



香港中文大學
The Chinese University of Hong Kong

CSCI2720 - Building Web Applications

Lecture 2: HTML

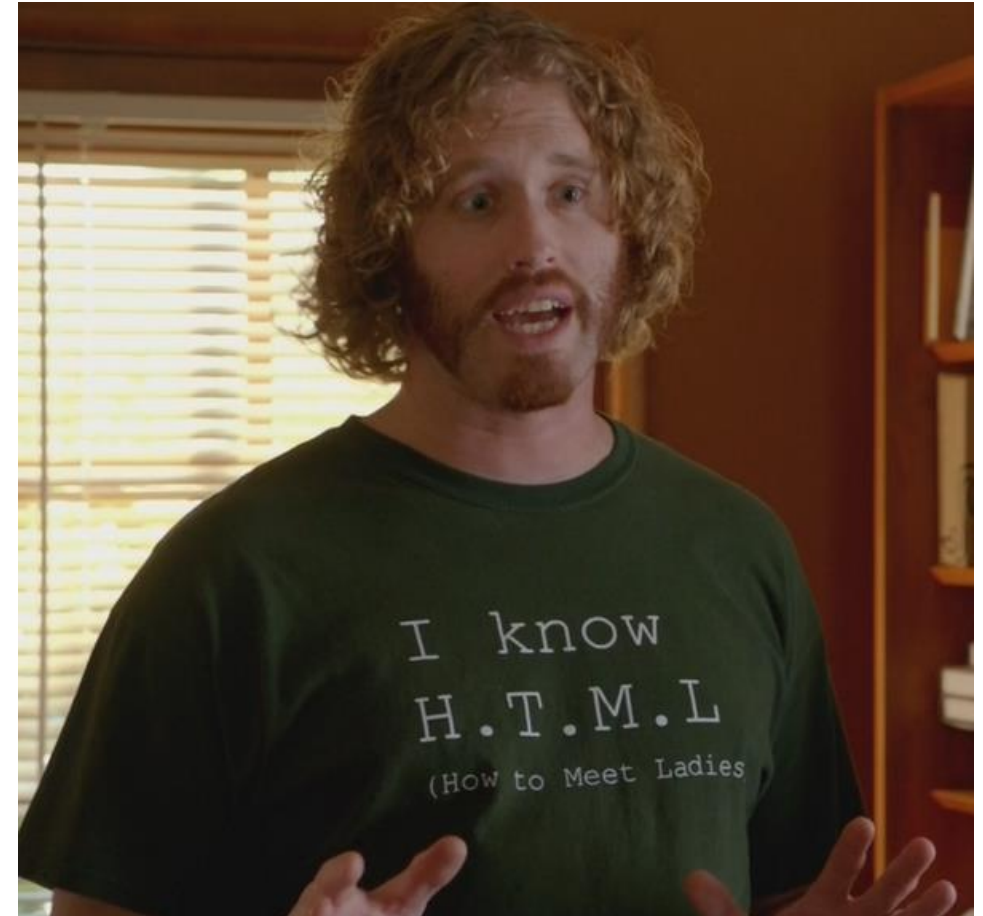
Dr Colin Tsang

Outline

- HTML Basics
- Marking up elements
- Hyperlinks
- Encoding special characters

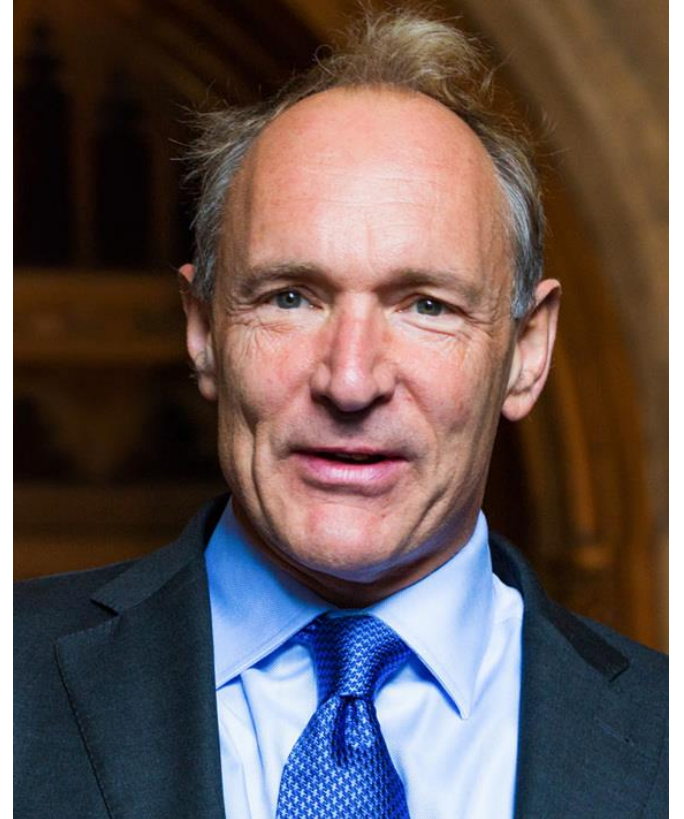
HTML Basics

- HTML – *H*ypert*e*xt *M*arkup *L*anguage
 - It is not a programming language
 - A rendering guideline for software
 - Rendering = Displaying HTML elements on a web browser
- The most fundamental code web browsers read to generate web contents

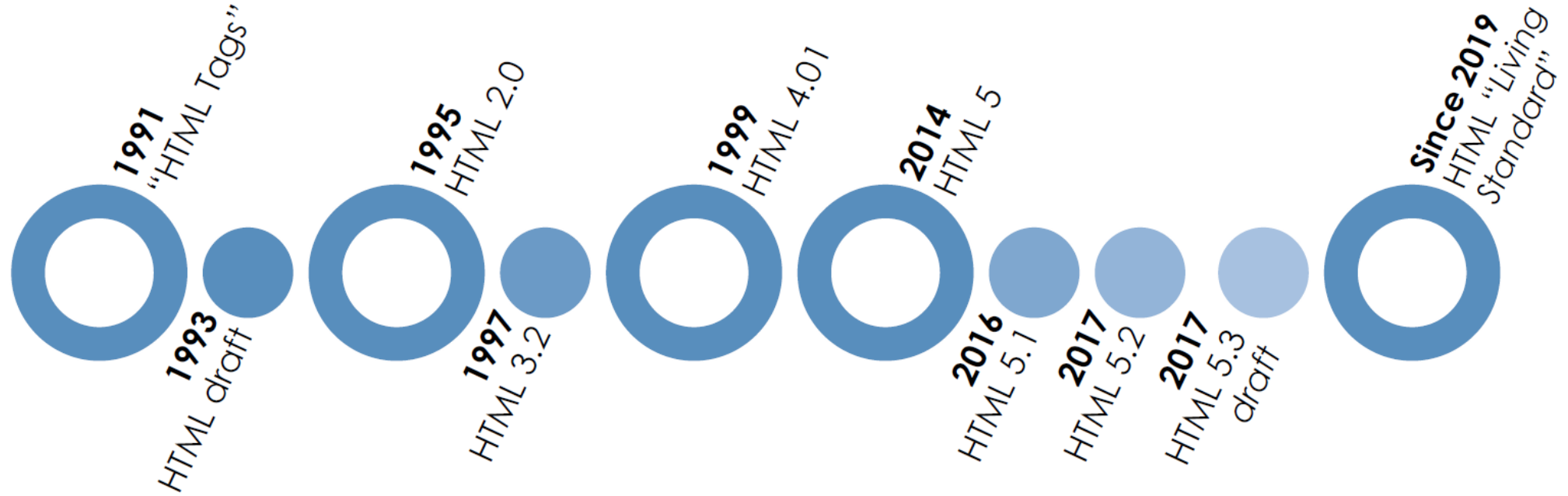


History of HTML

- Sir Tim Berners-Lee is the creator of the World Wide Web.
- In late 1980s, he was working at CERN and developed the first version of HTML, along with HTTP and URL standards.
- The original purpose is share and format scientific information among researchers.



History of HTML



Web Organizations

- W3C (since 1994)
 - *World Wide Web Consortium*
 - Founded by Sir Tim Berners-Lee
- Maintaining standards for WWW
 - Discuss and propose recommendations
 - Once a standard is approved by W3C, it is recommended for implementation and adoption by the web community.
- WHATWG (since 2004)
 - *Web Hypertext Application Technology Working Group*
 - Founded by people in leading web browser vendors, who think W3C's HTML development is slow.
- WHATWG work together with W3C since 2019 (i.e., HTML living standard).

Why HTML?

- HTML helps you to
 - *indicate the roles* of text or media on the page
 - set up *hyperlinks* to allow navigation between pages
- HTML is well supported by web browsers on multiple device platforms, allowing a *unified* experience.
 - One code for all device (e.g., phone, tablet, desktop, etc)
- Although people rarely write HTML directly, you need to learn basic concepts to generate a page using *scripts*.

A simple HTML example

- The `<!DOCTYPE>` declares the document type
 - `Html` represents an HTML5 file
- `<head>` section contains useful data but *not for displaying*.
- `<body>` section contains everything to be shown in the screen.
 - `<h1>` defines the first heading.
 - `<p>` defines a paragraph.
 - `<!-- -->` denotes comments which will be ignored when rendering.

```
<!DOCTYPE html>

<html>

<head>
  <title> CSCI2720 is fun </title>
</head>

<body>

  <h1>Hello World!</h1>
  <p>hello world!</p>
  <!-- Every CS course has use Hello World as the first example -->

</body>

</html>
```

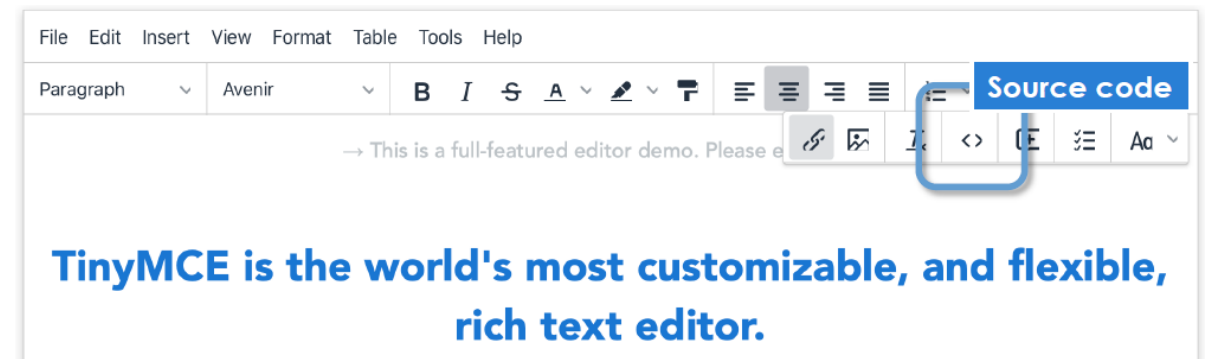


Hello World!

This is a paragraph

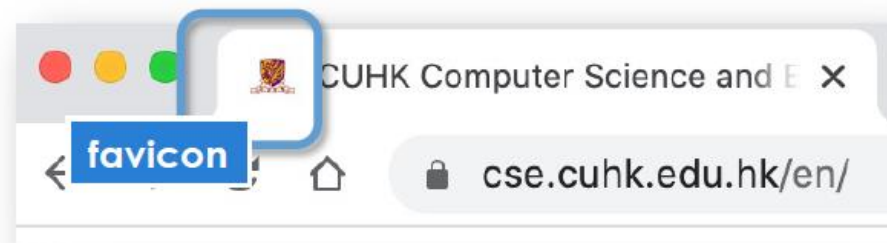
How to check the HTML code?

- Looking at the source code on any web page:
 - Chrome: right click and choose “*View Page Source*”
 - Firefox: right click and choose “*View Page Source*”
 - Safari: right click and choose “*Show Page Source*”
- “Source code” in some text editors



HTML Head

- Some items are *relevant* to a web page, but *not contents to be shown* in the page.
- For example:
 - Page title and “favicon” of a page.
 - Stylesheets, scripts or other external files.
 - Metadata like keywords for search engines.



HTML Body

- All contents in the body will be shown in the page
 - Headings, paragraphs, images, videos, tables, ...
- You can create a structure to present the contents in a semantic manner, using a header, sections, and a footer.
- Sometimes executable scripts are put at the end of the body.

Syntax of HTML elements

- All HTML *elements* are building blocks of the web page.
 - Could be in the *head (not shown)* or in the *body (shown)*
- Elements are created using tags in the code.
 - Tags may or may not have an attribute
- HTML is not case-sensitive, yet recommended for small letter tags
 - For example, the <a> tag is for hyperlink.

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

```
<a>Click me</a>
```

Marking up elements

- Headings <h1>, <h2>, ..., <h6>
- Paragraph <p> and line break

- Formatting
 - Bold , italic <i>, underline <u>
 - Subscript <sub>, superscript <sup>
- Pre-formatted <pre>. It is used to display text exactly as it appears in the code.

Heading 1

Heading 2

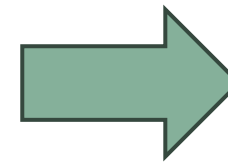
A paragraph with **bold text**, *italic text*, and underlined text with line break followed by _{sub}script and ^{super}script

```
Here are      some
              preformatted
text.
```

Marking up elements

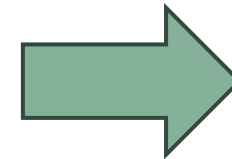
- List:
 - Ordered list
 - Unordered list
 - List items
- Some useful attributes:
 - type="square"
 - type="A"
 - start="5"

```
<ol type="I">  
  <li>First item</li>  
  <li>Second item</li>  
  <li>Third item</li>  
</ol>
```



I. First item
II. Second item
III. Third item

```
<ul type="circle">  
  <li>Red</li>  
  <li>Green</li>  
  <li>Blue</li>  
</ul>
```

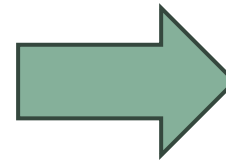


○ Red
○ Green
○ Blue

Marking up elements

- Tables `<table>`
 - Rows: `<tr>`
 - Cells: `<td>`
 - Optional table header: `<th>`

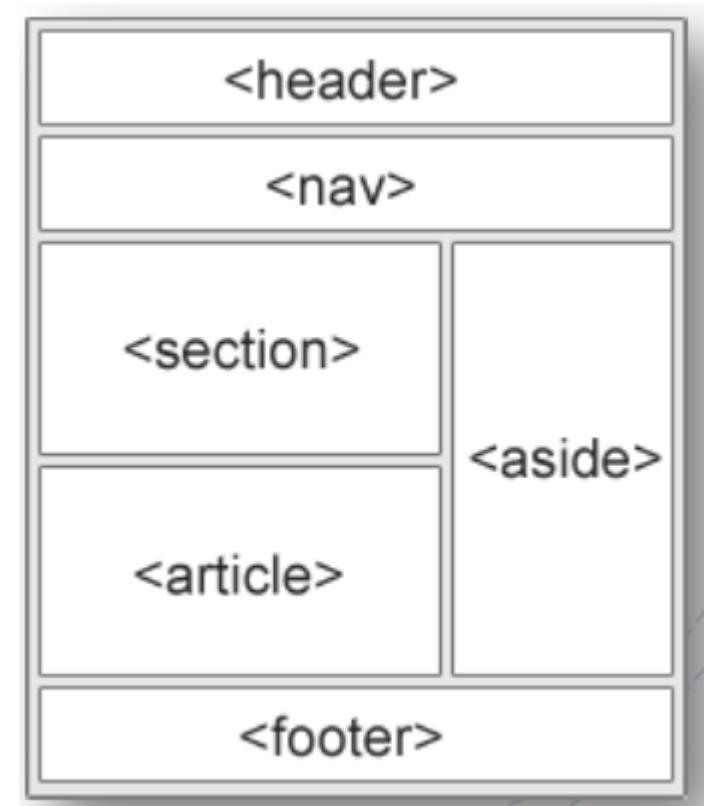
```
<table>
  <tr>
    <th>Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>John</td>
    <td>25</td>
  </tr>
  <tr>
    <td>Jane</td>
    <td>30</td>
  </tr>
</table>
```



Name	Age
John	25
Jane	30

Semantic elements

- In HTML5, it is recommended that the page contents are declared clearly into logical sections
 - e.g., sections `<section>`, navigation bar `<nav>`
- Browsers generally do not define how to render them
 - Easier for search engines and bots to know how data is organized on the page
 - Good anchor points for styling up the page with CSS
 - Expect bold ``, italic ``, etc, which have predefined behaviours in a browser
- Reading assignment:
 - https://www.w3schools.com/html/html5_semantic_elements.asp



Hyperlinks

Go to other webpages

- The hyperlink allows a “non-linear” manner of hypertext and hypermedia
 - *Inline links*: pointing to *another file*, which could be in the same server or another server.
 - *anchors*: point to *another part/section* in the *same file*.
- Usually displayed in different colours than normal text, depending on whether the link has been visited or not.

Hyperlinks

- The `<a>` element
 - `href` attribute = what to point to
 - `target` attribute = where to open, e.g., “_blank” opens the link in a new tab/window, “_self” open the link in the same tab/window

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

- Defining a fragment name using an id could be useful.

```
<a href="#section1">Go to Section 1</a>  
  
...  
  
<h2 id="section1">Section 1</h2>
```

← Anchors

Absolute paths VS Relative paths

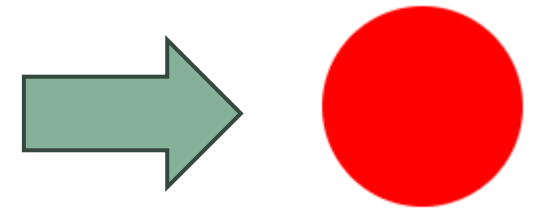
- Absolute paths
 - e.g., <http://www.cuhk.edu.hk/english/index.html>
- Using a complete URL
 - Protocol (http)
 - Domain (www.cuhk.edu.hk)
 - Port (80, not typed by default)
 - Path (/english/)
 - Filename (index.html)
- Relative paths
 - e.g., ../hello.html
- Using the current document as reference
- e.g., we are at the address <http://www.cuhk.edu.hk/english/index.html>
 - `` brings us to the chinese directory under “../” parent directory.

Including images

- Modern browsers support generally lost of image types, usually using ``
 - e.g., ``
- See: https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Image_types
- The special tag `<svg>` can be used for the scalar vector graphics
 - Specifying content of a graphic using elements, e.g., `<circle>`. `<line>`
- People also use `<picture>` for detailed control on responsiveness.
 - We will talk about it in later lectures.

<svg>

```
<svg width="200" height="200">  
  <circle cx="100" cy="100" r="50" fill="red" />  
</svg>
```



- cx and cy referred to the position of the circle center.
- r defines the radius.
- The fill attribute define what color to fill the circle.

Embedding audio and video

- The relatively newer elements of `<audio>` and `<video>` adds native multimedia support into browsers
- Since there are too many multimedia file formats out there, you can use multiple `<source>` tags to point to multiple files.

```
<video controls>
  <source src="video.mp4" type="video/mp4">
  <source src="video.webm" type="video/webm">
  <source src="video.ogv" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

Encoding special characters

- Browsers doesn't like to see `<` or `>` in the text as it is the same as the tags,
 - e.g., `<p> Hi, here is an inequality x < y </p>`
 - Modern browsers usually can guess it correctly.

- `<` is `<`

- `>` is `>`

```
<p> hi, here is an inequality x &lt; y </p>
```



hi, here is an inequality x < y

- A list of HTML entities can be found in: <https://dev.w3.org/html5/html-author/charref>

Handling space

- By default, consecutive whitespace (space, new line, tab, etc.) in an HTML file will be regarded as one, e.g.,
 - `<p> Hello World </p>` will be rendered as:
→ Hello World
- ` ` is used for crating fixed spaces within text that won't collapse or cause line breaks, even if the browser window is resized.

```
<p>This is an example&nbsp;of using the non-breaking space character.</p>
```


More about HTML.....

- We have only gone through some very basic features in HTML language
 - These are cornerstones which you will see again and again in this course
- We will learn more throughout this course.
- Check out a HTML cheat sheet: <https://www.wpkube.com/html5-cheat-sheet/>

Some self-learning materials

- HTML living standard:
 - <https://html.spec.whatwg.org>
- w3schools.com HTML5 tutorial
 - <https://www.w3schools.com/html/>
- MDN HTML guides and tutorials
 - <https://developer.mozilla.org/en-US/docs/Learn/HTML>