
35 <body onload="askName();" >
36 <!-- Page content begins here -->
```

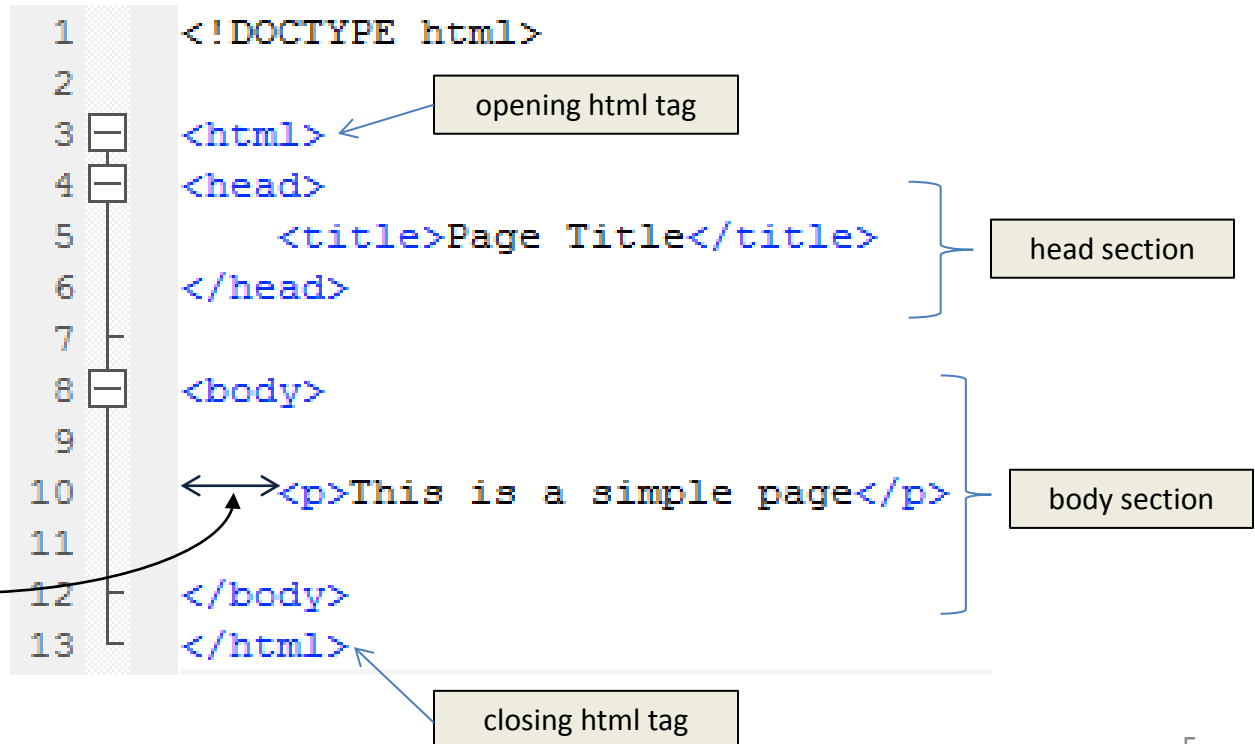
Komodo Edit

Structure of a Webpage

- HTML tags are placed inside angle brackets < >
- Most HTML tags work in pairs, with
 1. a start tag (e.g., <html>); and
 2. a end tag (e.g., </html>)

Each HTML file should have one and only one html tag, head section and body section

The indentation improves the visual layout of the HTML code so that it is easier for human to see the structure, but has no effect on the browser



HTML features to explore

- Adding a title of the webpage
- Creating headings with multiple levels
- Formatting paragraphs
- Generating bullet list and numbered list
- Inserting links to the webpage
- Displaying images on the webpage

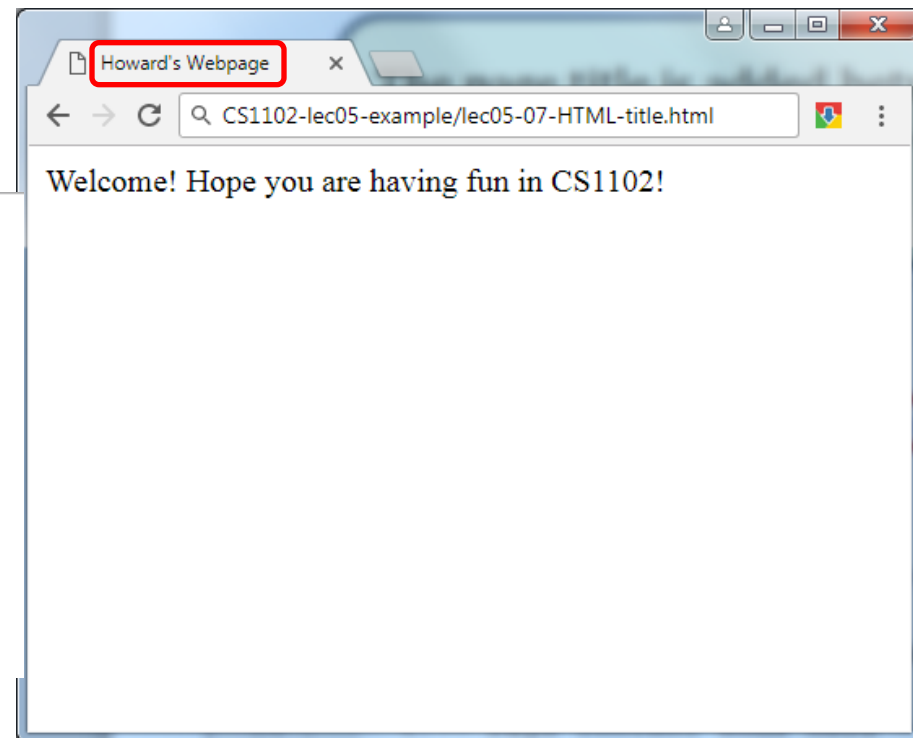
HTML: Title

- The page title can
 - help you quickly identify a webpage on a browser if you open many tabs
 - be used by search engine to determine the relevance with respect to keyword search
 - be used as the default title when you bookmark your webpage

The page title is added between the `<title>` and `</title>` tags in the HTML file

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Howard's Webpage</title>
5   </head>
6   <body>
7     <!-- Page content begins here -->
8     Welcome! Hope you are having fun in CS1102!
9     <!-- Page content ends here -->
10  </body>
11 </html>
```

Code Example: lec05-07-HTML-title.html



HTML: Headings

- Webpage headings highlight different sections on the webpage so visitors can quickly scan for information of interest.
- There are 6 levels of headings: `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`
 - `<h1>` is the most important heading
 - `<h6>` is the least important heading



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

<p><b>Tip:</b> Use h1 to h6 elements only for headings. Do not use them just to
make text bold or big. Use other tags for that.</p>

</body>
</html>
```

This is heading 1
This is heading 2
This is heading 3
This is heading 4
This is heading 5
This is heading 6
Tip: Use h1 to h6 elements only for headings. Do not use them just to make text bold or big. Use other tags for that.

https://www.w3schools.com/TAgS/tryit.asp?filename=tryhtml_headers

HTML: Headings Example

```
1 <!DOCTYPE html>
2 <html>
3   <head>!--
8     <title>Web Development Components</title>
9   </head>
10  <body>
11    <!-- Page content begins here -->
12    <h1>Web Development</h1>
13    Often you will find 3 components in a webpage source code:
14
15    <h2>HTML5</h2>
16    HyperText Markup Language (HTML) uses a set of codes called tags to
17    describe the structure of a webpage.
18
19    <h2>CSS</h2>
20    Cascading Style Sheets (CSS) describes how the HTML elements should
21    be displayed by specifying the fonts, colors, layout and placement of
22    these HTML elements.
23
24    <h2>JavaScript</h2>
25    Javascript is a programming language that can provide instructions
26    for a browser to dynamically generate content for a website or enhance
    the website interactivity.
27
28    <!-- Page content ends here -->
29  </body>
30 </html>
```

Code Example: lec05-09-HTML-headings.html



HTML: Paragraphs

The `<p>` and `</p>` tags are used to define a paragraph inside

Browser will automatically insert line spacing between successive `<p>` tags to enhance readability

Line breaks

Line breaks and line spacing in the HTML file are not matched with the way the corresponding text is displayed on the browser

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Web Development</title>
5 </head>
6 <body>
7 <!-- Paragraph 1 -->
8 <h1>Web Development</h1>
9 <p>Often you will find 3 components in a webpage source code:</p>
10
11 <h2>HTML5</h2>
12 <p>HyperText Markup Language (HTML) uses a set of codes called tags to describe the structure of a webpage. You should check to make sure that your HTML tags are valid, e.g., matching start tags with end tags.</p>
13
14 <p>The HTML tags are not displayed by the web browser in showing the webpage, but are used to determine how the document should be displayed. A webpage looks fine on one browser does not mean that it will always look the same on another browser so you should test on different browsers to ensure compatibility.</p>
15
16 <h2>CSS</h2>
17 <p>Cascading Style Sheets (CSS) describes how the HTML elements should be displayed by specifying the fonts, colors, layout and placement of these HTML elements. CSS can be embedded on the HTML file or can be defined and referred as an external file.</p>
18
19 <p>Although you could provide the same formatting and layout instructions using HTML, it is better to use CSS as a single CSS rule can be applied to multiple elements so it saves the efforts for defining the appearance of individual HTML elements one by one. The same CSS can also be applied across webpages under the same website.</p>
20
21 <h2>JavaScript</h2>
22 <p>JavaScript is a programming language that can provide instructions for a browser to dynamically generate content for a website or enhance the website interactivity. Like CSS, JavaScript can also be embedded on the HTML file or can be defined and referred as an external file.</p>
```

Web Development

Often you will find 3 components in a webpage source code:

HTML5

HyperText Markup Language (HTML) uses a set of codes called tags to describe the structure of a webpage. You should check to make sure that your HTML tags are valid, e.g., matching start tags with end tags.

The HTML tags are not displayed by the web browser in showing the webpage, but are used to determine how the document should be displayed. A webpage looks fine on one browser does not mean that it will always look the same on another browser so you should test on different browsers to ensure compatibility.

CSS

Cascading Style Sheets (CSS) describes how the HTML elements should be displayed by specifying the fonts, colors, layout and placement of these HTML elements. CSS can be embedded on the HTML file or can be defined and referred as an external file.

Although you could provide the same formatting and layout instructions using HTML, it is better to use CSS as a single CSS rule can be applied to multiple elements so it saves the efforts for defining the appearance of individual HTML elements one by one. The same CSS can also be applied across webpages under the same website

JavaScript

JavaScript is a programming language that can provide instructions for a browser to dynamically generate content for a website or enhance the website interactivity. Like CSS, JavaScript can also be embedded on the HTML file or can be defined and referred as an external file.

HTML: Lists

- A list enhances the presentation of information under a category with many items
 - E.g., which of the following webpage has better readability to you?

Web Development

Often you will find 3 components in a webpage source code:

1. HTML: HyperText Markup Language (HTML) uses a set of codes called tags to describe the structure of a webpage
2. CSS: Cascading Style Sheets (CSS) describes how the HTML elements should be displayed by specifying the fonts, colors, layout and placement of these HTML elements
3. Javascript: Javascript is a programming language that can provide instructions for a browser to dynamically generate content for a website or enhance the website interactivity

Web Development

Often you will find 3 components in a webpage source code:

1) HTML: HyperText Markup Language (HTML) uses a set of codes called tags to describe the structure of a webpage; 2) CSS: Cascading Style Sheets (CSS) describes how the HTML elements should be displayed by specifying the fonts, colors, layout and placement of these HTML elements; 3) Javascript: Javascript is a programming language that can provide instructions for a browser to dynamically generate content for a website or enhance the website interactivity

HTML: Lists (2)

- An unordered list displays items with bullets

`<ul>` and `</ul>` define the beginning and end of an unordered list
`<li>` and `</li>` encloses each list item

```
<h4>An Unordered List:</h4>
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

HTML

An Unordered List:

- Coffee
- Tea
- Milk

Browser display

https://www.w3schools.com/TAGs/tryit.asp?filename=tryhtml_lists4

- An ordered list displays items with automatic numbering

`<ol>` and `</ol>` define the beginning and end of an ordered list
`<li>` and `</li>` encloses each list item

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

HTML

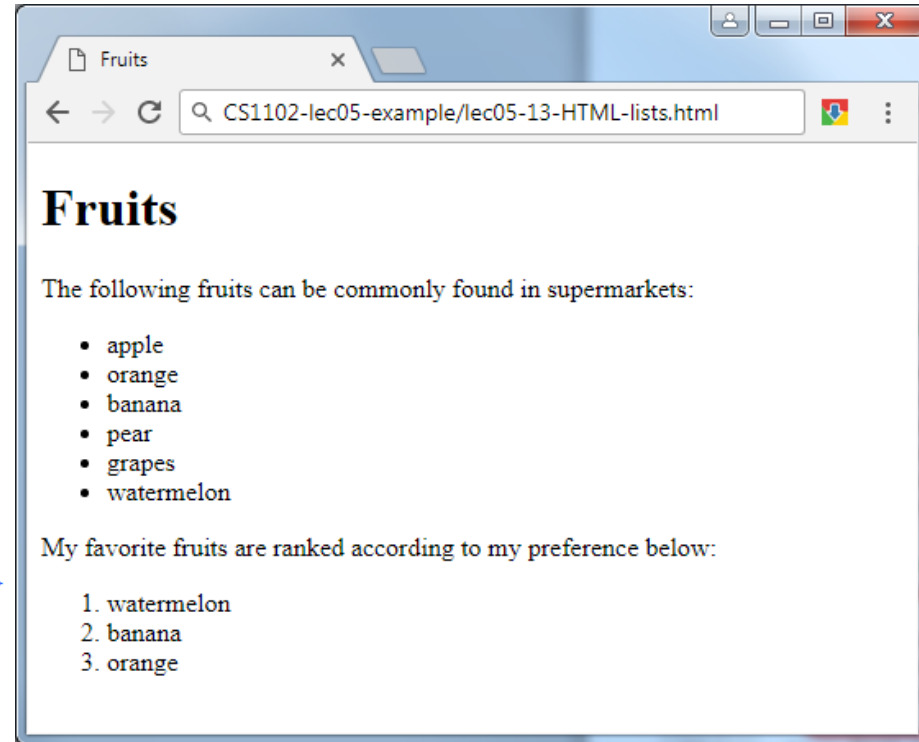
1. Coffee
2. Tea
3. Milk

Browser display

https://www.w3schools.com/TAGs/tryit.asp?filename=tryhtml_lists

HTML: Lists Example

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Fruits</title>
5   </head>
6   <body>
7     <!-- Page content begins here -->
8     <h1>Fruits</h1>
9     <p>The following fruits can be commonly found in supermarkets:</p>
10    <ul>
11      <li>apple</li>
12      <li>orange</li>
13      <li>banana</li>
14      <li>pear</li>
15      <li>grapes</li>
16      <li>watermelon</li>
17    </ul>
18    <p>My favorite fruits are ranked according to my preference below:</p>
19    <ol>
20      <li>watermelon</li>
21      <li>banana</li>
22      <li>orange</li>
23    </ol>
24    <!-- Page content ends here -->
25  </body>
26 </html>
```

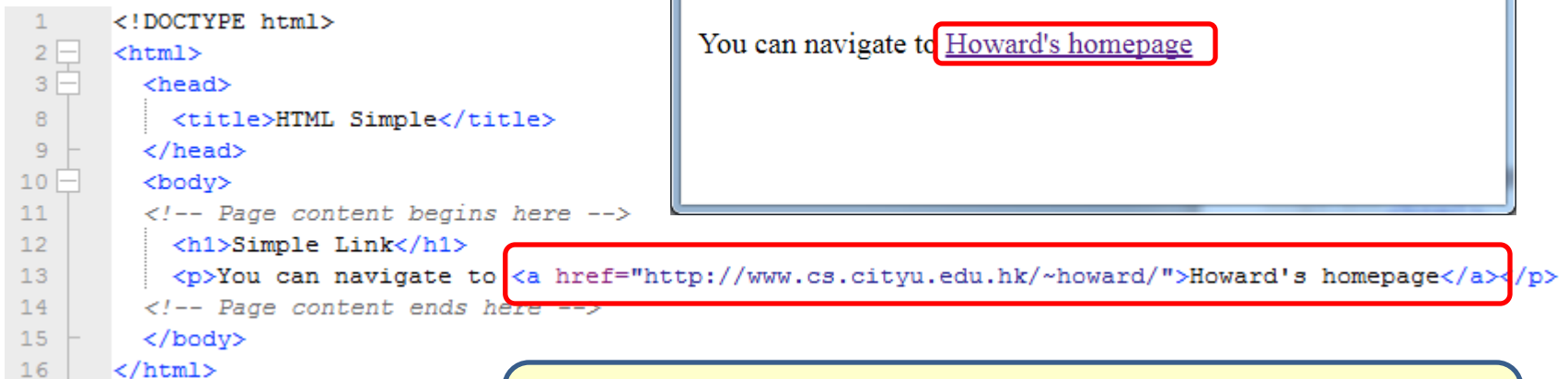


Code Example: lec05-13-HTML-lists.html

With lists, it would be easy to insert or delete items, or rearrange the order of the items. For numbered list, there is no need to worry about the numbering even when the items are modified because the numbering is automatic

HTML: Links

- A hyperlink, or simply called a link, can be text or image in a webpage that a user can click to navigate to another webpage

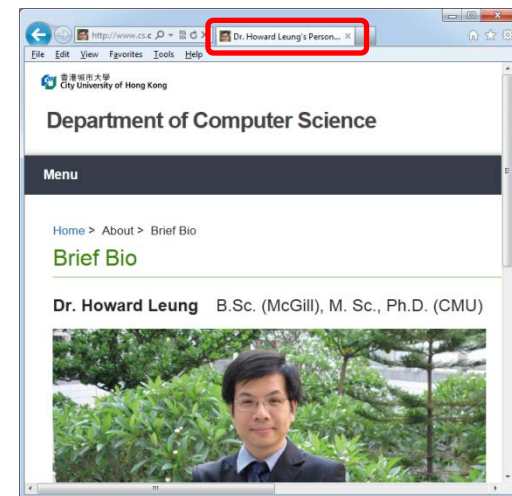
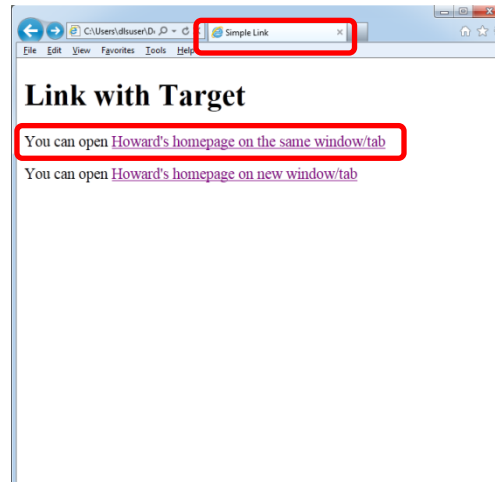


Code Example: lec05-14-HTML-simple-link.html

The `<a>` anchor tag is used to link to another hypertext location
The `href` attribute value contains the URL (Uniform Resource Locator), i.e., the link destination

HTML: Link Target

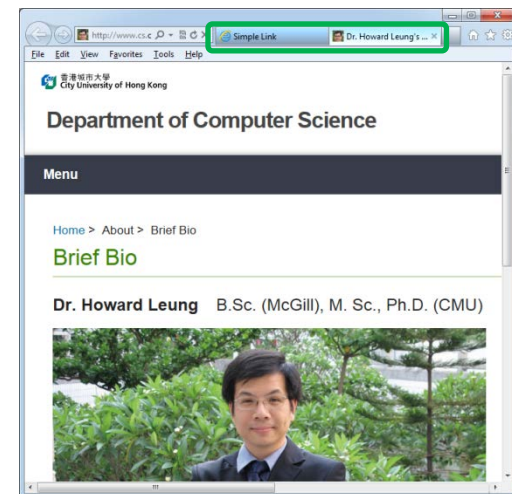
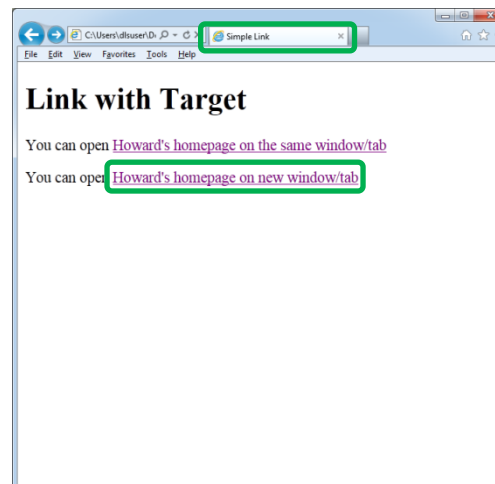
By default, the link destination will be opened in the same browser tab or window



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML Link Target</title>
5   </head>
6   <body>
7     <!-- Page content begins here -->
8     <h1>HTML Link with Target</h1>
9     <p>You can open <a href="http://www.cs.cityu.edu.hk/~howard/">Howard's homepage on the same window/tab</a></p>
10    <p>You can open <a href="http://www.cs.cityu.edu.hk/~howard/" target="_blank">Howard's homepage on new window/tab</a></p>
11    <!-- Page content ends here -->
12  </body>
13 </html>
```

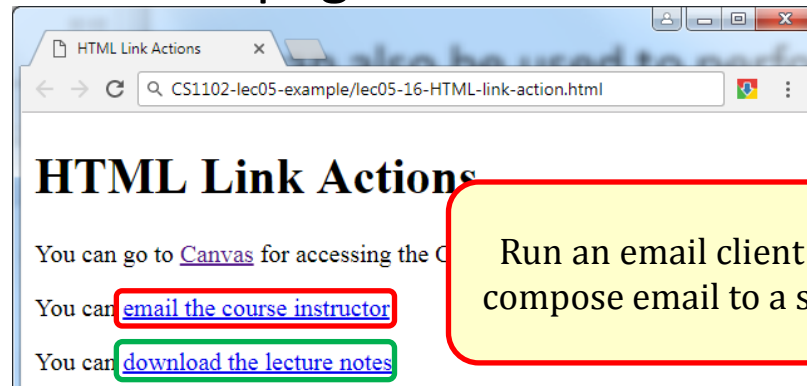
Code Example: lec05-15-HTML-link-target.html

With the attribute `target="_blank"`, the linked destination is opened in a new window or tab



HTML: Link Actions

- A hyperlink can also be used to perform other action than navigating to another webpage



Run an email client (if installed) to compose email to a specific recipient

Download a file

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML Link Actions</title>
5   </head>
6   <body>
7     <!-- Page content begins here -->
8     <h1>HTML Link Actions</h1>
9     <p>You can go to <a href="http://canvas.cityu.edu.hk/">Canvas</a> for accessing the CS1102 course page</p>
10    <p>You can <a href="mailto:howard@cityu.edu.hk">email the course instructor</a></p>
11    <p>You can <a href="CS1102_1718A_lec05-WebDevelopment.pdf">download the lecture notes</a></p>
12    <!-- Page content ends here -->
13  </body>
14 </html>
```

Code Example: lec05-16-HTML-link-action.html

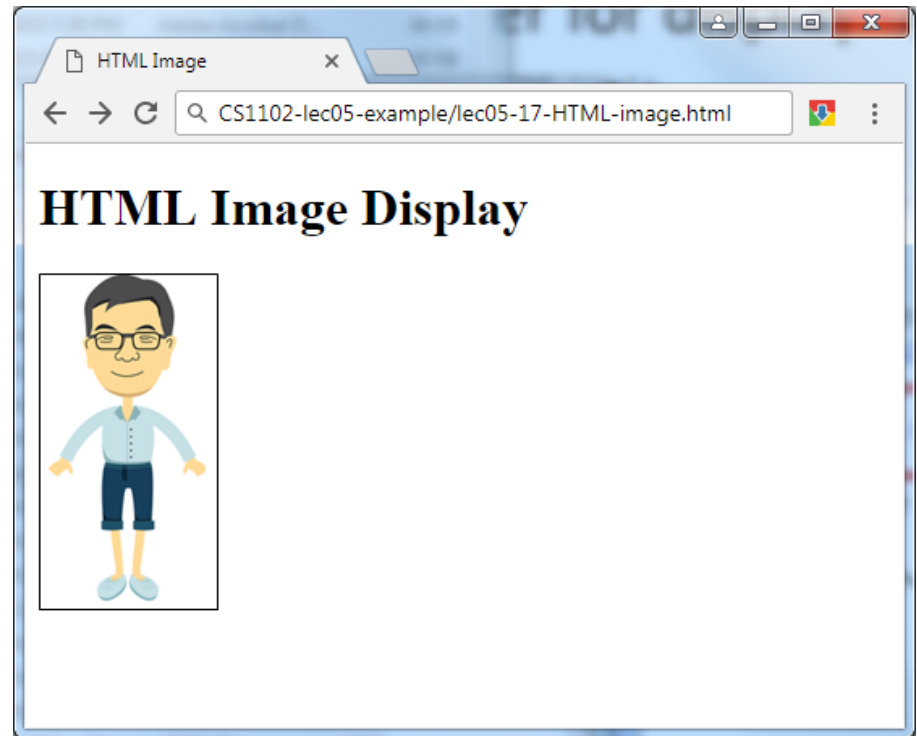
HTML: Images

- Images are always stored as separate files. The `<img>` tag can be used in the HTML code to provide instruction to the browser for displaying an image on a webpage

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML Image</title>
5   </head>
6   <body>
7     <!-- Page content begins here -->
8     <h1>HTML Image Display</h1>
9     
10    <!-- Page content ends here -->
11  </body>
12 </html>
```

folder

lec05-17-HTML-image.html
character.png



HTML: Images (2)

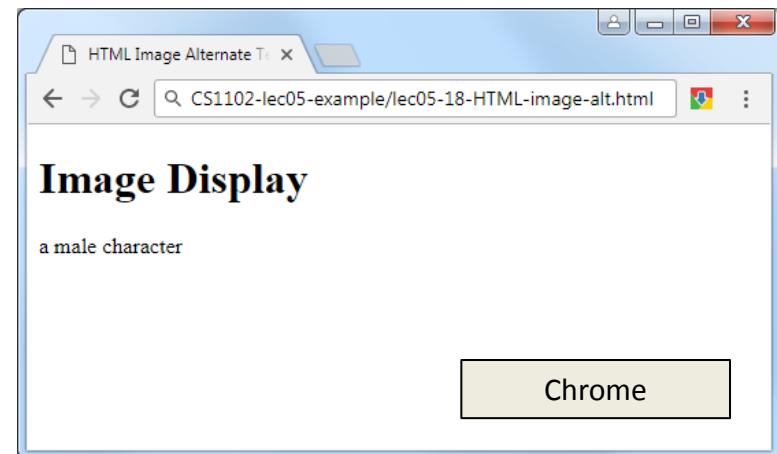
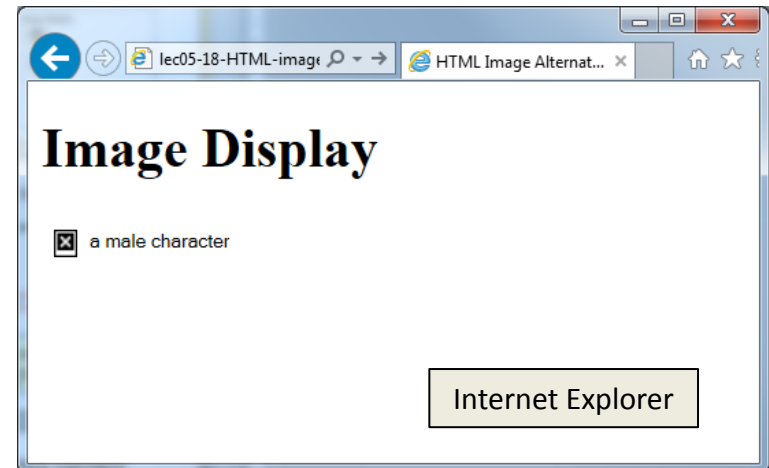
- The `alt` attribute is used to specify the alternate text for the image

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML Image Alternate Text</title>
5   </head>
6   <body>
7     <!-- Page content begins here -->
8     <h1>Image Display</h1>
9     
10    <!-- Page content ends here -->
11  </body>
12 </html>
```

Code Example: lec05-18-HTML-image-alt.html

The alternate text has the following functions:

1. Alternate text will be displayed instead of an image when the image cannot be displayed in case the image does not exist or there is a typo in the image filename.
2. Visually impaired users may not be able to view the image but are still able to understand the image content by using screen reading software to read aloud the alternate text for the image

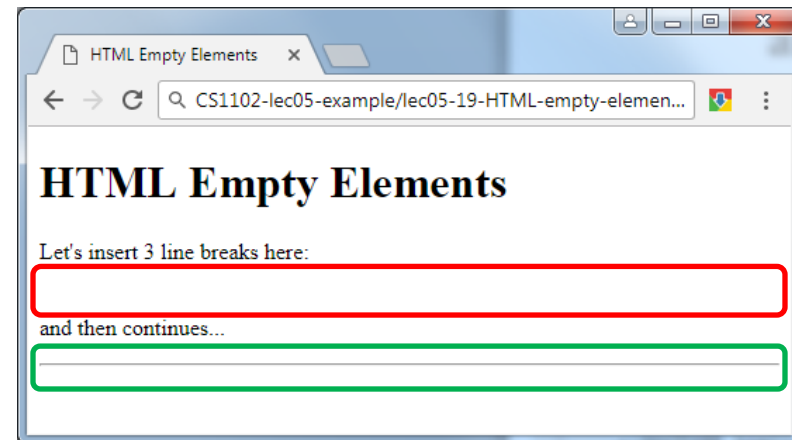


HTML: Empty Element

- Note that the `img` tag does not have a corresponding end tag
 - E.g., `` works alone and there is no such thing as `</img>`
- HTML elements that do not require an end tag are called empty elements
- Other examples of empty elements:
 - `<br />`: add a line break
 - `<hr />`: add a horizontal rule (line)

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML Empty Elements</title>
5   </head>
6   <body>
7     <!-- Page content begins here -->
8     <h1>HTML Empty Elements</h1>
9
10    <p>Let's insert 3 line breaks here: <br /><br /><br /> and then continues...</p>
11
12    <br />
13
14    <hr />
15
16    <!-- Page content ends here -->
17  </body>
18 </html>
```

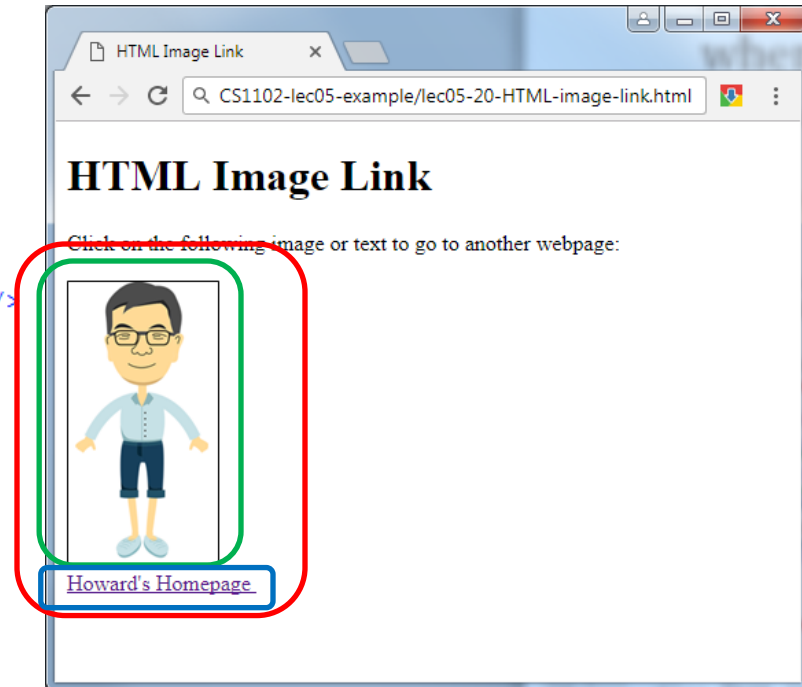
Code Example: lec05-19-HTML-empty-element.html



HTML: Image Link

- You have seen how to make a text hyperlink before. You can also make an image hyperlink so that the same link actions can be performed when the image is clicked

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML Image Link</title>
5   </head>
6   <body>
7     <!-- Page content begins here -->
8     <h1>HTML Image Link</h1>
9
10    <p>Click on the following image or text to go to another webpage:<br/>
11
12    <a href="http://www.cs.cityu.edu.hk/~howard">
13      <br />
14      
15      <br />
16      Howard's Homepage
17    </a>
18
19    </p>
20
21    <!-- Page content ends here -->
22  </body>
23 </html>
```



Code Example: lec05-20-HTML-image-link.html

Absolute URL vs Relative URL

- The URL can be specified either as an absolute URL or a relative URL

An absolute URL includes the full path

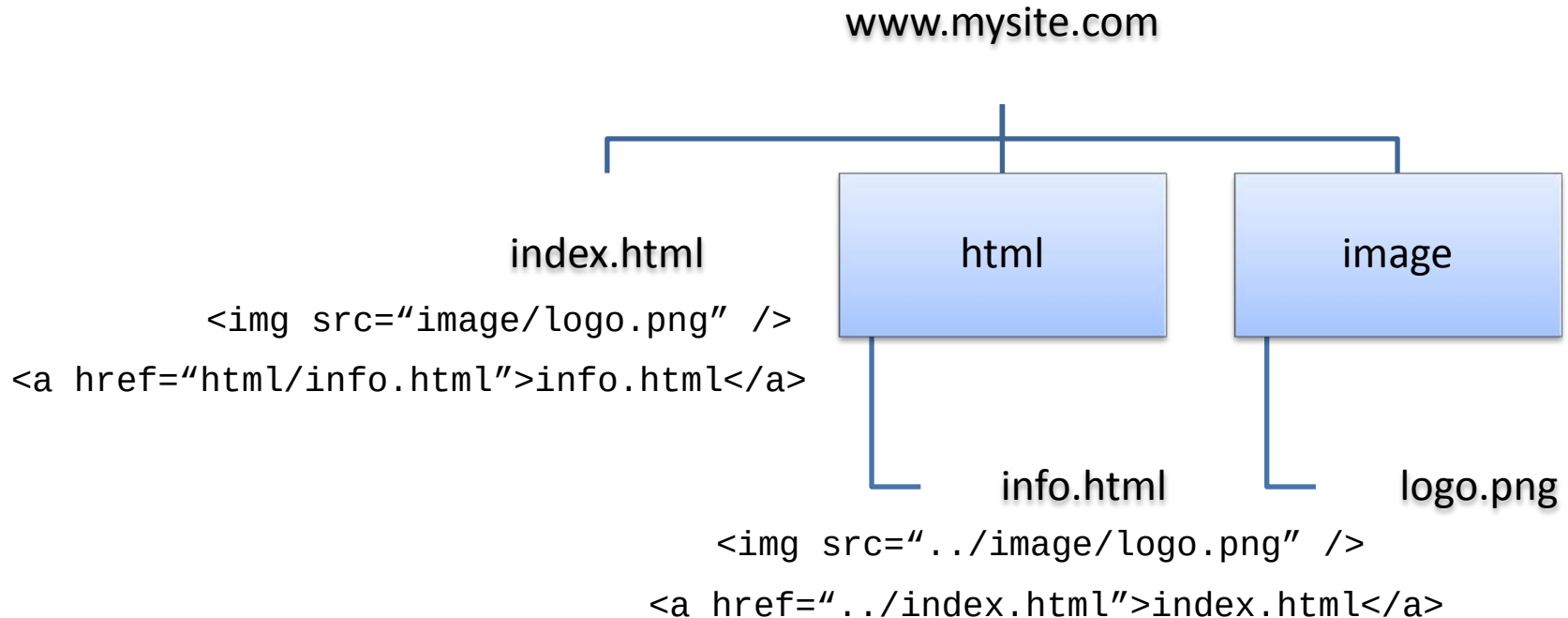
Usually absolute URL is used for linking to files from external websites (not on the same web server)

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML Link Actions</title>
5   </head>
6   <body>
7     <!-- Page content begins here -->
8     <h1>HTML Link Actions</h1>
9     <p>You can go to <a href="http://canvas.cityu.edu.hk/">Canvas</a> for accessing the CS1102 course page</p>
10    <p>You can <a href="mailto:howard@cityu.edu.hk">email the course instructor</a></p>
11    <p>You can <a href="CS1102_1718A_lec05-WebDevelopment.pdf">download the lecture notes</a></p>
12    <!-- Page content ends here -->
13  </body>
14 </html>
```

Relative URL specifies the location of a file relative to the location of the current file

File Path

- Files on a website may be placed on different folders and the path needs to be specified in order to link to a file placed at a different folder



`../` refers to the parent folder,
i.e., go up one level in the folder structure

CSS

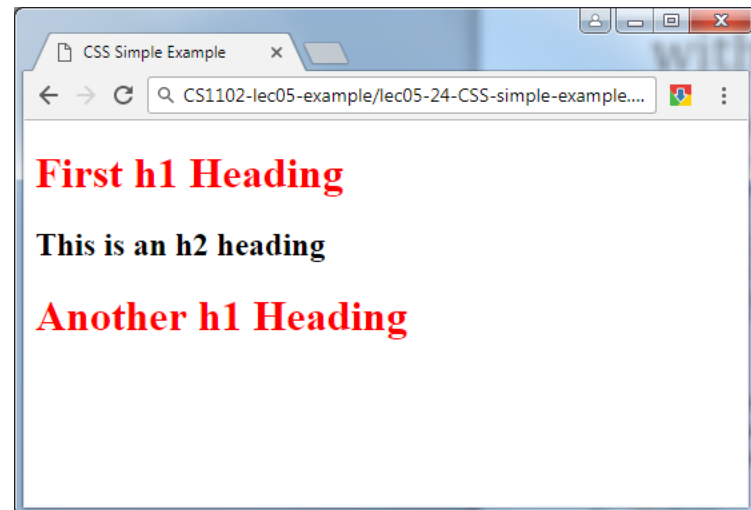
- Cascading Style Sheets (CSS) describes how the HTML elements should be displayed by specifying the fonts, colors, layout and placement of these HTML elements
- Although you could provide the same formatting and layout instructions using HTML, it is better to use CSS as a single CSS rule can be applied to multiple elements so it saves the efforts for defining the appearance of individual HTML elements one by one. The same CSS can also be applied across webpages under the same website

CSS: Simple Example

- CSS can be embedded in the head section of the webpage, with the styles defined inside the `<style> </style>` tags

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>CSS Simple Example</title>
5   <style>
6     h1 {
7       color: red;
8     }
9   </style>
10 </head>
11 <body>
12   <!-- Page content begins here -->
13   <h1>First h1 Heading</h1>
14   <h2>This is an h2 heading</h2>
15   <h1>Another h1 Heading</h1>
16   <!-- Page content ends here -->
17 </body>
18 </html>
```

This style sets all `<h1>` headings in red color



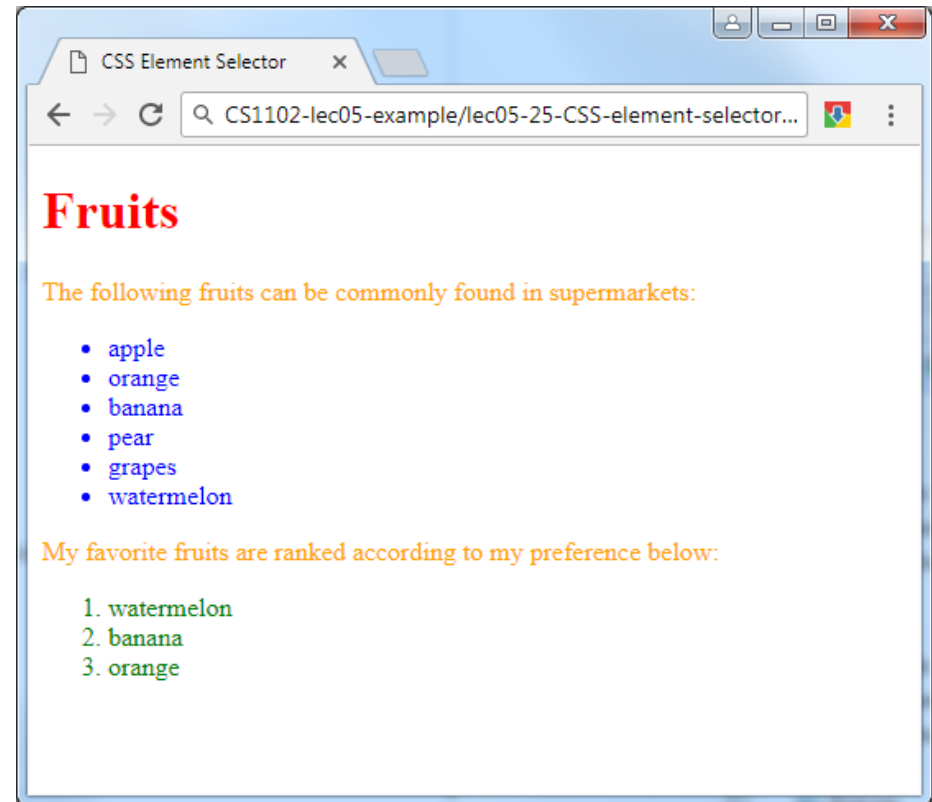
Code Example: lec05-24-CSS-simple-example.html

CSS: Element Selector

- A CSS style can be applied to different HTML elements by using the corresponding HTML tag as element selector

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>CSS Element Selector</title>
5 <style>
6   h1 {
7     color: red;
8   }
9   ul {
10    color: blue;
11  }
12   ol {
13    color: green;
14  }
15   p {
16    color: orange;
17  }
18 </style>
19 </head>
20 <body>
21 <!-- Page content begins here -->
22 <h1>Fruits</h1>
23 <p>The following fruits can be commonly found in supermarkets:</p>
24 <ul>
25   <li>apple</li>
26   <li>orange</li>
27   <li>banana</li>
28   <li>pear</li>
29   <li>grapes</li>
30   <li>watermelon</li>
31 </ul>
32 <p>My favorite fruits are ranked according to my preference below:</p>
33 <ol>
34   <li>watermelon</li>
35   <li>banana</li>
36   <li>orange</li>
37 </ol>
38 <!-- Page content ends here -->
39 </body>
40 </html>
```

Code Example: lec05-25-CSS-element-selector.html

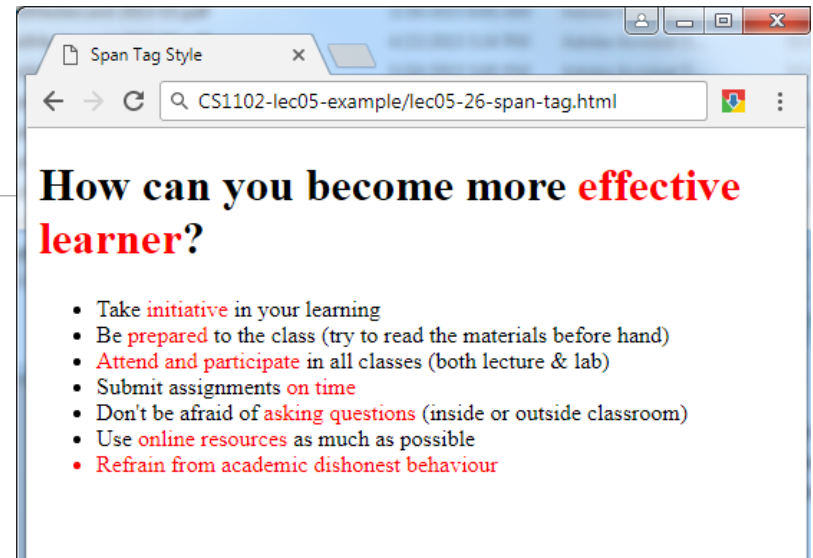


The tag

- The tag can be used to enclose a part of the text such that the style can be applied.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Span Tag Style</title>
5     <style>
6       span {
7         color: red;
8       }
9     </style>
10  </head>
11  <body>
12    <!-- Page content begins here -->
13    <h1>How can you become more <span>effective learner</span>?</h1>
14    <ul>
15      <li>Take <span>initiative</span> in your learning</li>
16      <li>Be <span>prepared</span> to the class (try to read the materials before hand)</li>
17      <li><span>Attend and participate</span> in all classes (both lecture & lab) </li>
18      <li>Submit assignments <span>on time</span></li>
19      <li>Don't be afraid of <span>asking questions</span> (inside or outside classroom)</li>
20      <li>Use <span>online resources</span> as much as possible</li>
21      <span><li>Refrain from academic dishonest behaviour</li></span>
22    </ul>
23    <!-- Page content ends here -->
24  </body>
25 </html>
```

Code Example: lec05-26-span-tag.html



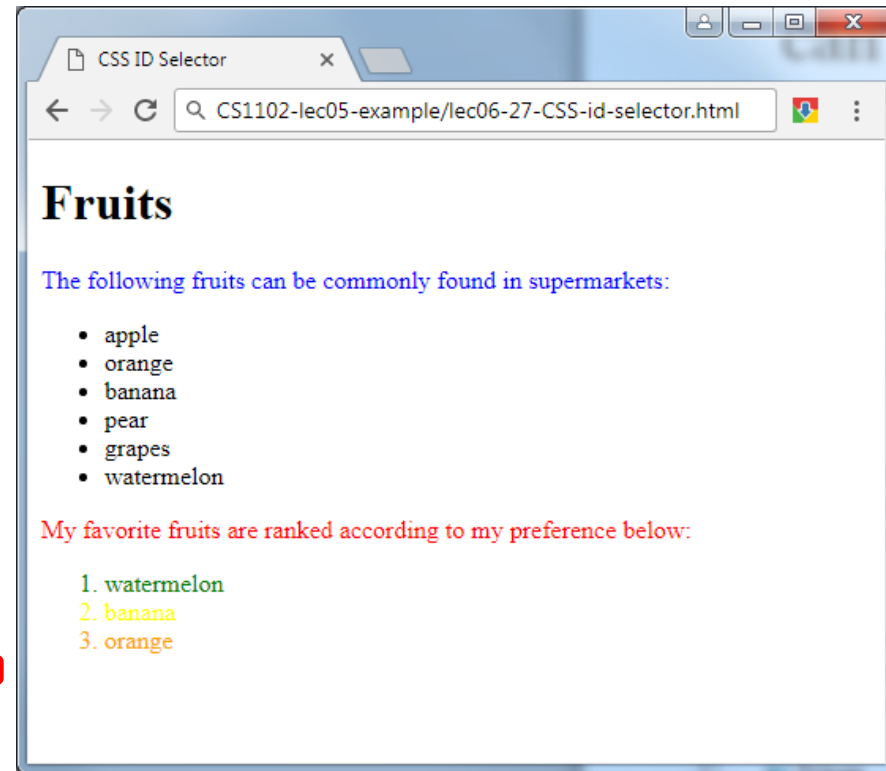
CSS: ID Selector

- Each html element can be labeled with an id. A CSS style can be applied to an element by specifying its id

Code Example: lec05-27-CSS-id-selector.html

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>CSS ID Selector</title>
5     <style>
6       #paragraph1 {
7         color: blue;
8       }
9       #paragraph2 {
10        color: red;
11      }
12      #favorite1 {
13        color: green;
14      }
15      #favorite2 {
16        color: yellow;
17      }
18      #favorite3 {
19        color: orange;
20      }
21    </style>
22  </head>
23  <body>
24    <!-- Page content begins here -->
25    <h1>Fruits</h1>
26    <p id="paragraph1">The following fruits can be commonly found in supermarkets:</p>
27    <ul>
28      <li>apple</li>
29      <li>orange</li>
30      <li>banana</li>
31      <li>pear</li>
32      <li>grapes</li>
33      <li>watermelon</li>
34    </ul>
35    <p id="paragraph2">My favorite fruits are ranked according to my preference below:</p>
36    <ol>
37      <li id="favorite1">watermelon</li>
38      <li id="favorite2">banana</li>
39      <li id="favorite3">orange</li>
40    </ol>
41    <!-- Page content ends here -->
42  </body>
43 </html>
```

The hash # followed by the id is used to specify an id selector



Each id used to label an html element should be unique, i.e., it can only be used once and no two html elements should have the same id

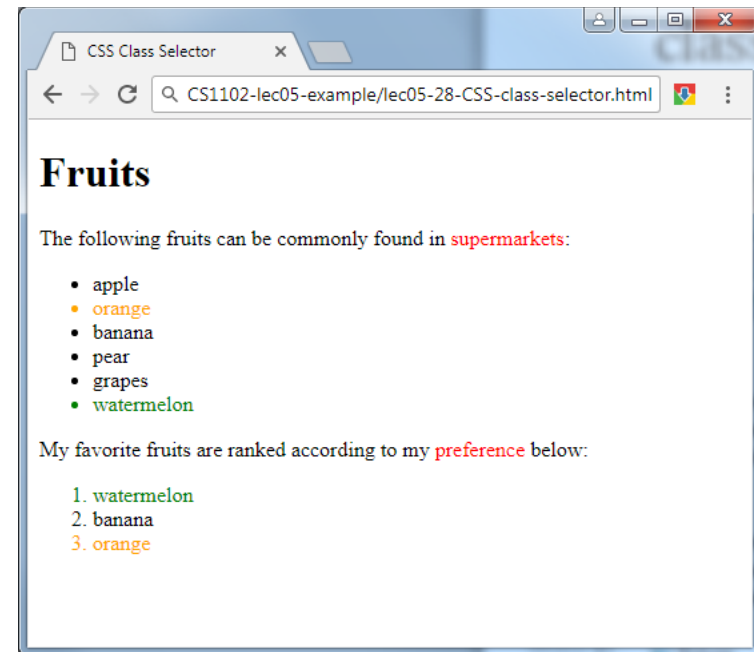
CSS: Class Selector

- A CSS style can be applied to all elements with the same class name

Code Example: lec05-28-CSS-class-selector.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>CSS Class Selector</title>
5 <style>
6   .highlight {
7     color: red;
8   }
9   .watermelon {
10    color: green;
11  }
12   .orange {
13     color: orange;
14   }
15 </style>
16 </head>
17 <body>
18 <!-- Page content begins here -->
19 <h1>Fruits</h1>
20 <p>The following fruits can be commonly found in <span class="highlight">supermarkets</span></p>
21 <ul>
22   <li>apple</li>
23   <li class="orange">orange</li>
24   <li>banana</li>
25   <li>pear</li>
26   <li>grapes</li>
27   <li class="watermelon">watermelon</li>
28 </ul>
29 <p>My favorite fruits are ranked according to my <span class="highlight">preference</span> below:</p>
30 <ol>
31   <li class="watermelon">watermelon</li>
32   <li>banana</li>
33   <li class="orange">orange</li>
34 </ol>
35 <!-- Page content ends here -->
36 </body>
37 </html>
```

The period . followed by a class name is used to specify a class selector



Different html elements can be labeled with the same class name

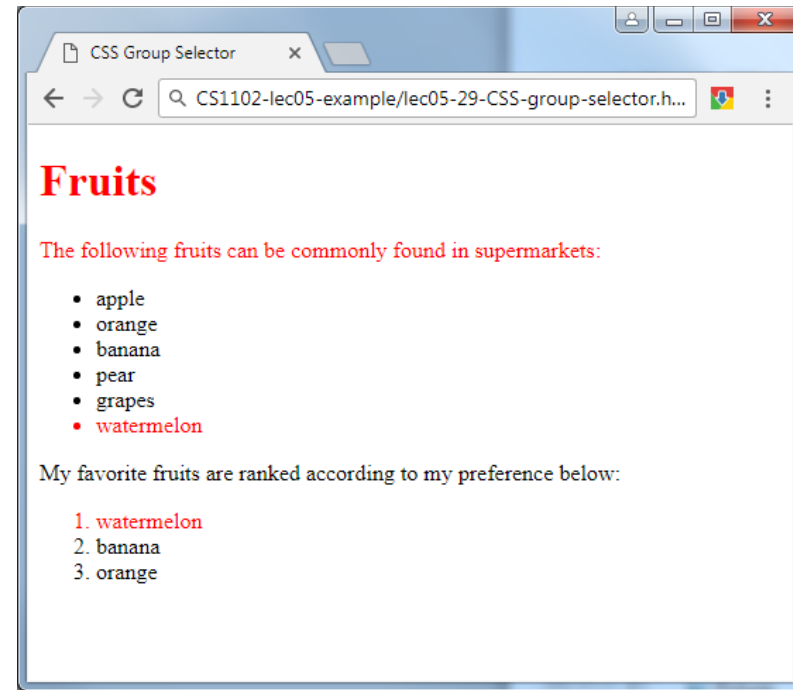
CSS: Group Selector

- A CSS style can be applied to a group of elements

Code Example: lec05-29-CSS-group-selector.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>CSS Group Selector</title>
5 <style>
6   h1, #paragraph1, .highlight
7     color: red;
8 </style>
9 </head>
10 <body>
11 <!-- Page content begins here -->
12 <h1>Fruits</h1>
13 <p id="paragraph1">The following fruits can be commonly found in supermarkets:</p>
14 <ul>
15   <li>apple</li>
16   <li>orange</li>
17   <li>banana</li>
18   <li>pear</li>
19   <li>grapes</li>
20   <li class="highlight">watermelon</li>
21 </ul>
22 <p>My favorite fruits are ranked according to my preference below:</p>
23 <ol>
24   <li class="highlight">watermelon</li>
25   <li>banana</li>
26   <li>orange</li>
27 </ol>
28 <!-- Page content ends here -->
29 </body>
30 </html>
```

The comma , is used to separate individual selectors to specify a group selector



CSS: Color

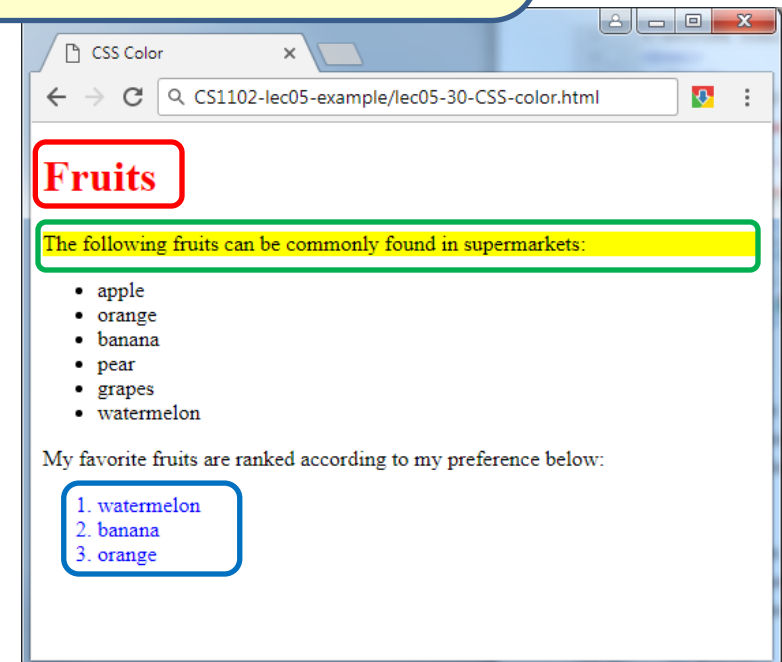
- There are other ways to set color properties in CSS

The color can be specified by the R,G,B values in the form of

- 3 decimal numbers ranging from 0-255, or
- 6 hexadecimal nibbles where every 2 hexadecimal nibbles represent each of R, G, B value respectively

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>CSS Color</title>
5 <style>
6   h1 {
7     color: rgb(255,0,0);
8   }
9   #paragraph1 {
10    background-color: yellow;
11  }
12   ol {
13     color: #0000FF;
14   }
15 </style>
16 </head>
17 <body>
18 <!-- Page content begins here -->
19 <h1>Fruits</h1>
20 <p id="paragraph1">The following fruits can be commonly found in supermarkets:</p>
21 <ul>
22   <li>apple</li>
23   <li>orange</li>
24   <li>banana</li>
25   <li>pear</li>
26   <li>grapes</li>
27   <li>watermelon</li>
28 </ul>
29 <p>My favorite fruits are ranked according to my preference below:</p>
30 <ol>
31   <li>watermelon</li>
32   <li>banana</li>
33   <li>orange</li>
34 </ol>
35 <!-- Page content ends here -->
36 </body>
37 </html>
```

Code Example: lec05-30-CSS-color.html



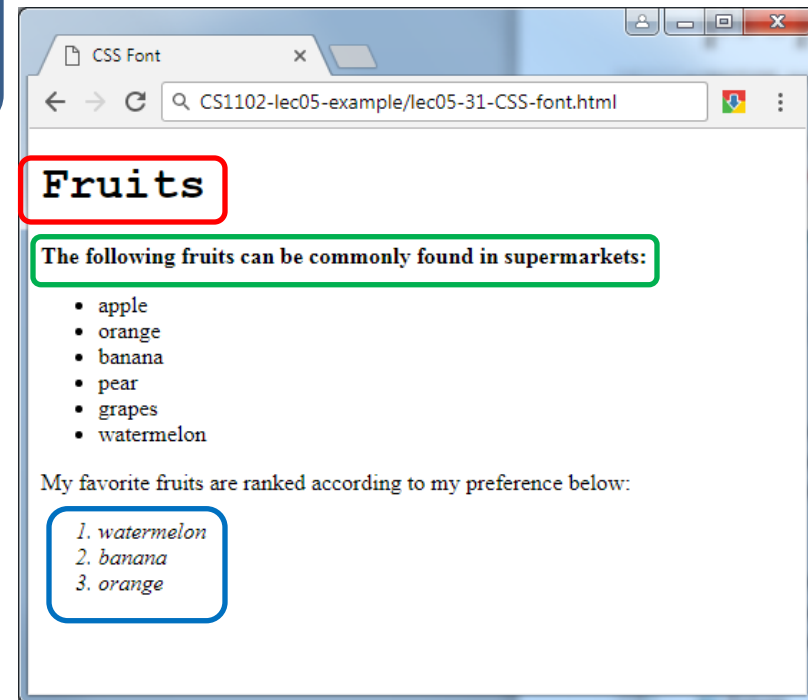
CSS: Font

- A CSS style can be applied to change various font properties

The font-family property can list fonts names. If a browser does not support the first font, it will then try the next font.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>CSS Font</title>
5     <style>
6       h1 {
7         font-family: "Courier New", Helvetica, Georgia;
8       }
9       #paragraph1 {
10        font-weight: bold;
11      }
12       ol {
13        font-style: italic;
14      }
15     </style>
16   </head>
17   <body>
18     <!-- Page content begins here -->
19     <h1>Fruits</h1>
20     <p id="paragraph1">The following fruits can be commonly found in supermarkets:</p>
21     <ul>
22       <li>apple</li>
23       <li>orange</li>
24       <li>banana</li>
25       <li>pear</li>
26       <li>grapes</li>
27       <li>watermelon</li>
28     </ul>
29
30     <p>My favorite fruits are ranked according to my preference below:</p>
31     <ol>
32       <li>watermelon</li>
33       <li>banana</li>
34       <li>orange</li>
35     </ol>
36     <!-- Page content ends here -->
37   </body>
38 </html>
```

Code Example: lec05-31-CSS-font.html

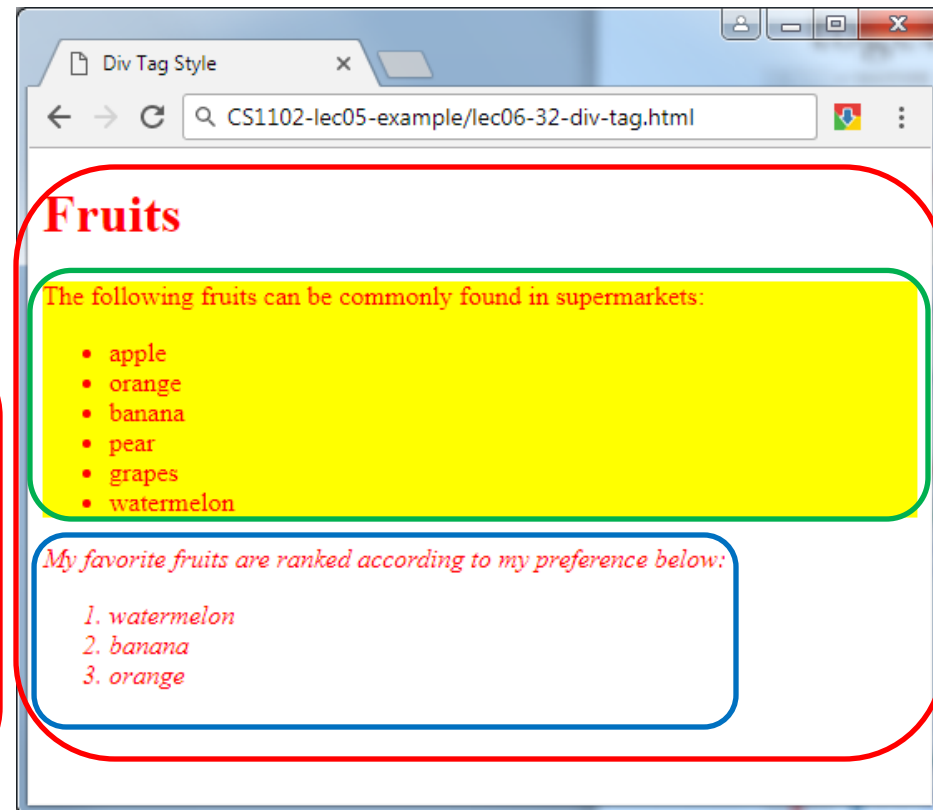


The <div> tag

- The <div> tag is often used to group different parts of a webpage together to form a section such that a style can be applied.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Div Tag Style</title>
5     <style>
6       #main_container {
7         color: red;
8       }
9       #common_fruit {
10        background-color: yellow;
11      }
12       #favorite_fruit {
13        font-style: italic;
14      }
15     </style>
16   </head>
17   <body>
18     <!-- Page content begins here -->
19     <div id="main_container">
20       <h1>Fruits</h1>
21       <div id="common_fruit">
22         <p>The following fruits can be commonly found in supermarkets:</p>
23         <ul>
24           <li>apple</li>
25           <li>orange</li>
26           <li>banana</li>
27           <li>pear</li>
28           <li>grapes</li>
29           <li>watermelon</li>
30         </ul>
31       </div>
32       <div id="favorite_fruit">
33         <p>My favorite fruits are ranked according to my preference below:</p>
34         <ol>
35           <li>watermelon</li>
36           <li>banana</li>
37           <li>orange</li>
38         </ol>
39       </div>
40     </div>
41     <!-- Page content ends here -->
42   </body>
43 </html>
```

Code Example: lec05-32-div-tag.html



Javascript

- Javascript is a programming language that can provide instructions for a browser to dynamically generate content for a website or enhance the website interactivity
- Javascript can be embedded in the head section of the webpage, with the code defined inside the `<script> </script>` tags

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>JS 1st Example</title>
5     <script>
6       function myclick() {
7         alert("Welcome to CS1102!");
8       }
9     </script>
10  </head>
11  <body>
12    <!-- Page content begins here -->
13    <h1>CS1102</h1>
14    <h2 onclick="myclick();">Click Me</h2>
15    <!-- Page content ends here -->
16  </body>
17 </html>
```

In Scratch, you drag and drop the blocks to form your code.
In Javascript, you have to type the code and you need to be careful to avoid typos which will cause errors in your program.

The semi-colon ; is placed at the end of each Javascript statement



Javascript: Website Interactivity

- Website interactivity can be enhanced by detecting a user event and defining the corresponding event handler to perform certain action, e.g., the following program has an event handler that detects if the user clicks on the h2 heading “Click Me” and then calls the function `myclick` to pop up a message.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
8 <title>Javascript First Example</title>
9 <script>
10     function myclick() {
11         alert("Welcome to CS1102!");
12     }
13 </script>
14 </head>
15 <body>
16 <!-- Page content begins here -->
17 <h1>CS1102</h1>
18 <h2 onclick="myclick();">Click Me</h2>
19 <!-- Page content ends here -->
20 </body>
21 </html>
```

Code Example:
lec05-34-JS-first-example.html

`myclick` is a self-defined function which is defined here

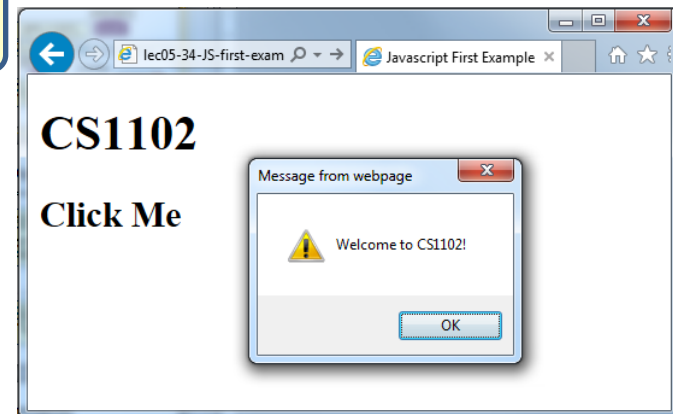
The statement `alert("[message]")` will pop up a window and display the message specified inside the parentheses ()

Scratch Analogy

`myclick`

The attribute `onclick` is an event handler that is invoked when this h2 element is clicked. In this case, the Javascript function `myclick()` will be called

Scratch Analogy



Javascript: Dynamic Content

- The innerHTML property allows the content of an HTML element to be changed dynamically

- Originally the “welcome” div has no content: `<div id=“welcome”></div>`
- After the page has finished loading, the on load function is invoked which will call the `showDynamicContent()` function
- The following statement
`document.getElementById(“welcome”).innerHTML = “<h2>Welcome!</h2>”`
replaces the current content of the “welcome” div with the string “<h2>Welcome!</h2>” such that the webpage will be displayed as if the html of the “welcome” div is
`<div id=“welcome”><h2>Welcome!</h2></div>`

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Javascript Dynamic Content</title>
5     <script>
6       function showDynamicContent() {
7         document.getElementById("welcome").innerHTML="<h2>Welcome!</h2>";
8       }
9     </script>
10  </head>
11  <body onload="showDynamicContent();">
12    <!-- Page content begins here -->
13    <h1>CS1102</h1>
14    <div id="welcome"></div>
15    <!-- Page content ends here -->
16  </body>
17 </html>
```

Code Example: lec05-35-JS-innerHTML.html

Usually the on load event handler is added as an attribute in the body tag and is used to call some Javascript function to carry out some tasks right after the webpage has finished loading to initialize some settings



Comments

- HTML, CSS and Javascript allow programmers to insert comments in the code but their syntax differs
- Comments will be ignored by the browser when the webpage is displayed but it is a good practice to insert comments to document what the code logic, such that it will be easy for others to understand your code or for you to revisit your code later

In CSS, comments can be placed between `/*` and `*/` and can span multiple lines

In Javascript, there are 2 ways to add comments:

- 1) Similar to CSS, Javascript comments can be placed between `/*` and `*/` and can span multiple lines
- 2) Javascript comments can also be placed after `//` until the end of line so this is a single line comment. Note that CSS does not support this single line comment style

In HTML, comments can be placed between `<!--` and `-->` and can span multiple lines

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Comments</title>
5   <style>
6     /* The following CSS style sets the corresponding div element
7      * with color red */
8     #course {
9       color: red;
10    }
11  </style>
12  <script>
13    function init() { // this function is called after the webpage has been loaded
14      /* The following statement dynamically replaces the content
15       * of the corresponding div element by the given string */
16      document.getElementById("course").innerHTML="<h2>Introduction to Computer Studies</h2>";
17    }
18  </script>
19 </head>
20 <body onload="init();" >
21   <!-- Page content begins here -->
22   <h1>CS1102</h1>
23   <!-- the following div element's content is empty in the HTML
24      * and will be assigned by Javascript after the webpage has been loaded -->
25   <div id="course"></div>
26   <!-- Page content ends here -->
27 </body>
28 </html>
```

Code Example: lec05-36-comment.html

Javascript: Prompting User Input

- Similar to the block  in Scratch, Javascript can use the `prompt ( )` function to actively ask the user to input something

In Javascript, variables can be declared by using the syntax `var` followed by the variable name. There are several rules for defining valid variable names:

1. Names are case sensitive, e.g., `name` and `Name` are different variables
2. Names can begin with letters, underscore `_`, dollar sign `$`, but not digits
3. Digits can be used in names as long as it is not the first character
4. Names cannot be the same as reserved words such as `var`, `prompt`, etc.

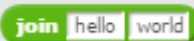
Javascript Reserved Words:

https://www.w3schools.com/js/js_reserved.asp

The `prompt()` function is a Javascript built-in function and it pops up a dialog box asking the user to provide input on a textbox. The output of this function is the string input by the user on the textbox

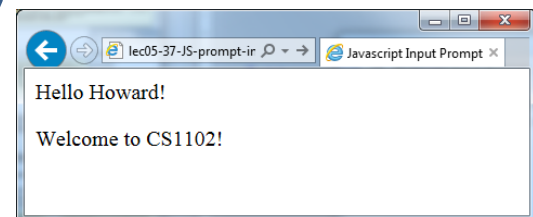
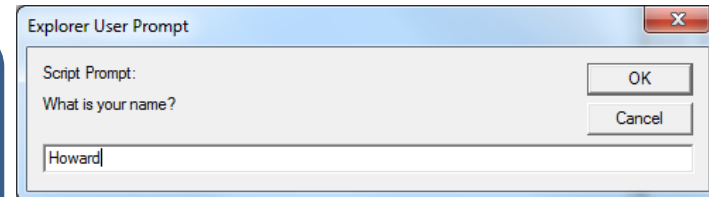
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Javascript Input Prompt</title>
5     <script>
6       function init() {
7         var name; // declare a variable
8         name = prompt("What is your name?"); // assign the user input to the variable
9         document.getElementById("hello").innerHTML = "Hello " + name + "!";
10      }
11    </script>
12  </head>
13  <body onload="init();" >
14    <!-- Page content begins here -->
15    <div id="hello"></div>
16    <p>Welcome to CS1102!</p>
17    <!-- Page content ends here -->
18  </body>
19 </html>
```

"Hello" + name + "!" concatenate these strings together to form a longer string. This is similar to the block



in Scratch

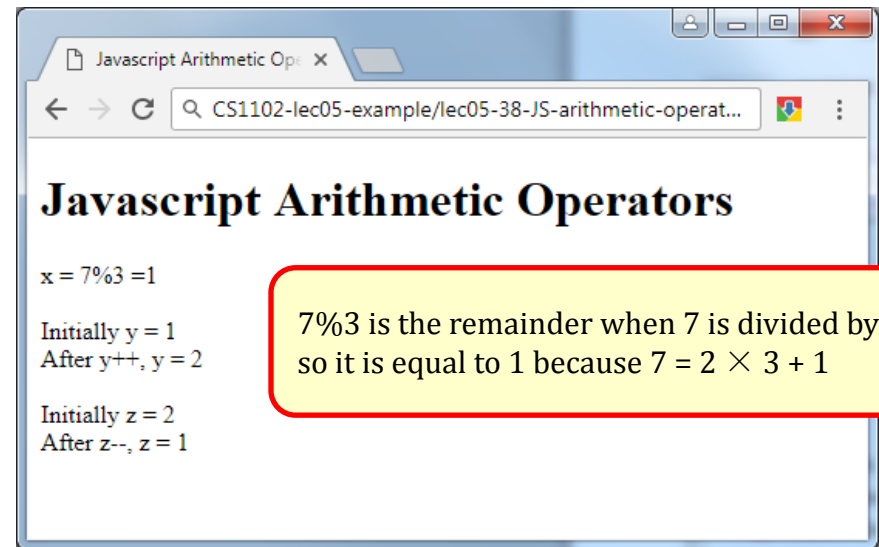
Code Example: lec05-37-JS-prompt-input.html



Javascript: Arithmetic Operators

- Javascript arithmetic operators:
 - The 4 operators +, -, *, / are intuitive
 - % is the modulus operator like 
 - ++ is the increment operator, x++ will increase the value of the variable x by 1
 - -- is the decrement operator, x-- will decrease the value of the variable x by 1

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Javascript Arithmetic Operators</title>
5 <script>
6     function init() {
7         var s = "";
8         var x,y,z;
9
10        x = 7%3;
11        s = s + "x = 7%3 = " + x + "<br /><br />";
12
13        y = 1;
14        s = s + "Initially y = " + y + "<br />";
15        y++;
16        s = s + "After y++, y = " + y + "<br /><br />";
17
18        z = 2;
19        s = s + "Initially z = " + z + "<br />";
20        z--;
21        s = s + "After z--, z = " + z + "<br /><br />";
22
23        document.getElementById("display").innerHTML = s;
24    }
25 </script>
26 </head>
27 <body onload="init();">
28 <!-- Page content begins here -->
29 <h1>Javascript Arithmetic Operators</h1>
30 <div id="display"></div>
31 <!-- Page content ends here -->
32 </body>
33 </html>
```



Code Example: lec05-38-JS-arithmetic-operators.html

Javascript: Comparison and Logical Operators

- Javascript comparison operators: `<`, `<=`, `>`, `>=`, `==`, `!=`
 - The first 4 operators `<`, `<=`, `>`, `>=` are intuitive
 - The operator `==` checks whether the expressions on the left hand side and right hand side are equal. This should not be confused with the assignment operator `=`
 - The operator `!=` checks whether the expressions are different (not equal to)

Scratch Analogy



- Javascript logical operators:

Scratch Analogy

- `&&` is the AND operator
- `||` is the OR operator
- `!` is the NOT operator



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Javascript Comparison and Logical Operators</title>
5   <script>
6     function init() {
7       var s = "";
8       var input = prompt("Enter a month (1-12):");
9       var month = Number(input);
10
11       if (month>8 && month <=12)
12         s = s+"This month is in Semester A\n";
13
14       if (month == 12 || month < 3)
15         s = s+"This is a winter month.\n";
16
17       if (month != 12)
18         s = s+"Santa Claus is not coming this month.\n";
19
20       alert(s);
21     }
22   }
23 </script>
24 </head>
25 <body onload="init();">
26   <!-- Page content begins here -->
27   <h1>Logical Operators</h1>
28   <!-- Page content ends here -->
29 </body>
30 </html>
```

The expression `(month>8 && month <=12)` is true if month > 8 AND month <=12, i.e., when month is equal to 9, 10, 11, 12

The expression `(month==12 || month < 3)` is true if month is equal to 12 OR month < 3, i.e., when month is equal to 12, 1, 2

The expression `(month!=12)` is true if month is NOT equal to 12, i.e., when month is equal to 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

Code Example: lec05-39-JS-comparison-logical-operators.html

Javascript: One-Way Conditional

Code Example: lec05-40-JS-one-way-conditional.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
8 <title>Javascript One-Way Conditional</title>
9 <script>
10 function init() {
11     var p, s;
12     var currentDate = new Date();
13     s = "";
14     p = prompt("What is your birth month? (1-12)");
15     if (Number(p) == currentDate.getMonth() + 1)
16         alert("Happy Birthday!");
17 }
18 </script>
19 </head>
20 <body onload="init();" >
21 <!-- Page content begins here -->
22 <h1>One Way Conditional</h1>
23 <!-- Page content ends here -->
24 </body>
25 </html>
```

The variable s is initialized to be the empty string ""

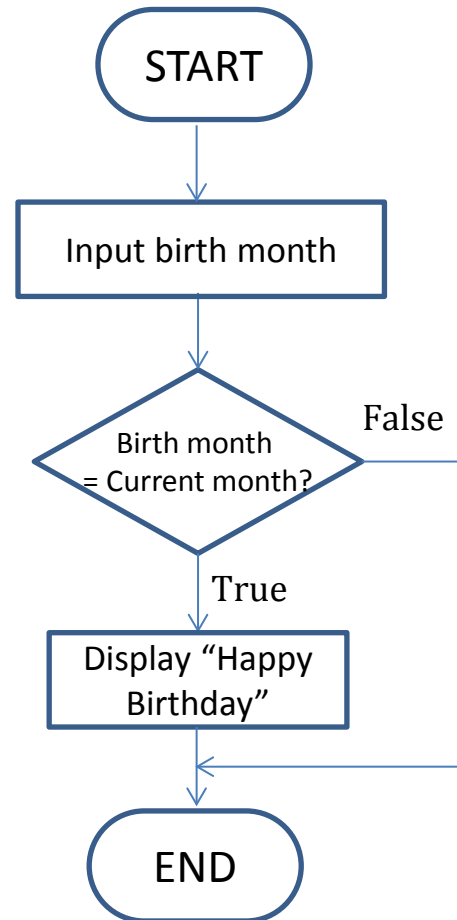
The if statement behaves in a similar way as the Scratch block



The expression `currentDate.getMonth()` returns the current month in the range from 0 to 11 denoting January to December. After adding 1, the expression `currentDate.getMonth() + 1` will thus have the range from 1 to 12. This is analogous to the Scratch block



The variable p is assigned as a string (text) from the output of the prompt function. The expression `Number(p)` converts the string p to its numerical value so that it can be manipulated as a number



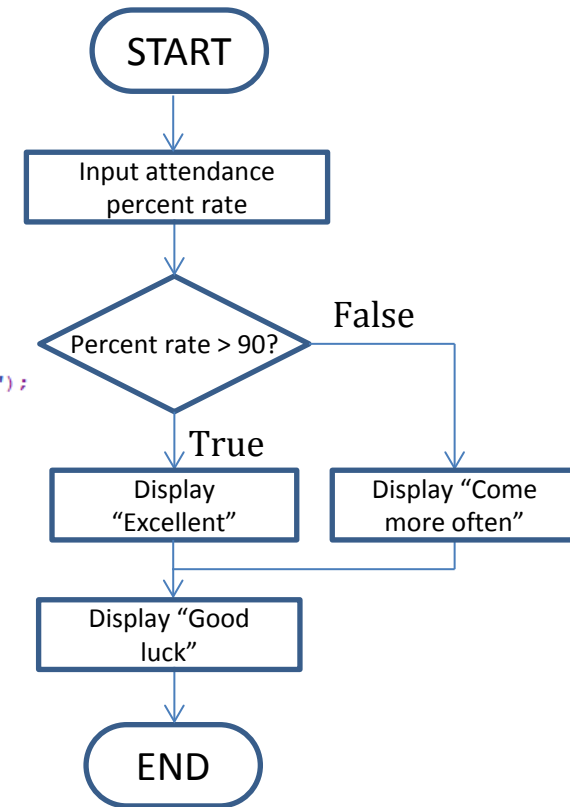
Javascript: Two-Way Conditional

Code Example: lec05-41-JS-two-way-conditional.html

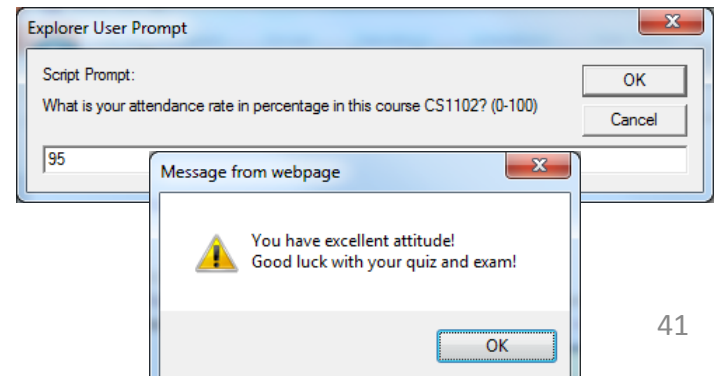
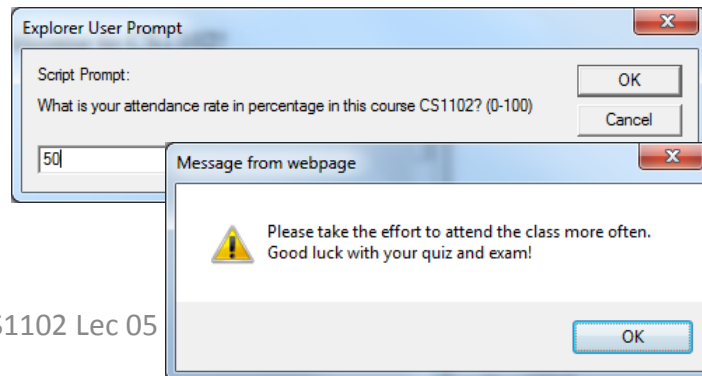
The if-else statement behaves in a similar way as the Scratch block



\n specifies that a line break should be added such that the subsequent text will be displayed in a new line when shown by the alert() function



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Javascript Two-Way Conditional</title>
5 <script>
6     function init() {
7         var p, s;
8         s = "";
9         p = prompt("What is your attendance rate in percentage in this course CS1102? (0-100)");
10        if (Number(p)>90)
11            s = s+"You have excellent attitude!\n";
12        else
13            s = s+"Please take the effort to attend the class more often.\n";
14            s = s+"Good luck with your quiz and exam!";
15        alert(s);
16    }
17 </script>
18 </head>
19 <body onload="init();">
20 <!-- Page content begins here -->
21 <h1>Two-Way Conditional</h1>
22 <!-- Page content ends here -->
23 </body>
24 </html>
```



Javascript: N-Way Conditional

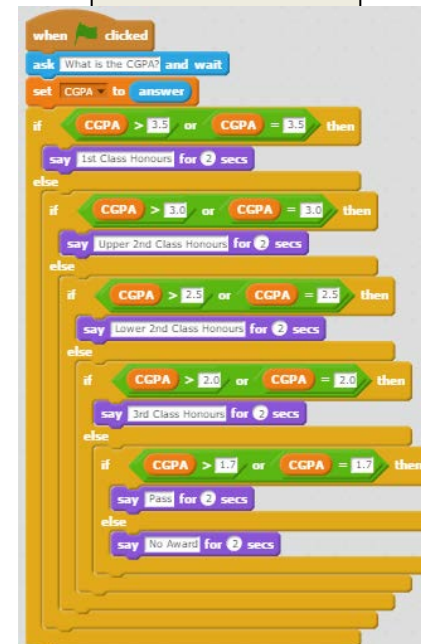
Code Example: lec05-42-JS-N-way-conditional.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
8 <title>Javascript N-Way Conditional</title>
9 <script>
10 function init() {
11     var p, s, cgpa;
12     s = "";
13     p = prompt("What is the CGPA");
14     cgpa = Number(p);
15     if (cgpa >= 3.5)
16         s = "1st Class Honours";
17     else if (cgpa >= 3.0)
18         s = "Upper 2nd Class Honours";
19     else if (cgpa >= 2.5)
20         s = "Lower 2nd Class Honours";
21     else if (cgpa >= 2.0)
22         s = "3rd Class Honours";
23     else if (cgpa >= 1.7)
24         s = "Pass";
25     else
26         s = "No Award";
27     alert(s);
28 }
29 </script>
30 </head>
31 <body onload="init();">
32 <!-- Page content begins here -->
33 <h1>N-Way Conditional</h1>
34 <!-- Page content ends here -->
35 </body>
36 </html>
```

an N-way conditional can be constructed by placing one if-else statement inside another one in a *nested* manner

CGPA	Boolean Expression	Award Classification
3.5 or above	CGPA>=3.5	1st Class Honours
3.0-3.49	CGPA>=3.0 AND CPGA<3.5	Upper 2nd Class Honours
2.5-2.99	CGPA>=2.5 AND CPGA<3.0	Lower2nd Class Honours
2.0-2.49	CGPA>=2.0 AND CPGA<2.5	3rd Class Honours
1.7-1.99	CGPA>=1.7 AND CPGA<2.0	Pass
<1.7	CPGA<1.7	No Award

Scratch Analogy



Javascript: For-Loop

- The for-loop is often used to carry out a task for a finite number of times

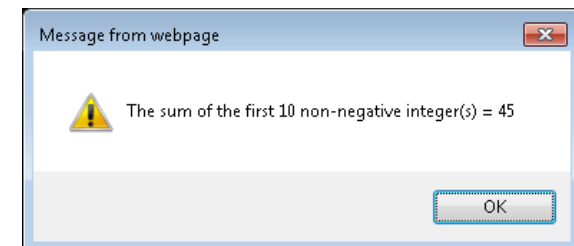
The for-statement, e.g. `for (i=0; i<N; i++)`, contains 3 parts inside the parentheses:

1. Initialization: the code is executed at the beginning of the loop
e.g., `i=0` assigns 0 to the variable `i` right after the statement `sum=0`;
 2. Continuation condition: the tasks specified in the loop are carried out if the continuation condition in the form of Boolean expression is true, otherwise the loop ends if the continuation condition is false
e.g., `i<10` is true when `i=0, 1, 2, 3, 4, 5, 6, 7, 8, 9` but is false when `i=10`;
 3. Increment statement: this part is executed at the end of each iteration of the loop
e.g., `i++` means that the variable `i` is increased by 1 and it is executed at the end of each iteration of the loop after the statement `sum = sum + i`;
- Note that there should be no semi-colon after the parenthesis, i.e. `for (i=0; i<N; i++);` is wrong

Code Example: lec05-43-JS-for-loop.html

```
1  <!DOCTYPE html>
2  <html>
3  <head>
8    <title>Javascript For-Loop</title>
9    <script>
10     function init() {
11         var i, N, sum;
12         N = 10;
13         sum = 0;
14         for (i=0; i<N; i++) {
15             sum = sum + i;
16         }
17         alert("The sum of the first "+N+" non-negative integer(s) = "+sum);
18     }
19 </script>
20 </head>
21 <body onload="init();">
22 <!-- Page content begins here -->
23 <p>Adding the first N integers</p>
24 <!-- Page content ends here -->
25 </body>
26 </html>
```

The curly brackets after the for-statement enclose the statements that are executed at each iteration of the for-loop, in this example, `sum = sum + i`;
You can put multiple statements inside the curly brackets such that all of them will be executed at each iteration of the loop



Javascript: For-Loop (cont.)

```
N = 10;  
sum = 0;  
for (i=0; i<N; i++) {  
    sum = sum + i;  
}
```

The code on the left is executed according to the following sequence of operations:

1. N=10
2. sum=0
3. i=0
4. $i < N \Leftrightarrow 0 < 10 = \text{true}$ so the loop will continue to run
5. $\text{sum} = \text{sum} + i \Leftrightarrow \text{sum} = 0 + 0 = 0$
6. $i++ \Leftrightarrow i = 0 + 1 = 1$
7. $i < N \Leftrightarrow 1 < 10 = \text{true}$ so the loop will continue to run
8. $\text{sum} = \text{sum} + i \Leftrightarrow \text{sum} = 0 + 1 = 1$
9. $i++ \Leftrightarrow i = 1 + 1 = 2$
10. $i < N \Leftrightarrow 2 < 10 = \text{true}$ so the loop will continue to run
11. $\text{sum} = \text{sum} + i \Leftrightarrow \text{sum} = 1 + 2 = 3$
12. $i++ \Leftrightarrow i = 2 + 1 = 3$
13. $i < N \Leftrightarrow 3 < 10 = \text{true}$ so the loop will continue to run
14. $\text{sum} = \text{sum} + i \Leftrightarrow \text{sum} = 3 + 3 = 6$
15. $i++ \Leftrightarrow i = 3 + 1 = 4$
16. $i < N \Leftrightarrow 4 < 10 = \text{true}$ so the loop will continue to run
17. $\text{sum} = \text{sum} + i \Leftrightarrow \text{sum} = 6 + 4 = 10$
18. $i++ \Leftrightarrow i = 4 + 1 = 5$
19. $i < N \Leftrightarrow 5 < 10 = \text{true}$ so the loop will continue to run
20. $\text{sum} = \text{sum} + i \Leftrightarrow \text{sum} = 10 + 5 = 15$
21. $i++ \Leftrightarrow i = 5 + 1 = 6$
22. $i < N \Leftrightarrow 6 < 10 = \text{true}$ so the loop will continue to run
23. $\text{sum} = \text{sum} + i \Leftrightarrow \text{sum} = 15 + 6 = 21$
24. $i++ \Leftrightarrow i = 6 + 1 = 7$
25. $i < N \Leftrightarrow 7 < 10 = \text{true}$ so the loop will continue to run
26. $\text{sum} = \text{sum} + i \Leftrightarrow \text{sum} = 21 + 7 = 28$
27. $i++ \Leftrightarrow i = 7 + 1 = 8$
28. $i < N \Leftrightarrow 8 < 10 = \text{true}$ so the loop will continue to run
29. $\text{sum} = \text{sum} + i \Leftrightarrow \text{sum} = 28 + 8 = 36$
30. $i++ \Leftrightarrow i = 8 + 1 = 9$
31. $i < N \Leftrightarrow 9 < 10 = \text{true}$ so the loop will continue to run
32. $\text{sum} = \text{sum} + i \Leftrightarrow \text{sum} = 36 + 9 = 45$
33. $i++ \Leftrightarrow i = 9 + 1 = 10$
34. $i < N \Leftrightarrow 10 < 10 = \text{false}$ so the loop will end

1st iteration

2nd iteration

3rd iteration

4th iteration

5th iteration

6th iteration

7th iteration

8th iteration

9th iteration

10th iteration

Javascript: While-Loop

- The while-loop is used to carry out a task repeatedly as long as a continuation condition is true

Code Example: lec05-45-JS-while-loop.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Javascript While-Loop</title>
5 <script>
6
7     function init() {
8         var isInputValid, number;
9         isInputValid = false;
10        while (!isInputValid) {
11            number = prompt("Input a positive integer");
12            if (isNaN(number)) {
13                alert("Please enter a NUMBER!");
14            }
15            else if (Number(number) <= 0) {
16                alert("Please enter a POSITIVE number!");
17            }
18            else {
19                isInputValid = true;
20            }
21            alert("The positive number that you entered is "+number);
22        }
23    }
24 </script>
25 </head>
26 <body onload="init();">
27 <!-- Page content begins here -->
28 <p>Checking for Positive Number</p>
29 <!-- Page content ends here -->
30 </body>
31 </html>
```

isInputValid is a Boolean variable which has value true or false

- it is set to be false initially
- it will be set to true if the user inputs a positive number

! is the NOT operator and will negate its subsequent Boolean expression

isInputValid	!isInputValid
true	false
false	true

isNaN is Javascript built-in function and is used to check whether the given parameter is not a number. It will return true if it is not a number and false if it is a number, e.g.,

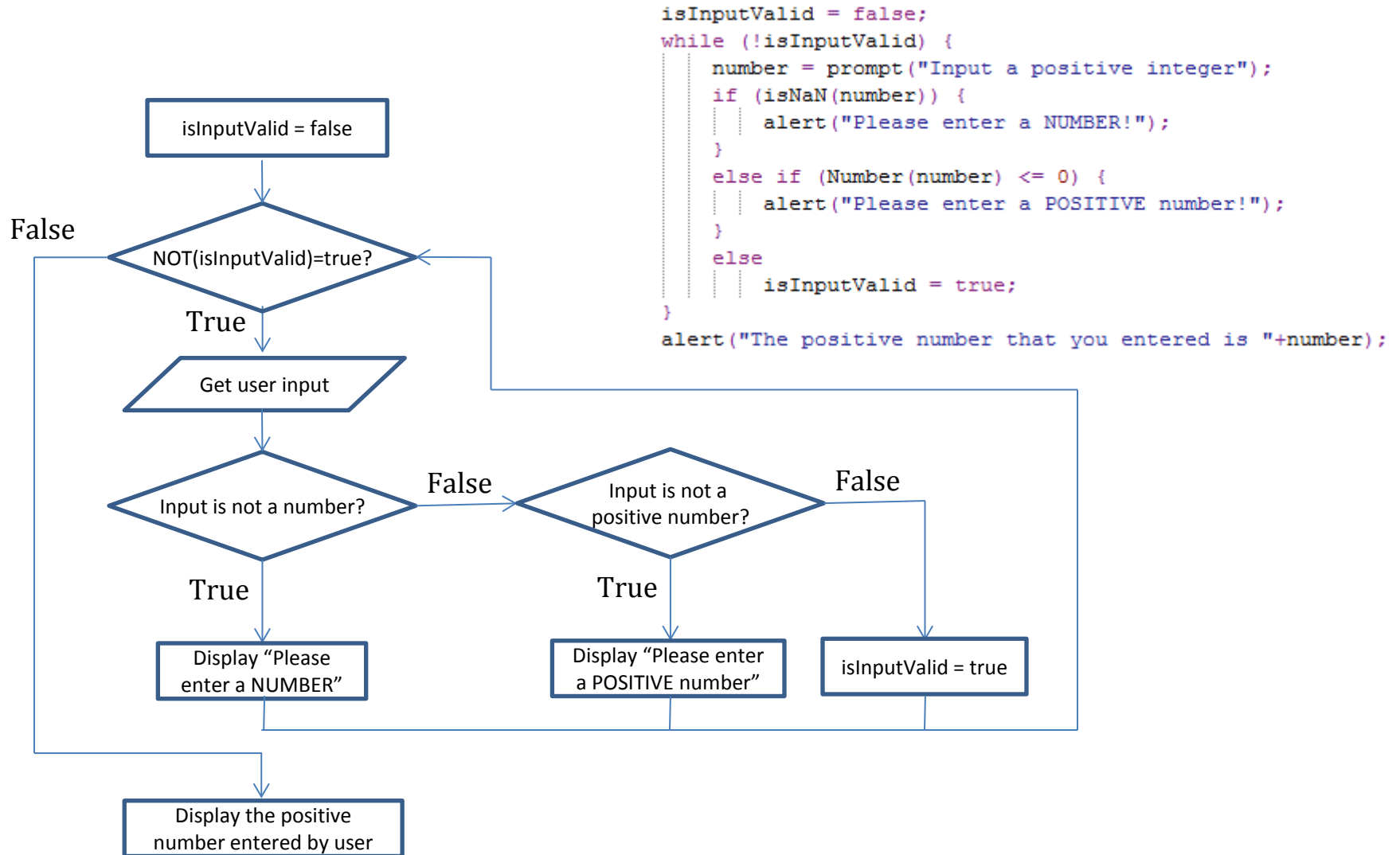
isNaN(234)=false
isNaN("abc")=true

Number is Javascript built-in function and is used to convert the given parameter to a number according to its value such that numeric calculations can be applied, e.g.,

Number("123")=123
Number("123")+1 = 124
however, "123"+1 = "1231"

The curly brackets after the while-statement enclose the statements that are executed at each iteration of the while-loop

Javascript: While-Loop (cont.)



Javascript: Button

- A button can be created by using the HTML tag <button>
- The attribute onclick can be used to handle the event when the button is clicked

Code Example: lec05-47-JS-button.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
8 <title>Javascript Button</title>
9 <script>
10
11 function init() {
12     document.getElementById("number").style = "font-size:100px;";
13 }
14 function throw_dice() {
15     var n;
16     n = Math.floor(Math.random()*6)+1;
17     document.getElementById("number").innerHTML = n;
18 }
19 </script>
20 </head>
21 <body onload="init();" >
22 <!-- Page content begins here -->
23 <h1>Javascript Button</h1>
24 <button id="dice" onclick="throw_dice();" >Throw Dice</button>
25 <br/>
26 <div id="number" ></div>
27 <!-- Page content ends here -->
28 </body>
</html>
```

The style can be set by Javascript

The expression `Math.floor(Math.random()*6)+1` will generate a random integer between 1 and 6

The attribute onclick is an event handler that is invoked when this button is clicked. In this case, the Javascript function `throw_dice()` will be called

Javascript built-in function

`Math.floor()`

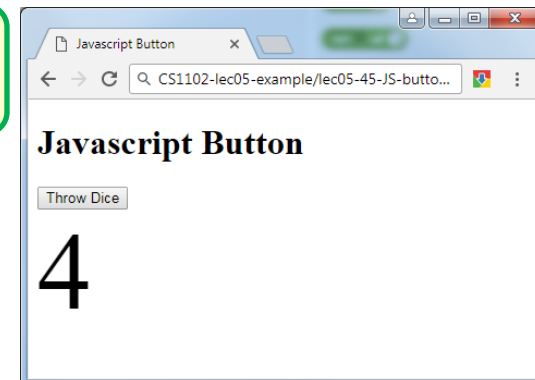
`Math.random()`

Scratch Analogy

floor of

pick random 0 to 1.0

`Math.random()` outputs a pseudo-random floating-point number between 0 (inclusive) and 1 (exclusive)



Lesson Summary

- HyperText Markup Language (HTML) uses a set of codes called tags to describe the structure of a webpage
- Cascading Style Sheets (CSS) describes how the HTML elements should be displayed by specifying the fonts, colors, layout and placement of these HTML elements
- Javascript is a programming language that can provide instructions for a browser to dynamically generate content for a website or enhance the website interactivity
- You will not be able to learn everything about HTML, CSS and Javascript after only one or two lectures, but it serves as a starting point such that you can explore other features on your own

Reference

[1] HTML5 Tutorial

- <http://www.w3schools.com/html>

[2] Web Hypertext Application Technology Working Group (WHATWG)

- <http://www.whatwg.org>

[3] CSS3 Introduction

- http://www.w3schools.com/css/css3_intro.asp

[4] Javascript Tutorial

- <http://www.w3schools.com/js/>