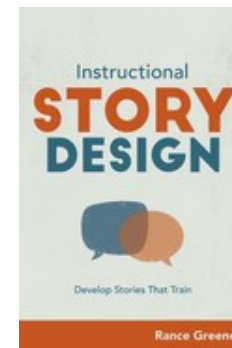
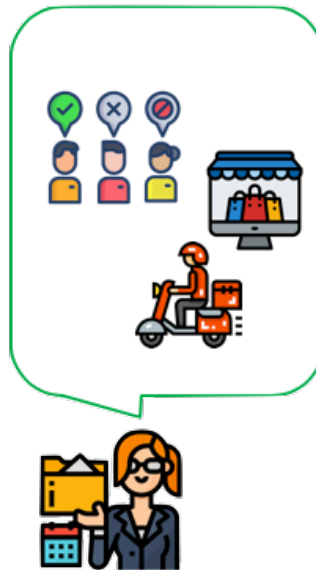
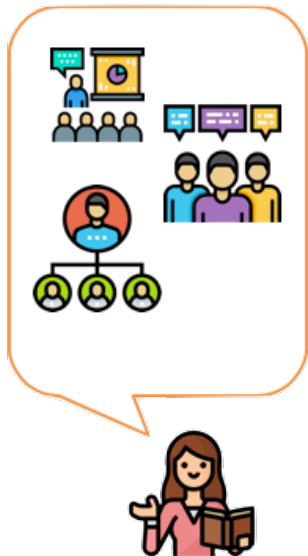


ICT239 Web Programming

Introduction to Web Programming Seminar 1

Jean Looi cklooi001@suss.edu.sg



Inspired by

Instructional Story Design:
Develop Stories That Train
- Rance Greene

PART I
Discover

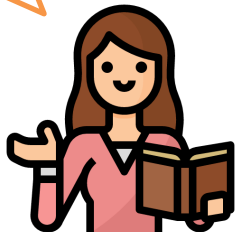
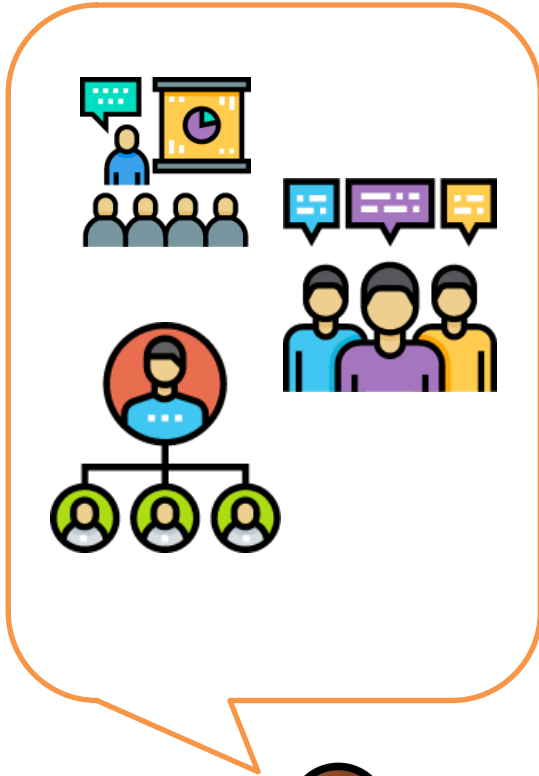
PART II
Design

PART III
Deliver

PART IV
Overcome
Barriers

Scenarios: Teacher, Factory Manager, Web CEO

https://docs.google.com/spreadsheets/d/1fLdwMjcA3nu_zHP4BcxTv-4NDxyZGSj3CtVwxrxHVmM/edit#gid=0



Group: School



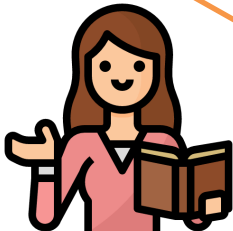
As a Form Teacher:

“I would like to manage the students in my class”

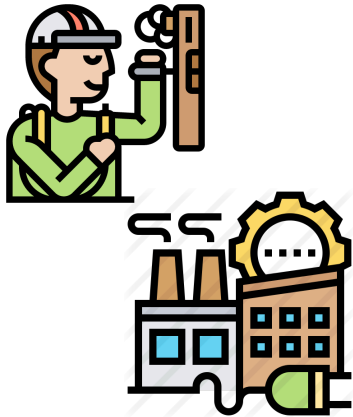
“What kind of subjects are they taking ?”

“Which other teachers are teaching in my class”

“What is the performance of my students ?”



Group: Factory



As the Factory Managing Director:

“I would like to monitor my Factory’s output”

“How many components are my factory producing ?”

“Which parts are in shortages or in excess”

“Who are the suppliers supplying to the current demand?”



Group: Web Store

As the CEO:

“I would like to know who are the top customers”

“How many different products are they buying regularly?”

“Is the Web Store profitable ?”

“Are our products delivered on time to our customers ?”



COURSE OVERVIEW


INTRODUCTION TO WEB PROGRAMMING

- Seminar 1 - Introduction to Web Programming
- Seminar 2 – Front End Development – HTML & CSS
- Seminar 3 – Front End Development – JavaScript
- Seminar 4 – Backend Development – Flask I
- Seminar 5 – Backend Development – Flask II
- Seminar 6 – Backend Development – Flask III


Seminar session will be held in Virtual Zoom / Collaborate

COURSE OVERVIEW

BMI Calculator



[Log BMI](#)



My id is: abc@def.com
My gender and weight

BMI Calculator

Weight (kilogram)

Height

☒ m ☐ cm

The Output Area

Watch the following YouTube:

<https://www.youtube.com/watch?v=5xLGyOYvDrU>

COURSE OVERVIEW

Learning Mode

- Self-study guided by the study guide units. Independent study will require at **least 3 hours per week**.
- Working on assignments
- Classroom Seminar sessions (3 hours each session, 6 sessions in total).
- Online Office hours (2 hours each session, 6 sessions in total)

Seminar Activities

- Lecture & Concepts
- Sharing Exercises (Pre-allocated or Volunteer?)
- Specific Questions (Ask before the seminar) and Clarifications
- Related Topics (if any)
- (Not for debugging!)

COURSE OVERVIEW

Learning Material

- Data visualization with Python and Javascript. Crafting a Data-visualisation Toolchain for the Web
Kyran Dale (2016) Published by O'Reilly Media, Inc. , 1005 Gravenstein Highway North, Sebastopol, CA 95472.
- Python for Everybody. Exploring Data Using Python 3
Charles R. Severance. Copyright ~2009- Charles Severance.

Websites

- <http://www.dr-chuck.com/>
- <https://developer.mozilla.org/en-US/docs/Learn>

COURSE OVERVIEW

iStudyGuide

Interaction with Instructor and Fellow Students

- Collaborate and Share using Discussion Forum
- Online Office hours after the Seminar

Academic Integrity

*Self-Help Chunks

ICT239_JAN20_T01 > Discussions > ICT239 T01 Group Question and Answer Discussion Forum

2020_JAN_T2_PT_6

Home
Announcements
Discussions
People
Modules
Syllabus
Assignments
Grades
SUSS Gradebook
Report a Problem
Accessibility Check
New Analytics



ICT239 T01 Group Question and Answer Discussion Forum

TAN MENG CHOON (CHEN MINGCHUN)

All sections

Dear Class,
Welcome to ICT239 Web Programming Class!
Kindly use this Discussion Forum to post **ANY** questions you may have on this course or Web Programming in general. I will respond to your questions as soon as I can.
See you in class.

Instructor
Tan Meng Choon



COURSE OVERVIEW

Assessments

Assessment	Description	Weight Allocation
Assignment 1	Pre-Class Quiz	6%
Assignment 2	Online Quiz	6%
Assignment 3	Tutor-Marked Assignment	18%
Examination	ECA	70%
TOTAL		100%

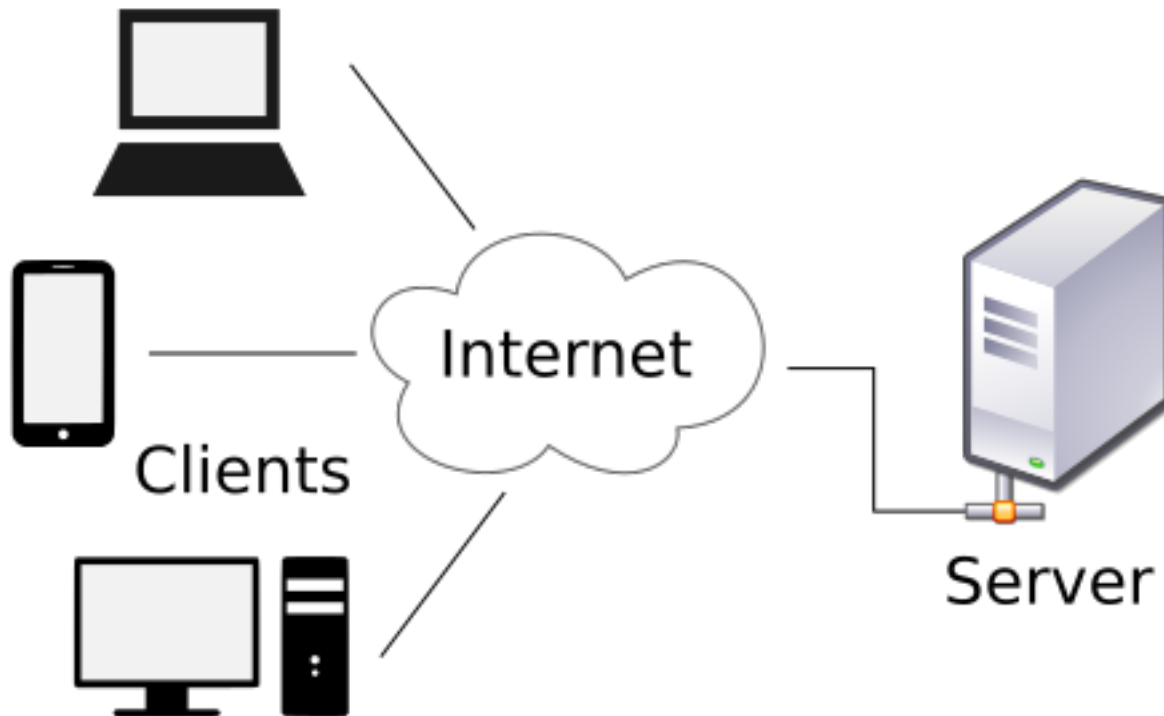
SEMINAR OVERVIEW

INTRODUCTION TO WEB PROGRAMMING – LEARNING OBJECTIVES

1. Understand fundamental of Web Technologies - Internet, Browsers, Web Servers, Web Applications, Databases
2. Web Application Development Tools - Testing & Debugging
3. Basic Web Page Elements and Learn how Web Pages are deployed

FUNDAMENTAL OF WEB TECHNOLOGY

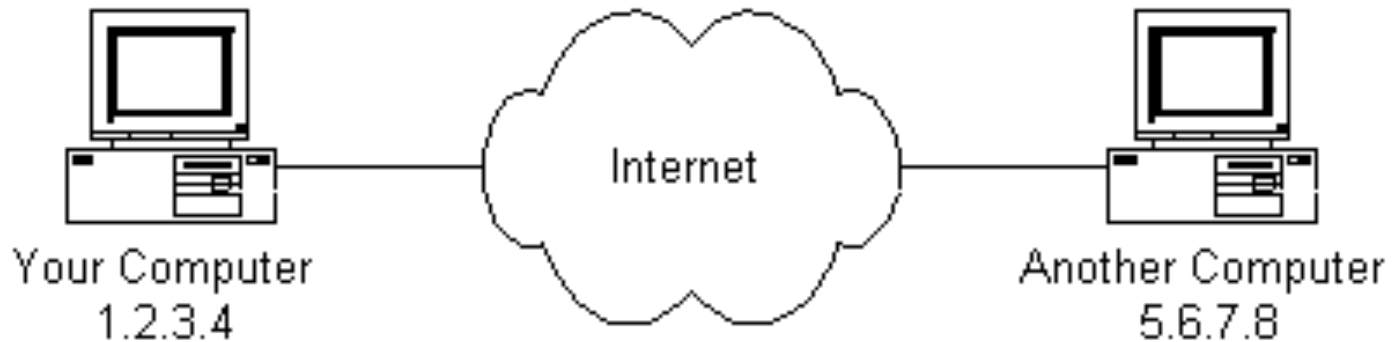
Client-Server Model



https://www.wikiwand.com/en/Client%E2%80%93server_model

FUNDAMENTAL OF WEB TECHNOLOGY

Client-Server Model



The elements of an URL

http://	www.	copahost.com	/blog/page.html
Protocol	Subdomain	Domain name	Path + page



<https://web.stanford.edu/class/msande91si/www-spr04/readings/week1/InternetWhitepaper.htm>

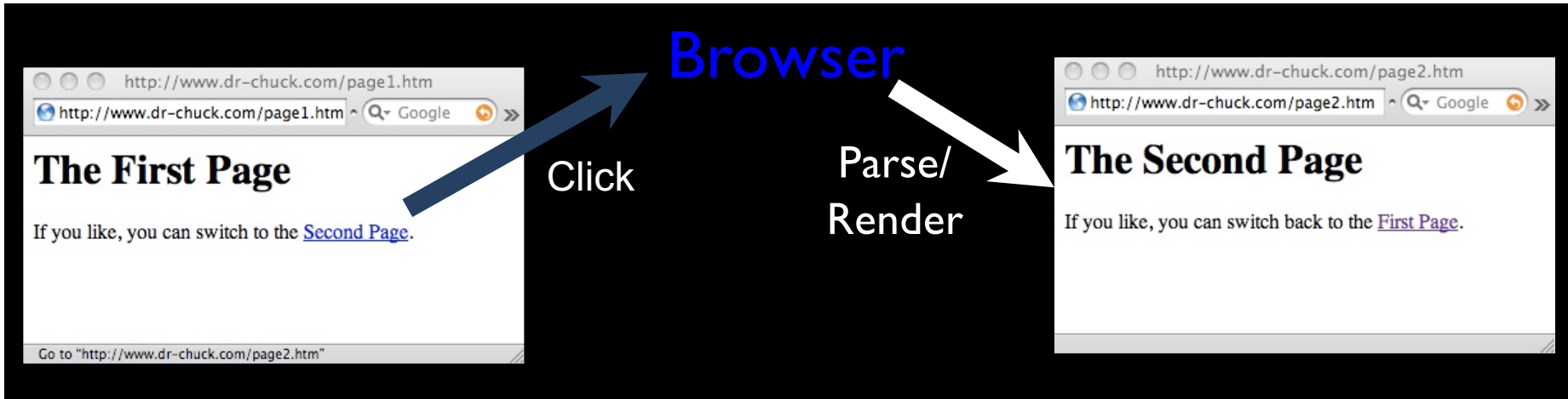
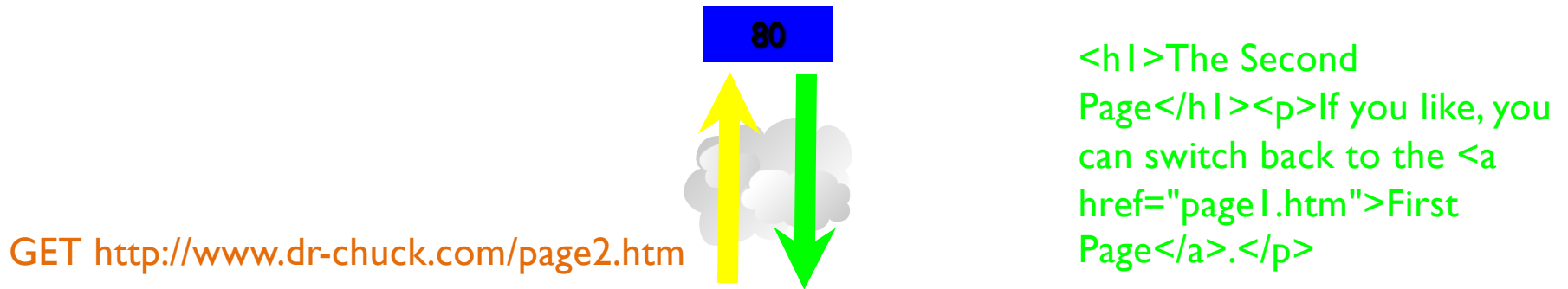
FUNDAMENTAL OF WEB TECHNOLOGY

Client-Server Model

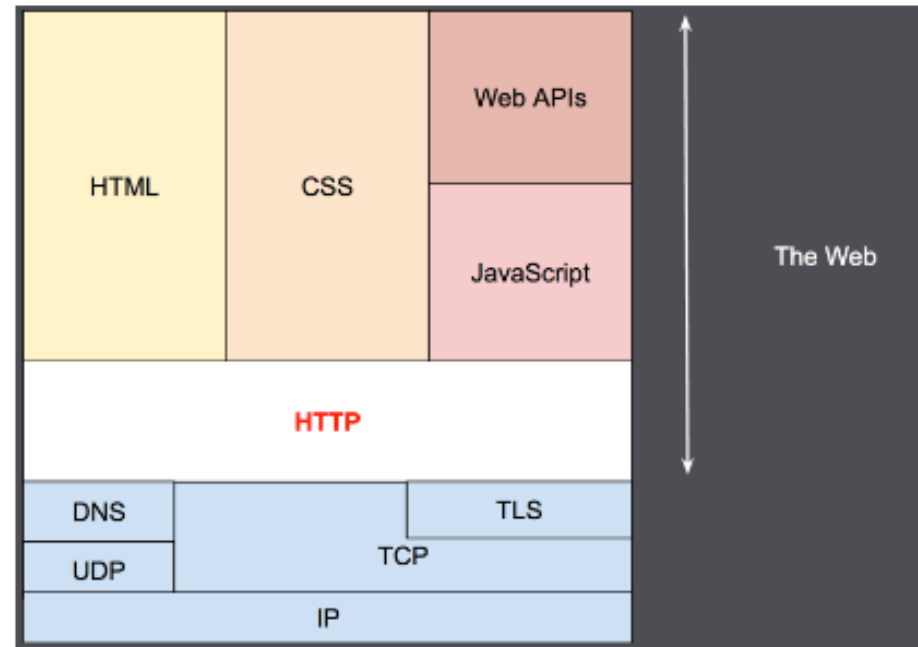
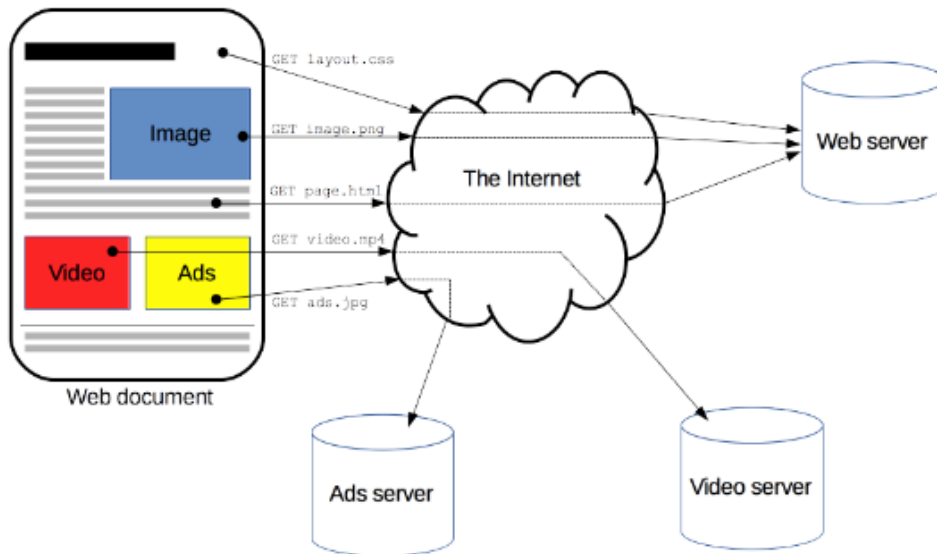
Request

Web Server

Response



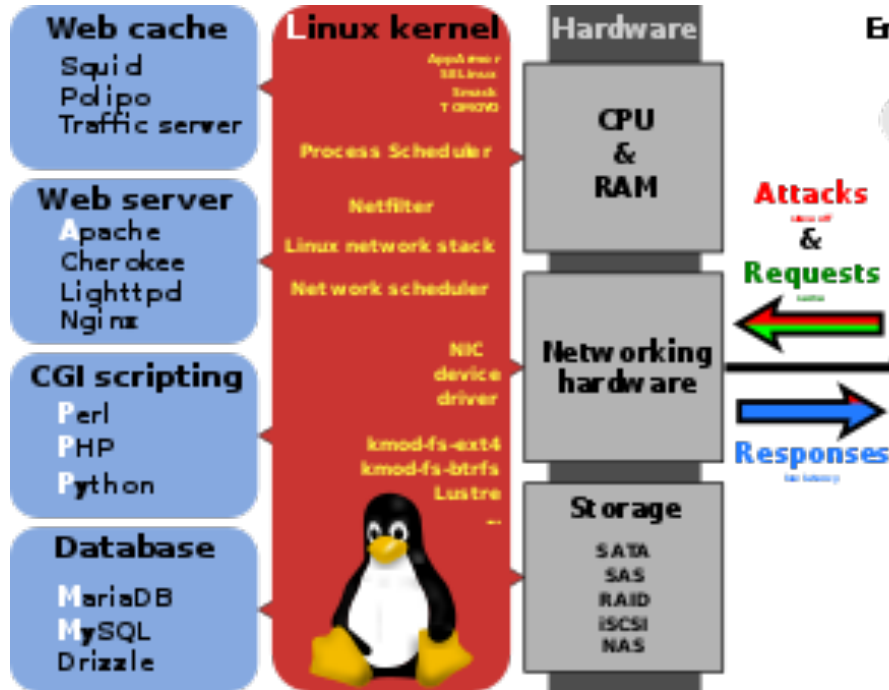
FUNDAMENTAL OF WEB TECHNOLOGY



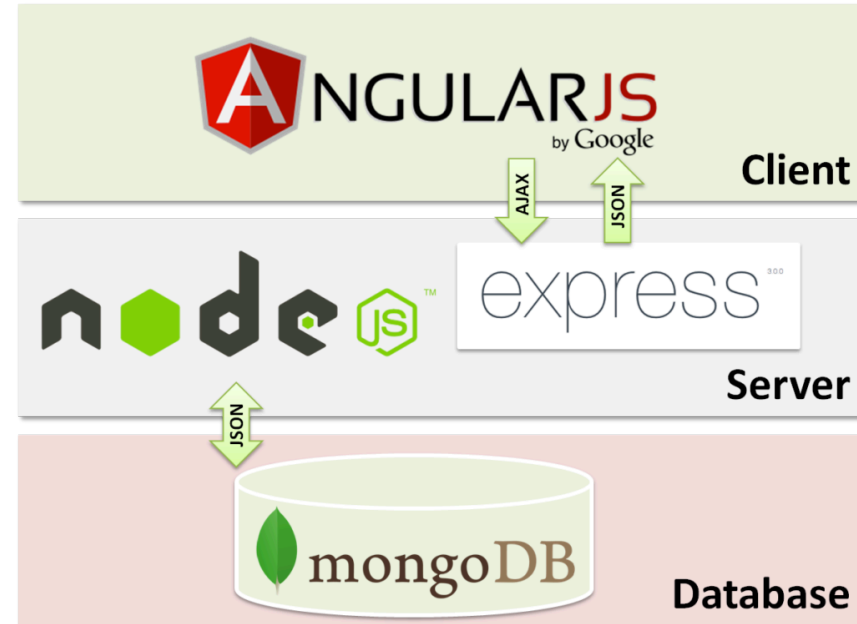
WEB PROGRAMMING STACK

Frontend and Backend Components

“LAMP” Stack



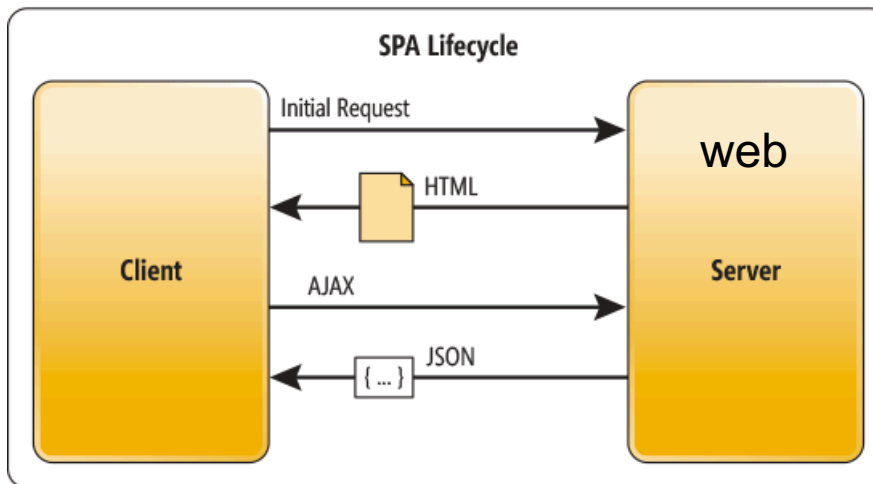
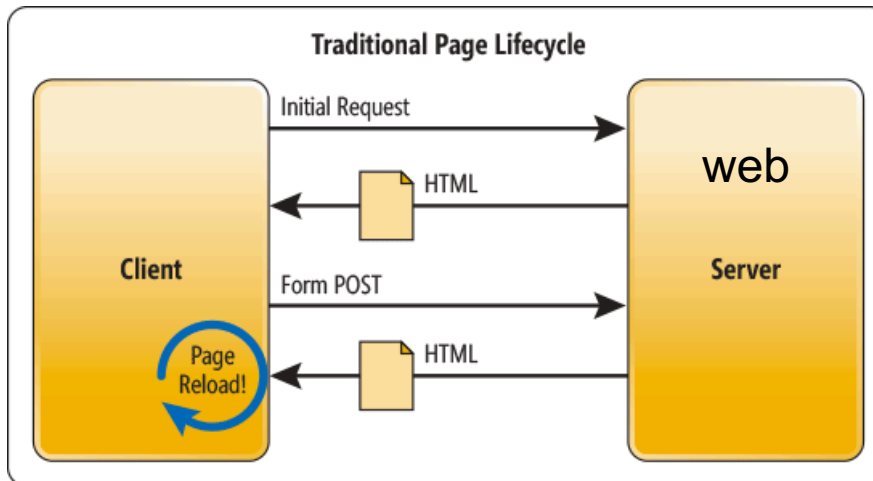
“MEAN” Stack



WEB PROGRAMMING STACK

Topic 2: Web Programming Stack

Single Page Applications



Loading all required HTML/CSS/JS in a single page load.

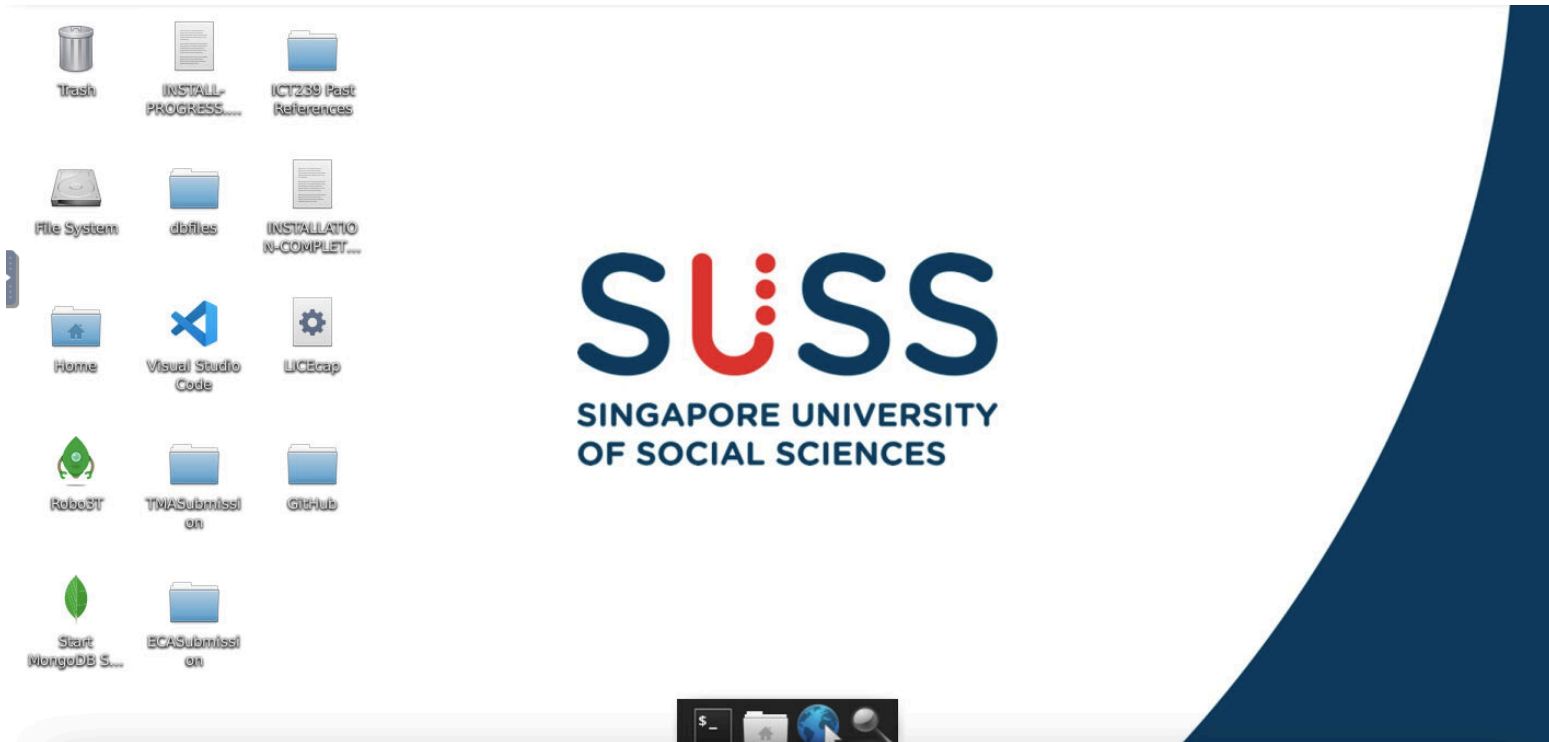
Other resources are loaded asynchronously – ie in response to a user action

Eg using a search function



TOOLING UP: DEVELOPMENT SET UP

Introduction to Vocareum



TOOLING UP: DEVELOPMENT SET UP

Tooling Up: Development Set Up

- Text Editor
 - Syntax Highlight
 - Indentation levels
 - Multiple window/panes
 - Code Linting
 - Eg Sublime Text, Atom, VS Code, vim
- Browser with Development Tools
 - Elements
 - CSS Styling / Computed
 - Client side console
- Terminals or Command prompts
 - Server side console
 - Debug / Output logs

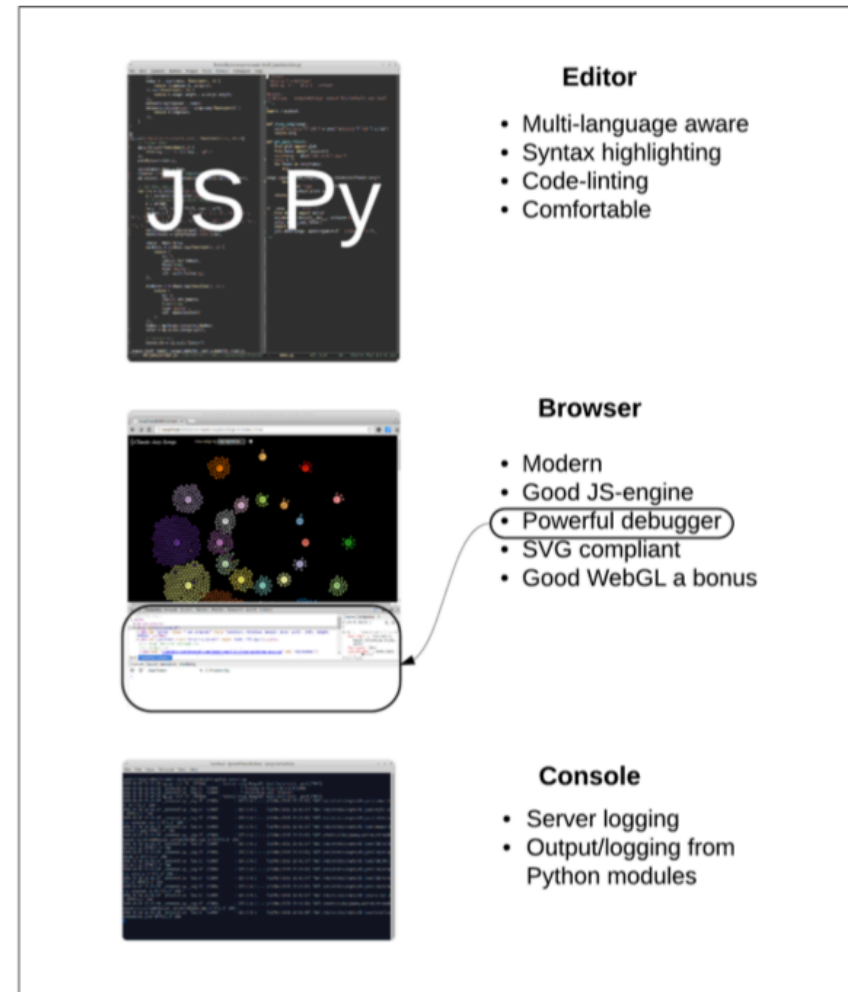


Figure 4-1. Primary Webdev Tools

TOOLING UP: DEVELOPMENT SET UP

Building a Web-page – the Basic Elements

- Create a html file
- **Python 2.7**
 - `python -m SimpleHTTPServer`
- **Python 3**
 - `python -m http.server`
- View the output from browser (client)
- View the output from console (server)
- HTML skeleton

```
<!DOCTYPE html> <meta charset="utf-8">
<style>
<!-- CSS -- > </style>
<body>
<!-- page content -->
<script>
<!-- JavaScript -->
</script> </body>
```

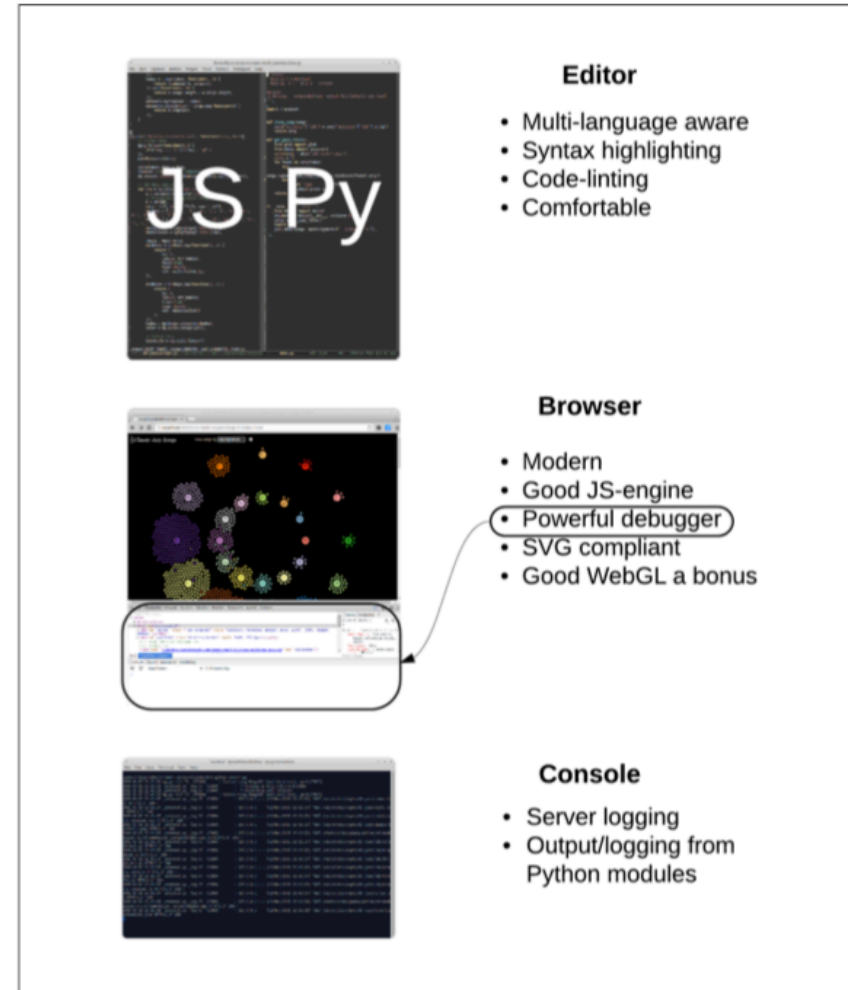
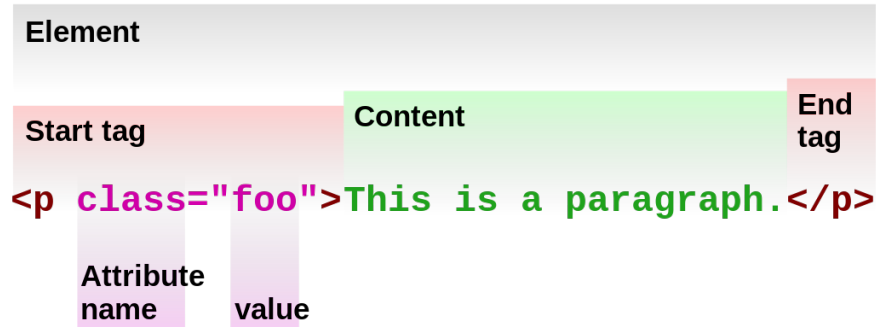


Figure 4-1. Primary Webdev Tools

FRONT END DEVELOPMENT

Structuring the Web

HTML elements



HTML - is the markup language that we use to structure and give meaning to our web content, for example defining paragraphs, headings, and data tables, or embedding images and videos in the page.

Anatomy of HTML Document

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>My test page</title>
6    </head>
7    <body>
8      
9    </body>
10 </html>
```

FRONT END DEVELOPMENT

Styling the Web

CSS examples

```
1 h1 {  
2   color: blue;  
3   background-color: yellow;  
4   border: 1px solid black;  
5 }  
6  
7 p {  
8   color: red;  
9 }
```

Language of style rules that we use to apply styling to our HTML content, for example setting background colors and fonts, and laying out our content in multiple columns.

HTML Document

```
1 <!DOCTYPE html>  
2 <html>  
3   <head>  
4     <meta charset="utf-8">  
5     <title>My CSS experiment</title>  
6     <link rel="stylesheet" href="style.css">  
7   </head>  
8   <body>  
9     <h1>Hello World!</h1>  
10    <p>This is my first CSS example</p>  
11  </body>  
12 </html>
```



FRONT END DEVELOPMENT

Making the Web (Client) Dynamic

Example

```
1 | <p>Player 1: Chris</p>
```

```
1 | p {  
2 |   font-family: 'helvetica neue', helvetica, sans-serif;  
3 |   letter-spacing: 1px;  
4 |   text-transform: uppercase;  
5 |   text-align: center;  
6 |   border: 2px solid rgba(0,0,200,0.6);  
7 |   background: rgba(0,0,200,0.3);  
8 |   color: rgba(0,0,200,0.6);  
9 |   box-shadow: 1px 1px 2px rgba(0,0,200,0.4);  
10 |  border-radius: 10px;  
11 |  padding: 3px 10px;  
12 |  display: inline-block;  
13 |  cursor: pointer;  
14 | }
```

```
1 | const para = document.querySelector('p');  
2 |  
3 | para.addEventListener('click', updateName);  
4 |  
5 | function updateName() {  
6 |   let name = prompt('Enter a new name');  
7 |   para.textContent = 'Player 1: ' + name;  
8 | }
```

The scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else.

PLAYER 1: CHRIS

An embedded page at mdn.mozillademos.org says

Enter a new name

Cancel

OK

PLAYER 1: JESSICA!

FRONT END DEVELOPMENT

Programming the Web Page

Document Object Model (DOM)

The HTML DOM Tree of Objects

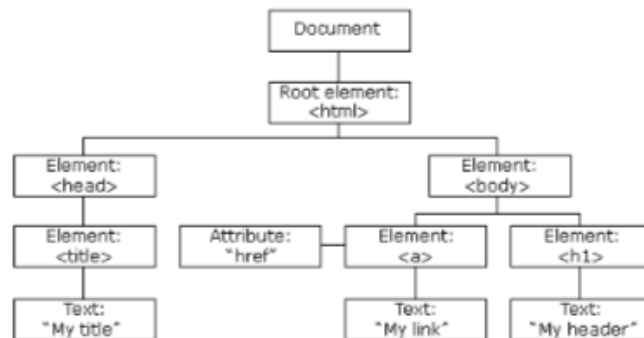


Figure 1: HTML DOM

Source: www.w3schools.com

A programming interface for HTML. Represents the page so that programs can change the document structure, style, and content. The DOM represents the document as nodes and objects. That way, programming languages can connect to the page.

```
1 var table = document.getElementById("table");
2 var tableAttrs = table.attributes; // Node/Element interface
3 for (var i = 0; i < tableAttrs.length; i++) {
4     // HTMLTableElement interface: border attribute
5     if(tableAttrs[i].nodeName.toLowerCase() == "border")
6         table.border = "1";
7 }
8 // HTMLTableElement interface: summary attribute
9 table.summary = "note: increased border";
```

FRONT END DEVELOPMENT

Topic 1: Building a Web-page – the Basic Elements

- Content blocks
 - <DIV>
- Headers
 - <h1> .. <h6>
- Blocks elements
 - <p> paragraphs
 - images
 - <a> hyper links
 - inline text
 - list
 - <table>

How to create this page ?

ICT239 COURSE GUIDE

3. Learning Outcomes

Knowledge & Understanding (Theory Component)

By the end of this course, you should be able to:

- Analyse the HTTP protocol in a sequence of requests for coherent communication
- Apply programming methods to present information in HTML
- Demonstrate the visualisation of data on a web presentation
- Employ web programming framework for developing website

Key Skills (Practical Component)

By the end of this course, you should be able to:

- Construct a prototype website to present information from multiple sources upon users requests
- Experiment with new methods of visualisation the data to suit particular information needs

4. Learning Materials

The following is a list of the required learning materials to complete this course.

Required Textbook(s)

Data visualization with Python and Javascript: Crafting a Data-visualisation Toolchain for the Web. Kyran Dale (2016) Published by O'Reilly Media, Inc. , 1005 Gravenstein Highway North, Sebastopol, CA 95472.
Python for Everybody. Exploring Data Using Python 3. Charles R. Severance. Copyright ~2009- Charles Severance.

Other recommended study material (Optional)

The following learning materials may be required to complete the learning activities:

Website(s):

<https://developer.mozilla.org/en-US/docs/Learn/Server-side/>
<https://developer.mozilla.org/en-US/docs/Learn/JavaScript/>

CHROME DEVELOPER TOOLS & DEBUGGING

- The elements Tab
- The Source Tab
- Other Tools
 - network
 - memory
 - application data

Exercise:

<https://developers.google.com/web/tools/chrome-devtools/open>

- DevTools for Beginners
- DOM
- CSS
- JS
- Console

Chapter 3: Scalable Vector Graphics (SVG)

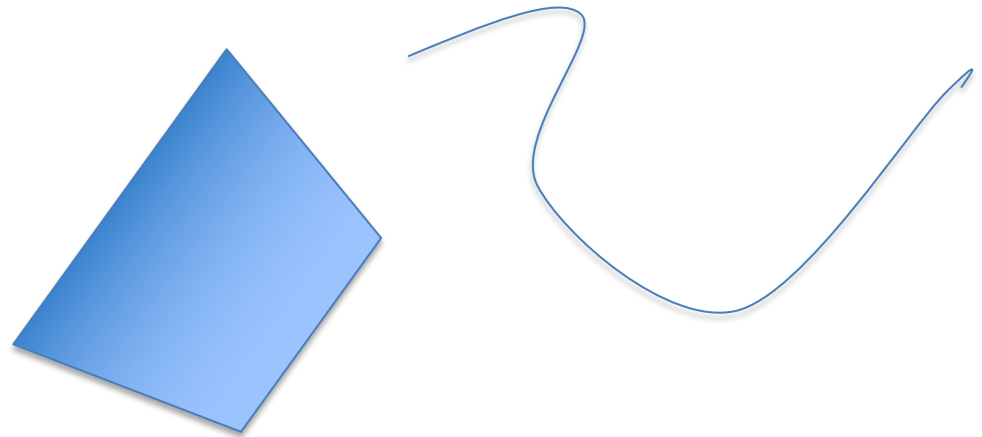
Topic 1: Introduction to Web Images

Scalable Vector Graphics (SVG) is an XML-based vector image format for two-dimensional graphics

- Support interactivity and animation
- Images and their behaviours are defined in XML text files.

```
<!DOCTYPE <html>
<body>
  <svg width=100 height=100>
    <circle cx="50" cy="50" r="10" />
  </svg>
</body> </html>
```

```
<style>
circle {
  fill: blue;
  stroke: red;
  stroke-width: 2px;
}
</style>
```



Refer to Study Guide Topic 1: SVG code examples

- Circle
- Line
- Rect
- Polygon

Chapter 3: Scalable Vector Graphics (SVG)

Topic 1: Introduction to Web Images

Scalable Vector Graphics (SVG) is an XML-based vector image format for two-dimensional graphics

- Support interactivity and animation
- Images and their behaviours are defined in XML text files.

Refer to Study Guide Topic 1: SVG code examples - Advance

- Path
- Curves
- Scaling & Rotation
- Groups
- Layering & Transparency



Figure 10: SVG transforms
Source: Developer



Figure 11: SVG opacity
Source: Developer



Chapter 3: Scalable Vector Graphics (SVG)

Topic 2: Javascript SVG using D3.js

Javascript visualisation library for

- manipulating documents (DOM) based on data.
- Create charts , maps, diagrams (<https://github.com/d3/d3/wiki/Gallery>)

Refer to Study Guide Section 3.2 : D3JS code examples

- D3js Bar Charts

SEMINAR OVERVIEW

INTRODUCTION TO WEB PROGRAMMING – LEARNING OBJECTIVES

1. Understand fundamental of Web Technologies - Internet, Browsers, Web Servers, Web Applications, Databases
2. Web Application Development Tools - Testing & Debugging
3. Basic Web Page Elements and Learn how Web Pages are deployed

TODO BEFORE NEXT SEMINAR

Reminder

- Read Study Unit 2,3 – they are related
- References
 - Data visualization with python and javascript - Crafting a Data-visualisation Toolchain for the Web – Chap 4
 - https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web
 - <https://developer.mozilla.org/en-US/docs/Learn/HTML>
 - <https://developer.mozilla.org/en-US/docs/Learn/CSS>
- Try out the exercises! (In the Study Unit)
- Use your Canvas resources
 - Study Guide
 - Discussion Forums
 - Course Textbook / Google

PRE-READ

Structuring the web with HTML

<https://developer.mozilla.org/en-US/docs/Learn/HTML> (dated 28 Mar 2020)

Section	Note
Introduction to HTML Overview	Getting started with HTML What's in the head? Metadata in HTML HTML text fundamentals Creating hyperlinks Advanced text formatting Document and website structure Debugging HTML Assessment: Marking up a letter Assessment: Structuring a page of content
Multimedia and Embedding	Multimedia and embedding overview Images in HTML Video and audio content Adding vector graphics to the Web Assessment: Mozilla splash page
HTML Tables	HTML tables overview HTML table basics Assessment: Structuring planet data

PRE-READ

Learn to style HTML using CSS

<https://developer.mozilla.org/en-US/docs/Learn/CSS> (dated 28 Mar 2020)

Section	Note
CSS First Steps	CSS first steps overview What is CSS? Getting started with CSS How CSS is structured How CSS works Using your new knowledge
CSS Building Blocks	CSS building blocks overview Cascade and inheritance CSS selectors The box model Backgrounds and borders Values and units Sizing items in CSS Images, media, and form elements Styling tables Debugging CSS
Use CSS to solve common problems (Extra)	Common use cases Uncommon and advanced techniques

Thank You.