

# ETC5523: Communicating with Data

## Introduction to web technologies

Lecturer: *Emi Tanaka*

Department of Econometrics and Business Statistics

✉ [emi.tanaka@monash.edu](mailto:emi.tanaka@monash.edu)

📅 Week 2

🌐 [cwd.numbat.space](http://cwd.numbat.space)




## Aim

- Understand the make up of web documents, i.e. HTML, CSS and JS
- Write basic HTML and CSS with valid syntax
- Use CSS selectors to apply new styles
- Integrate CSS and JS to R markdown documents with HTML output

## Why

- Basic web development skills are necessary for customising HTML documents
- This hard skill will be necessary later for when you are communicating with web documents and web apps

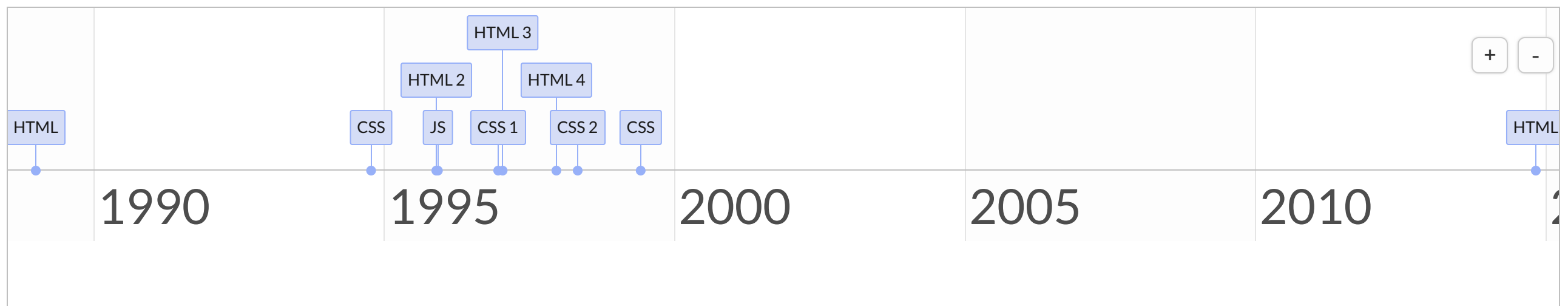
# World Wide Web (WWW)

- WWW (or the **Web**) is the information system where documents (web pages) are identified by Uniform Resource Locators (**URLs**)
- A web page consists of:
  -  **HTML** provides the basic structure of the web page
  -  **CSS** controls the look of the web page (optional)
  -  **JS** is a programming language that can modify the behaviour of elements of the web page (optional)



# Web Documents are Handy

- HTML documents are really handy for including interactive elements and supported in almost all computer devices.
- Naturally, this ties in well with [interactive data visualisation](#).
- Below is an interactive timeline visualisation of historical developments of HTML/CSS/JS:



So what exactly is  
**HTML**, **CSS**, and **JS**?

# Hyper Text Markup Language

- HTML files have the extension `.html`.
- HTML files are often rendered using a web browser via an URL.
- HTML files are just text files that follows a special syntax that alerts web browsers how to render it.

# Simple HTML example

 [simple-example.html](#)



```
1 <html>
2 <body>
3 <h1>ETC5523: Communicating with Data</h1>
4 <h2>Lecturer</h2>
5 <ul>
6 <li>Emi Tanaka (Chief Examiner)</li>
7 </ul>
8 <h2>Tutor</h2>
9 <ul>
10 <li>Weihao (Patrick) Li</li>
11 </ul>
12 </body>
13 </html>
```



## ETC5523: Communicating with Data

### Lecturer

- Emi Tanaka (Chief Examiner)

### Tutor

- Weihao (Patrick) Li

# HTML Structure

```
1 <html>
2 <!--This is a comment and ignored by web client.-->
3 <head>
4   <!--This section contains web page metadata.-->
5   <title>Communicating with Data</title>
6   <meta name="author" content="Emi Tanaka">
7   <link rel="stylesheet" href="css/styles.css">
8 </head>
9
10 <body>
11 <!--This section contains what you want to display on your web page.-->
12 <h1>I'm a first level header</h1>
13 <p>This is a paragraph.</p>
14 </body>
15 </html>
```



# HTML Syntax



```
1 <span style="color:blue;">Author content</span>
```

## Author content

The breakdown of this HTML syntax:

start tag: `<span style="color:blue;">Author content</span>`

end tag: `<span style="color:blue;">Author content</span>`

content: `<span style="color:blue;">Author content</span>`

element name: `<span style="color:blue;">Author content</span>`

attribute: `<span style="color:blue;">Author content</span>`

attribute name: `<span style="color:blue;">Author content</span>`

attribute value: `<span style="color:blue;">Author content</span>`

# Not all HTML tags have an end tag



```
1 
```



# Some HTML elements

block element:	<code>&lt;div&gt;content&lt;/div&gt;</code>
inline element:	<code>&lt;span&gt;content&lt;/span&gt;</code>
paragraph:	<code>&lt;p&gt;content&lt;/p&gt;</code>
header level 1:	<code>&lt;h1&gt;content&lt;/h1&gt;</code>
header level 2:	<code>&lt;h2&gt;content&lt;/h2&gt;</code> (note: only up to 6 levels)
italic:	<code>&lt;i&gt;content&lt;/i&gt;</code>
emphasised text:	<code>&lt;em&gt;content&lt;/em&gt;</code>
bold:	<code>&lt;b&gt;content&lt;/b&gt;</code>
strong importance:	<code>&lt;strong&gt;content&lt;/strong&gt;</code>
link:	<code>&lt;a href="https://cwg.numbat.space/"&gt;content&lt;/a&gt;</code>
insert new line:	<code>&lt;br&gt;</code>
unordered list:	<code>&lt;ul&gt;</code> <code>&lt;li&gt;item 1&lt;/li&gt;</code> <code>&lt;li&gt;item 2&lt;/li&gt;</code> <code>&lt;/ul&gt;</code>

How these are rendered to the browser depends on the **browser default style values, style attribute or CSS...**

# Cascading Style Sheet (CSS)

- CSS files have the extension `.css` and styles also XHTML, plain XML, SVG and XUL.
- There are 3 ways to style elements in HTML:
  - inline by using the `style` attribute inside HTML start tag:

```
1 <h1 style="color:blue;">Blue Header</h1>
```

- externally by using the `<link>` element:

```
1 <link rel="stylesheet" href="styles.css">
```

- internally by defining within `<style>` element:

```
1 <style type="text/css"> h1 { color: blue; } </style>
```

# CSS Syntax



```
1 <style type="text/css">
2 h1 { color: blue; }
3 </style>
4
5 <h1>This is a header</h1>
```



# This is a header

The breakdown of the CSS syntax:

selector:    **h1** { color: blue; }

property:    h1 { **color: blue;** }

property name: h1 { **color:** blue; }

property value: h1 { color: **blue;** }

# You may have multiple properties for a single CSS selector

```
1 h1 {  
2   color: blue;  
3   font-family: monospace;  
4   background: yellow;  
5 }
```

**This is a header**

# Some CSS properties

`<div> Sample text </div>`

background color:	<code>div { background-color: yellow; }</code>	Sample text
text color:	<code>div { color: purple; }</code>	Sample text
border:	<code>div { border: 1px dashed brown; }</code>	Sample text
left border only:	<code>div { border-left: 10px solid pink; }</code>	Sample text
text size:	<code>div { font-size: 10pt; }</code>	Sample text
padding:	<code>div { background-color: yellow; padding: 10px; }</code>	Sample text
margin:	<code>div { background-color: yellow; margin: 10px; }</code>	Sample text
horizontally center text:	<code>div { background-color: yellow; padding-top: 20px; text-align: center; }</code>	Sample text
font family:	<code>div { font-family: Marker Felt, times; }</code>	<b>Sample text</b>
strike:	<code>div { text-decoration: line-through; }</code>	<del>Sample text</del>
underline:	<code>div { text-decoration: underline; }</code>	<u>Sample text</u>
opacity:	<code>div { opacity: 0.3; }</code>	Sample text

# CSS Selector

<code>*</code>	selects all elements
<code>div</code>	selects all <code>&lt;div&gt;</code> elements
<code>div, p</code>	selects all <code>&lt;div&gt;</code> and <code>&lt;p&gt;</code> elements
<code>div p</code>	selects all <code>&lt;p&gt;</code> within <code>&lt;div&gt;</code>
<code>div &gt; p</code>	selects all <code>&lt;p&gt;</code> one level deep in <code>&lt;div&gt;</code>
<code>div + p</code>	selects all <code>&lt;p&gt;</code> immediately after a <code>&lt;div&gt;</code>
<code>div ~ p</code>	selects all <code>&lt;p&gt;</code> preceded by a <code>&lt;div&gt;</code>
<code>.classname</code>	selects all elements with the attribute <code>class="classname"</code> .
<code>.c1.c2</code>	selects all elements with <i>both</i> <code>c1</code> and <code>c2</code> within its class attribute.
<code>.c1 .c2</code>	selects all elements with class <code>c2</code> that is a descendant of an element with class <code>c1</code> .
<code>#idname</code>	selects all elements with the attribute <code>id="idname"</code> .



# JavaScript (JS)

- JS is a programming language and enable interactive components in HTML documents.
- You can insert JS into a HTML document in two ways:
  - **internally** by defining within `<script>` element:

```
1 <script>
2 document.getElementById("p1").innerHTML = "content";
3 </script>
```

- **externally** by using the `src` attribute to refer to the external file:

```
1 <script src="myjava.js"></script>
```

- You are *not* expected to be able to do any JS in this course.

# HTML outputs with Rmd documents

```
1 ---  
2 output: html_document  
3 ---
```

```
1 ---  
2 output: bookdown::html_document2  
3 ---
```

# Inserting CSS in Rmd documents Part 1

- There is a **CSS** engine:

```
1  ```{css}
2  h1 { color: red; }
3  ```
```

- This inserts the following output into the document:

```
1  <style type="text/css">
2  h1 { color: red; }
3  </style>
```

- If the output is a HTML document then the defined styles will apply to the output document.

# Inserting CSS in Rmd documents Part 2

- If you have an external file, say `styles.css`, that you define the styles, then most HTML outputs will support this with YAML argument `css`

```
1 ---
2 output:
3   html_document:
4     css: ["styles.css"]
5 ---
```

or say

```
1 ---
2 output:
3   bookdown::html_document2:
4     css: ["styles.css", "custom.css"]
5 ---
```

# Inserting JS into Rmd documents Part 1

- There is a `js` engine:

```
1 ```{js}
2 document.getElementById("p1").innerHTML = "content";
3 ```
```

which inserts:

```
1 <script type="text/javascript">
2 document.getElementById("p1").innerHTML = "content";
3 </script>
```

- If you have an external file, say `myjava.js`, then you can directly insert this in the body of the Rmd file as:

```
1 <script src="myjava.js"></script>
```

# Inserting JS into Rmd documents Part 2

If you need to insert at a specific location within the document then you can use `includes`:

```
1 ---
2 output:
3   html_document:
4     includes:
5       in_header: ["header.html"]
6       before_body: ["before_body.html"]
7       after_body: ["after_body.html"]
8 ---
```

where `header.html`, `before_body.html`, `after_body.html` includes the JS code, e.g.

```
1 <script src="myjava.js"></script>
2 <script type="text/javascript">
3   document.getElementById("p1").innerHTML = "content";
4 </script>
```

# Community Web Enhancements

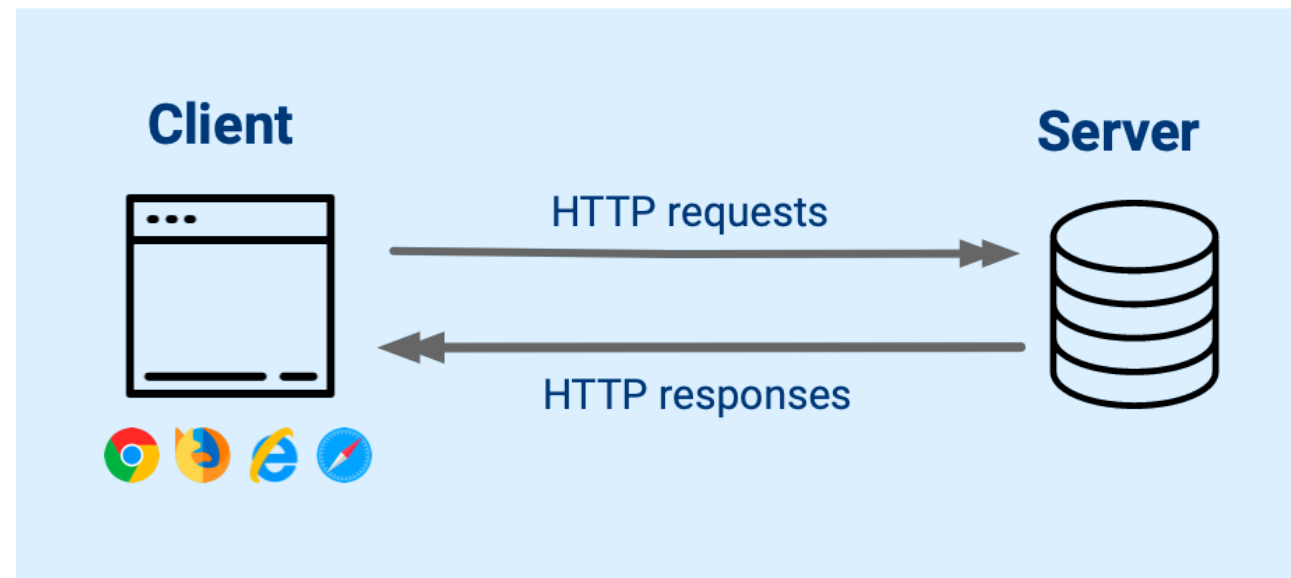
- [Bootstrap](#) is a free open-source CSS and JS that is widely adopted.<sup>1</sup>
- [jQuery](#) is a widely used JS library for object selection and manipulation.<sup>1</sup>
- [MathJax](#) is a JS library for displaying mathematics.<sup>1</sup>
- [Font Awesome](#) inserts icons to web using their CSS.<sup>2</sup>
- [D3.js](#) is one of the most popular JS library for interactive data visualisation.

1. Included in the default [rmarkdown::html\\_document](#).

2. Sometimes included in certain Rmd HTML documents.

# Communication in the Web

- Hypertext Transfer Protocol (**HTTP**) functions as the communicator in the Web
- HTTPS is the secure version of HTTP where communications are encrypted









## Warning

Different clients may work differently! E.g. Internet Explorer and Chrome may render the same web page differently.





# Web Standard

- There are 3 major groups that govern the standard for the Web:
  - World Wide Web Consortium (**W3C**) formed in 1994 maintains the **CSS specifications** 
  - Web Hypertext Application Technology Working Group (**WHATWG**) formed in 2004 and is the publisher of the **HTML and DOM standards** 
  - **TC39 technical committee** of **Ecma International**, renamed from European Computer Manufacturers Association (ECMA) in 1994, maintains the standards for JS 
- These groups consist of Mozilla,  Apple,  Google,  Microsoft and other invited members.



# MDN Web Docs

LIVE DEMO

<https://developer.mozilla.org/>

- Documentation of web standards maintained by the community.
- Includes status of use:



#### Obsolete

This feature is obsolete. Although it may still work in some browsers, its use is discouraged since it could be removed at any time. Try to avoid using it.

- As well as its compatibility with web clients:

#### Browser compatibility

[Update compatibility data on GitHub](#)

	Desktop						Mobile					
	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari	Android webview	Chrome for Android	Firefox for Android	Opera for Android	Safari on iOS	Samsung Internet
<code>basefont</code>	No	12 — 79	No	Yes	No	No	No	No	No	No	No	No

[What are we missing?](#)

Full support
 No support

Deprecated. Not for use in new websites.

# Writing HTML

You can play at <https://htmltidy.net/>



**HTML Tidy .Net**

File Edit Insert View Format Table Tools

Formats B I U

This is a **sample text** you can play with!

**HTML Tidy .Net** instant HTML beautifier offers you many code editing options:

he following Tidy options are available:

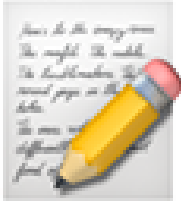
- Inline styles
- Classes, ID's
- Empty tags
- Tags with one space
- Successive spaces
- Remove comments
- Tag attributes
- Make plain text

To adjust the Tidy options, click the  button 😊

Source: 955

```

1 <!-- ##### This is a comment ##### -->
2 <h3>This is a <span style="background-color: #89bbaa; color:
  #ffffff; padding: 0 3px;">sample text</span> you can play
  with!</h3>
3 <p><strong>&nbsp;
  &nbsp;instant HTML beautifier offers you many code editing
  options:</strong></p>
4 <p>he following Tidy options are available:</p>
5 <ul style="background-color: #89bbaa; color: #ffffff; font-
  weight: bold; padding: 5px 40px;">
6 <li>Inline styles</li>
7 <li>Classes, ID's</li>
8 <li>Empty tags</li>
9 <li>Tags with one space</li>
10 <li>Successive spaces</li>
11 <li>Remove comments</li>
12 <li>Tag attributes</li>
13 <li>Make plain text</li>
14 </ul>
15 <p>To adjust the Tidy options, click the  button </p>
  
```



# Interactive Cheatsheets

 HTML Cheatsheet <https://htmlcheatsheet.com>

 CSS Cheatsheet <https://htmlcheatsheet.com/css/>

 JS Cheatsheet <https://htmlcheatsheet.com/js/>

# Inspect Element LIVE DEMO

Below GIF shows interactive use of Inspect Element<sup>1</sup> available from the menu in most web browsers when you right click on the web page<sup>2</sup>:



~

1. Or called Inspect.

2. Note: this is for inspection and doesn't actual make changes to the files.

# Week 2 Lesson

## Summary

- We went through the basics of HTML, CSS and JS
- You should be more comfortable with writing HTML and CSS
- You learnt how to include CSS and JS into HTML documents created from R Markdown
- You know the tools to interactively learn HTML and CSS

## Resources

- [Introduction to Data Technologies Ch 2-4 by Paul Murrell](#)
- [Interactive HTML/CSS/JS cheatsheets](#)
- [HTML Tidy .Net](#)