

A Brief Introduction to Web Technologies

Yong Wang

College of Computing and Data Science
Nanyang Technological University

Outline

- HTML
- CSS
- SVG
- JavaScript

HTML

HTML

(Hypertext Markup Language)

- Used to mark up the content of a web page by adding a structure to the elements in the web page
- Elements
 - Paragraph, division, ordered and unordered list, headings, links, body, head, title, etc., and the root html
 - Elements are created by tags, for example,
 - `<p>` defines the beginning of a paragraph
 - `</p>` closes the paragraph

A Simple HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Page Title</h1>
    <p>This is a really interesting paragraph.</p>
  </body>
</html>
```

A List of Common Elements

`<!DOCTYPE html>`

The standard document type declaration.
Must be the first thing in the document.

`html`

Surrounds all HTML content in a document.

`head`

The document `head` contains all metadata about the document, such as its `title` and any references to external stylesheets and scripts.

`title`

The title of the document. Browsers typically display this at the top of the browser window and use this title when bookmarking a page.

`body`

Everything not in the `head` should go in the `body`. This is the primary visible content of the page.

`h1`, `h2`, `h3`, `h4`

These let you specify headings of different levels. `h1` is a top-level heading, `h2` is

`p`

A paragraph!

`ul`, `ol`, `li`

Unordered lists are specified with `ul`, most often used for bulleted lists.

Ordered lists (`ol`) are often numbered.

Both `ul` and `ol` should include `li` elements to specify list items.

`em`

Indicates emphasis. Typically rendered in *italics*.

`strong`

Indicates additional emphasis. Typically rendered in **boldface**.

`a`

A link. Typically rendered as underlined, blue text, unless otherwise specified.

`span`

An arbitrary `span` of text, typically within a larger containing element like `p`.

`div`

An arbitrary *division* within the document. Used for grouping and containing related elements.

Comments, Classes, and IDs

- You can add comments to your html document with `<!-- this is a comment -->`
- Elements can be identified by their classes or IDs (important for CSS and Javascript)
- Classes:

```
<p class="uplifting">Brilliant paragraph</p>
<p class="uplifting">Insightful paragraph</p>
<p class="uplifting awesome">Awe-inspiring paragraph</p>
```

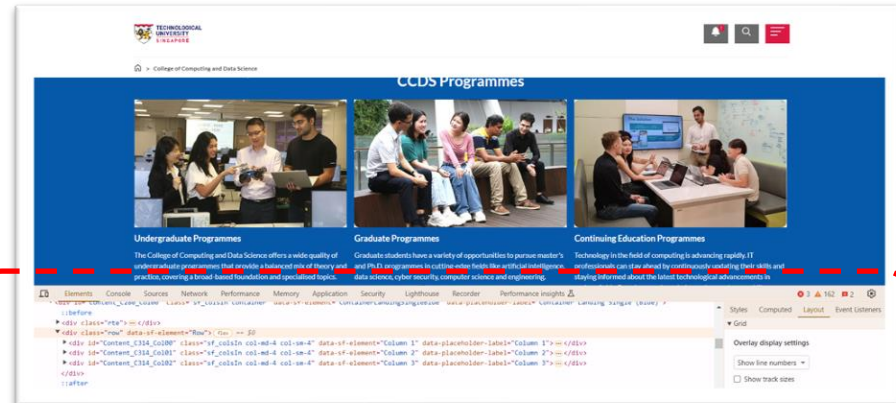
- IDs: (only used for one element and only once in a page)

```
<div id="content">
  <div id="visualization"></div>
  <div id="button"></div>
</div>
```

Document Object Model (DOM)

- Describes the hierarchical structure of HTML
 - The parent, child, sibling, ancestor, descendant relationships among the HTML elements, which is also called *DOM Tree*
- Open the development tool of your browser to check the DOM of any webpage

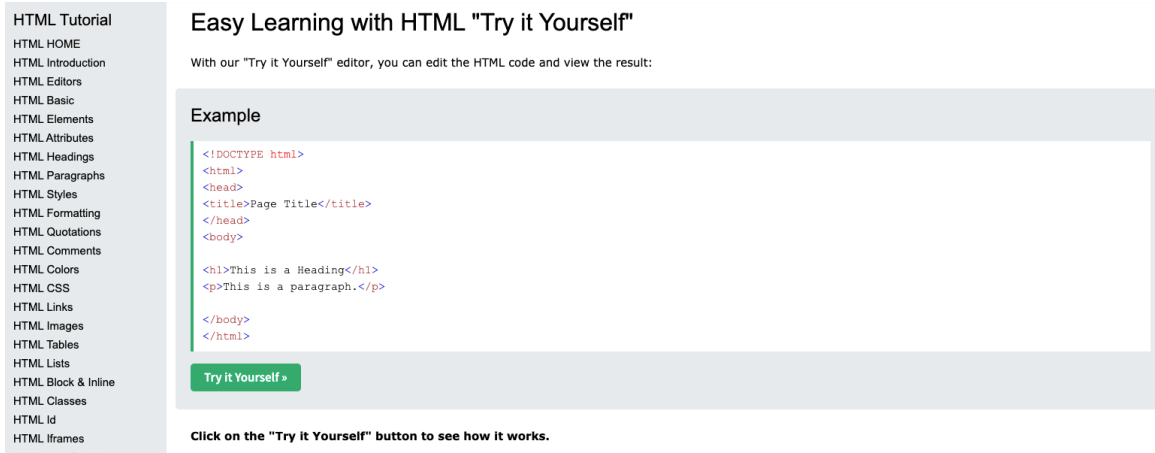
By right-clicking a webpage, you can select “Inspect” and then you will be able to see the DOM of the webpage.



Try and Learn More about HTML

- HTML tutorial on W3Schools

<https://www.w3schools.com/html/default.asp>



The screenshot displays the W3Schools HTML tutorial interface. On the left is a vertical navigation menu with links such as 'HTML Tutorial', 'HTML HOME', 'HTML Introduction', 'HTML Editors', 'HTML Basic', 'HTML Elements', 'HTML Attributes', 'HTML Headings', 'HTML Paragraphs', 'HTML Styles', 'HTML Formatting', 'HTML Quotations', 'HTML Comments', 'HTML Colors', 'HTML CSS', 'HTML Links', 'HTML Images', 'HTML Tables', 'HTML Lists', 'HTML Block & Inline', 'HTML Classes', 'HTML Id', and 'HTML Iframes'. The main content area is titled 'Easy Learning with HTML "Try it Yourself"' and includes the text: 'With our "Try it Yourself" editor, you can edit the HTML code and view the result:'. Below this is an 'Example' section containing a code editor with the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

 At the bottom of the code editor is a green button labeled 'Try it Yourself »'. Below the code editor, a text prompt reads: 'Click on the "Try it Yourself" button to see how it works.'

- You are highly recommended to try some of those online examples by clicking “*Try it Yourself*”

Demo: How to Run/Render a HTML File?

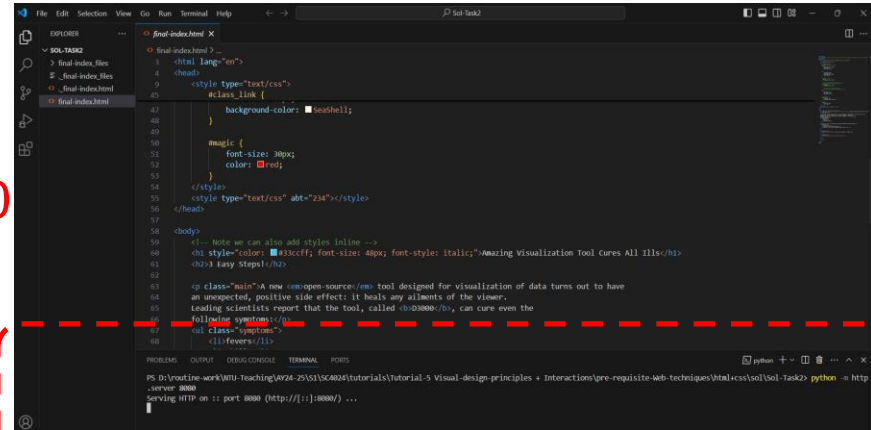
- Directly open the HTML file in a browser
- Run a server on your local computer:
 - For Python3 users:

```
# python3 -m http.server 8080
```
 - For Python2 users:

```
# python3 -m http.server 8080
```

```
# python -m SimpleHTTPServer 8080
```

You are recommended to use VS code to run all the demo code, which provides the terminal for you to type in the python command to run a web server.



Task 1: Create Your First HTML Page

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Page Title</h1>
    <p>This is a really interesting paragraph.</p>
  </body>
</html>
```

Following the template above, you are asked to create the following HTML page by using “Task1-starting-code”:

Amazing Visualization Tool Cures All Ills (h1)

A new open-source tool designed for visualization of data turns out to have positive an unexpected side effect: it heals any ailments of the viewer. Leading scientists report that the tool, called D3000, can cure even the following symptoms:

- fevers
- chills
- general malaise

(ul)

It achieves this end with a patented, three-step process.

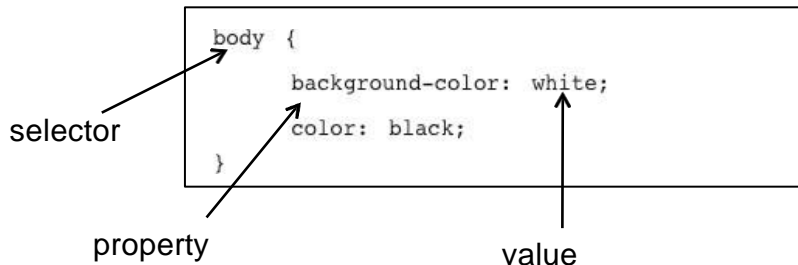
1. Load in data.
2. Generate a visual representation.
3. Activate magic healing function.

(ol)

CSS

Cascading Style Sheets (CSS)

- To style the visual presentation of DOM elements



- Selectors:
 - DOM elements : body, h1, p, div, em, etc.
 - Descendant selectors: div p (*p elements contained in a div*)
 - Class selectors: example: .caption, .label, .axis (*caption, label, and axis are class names*)
 - You can string the classes together: .main.steps (*select one element that has the class names of both “main” and “steps”*)
 - ID selectors: e.g. #nav #export

Properties

- There are tons of properties in CSS
- Common properties: font-family, font-size, background-color, background-image, border, etc.
- An exhaustive list of CSS properties:
<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

It is not necessary to remember all of them! You can always google CSS properties when you need to use them!

Apply CSS rules

- Embed CSS in HTML

```
<html>
  <head>
    <style type="text/css">
      p {
        font-size: 24px;
        font-weight: bold;
        background-color: red;
        color: white;
      }
    </style>
  </head>
  <body>
    <p>If I were to ask you, as a mere
    paragraph, would you say that I
    have style?</p>
  </body>
</html>
```

Apply CSS rules

- Reference an external file

```
<html>
  <head>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <p>If I were to ask you, as a mere
    paragraph, would you say that
    I have style?</p>
  </body>
</html>
```


Apply CSS rules

- Attach inline styles

```
<p style="color: blue; font-size: 48px; font-style: italic;">Inline  
styles  
are kind of a hassle</p>
```

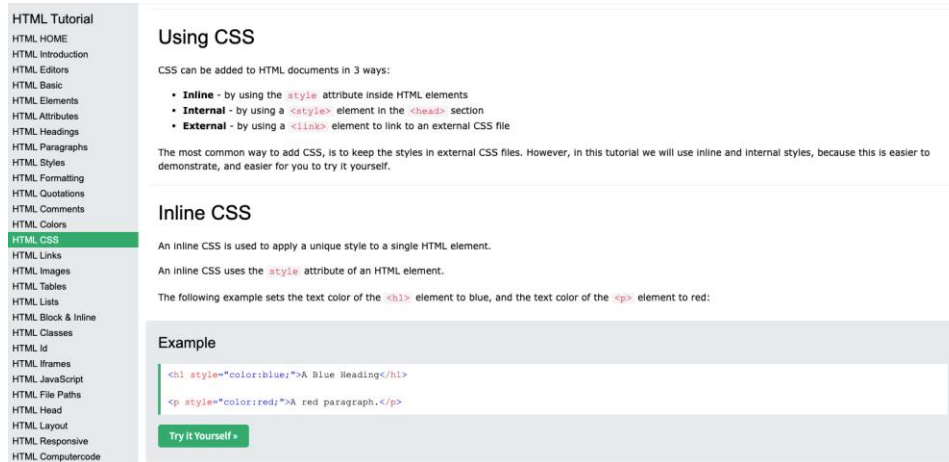


Inline styles are kind of a hassle

Try and Learn More about CSS

- CSS tutorial on W3Schools

https://www.w3schools.com/html/html_css.asp



- You are highly recommended to try some of those online examples by clicking “*Try it Yourself*”

Task 2: CSS Practices

- Add CSS to the “index.html” file of “Task2-starting-code” folder and change its styles as follows:

Amazing Visualization Tool Cures All Ills


3 Easy Steps!

A new *open-source* tool designed for visualization of data turns out to have an unexpected, positive side effect: it heals any ailments of the viewer. Leading scientists report that the tool, called D3000, can cure even the following symptoms:

- fevers
- chills
- general malaise

It achieves this end with a patented, three-step process.

1. Load in data.
2. Generate a visual representation.
3. Activate *magic healing function*.

 NANYANG
TECHNOLOGICAL
UNIVERSITY
SINGAPORE

[This is a link](#)

SVG

Scalable Vector Graphics (SVG)

- We can use D3 to produce SVG
- SVG can be directly included in a HTML document

- How to write SVG?

- Create a SVG element
- Between the svg tags, include your visual elements
 - rect, circle, ellipse, line, text, and path
- (0,0) is the top left corner

- rect `<rect x="0" y="0" width="500" height="50"/>`
- circle `<circle cx="250" cy="25" r="25"/>`
- ellipse `<ellipse cx="250" cy="25" rx="100" ry="25"/>`
- text `<text x="250" y="25">Easy-peasy</text>`
- path anything more complex than the preceding shapes

```
<svg width="500" height="50">
</svg>
```

Styling SVG

`fill`

A color value. Just as with CSS, colors can be specified as named colors, hex values, or RGB or RGBA values.

`stroke`

A color value.

`stroke-width`

A numeric measurement (typically in pixels).

`opacity`

A numeric value between 0.0 (completely transparent) and 1.0 (completely opaque).

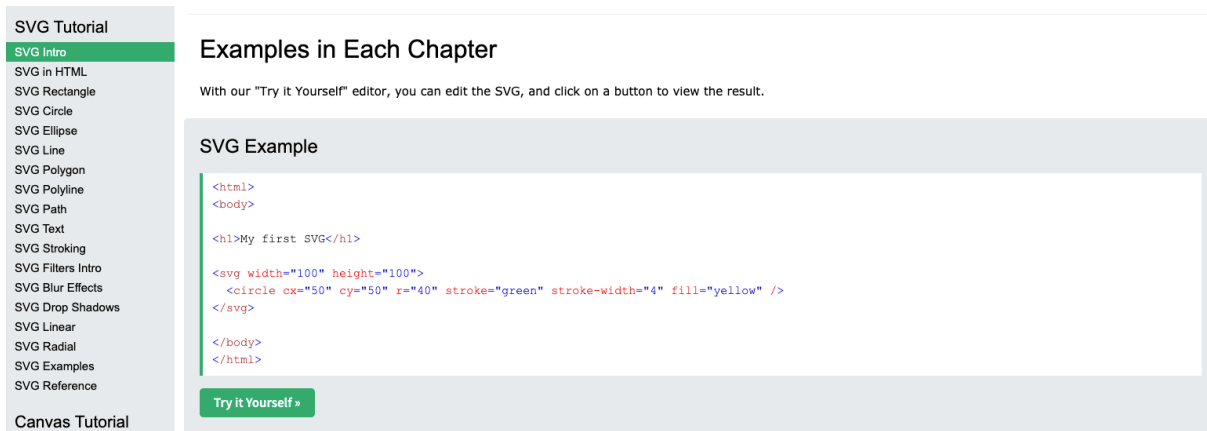
With `text`, you can also use these properties, which work just like in CSS:

- `font-family`
- `font-size`

Try and Learn More about SVG

- SVG tutorial on W3Schools

https://www.w3schools.com/graphics/svg_intro.asp



SVG Tutorial

SVG Intro

SVG in HTML

SVG Rectangle

SVG Circle

SVG Ellipse

SVG Line

SVG Polygon

SVG Polyline

SVG Path

SVG Text

SVG Stroking

SVG Filters Intro

SVG Blur Effects

SVG Drop Shadows

SVG Linear

SVG Radial

SVG Examples

SVG Reference

Canvas Tutorial

Examples in Each Chapter

With our "Try it Yourself" editor, you can edit the SVG, and click on a button to view the result.

SVG Example

```
<html>
<body>

<h1>My first SVG</h1>

<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>

</body>
</html>
```

Try it Yourself »

- SVG Tutorial by Mozilla Developer Network

<https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial>

Task 3: SVG

- Run “svg-index.html” and you will see the following yellow circle within an SVG
- Do the following changes and see the final visualization effect
 - Change the SVG width and height to 400
 - Move the center of the circle to (200,200)
 - Change the fill color to green
 - Change the radius to 150
 - Change the border color to red

My first SVG



Javascript

Javascript

- Created in 10 days in May 1995 by Brendan Eich
- Originally developed as a prototype language for web browser (Client-side)
- Now used in server-side (Node.js) as well
- Not related to Java, just named similarly for marketing purpose
- C style syntax but got inspiration from Functional programming
 - *for, while, continue, break, if/else, switch* are similar to C or Python
 - operators (+, -, *, /, %) are also similar (except ==, !=, ||)
 - include function operations such as *map, reduce, forEach*.

Quick Review of JS syntax

- Print message to the console (in the development window)
 - `console.log("hello world!");`
- Declare a variable
 - `var number = 5;`
 - You can later change the variable content to a value of different type
 - `number = "hello";`
 - JS is a loosely typed language
- Declare an array (useful for you to try some visualization)
 - `var numbers = [1,2,3,4,5];`
- Objects

```
var fruit = {  
  kind: "grape",  
  color: "red",  
  quantity: 12,  
  tasty: true  
};
```

```
fruit.kind    //Returns "grape"  
fruit.color   //Returns "red"  
fruit.quantity //Returns 12  
fruit.tasty   //Returns true
```

Objects

Quick Review of JS syntax

- Mathematical Operators

<code>==</code>	<i>//Equal to</i>	<code>+</code>	<i>//Add</i>
<code>!=</code>	<i>//Not equal to</i>	<code>-</code>	<i>//Subtract</i>
<code><</code>	<i>//Less than</i>	<code>*</code>	<i>//Multiply</i>
<code>></code>	<i>//Greater than</i>	<code>/</code>	<i>//Divide</i>
<code><=</code>	<i>//Less than or equal to</i>		
<code>>=</code>	<i>//Greater than or equal to</i>		

- Control structures

```
if (3 < 5) {  
    console.log("Eureka! Three is less than five!");  
}  
  
for (var i = 0; i < 5; i++) {  
    console.log(i); //Prints value to console  
}
```

- Functions (a chunk of reusable code)

- Comments

```
var calculateGratuity = function(bill) {  
    return bill * 0.2;  
};  
  
/* JavaScript supports CSS-style comments like this. */  
  
// But double-slashes can be used as well.
```

Javascript

- Putting javascript code in your HTML
 - External source file:

```
<!-- below is how you are going to load your javascript file -->
```

```
<script type="text/javascript" src="myExample.js"></script>
```

- Direct put in your HTML:

```
<script type="text/javascript">
```

```
    //Width and height  
    var w = 600;  
    ... var h = 250;
```

```
</script>
```

Javascript Tutorials

- W3School Tutorial
 - https://www.w3schools.com/js/js_operators.asp

Task 4: Use JavaScript to Change SVG styles

- Run “js-index.html” and follow the example code to update the circle styles
 - Change the fill color to red
 - Change the radius to 150
 - Change the border color to yellow
 - Change the stroke width to 10
 - Move the circle center to (150,150)

My first SVG

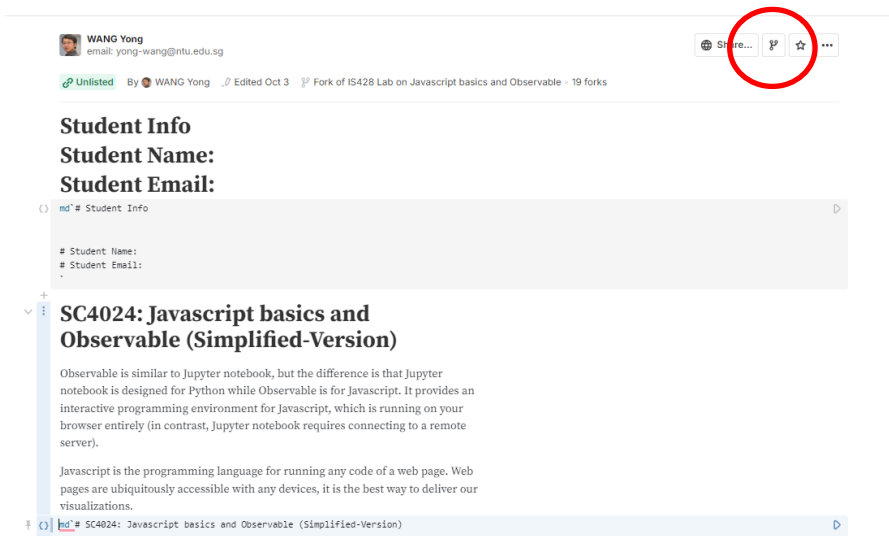


Optional Practices on JavaScript!

Link: <https://observablehq.com/d/4e3a3396fcd56802>

Notes on Writing JS Code on Observable Workbook

- You need to duplicate the give observable workbook by “forking” it
- Save your code before refreshing the webpage!



The screenshot displays the Observable notebook interface. At the top, the user profile for 'WANG Yong' is visible, along with a 'Share...' button and a 'Fork' button, both of which are circled in red. Below the header, the notebook content is shown, starting with 'Student Info' and 'Student Name:'. The main section is titled 'SC4024: Javascript basics and Observable (Simplified-Version)' and contains text explaining the difference between Jupyter and Observable notebooks, and the benefits of JavaScript for web-based visualizations.

WANG Yong
email: yong-wang@ntu.edu.sg

Unlisted By WANG Yong Edited Oct 3 Fork of IS428 Lab on Javascript basics and Observable · 19 forks

Student Info
Student Name:
Student Email:

```
md # Student Info
```

```
# Student Name:  
# Student Email:  
.
```

SC4024: Javascript basics and Observable (Simplified-Version)

Observable is similar to Jupyter notebook, but the difference is that Jupyter notebook is designed for Python while Observable is for Javascript. It provides an interactive programming environment for Javascript, which is running on your browser entirely (in contrast, Jupyter notebook requires connecting to a remote server).

Javascript is the programming language for running any code of a web page. Web pages are ubiquitously accessible with any devices, it is the best way to deliver our visualizations.

```
md # SC4024: Javascript basics and Observable (Simplified-Version)
```