

# CS2204 Fundamentals of Internet Applications Development

## Lecture 2 HTML – Part 2

*Computer Science, City University of Hong Kong*

*Semester B 2024-25*

# A quick review from the lecture 01

The concepts and terms in computer network and communication system

Network protocols: TCP/IP, HTTP, SSH ...

How to identify the address of a website or a web applications on the Internet

Examples of Internet applications

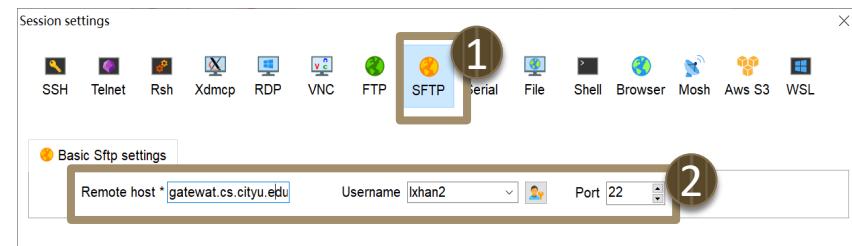
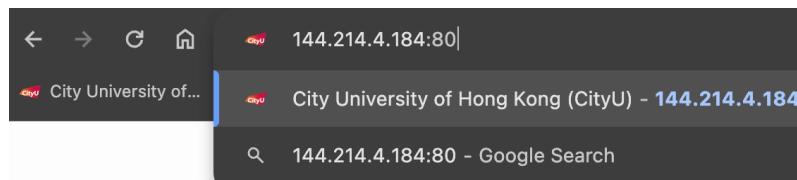
HTML Introduction

# Clarification about port number

You can access CityU website either by <https://www.cityu.edu.hk/> or **144.214.4.184**

- The **default** port number that is listening to the **webpage visit** request is **80**, thus, visiting **144.214.4.184** is the same as visiting **144.214.4.184: 80**

In Lab 1, when we access a remote server using port **22**, that is the **default** port set for **file transfer** with SFTP





## Question time

1. What are the full forms of URL and DNS?
2. What are the differences between IPv4 and IPv6?

# Agenda

**Webpage and Source Code Organization**

**HTML: Commonly Used Tags**

**File Path**

**Web Accessibility Design**

# Agenda

## **Webpage and Source Code Organization**

### HTML: Commonly Used Tags

### File Path

### Web Accessibility Design

# Recap: Structure of a HTML page

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

The diagram shows a snippet of HTML code with line numbers on the left. Annotations with arrows point to specific parts of the code:

- An arrow points from a box labeled "opening html tag" to the line <html>.
- A bracket on the right side of the code is labeled "head section" and groups the <head> and <title>Introduction</title> tags.
- A bracket on the right side of the code is labeled "body section" and groups the <body> tag and its three <p> tags.
- An arrow points from a box labeled "closing html tag" to the line </html>.
- An arrow points from a box labeled "Code Example: lec01-01-example.html" to the code.

```
1  <!DOCTYPE html>
2
3  <html>
4  <head>
5      <title>Introduction</title>
6  </head>
7
8  <body>
9      <p>This is a sample page</p>
10     <p>This is a sample page</p>
11     <p>This is a sample page</p>
12 </body>
13 </html>
```

This is a simple example.

This is a simple example.

This is a simple example.

# Webpage structure & Source Code

A webpage that displays information with text, images, audio, video can be viewed on a browser.

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** (and CSS, JavaScript, which we will cover later in the course)

Welcome to CS2204!

**Hello student**

The Course Intended Learning Outcomes (CILOs) of CS2204 are:

- Explain the development of WEB and its current trends.
- Use of Internet development tools such as XHTML editor
- Design and implement static Web pages using WEB standards.
- Create and set up Web sites and write interactive Web pages.
- Understand, compare and evaluate the design techniques required for Internet applications.

Hope you have a positive learning experience and *have fun* in this course!

Code Example: lec02-01-webpage-and-sourceCode.html

JavaScript, within  
<script></script>

HTML, within  
<body></body>

```
Line wrap □
1 !DOCTYPE html>
2 <html>
3   <head>
4     <title>Webpage and Source Code</title>
5     <style>
6       #main_container {
7         width: 900px;
8       }
9       #leftbox {
10        width: 150px;
11      }
12      #rightbox {
13        border-style: solid;
14        border-width: thin;
15        border-color: blue;
16        float: right;
17        padding: 5px;
18      }
19      #fun {
20        font-style: italic;
21      }
22      #footer {
23        clear: both;
24      }
25    </style>
26    <script>
27      function askName(){
28        document.getElementById("guest").innerHTML = prompt("What is your name?");
29      }
30    </script>
31  </head>
32  <body onload="askName();">
33    <div id="main_container">
34      <h1>Welcome to CS2204!</h1>
35      <div id="leftbox">
36        <h2>Hello <span id="guest"></span></h2>
37        <ul>
38          <li>The Course Intended Learning Outcomes (CILOs) of CS2204 are:
39            <ul>
40              <li>Explain the development of WEB and its current trends.</li>
41              <li>Use of Internet development tools such as XHTML editor</li>
42              <li>Design and implement static Web pages using WEB standards.</li>
43              <li>Create and set up Web sites and write interactive Web pages.</li>
44              <li>Understand, compare and evaluate the design techniques required for Internet app
45            </ul>
46          </li>
47        </ul>
48        <div id="rightbox">
49          <div>
50            
51          </div>
52        </div>
53      <div id="footer">
54        <p>Hope you have a positive learning experience and <span id="fun">have fun</span> in
55        </p>
56      </div>
57    </div>
58  </body>
59</html>
```

CSS, within <style></style>

# Doctype

An instruction that specifies the version of HTML used in a web document.

- `<!DOCTYPE>` declaration is based on a Document Type Definition (DTD)
- `<!DOCTYPE html>` means the current document is an html
- Before HTML 5, defining the document type can be very tedious

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

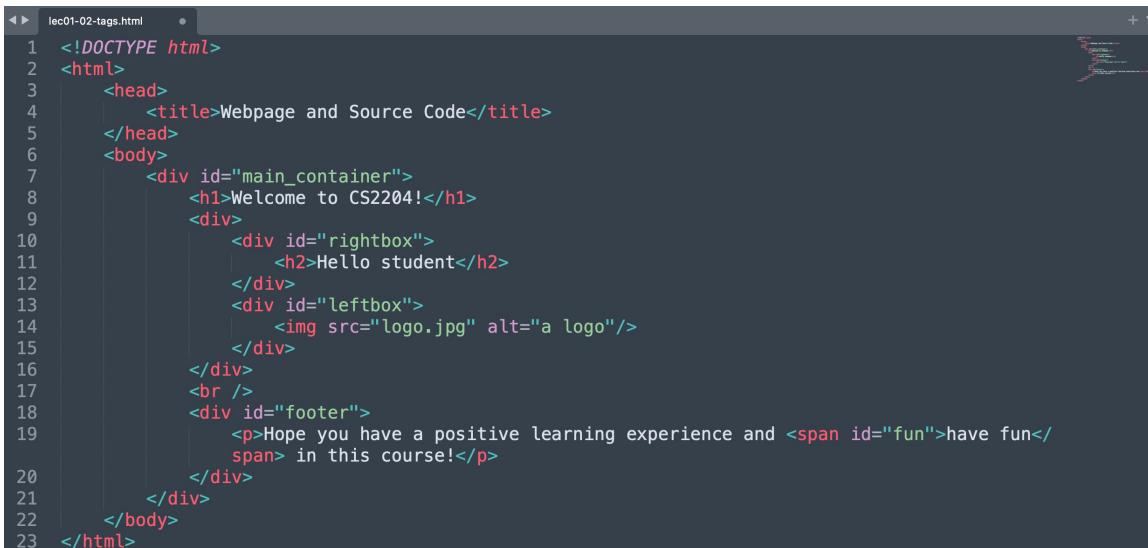
- HTML 5 later simplifies the doctype line

```
1  <!DOCTYPE html>  
2  
3  <html>  
4  |  <head>  
5  |  |  <title>Introduction</title>  
6  |  </head>  
7  |  
8  <body>  
9  |  <p>This is a sample page</p>  
10 |  <p>This is a sample page</p>  
11 |  <p>This is a sample page</p>  
12 |</body>  
13 </html>
```

# Recap: View webpage source code in a code editor

## Recommendations

- Visual Studio Code (<https://code.visualstudio.com/download>)
- Sublime: <https://www.sublimetext.com/>



The screenshot shows a code editor window with the file 'lec01-02-tags.html' open. The code displays a simple HTML structure with various tags and comments. The code is color-coded for readability, with tags like <html>, <head>, and <body> in blue, and <title> and <img> in green. Line numbers are visible on the left side of the code area.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Webpage and Source Code</title>
5   </head>
6   <body>
7     <div id="main_container">
8       <h1>Welcome to CS2204!</h1>
9       <div>
10         <div id="rightbox">
11           <h2>Hello student</h2>
12         </div>
13         <div id="leftbox">
14           
15         </div>
16       </div>
17       <br />
18       <div id="footer">
19         <p>Hope you have a positive learning experience and <span id="fun">have fun</span> in this course!</p>
20       </div>
21     </div>
22   </body>
23 </html>
```

## Advantages:

- Readability: color-coded tags
- Automatic indentation
- Line number
- Some editors also supports syntax checking

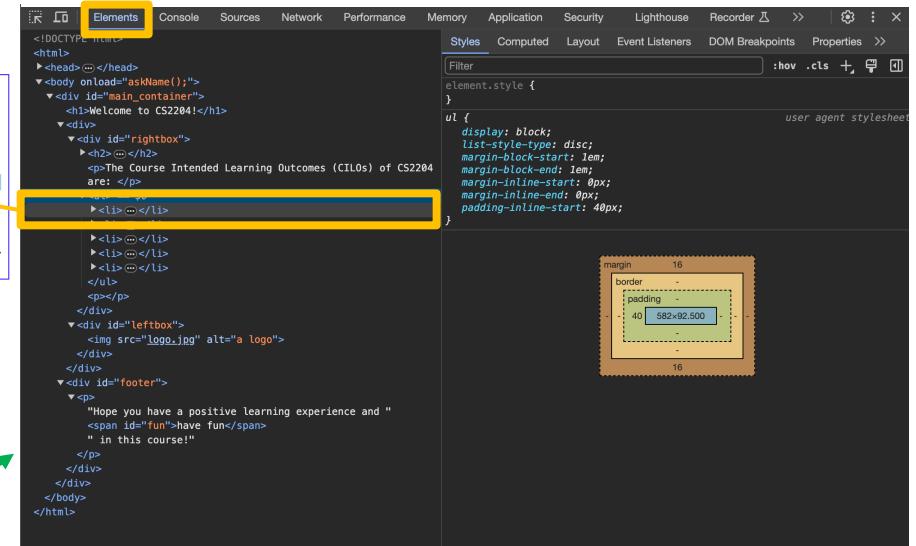
# View webpage source code in a browser

Right click the mouse on the page and select “Inspect”, a right panel will open up, showing how the HTML elements are placed in the current page

Later in the **style** panel, we can view CSS properties of each element; In **console** panel, we can debug JavaScript

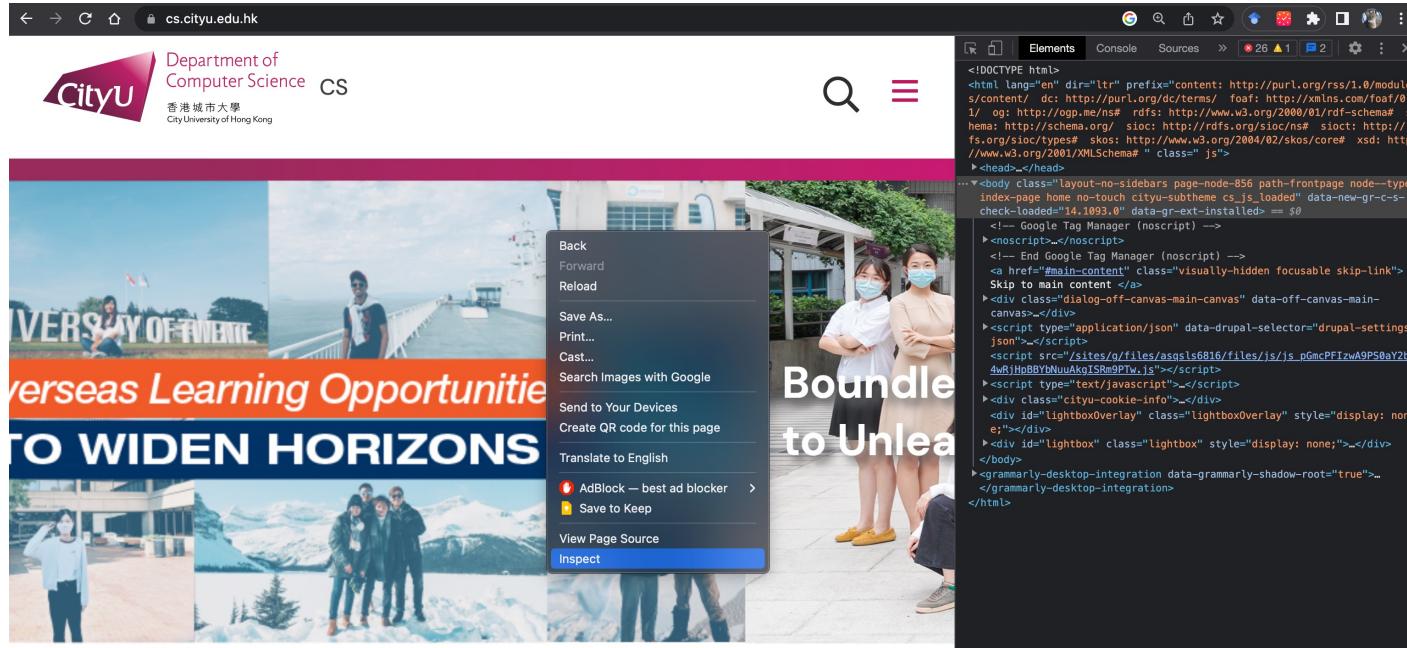
## Welcome to CS2204!

The CityU logo is visible in the top left corner. The main content area displays "Hello Yuhan" and a list of learning outcomes (CILOs) for CS2204. The context menu includes options like Back, Forward, Reload, Save As..., Print..., Cast..., Search Images with Google, Translate to English, Open in Reading mode, AdBlock — best ad blocker, Save to Keep, View Page Source, and Inspect.



# View the source code of any webpages

You can view the static source code of any webpage on a browser by “inspect” them



# Key components of 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

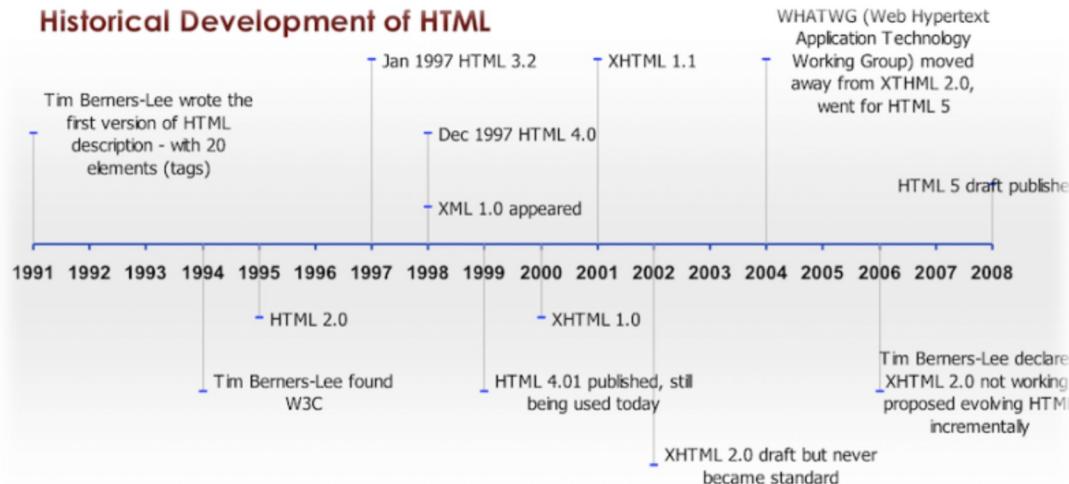
## 3. JavaScript

- Javascript is a programming language that can provide instructions for a browser to **dynamically** generate content for a website and enhance the website's **interactivity**

# Versions of HTML

Three main versions

- HTML (1991): HyperText Markup Language
- XHTML (2000): Extensible HTML, better suit for desktop
- HTML5 (2008): A newer version of the HTML, better suit for mobile



# Agenda

Webpage and Source Code Organization

**HTML: Commonly Used Tags**

File Path

Web Accessibility Design

# Recap: HTML Tags

The angle brackets `< >` together with the enclosed name are called a **tag** e.g., `<body>`, `<div>`

Most tags work **in pairs**, with an opening tag and a closing tag

- e.g., opening tag `<head>` and closing tag `</head>`
- Content - the text enclosed by the opening & closing tags

Some tags are **NOT** in pair, called **empty tags**, e.g., `<br>`, `<hr>`

- An empty tag therefore has no content.

Tags can also have properties, formally known as **attributes** used to specify its characteristics, e.g., `<div id="me">` div's content `</div>`

The diagram shows a snippet of HTML code with several annotations:

- A box labeled "attribute" with an arrow pointing to the attribute value "Webpage and Source Code" in the title tag.
- A box labeled "Opening tag" with an arrow pointing to the start of the main container div tag: `<div id="main_container">`.
- A box labeled "closing tag" with an arrow pointing to the end of the main container div tag: `</div>`.
- A box labeled "Empty tag" with an arrow pointing to the self-closing rightbox div tag: `<div id="rightbox"></div>`.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Webpage and Source Code</title>
  </head>
  <body>
    <div id="main_container">
      <h1>Welcome to CS2204!</h1>
      <div>
        <div id="rightbox">
          <h2>Hello student</h2>
        </div>
        <div id="leftbox">
          
        </div>
        <br />
        <div id="footer">
          <p>Hope you have a positive learning experience and <span id="fun">have fun</span> in this course!</p>
        </div>
      </div>
    </body>
  </html>
```

Code Example: lec01-02-tags.html

# Tags to explore

Head section <head>... </head>

- Adding a **title** of the webpage
- Define **meta** info

Body section <body>... </body>

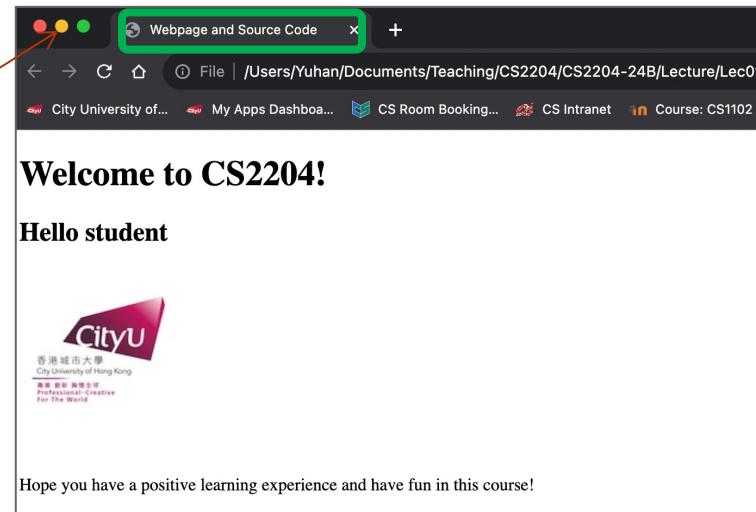
- Creating **headings** with **multiple levels**
- Formatting **paragraphs**
- Structure information in **list** and **table**
- Inserting **links** to the webpage
- Displaying **images** on the webpage
- Using **divs** to divide content

Footer section <footer>... </footer>

- The tags can be used are similar to the body section
- Usually displays authorship and copyright info, not required in a webpage

# Recap: Title (head section)

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Webpage and Source Code</title>
5   </head>
6   <body>
7     <div id="main_container">
8       <h1>Welcome to CS2204!</h1>
9       <div>
10         <div id="rightbox">
11           <h2>Hello student</h2>
12         </div>
13         <div id="leftbox">
14           
15         </div>
16       </div>
17       <br />
18       <div id="footer">
19         <p>Hope you have a positive learning experience and <span id="fun">have fun</span> in this course!</p>
20       </div>
21     </div>
22   </body>
23 </html>
```



The screenshot shows a web browser window titled "Webpage and Source Code". The page content includes a large "Welcome to CS2204!" heading, a "Hello student" sub-heading, and the CityU logo. A message at the bottom encourages a positive learning experience.

Code Example: lec01-02-tags.html

The page **title** can

- help 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** the webpage

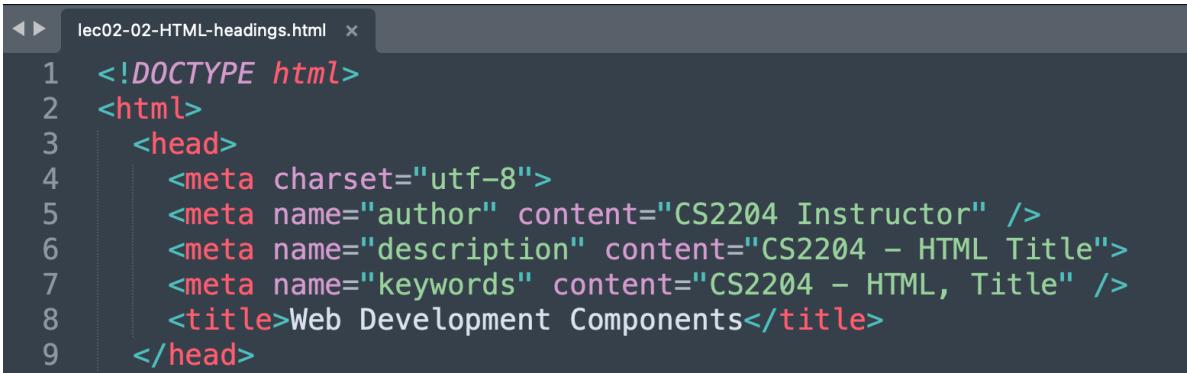
The page title is added between the `<title>` and `</title>` tags in the HTML file

# Meta (head section)

The <meta> tag provides information about your page, but it will **not** be shown on the webpage

The **name** attribute determines the usage, e.g., **author**, **description** or **keywords**. The **content** attribute contains the actual value of each name

- Description and keywords allow you to influence crawler/spider programs that support the tags, so that search engines can index your pages better and hopefully the quality of search results may be improved
- Most meta tags are **optional**, but the keywords, descriptions, etc. are usually used



```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <meta name="author" content="CS2204 Instructor" />
6          <meta name="description" content="CS2204 - HTML Title">
7          <meta name="keywords" content="CS2204 - HTML, Title" />
8          <title>Web Development Components</title>
9      </head>
```

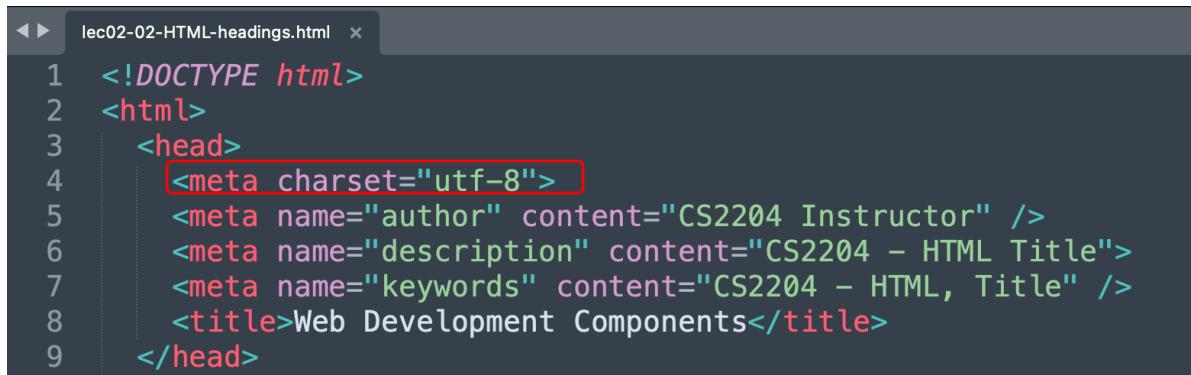
# Multi-Language Handling

**Multi-language support:** make sure the **encoding** is correct and then set the **character set** with `<meta>`

**Encoding** is the way characters (charset) are represented internally in computers.

**Unicode** is a **universal system** for encoding all characters from the languages worldwide. Each character in each language is assigned a unique code.

- **UTF-8** (Unicode Transformation Format): encode characters into 8-bit bytes (like ASCII)
- Other schemes: UTF-16, UTF-32
- These schemes differ in their encoding method, storage efficiency, and compatibility



```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <meta name="author" content="CS2204 Instructor" />
6      <meta name="description" content="CS2204 - HTML Title">
7      <meta name="keywords" content="CS2204 - HTML, Title" />
8      <title>Web Development Components</title>
9    </head>
```

# Other Elements in Head Section

Common Element	Description
base	It specifies an explicit URL used to resolve links and references to external sources, such as images, and style sheet
link	It enables the current document to establish links to external documents, such as style sheets
meta	It conveys information about the document to the server and the client
script	It specifies a script (e.g., JavaScript) for the page that is interpreted by a script engine
style	It specifies a style sheet (e.g., CSS) for the page
title	It contains the title of the document

# Tags to explore

Head section <head>... </head>

- Adding a title of the webpage
- Define meta info

Body section <body>... </body>

- Creating **headings** with **multiple levels**
- Formatting **paragraphs**
- Structure information in **list** and **table**
- Inserting **links** to the webpage
- Displaying **images** on the webpage
- Using **divs** to divide content

Footer section <footer>... </footer>

- The tags can be used are similar to the body section
- Usually displays authorship and copyright info, not required in a webpage

# 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

The screenshot shows a code editor interface with a toolbar at the top featuring icons for home, file, and run, followed by a green "Run" button. The left pane displays the following HTML code:

```
<!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>
```

The right pane shows the rendered output of the HTML code. The headings are displayed as follows:

- This is heading 1**
- This is heading 2**
- This is heading 3**
- This is heading 4**
- This is heading 5**
- This is heading 6**

A tip message is also present in the rendered output:

**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.

# Headings Example

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

Code Example: lec02-02-HTML-headings.html

## 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.

### 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.

### 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.

# Paragraphs

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Pa
5  </head>
6  <body>
7  <!-- Pa
8  <h1>W
9  <p>Of
10 <h2>HTM
11 <p>HyperText Markup Language (HTML) uses a set of codes called tags to
12 describe the structure of a webpage. You should check to make sure that
13 your HTML tags are valid, e.g., matching start tags with end tags.</p>
14
15 <p>The HTML tags are not displayed by the web browser in showing the
16 webpage, but are used to determine how the document should be displayed.
17 A webpage looks fine on one browser does not mean that it will always
18 look the same on another browser so you should test on different browsers
19 to ensure compatibility.</p>
20
21 <h2>CSS</h2>
22 <p>Cascading Style Sheets (CSS) describes how the HTML elements should be
23 displayed by specifying the fonts, colors, layout and placement of these
24 HTML elements. CSS can be embedded on the HTML file or can be defined and
25 referred as an external file.</p>
26
27 <p>Although you could provide the same formatting and layout instructions
28 using HTML, it is better to use CSS as a single CSS rule can be applied to
29 multiple elements so it saves the efforts for defining the appearance of
30 individual HTML elements one by one. The same CSS can also be applied
31 across webpages under the same website</p>
32
33 <h2>JavaScript</h2>
34 <p>JavaScript is a programming language that can provide instructions for
35 a browser to dynamically generate content for a website or enhance the
36 website interactivity. Like CSS, JavaScript can also be embedded on the
37 HTML file or can be defined and referred as an external file.</p>
```

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

## 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



## Web Development

Often you will find 3 components in a web

## 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.

# Lists: overview

A **list** enhances the presentation of information under a **category** with many items

- Which of the following webpage has better **readability**?

## 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

# Lists: unorder vs. ordered

An **unordered list (ul)** displays items with bullets (by default)

<ul> and </ul> define the beginning and end of an unordered list  
<li> and </li> enclose 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

An **ordered list (ol)** displays items with automatic numbering

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

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

HTML

1. Coffee
2. Tea
3. Milk

Browser display

# Lists: an 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 |
19 |   |       <p>My favorite fruits are ranked according to my preference below:</p>
20 |   |           <ol>
21 |   |               <li>watermelon</li>
22 |   |               <li>banana</li>
23 |   |               <li>orange</li>
24 |   |           </ol>
25 |
26 |   |       <!-- Page content ends here -->
27 |   </body>
28 </html>
```

Code Example: lec02-04-HTML-lists.html

## Fruits

The following fruits can be commonly found in supermarkets:

- apple
- orange
- banana
- pear
- grapes
- watermelon

My favorite fruits are ranked according to my preference below:

1. watermelon
2. banana
3. orange

Lists make it easy to **insert**, **delete**, or  
**rearrange** the order of the items.

# List: attributes

Sometimes an attribute can be defined in the start tag with the format `name="value"` to provide additional information

```
<ol type="A">
  <li>apple</li>
  <li>orange</li>
  <li>banana</li>
  <li>pear</li>
  <li>grapes</li>
  <li>watermelon</li>
</ol>
```

- A. apple
- B. orange
- C. banana
- D. pear
- E. grapes
- F. watermelon

The attribute `type` can be set with value "A" for an ordered list so that the number style will be A,B,C,... instead of the default 1,2,3,...

```
<ol start="3">
  <li>orange</li>
  <li>apple</li>
  <li>grape</li>
</ol>
```

- 3. orange
- 4. apple
- 5. grape

The attribute `start` can be set with a numeric value for an ordered list that corresponds to the start number of the first item instead of the default value of 1

# Definition List (Description List)

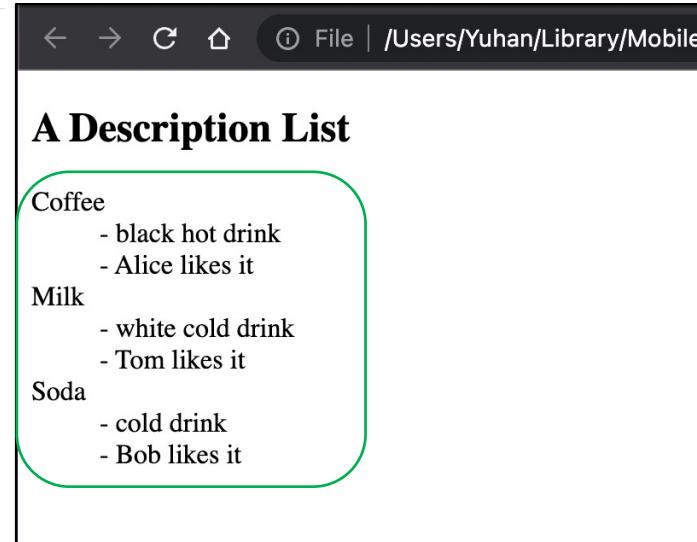
A description list is a list of terms, with a **description** (e.g., title)

- <dl> for **list**, <dt> for **title**, and <dd> for **data**
- There can be multiple <dd> under the same <dt>

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>HTML Description List</title>
  </head>
  <body>
    <!-- Page content begins here -->
    <h2>A Description List</h2>

    <dl>
      <dt>Coffee</dt>
      <dd>- black hot drink</dd>
      <dd>- Alice likes it</dd>
      <dt>Milk</dt>
      <dd>- white cold drink</dd>
      <dd>- Tom likes it</dd>
      <dt>Soda</dt>
      <dd>- cold drink</dd>
      <dd>- Bob likes it</dd>
    </dl>

    <!-- Page content ends here -->
  </body>
</html>
```



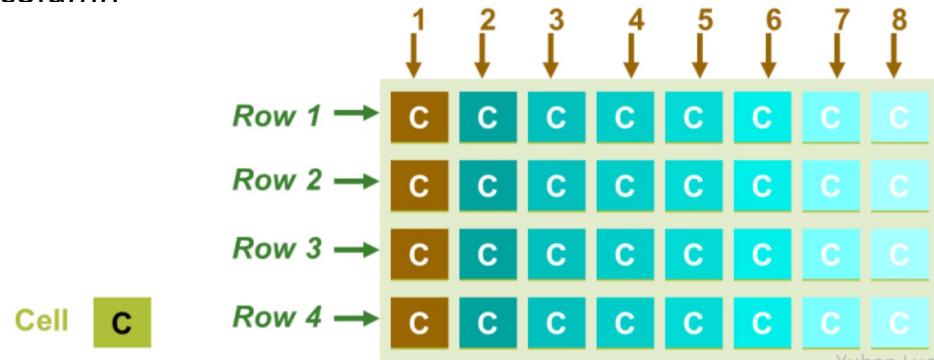
# Table

A table is a matching structure for displaying tabular information (**row** by **column**)

- A web system is often a front end to a database. Most databases are relational databases with records stored as tables

Basic components:

- Table **cells/data** (<td>): the basic structural unit of a table, containing data
- Table **rows** (<tr>): one horizontal row of cells
- Table **head** (<th>): the head of each column



# Table (Example)

```
1 <!DOCTYPE html>
2 <html>
3 <style>
4 table, th, td {
5   border:1px solid black;
6 }
7 </style>
8 <body>
9
10 <h2>A basic HTML table</h2>
11
12 <table>
13   <tr>
14     <th>Company</th>
15     <th>Contact</th>
16     <th>Country</th>
17   </tr>
18
19   <tr>
20     <td>Alfreds Futterkiste</td>
21     <td>Maria Anders</td>
22     <td>Germany</td>
23   </tr>
24
25   <tr>
26     <td>Centro comercial Moctezuma</td>
27     <td>Francisco Chang</td>
28     <td>Mexico</td>
29   </tr>
30
31 </table>
32
33 <p>To understand the example better, we have added borders to the table.</p>
34
35 </body>
</html>
```

To be covered  
later in CSS

Table head

Table row 1

Table row 2

## A basic HTML table

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico

To understand the example better, we have added borders to the table.

# Table Details (1)

## Caption <caption> </caption>

- Provides a **short description** of table's purpose
- **Only** permitted **immediately after** the <table> tag
- **Only one** <caption> element in a table

## Header <thead> </thead>

- Contains heading information of each column
- Can have multiple rows inside, i.e., with more than one
- <th> is usually used in header, instead of <td>

## Body <tbody> </tbody>

- Contains the rows showing table's contents

## Footer <tfoot> </tfoot>

- Contains table's footnote information
- Useful for a summary

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style>
5   table, th, td {
6     border: 1px solid black;
7   }
8 </style>
9 </head>
10 <body>
11
12 <h1>The caption,thead, tbody, and tfoot elements</h1>
13
14 <table>
15   <caption>Monthly savings</caption>
16   <thead>
17     <tr>
18       <th>Month</th>
19       <th>Savings</th>
20     </tr>
21   </thead>
22   <tbody>
23     <tr>
24       <td>January</td>
25       <td>$100</td>
26     </tr>
27     <tr>
28       <td>February</td>
29       <td>$80</td>
30     </tr>
31   </tbody>
32   <tfoot>
33     <tr>
34       <td>Sum</td>
35       <td>$180</td>
36     </tr>
37   </tfoot>
38 </table>
39
40 </body>
41 </html>
```

### The caption,thead, tbody, and tfoot

Monthly savings	
Month	Savings
January	\$100
February	\$80
Sum	\$180

# Table Details (2)

## Advantages of separating table heading, body, and footer

- Standard format is set by default (e.g., `<thead>` sets the heading font bold)
- Easy to define and manage the styles for each section
- Storing data in a more reasonable way (e.g., data cells are within a hierarchy)

What if I want to ...?

1. add another row (e.g., March) under February
2. add another column (e.g., spending) at the right

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <style>
5 table, th, td {
6   border: 1px solid black;
7 }
8 </style>
9 </head>
10 <body>
11
12 <h1>The caption,thead, tbody, and tfoot elements</h1>
13
14 <table>
15   <caption>Monthly savings</caption>
16   <thead>
17     <tr>
18       <th>Month</th>
19       <th>Savings</th>
20     </tr>
21   </thead>
22   <tbody>
23     <tr>
24       <td>January</td>
25       <td>$100</td>
26     </tr>
27     <tr>
28       <td>February</td>
29       <td>$80</td>
30     </tr>
31   </tbody>
32   <tfoot>
33     <tr>
34       <td>Sum</td>
35       <td>$180</td>
36     </tr>
37   </tfoot>
38 </table>
39
40 </body>
41 </html>
```

### The caption,thead, tbody, and tfoot

Monthly savings	
Month	Savings
January	\$100
February	\$80
Sum	\$180

# Table Cells Merging (1)

```
<table style="width:80%">
<tr>
  <td colspan="2">1</td>
  <td>3</td>
</tr>
<tr>
  <td>4</td>
  <td>5</td>
  <td rowspan="2">6</td>
</tr>
<tr>
  <td>7</td>
  <td>8</td>
</tr>
</table>
<br />
<table style="width:80%">
<tr>
  <td colspan="2">1</td>
  <td>2</td>
  <td>3</td>
</tr>
<tr>
  <td>4</td>
  <td>5</td>
  <td rowspan="2">6</td>
</tr>
<tr>
  <td>7</td>
  <td>8</td>
  <td>9</td>
</tr>
</table>
```

Cell merging across *columns/rows* can be achieved with the **attributes** `rowspan` and `colspan` within `<td>`

A basic HTML table

Why is there a missing cell?

# Table Cells Merging (2)

```
<table style="width:80%">
<tr>
  <td colspan="2">1</td>
  <td>2</td>
  <td>3</td>
</tr>
<tr>
  <td>4</td>
  <td>5</td>
  <td rowspan="2">6</td>
</tr>
<tr>
  <td>7</td>
  <td>8</td>
  <td>9</td>
</tr>
</table>
```

Method 1: 3 x 3 table

1		3
4	5	
7	8	6

Code Example: lec02-09-HTML-table-cell-merging.html

How to re-organize the table so the rows/columns are aligned?

1		2	3
4	5		6
7	8		9

```
<table style="width:80%">
<tr>
  <td colspan="2">1</td>
  <td>3</td>
</tr>
<tr>
  <td>4</td>
  <td>5</td>
  <td rowspan="2">6</td>
</tr>
<tr>
  <td>7</td>
  <td>8</td>
</tr>
</table>
<br />
```

# Table Cells Merging (3)

```
<table style="width:80%">
<tr>
  <td colspan="2">1</td>
  <td>2</td>
  <td>3</td>
</tr>
<tr>
  <td>4</td>
  <td>5</td>
  <td rowspan="2">6</td>
</tr>
<tr>
  <td>7</td>
  <td>8</td>
  <td>9</td>
</tr>
</table>
```

Method 2: 3 x 4 table

1		2	3
4	5	6	6+
7	8		9

How to re-organize the table so the rows/columns are aligned?

1		2	3
4	5	6	
7	8		9

```
<table style="width:80%">
<tr>
  <td colspan="2">1</td>
  <td>2</td>
  <td>3</td>
</tr>
<tr>
  <td>4</td>
  <td>5</td>
  <td rowspan="2">6</td>
  <td>6+</td>
</tr>
<tr>
  <td>7</td>
  <td>8</td>
  <td>9</td>
</tr>
</table>
```

- **rowspan/colspan = "x"** takes x cells in the table
- Make sure the number of cells is consistent across rows/columns



# Question time

How to merge cells across **rows** and **columns** at the **same** time and why?

1	2	3
4	5	6
7	8	

```
<table style="width:80%">
  <tr>
    <td colspan="2">1</td>
    <td>2</td>
    <td>3</td>
  </tr>
  <tr>
    <td>4</td>
    <td>5</td>
    <td colspan="2" rowspan="2">6</td>
  </tr>
  <tr>
    <td>7</td>
    <td>8</td>
  </tr>
</table>
```

A

```
<table style="width:80%">
  <tr>
    <td colspan="2">1</td>
    <td>2</td>
    <td>3</td>
  </tr>
  <tr>
    <td>4</td>
    <td>5</td>
    <td rowspan="2" colspan="2">6</td>
  </tr>
  <tr>
    <td>7</td>
    <td>8</td>
    <td colspan="2">6</td>
  </tr>
</table>
```

B

# Recap: Links

A **hyperlink**, or simply called a **link**, can be text or image in a webpage that a user can click to navigate to a different webpage or a file

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML Simple</title>
5   </head>
6   <body>
7     <!-- Page content begins here -->
8     <h1>Simple Link</h1>
9     <p>You can navigate to <a href="http://www.cs.cityu.edu.hk/">CS Department's homepage</a></p>
10    <!-- Page content ends here -->
11  </body>
12 </html>
```

Code Example: lec01-03-HTML-simple-link.html

## Simple Link

You can navigate to [CS Department's homepage](http://www.cs.cityu.edu.hk/)

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



## Question time

What should we do to make the target website open in a new tab instead of overwrite the existing one?

In addition to navigating us to another webpage, what else can `<a>...</a>` do (provide at least two functions)?

# Recap: Link Target

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

## HTML Link with Target

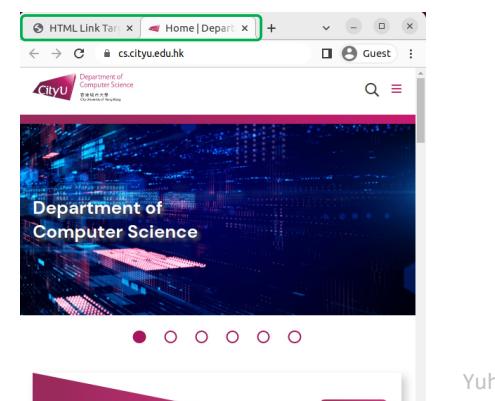
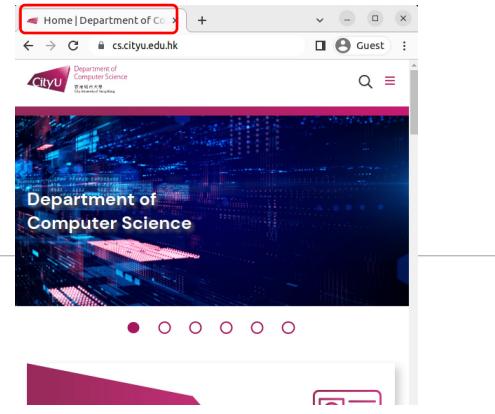
You can open [CS Department's homepage on the same window/tab](http://www.cs.cityu.edu.hk/)  
You can open [CS Department's homepage on new window/tab](http://www.cs.cityu.edu.hk/)

```
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/">CS Department's homepage on the same window/tab</a></p>
10    <p>You can open <a href="http://www.cs.cityu.edu.hk/" target="_blank">CS Department's homepage on new window/tab</a></p>
11    <!-- Page content ends here -->
12  </body>
13 </html>
```

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

## HTML Link with Target

You can open [CS Department's homepage on the same window/tab](http://www.cs.cityu.edu.hk/)  
You can open [CS Department's homepage on new window/tab](http://www.cs.cityu.edu.hk/)



Code Example: lec01-04-HTML-link-target.html

# Recap: Link Target

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

## HTML Link with Target

You can open [CS Department's homepage on the same window/tab](http://www.cs.cityu.edu.hk/)

You can open [CS Department's homepage on new window/tab](http://www.cs.cityu.edu.hk/)

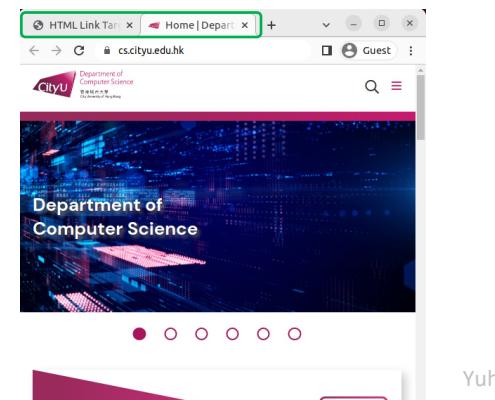
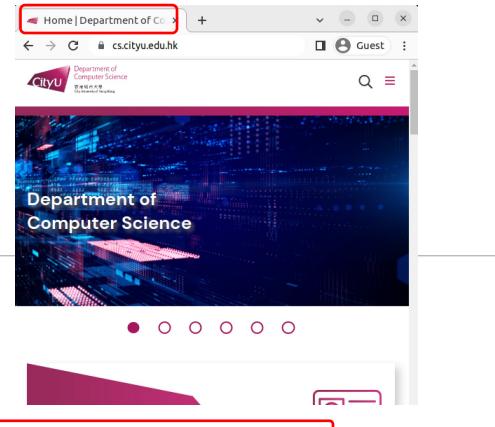
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>HTML Link Target</title>
5   </head>
6   <body>
7     <!-- Page content begins here -->
8     <h1>HTML Link with Target</h1>
9     <p>You can open <a href="http://www.cs.cityu.edu.hk/">CS Department's homepage on the same window/tab</a></p>
10  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

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

## HTML Link with Target

You can open [CS Department's homepage on the same window/tab](http://www.cs.cityu.edu.hk/)

You can open [CS Department's homepage on new window/tab](http://www.cs.cityu.edu.hk/)



# Recap: Link Actions

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

- Sending an email: <a href="mailto:abcdef@cityu.edu.hk">
- Download a file: <a href="2-HTML.pdf">

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>HTML Link Actions</title>
5    </head>
6    <body>
7      <!-- Page content begins here --&gt;
8
9      &lt;h1&gt;HTML Link Actions&lt;/h1&gt;
10
11     &lt;p&gt;You can email to &lt;a href="mailto:abc@cityu.edu.hk"&gt;
12       the course instructor&lt;/a&gt;&lt;/p&gt;
13
14     &lt;p&gt;You can download &lt;a href="2-HTML.pdf"&gt;the lecture
15       slides&lt;/a&gt;&lt;/p&gt;
16
17   &lt;!-- Page content ends here --&gt;
18 &lt;/body&gt;
19 &lt;/html&gt;</pre>

Code Example: lec01-05-HTML-link-action.html


```

## HTML Link Actions

You can go to [Canvas](#) for accessing the CS2204 course

You can [email the course instructor](#)

You can [download the lecture notes](#)

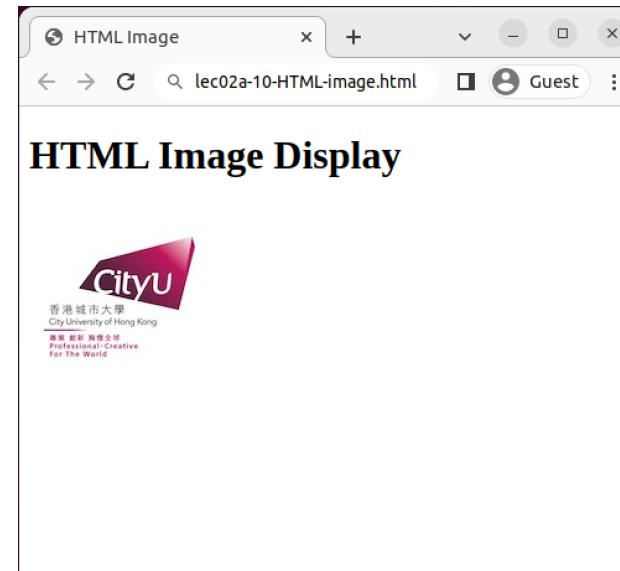
Run an email client (if installed) to  
compose email to a specific  
recipient

Download a file

# 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    
11
12    <!-- Page content ends here -->
13  </body>
14 </html>
```



Code Example: lec02-10-HTML-image.html

# Images: alternate text

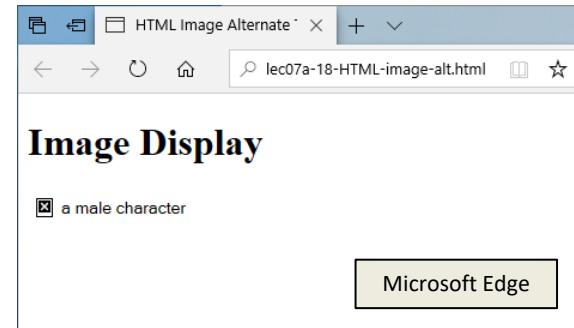
The `alt` attribute is used to specify the **alternate text** for the image, and is strongly recommended for accessibility purposes.

```
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    
11
12    <!-- Page content ends here -->
13  </body>
14</html>
15
```

Code Example: lec02-11-HTML-image-alt.html

The alternate text has the following functions:

- 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**
- 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



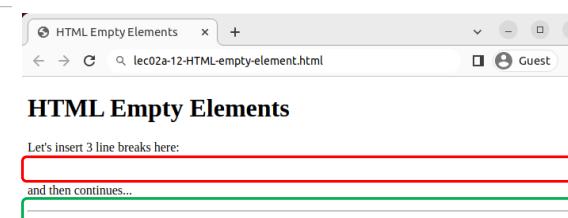
# Empty Element

img tag does **not** have a corresponding end tag

- <img> (or <img/>)
- 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> (or <br/>): add a **line break**
  - <hr> (or <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    <hr />
13
14  <!-- Page content ends here -->
15  </body>
16 </html>
```

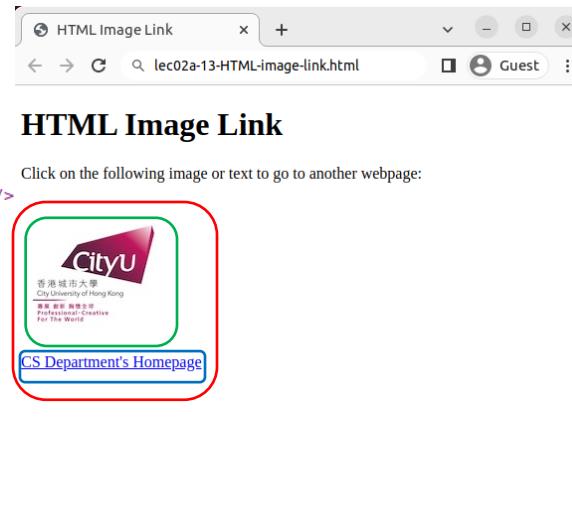
Code Example: lec02a-12-HTML-empty-element.html



# Image Link

Links can be added to **image** 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      <a href="http://www.cs.cityu.edu.hk/">
12        <br />
13        
14        <br />
15        CS Department's Homepage
16        <br />
17      </a>
18
19    </p>
20
21    <!-- Page content ends here -->
22  </body>
23</html>
```



Note that to add a hyperlink to an image, the image element should be placed **within** the `<a></a>` tag.

# Divide the webpage

The `<div></div>` tag is often used to group different parts of a webpage (e.g, heading, paragraph, image) together to form a section, such that a style can be applied

```
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 04a - Div Tag">
7     <meta name="keywords" content="CS1102 Lecture, Div Tag" />
8     <title>Div Tag Style</title>
9     <style>
10       div {color: red;}
11       ul {background-color: yellow;}
12       li {font-style: italic;}
13     </style>
14   </head>
15   <body>
16     <!-- Page content begins here -->
17     <div>
18       <h1>Fruits</h1>
19       <div>
20         <p>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       </div>
30       <div>
31         <p>My favorite fruits are ranked according to my preference below:</p>
32         <ol>
33           <li>watermelon</li>
34           <li>banana</li>
35           <li>orange</li>
36         </ol>
37       </div>
38     </div>
39     <!-- Page content ends here -->
40   </body>
41 </html>
```

To be covered later  
in CSS

## Fruits

The following fruits can be commonly found in supermarkets:

- *apple*
- *orange*
- *banana*
- *pear*
- *grapes*
- *watermelon*

My favorite fruits are ranked according to my preference below:

1. *watermelon*
2. *banana*
3. *orange*

# Comments in HTML

## What is comment?

Text note that is **ignored by the compiler or interpreter** (thus does not influence the display or code performance) and is **intended only for human readers**.

**Syntax:** <!-- -->

<!-- comments in the same line -->

**or**

<!--

comments may occupy multiple lines

comments may occupy multiple lines

-->

In a code editor, you can also select all the code you want to comment, and press **command/ctrl + ?**

# Escape characters

Characters difficult to input or represent directly

- **Syntax:** &codes;
- When to use
  - Reserved by HTML, e.g., the left angle bracket <, or
  - Special character, e.g., ©

Meaning	Escape Char.	Meaning	Escape Char.
<	&lt;	×	&times;
>	&gt;	÷	&divide;
“	&ldquo;	Space	&ampnbsp
”	&rdquo;	©	&copy;

# Escape characters (examples)

```
lec02-15-HTML-escape-character.html < />
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h1>HTML Entity Example</h1>
6
7  <p>The less-than sign: &lt;;</p>
8  <p>The greater-than sign: &gt;;</p>
9
10 <p>The registered trade mark sign: &reg;;</p>
11 <p>The copyright sign: &copy;;</p>
12 <p>The ampersand (and) sign: &amp;;</p>
13
14 <p>The double quote sign: &quot;;</p>
15 <p>The single quote sign: &apos;;</p>
16
17 <p>The cent sign: &cent;;</p>
18 <p>The pound sign: &pound;;</p>
19 <p>The yen sign: &yen;;</p>
20 <p>The euro sign: &euro;;</p>
21
22 </body>
23 </html>
```

## HTML Entity Example

The less-than sign: <

The greater-than sign: >

The registered trade mark sign: ®

The copyright sign: ©

The ampersand (and) sign: &

The double quote sign: "

The single quote sign: '

The cent sign: ¢

The pound sign: £

The yen sign: ¥

The euro sign: €

# Agenda

Webpage and Source Code Organization

HTML: Commonly Used Tags

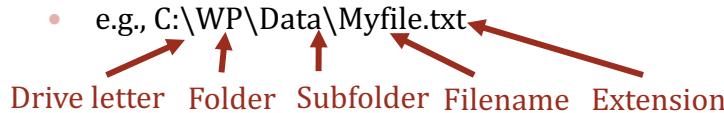
**File Path**

Web Accessibility Design

# File Path

A **path** is the general form of a file or directory name, giving a file's name and its **unique location** in a computer system

- **Full path / absolute path:** points to the same location on one file system **regardless of the working directory**, written **in reference to a root directory**



- **Relative path:** a path **relative to the current working directory**

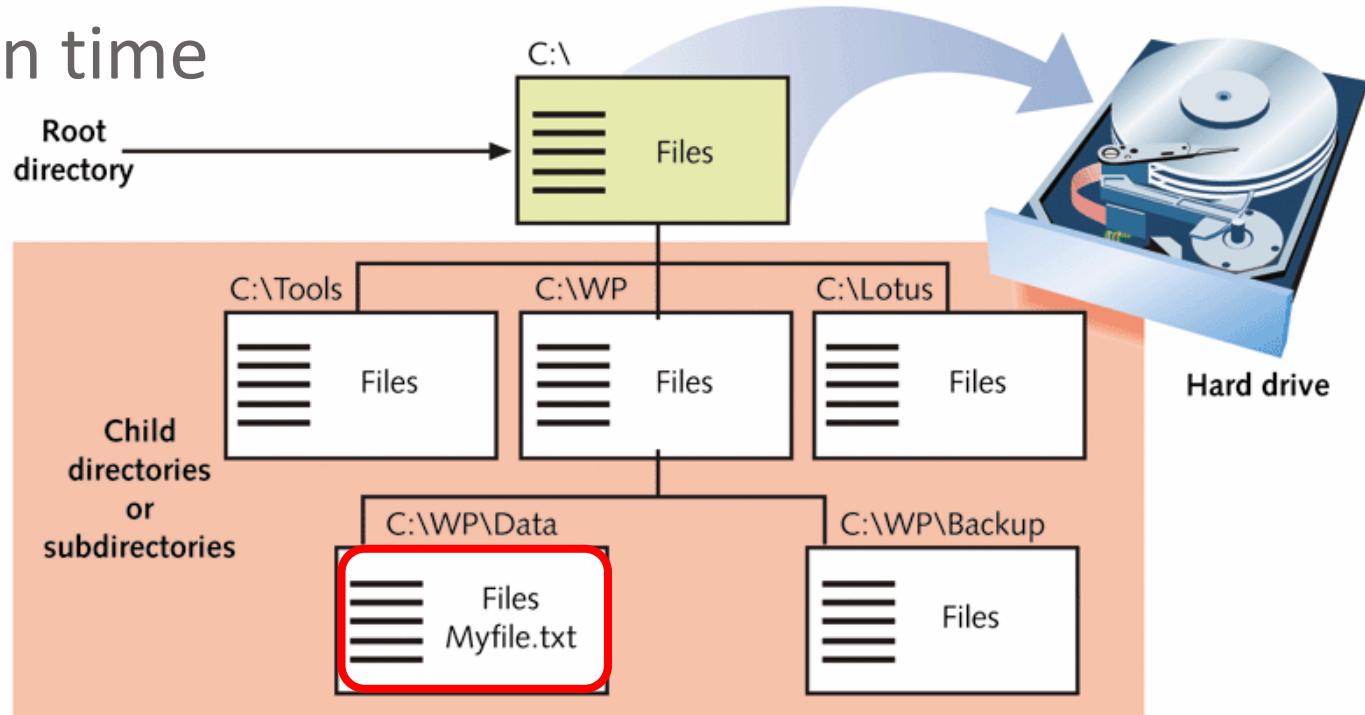
- e.g., ../image/logo.jpg
- A **single dot** (".") refers to the current directory
- **Double dots** ("..") refer to the parent directory
- Files in the same working directory are listed without any preceding slashes



# Question time

1. What is the **absolute** path of [Myfile.txt](#)?

2. Suppose you are now in [C:\WP\Backup](#) (i.e., it is your working directory), what is the **relative** path of [Myfile.txt](#) file?



A hard drive is organized into groups of files stored in directories. The first directory is called the root directory. All directories can have child directories or subdirectories. Under Windows, a directory is called a folder.

# 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

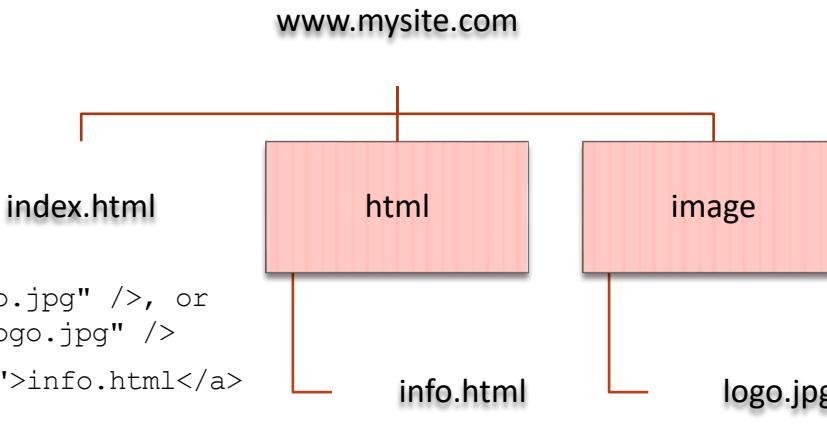
By naming the **home page** index.html under the **website's root folder** (e.g., www), people can access the page without specifying its file name:

www.mysite.com is the same as  
www.mysite.com/index.html

```
, or  
  
<a href="html/info.html">info.html</a>
```

For other pages, specifying the file name is necessary:

www.mysite.com/html/info.html



```
  
<a href=".index.html">index.html</a>
```

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

# Absolute URL vs Relative URL

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

- <a href="<https://www.cs.cityu.edu.hk/>">
- <a href=". /index.html">

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)

A **relative** URL specifies the location of a file relative to the location of the current file

Usually, relative URL is used for linking to files from **internal** websites

# Agenda

Webpage and Source Code Organization

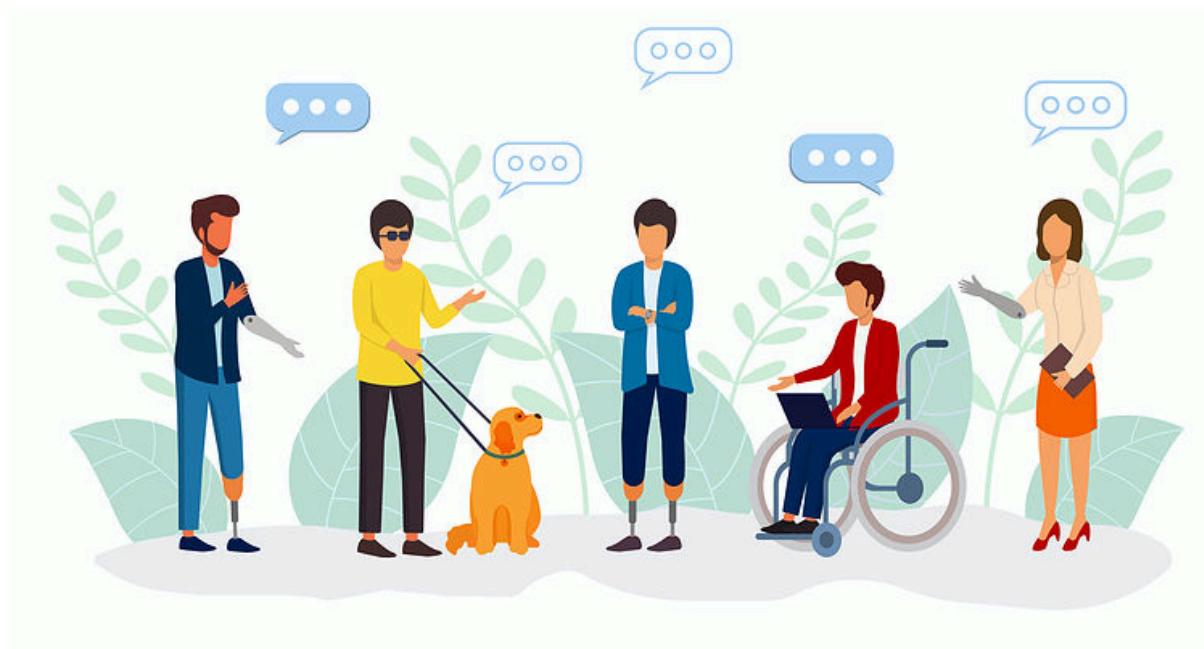
HTML: Commonly Used Tags

File Path

**Web Accessibility Design**

# What is web accessibility?

The inclusive practice to **minimize the barriers** that prevent interaction with, or access to websites by people with *physical disabilities*, *situational disabilities*, and *socio-economic restrictions*



# Why web accessibility is important?

Physical disabilities: visual impairment, motor impairment

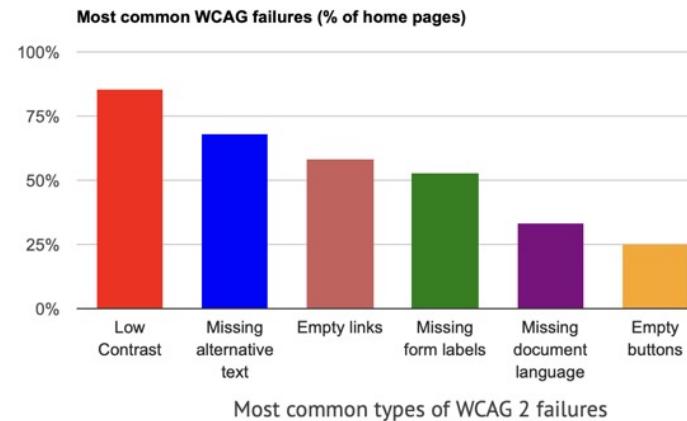
Situational disabilities: multi-tasking (cannot use two hands)

Socio-economic restrictions: limited bandwidth, internet speed

Education: Technology literacy

Language

...



Most common types of WCAG 2 failures

WCAG Failure Type	# of home pages	% of home pages
Low contrast text	852,868	85.3%
Missing alternative text for images	679,964	68%
Empty links	581,408	58.1%
Missing form input labels	528,482	52.8%
Missing document language	329,612	33.1%
Empty buttons	250,367	25%

# Web accessibility: principles

## Perceivable

- Information and user interface components must be presented to users in a way **that they can perceive**

## Operable

- UI components and navigation must be **operable**
- Cannot require interactions that the user cannot perform

## Understandable

- Information and the operation of the user interface must be **understandable**
- Content or operations cannot be beyond their understanding

## Robust

- Content must be robust enough to be **reliably interpreted** by a variety of platforms (e.g., different browsers)
- Content should remain accessible as technologies and user agents evolve

# How to ensure web accessibility?

Include **alt** text for all images

Allow user to adjust **font size**

Choose **colors** carefully, preferably using high-contrast colors

Provide **subtitles** if using video or audio

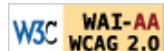
**Multiple** ways to navigate to a webpage

**Descriptive** web link

Accurate page title and link description

...

More guidelines are available in the [W3C](#) for different levels of conformance as it take a lot of effort to fulfill all requirements





## Lecture recap: Question time (1)

Which one of the following can direct users to  
<https://www.cs.cityu.edu.hk> ?

- A. `<a link = "https://www.cs.cityu.edu.hk ">This is a link</a>`
- B. `<a href = "https://www.cs.cityu.edu.hk ">This is a link</a>`



## Lecture recap: Question time (2)

Which one of the following exercises the **best** web accessibility and why?

- A. `<a href = "https://www.cs.cityu.edu.hk"><img src = "logo.jpg"></a>`
- B. `<a href = "https://www.cs.cityu.edu.hk"><img src = "logo.jpg" alt = "a logo"></a>`
- C. `<a href = "https://www.cs.cityu.edu.hk" title = "CS department" ><img src = "logo.jpg" alt = "a logo"></a>`



## Lecture recap: Question time (3)

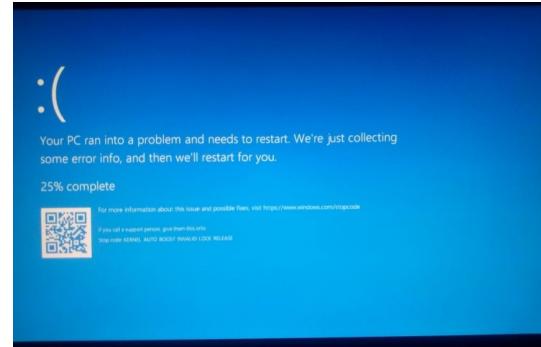
Which of the following **red tag** adds a line break in the middle of the paragraph?

- A. <p>My Bonnie lies <br> over the ocean.</p>
- B. <p>My Bonnie lies <br> over the ocean. </br></p>
- C. <p>My Bonnie lies <br/> over the ocean.</p>

# Is faulty-tolerant good?



A faulty-tolerant system



A faulty-vulnerable system

However, as a programmer, it's NOT an excuse to write crappy code just because it works in most cases. In real-world settings, there are always unexpected user scenarios (e.g., old-version browsers are less faulty-tolerant)

# Web Page Validation

How do we know our Web page is written correctly?

- Browser knows, but modern browsers tend to tolerate and recover errors (if possible). Since all mark-up languages have a well-defined grammar, they can be checked

Validation: verify mark-up pages according to the version

- Most editing tools perform syntax checking, but some subtle errors can only be found by checking against the standard or DTD (document type definition)

W3C provides a validation page with three options:

- by URL
- by uploading files
- by direct input

Convenient to use file upload in your testing, and URL will be blocked by the CS Lab firewall

- [W3C validation page: https://validator.w3.org/](https://validator.w3.org/)

# Lecture summary

The source code of a Webpage usually consists of HTML, CSS, and JavaScript

HyperText Markup Language (HTML) uses a set of codes called **tags** to describe the structure of a webpage (e.g., `<h1>`, `<h2>`, ... `<h5>`, `<p>`, `<ul>`, `<ol>`, `<table>`, `<a>`, `<image>`, `<br>`, `<hr>`, `<div>`).

HTML encode a webpage with tags, which come in pair most of the time, but there are also several empty tags

Links can be added to text, image, or any other web elements; and there are different ways of linking to the same source file (absolute vs. relative)

Web accessibility is important to ensure user experience and we should carefully consider web users who have special accessibility needs

Note that modern browsers are designed to be fault-tolerant but it's not an excuse as a programmer to write crappy code

# Reference

[1] HTML5 Tutorial

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

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

<http://www.whatwg.org>