Web Technology

SHIV NADAR
UNIVERSITY
CHENNAI

Introduction

Department of Computer Science and Engineering
Shiv Nadar University

Marks Split-up

1. **Theory**
   - **Mid Semester - 30 marks**
   - **End Semester - 50 marks**
   - **CT/Assignment - 20 marks**

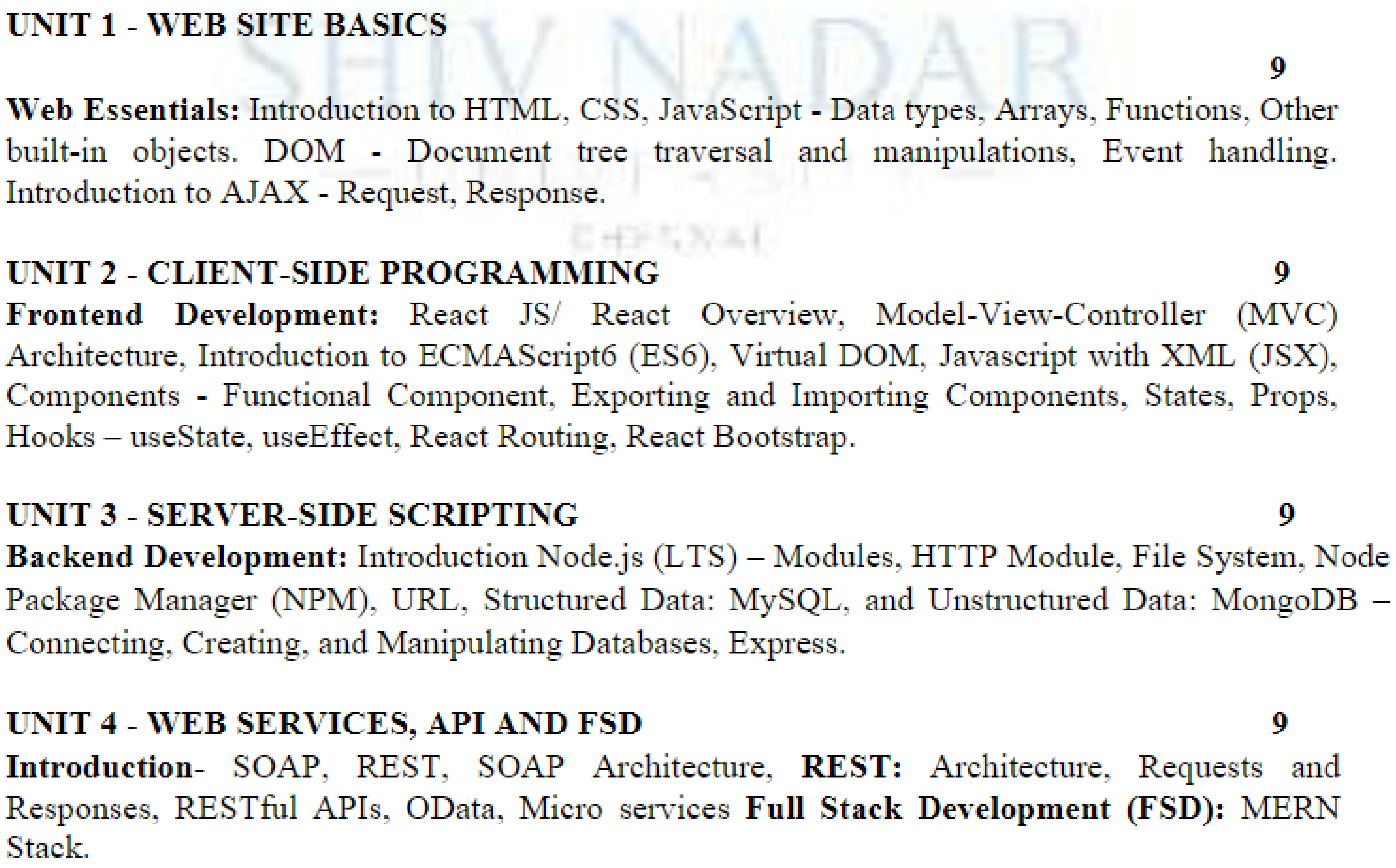
2. **Lab**
   - **Model Exams - 20 marks**
   - **End Semester Exams – 80 marks**

Course Outcomes

- **CO1 - Design Workflows, Websites, and Web Components.**
- **CO2 - Front-End Development (Client-Side Programming).**
- **CO3 - Back-End Development (Server-Side Programming).**
- **CO4 - Build Web Services APIs.**
- **CO5 - Full-Stack Development.**

K1

Syllabus Coverage



UNIT 1 - WEB SITE BASICS 9

Web Essentials: Introduction to HTML, CSS, JavaScript - Data types, Arrays, Functions, Other built-in objects. DOM - Document tree traversal and manipulations, Event handling. Introduction to AJAX - Request, Response.

UNIT 2 - CLIENT-SIDE PROGRAMMING 9

Frontend Development: React JS/ React Overview, Model-View-Controller (MVC) Architecture, Introduction to ECMAScript6 (ES6), Virtual DOM, Javascript with XML (JSX), Components - Functional Component, Exporting and Importing Components, States, Props, Hooks – useState, useEffect, React Routing, React Bootstrap.

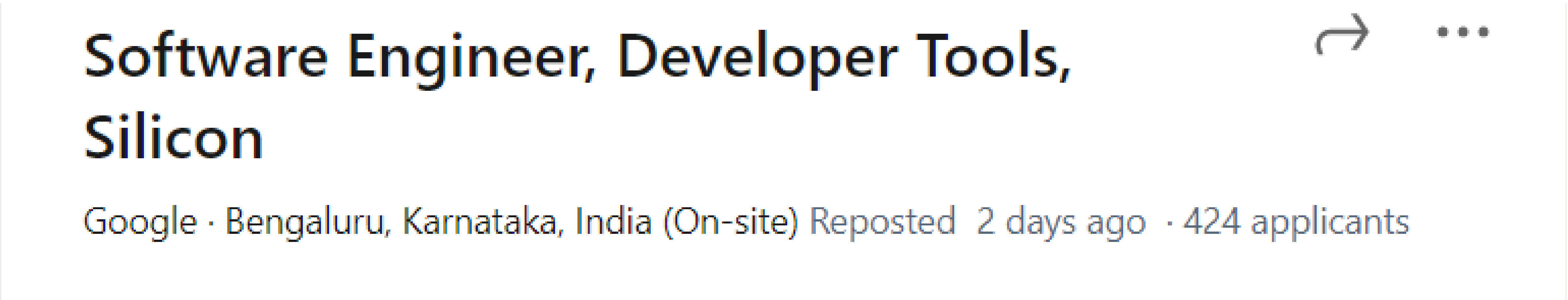
UNIT 3 - SERVER-SIDE SCRIPTING 9

Backend Development: Introduction Node.js (LTS) – Modules, HTTP Module, File System, Node Package Manager (NPM), URL, Structured Data: MySQL, and Unstructured Data: MongoDB – Connecting, Creating, and Manipulating Databases, Express.

UNIT 4 - WEB SERVICES, API AND FSD 9

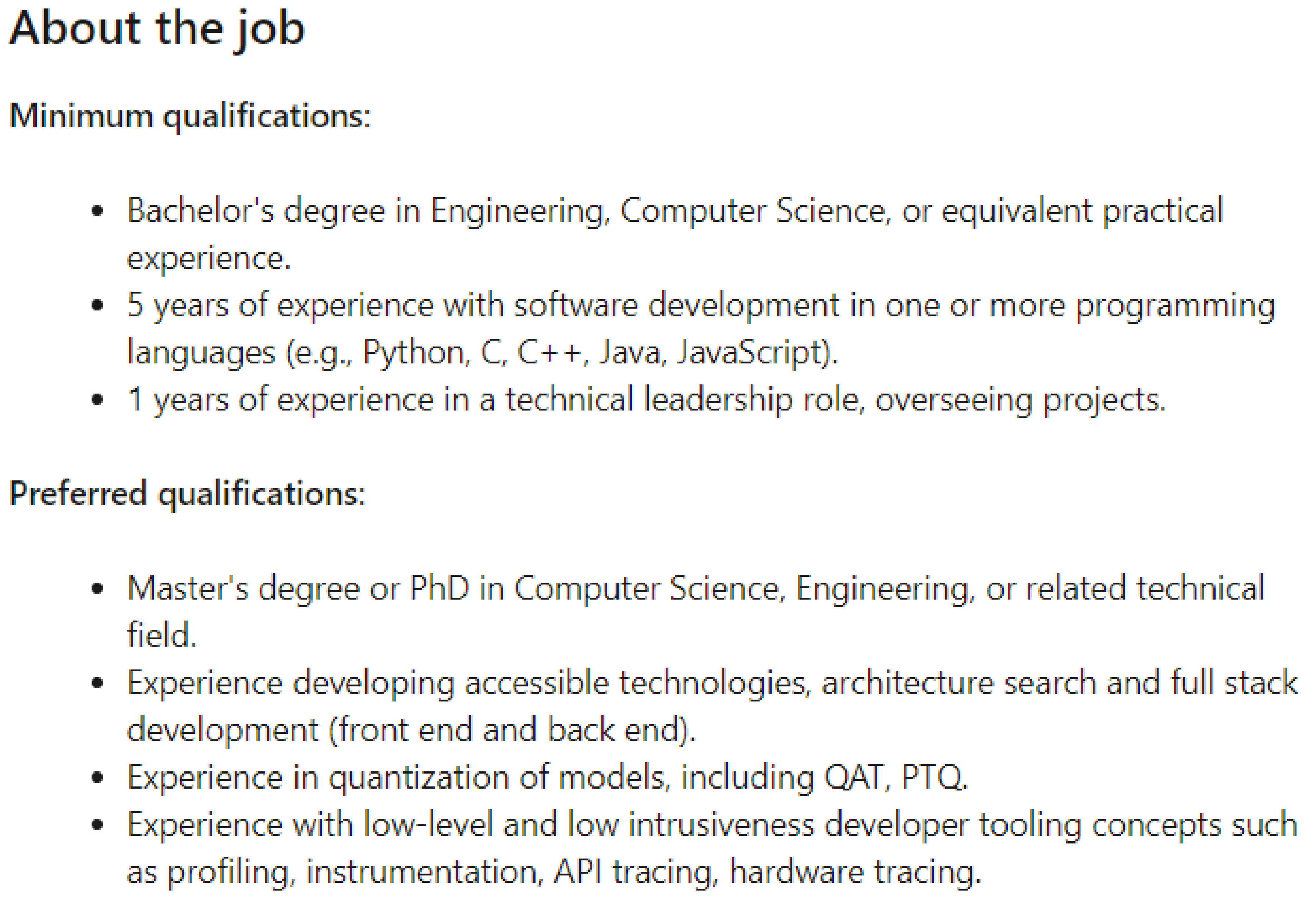
Introduction- SOAP, REST, SOAP Architecture, **REST:** Architecture, Requests and Responses, RESTful APIs, OData, Micro services **Full Stack Development (FSD):** MERN Stack.

Full Stack Development Careers



Software Engineer, Developer Tools, Silicon

Google · Bengaluru, Karnataka, India (On-site) Reposted 2 days ago · 424 applicants



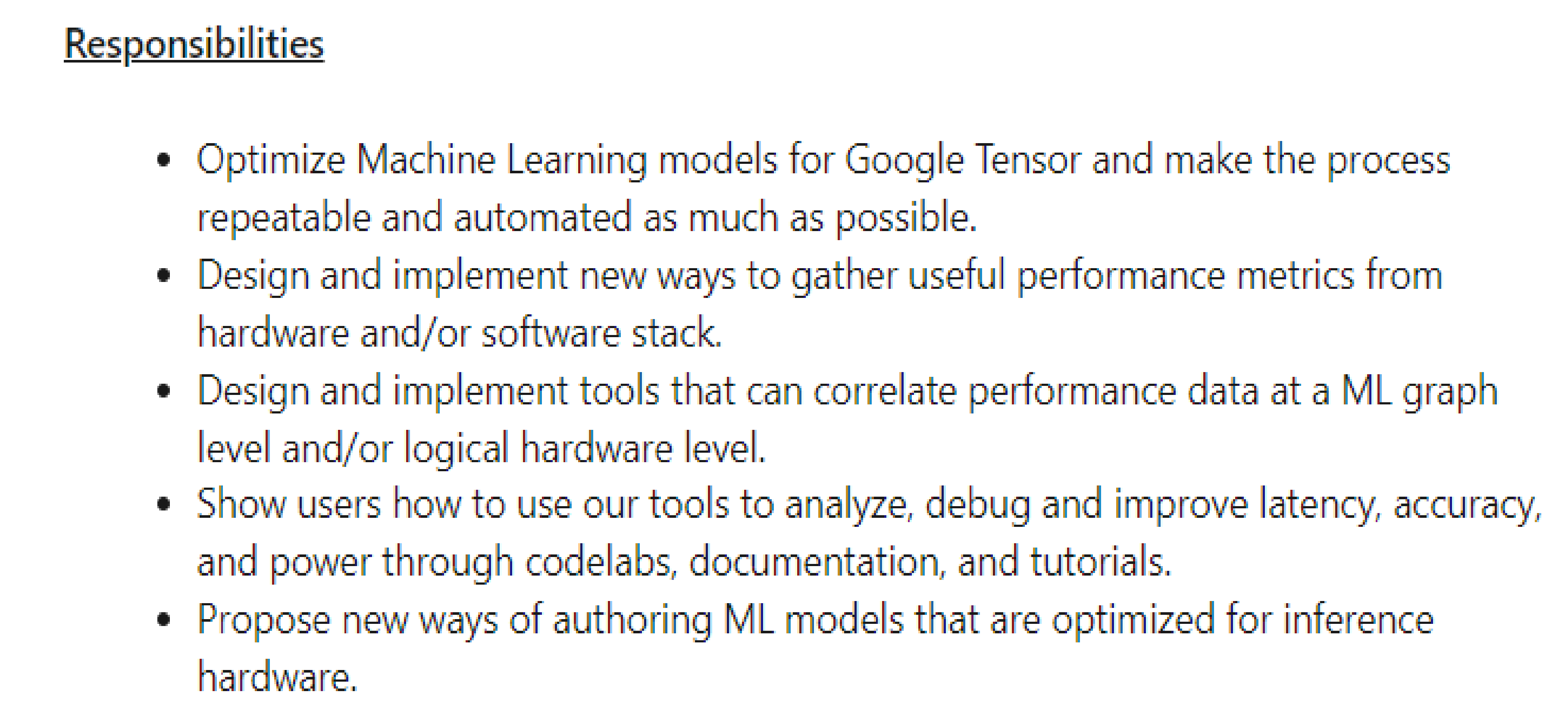
About the job

Minimum qualifications:

- Bachelor's degree in Engineering, Computer Science, or equivalent practical experience.
- 5 years of experience with software development in one or more programming languages (e.g., Python, C, C++, Java, JavaScript).
- 1 years of experience in a technical leadership role, overseeing projects.

Preferred qualifications:

- Master's degree or PhD in Computer Science, Engineering, or related technical field.
- Experience developing accessible technologies, architecture search and full stack development (front end and back end).
- Experience in quantization of models, including QAT, PTQ.
- Experience with low-level and low intrusiveness developer tooling concepts such as profiling, instrumentation, API tracing, hardware tracing.

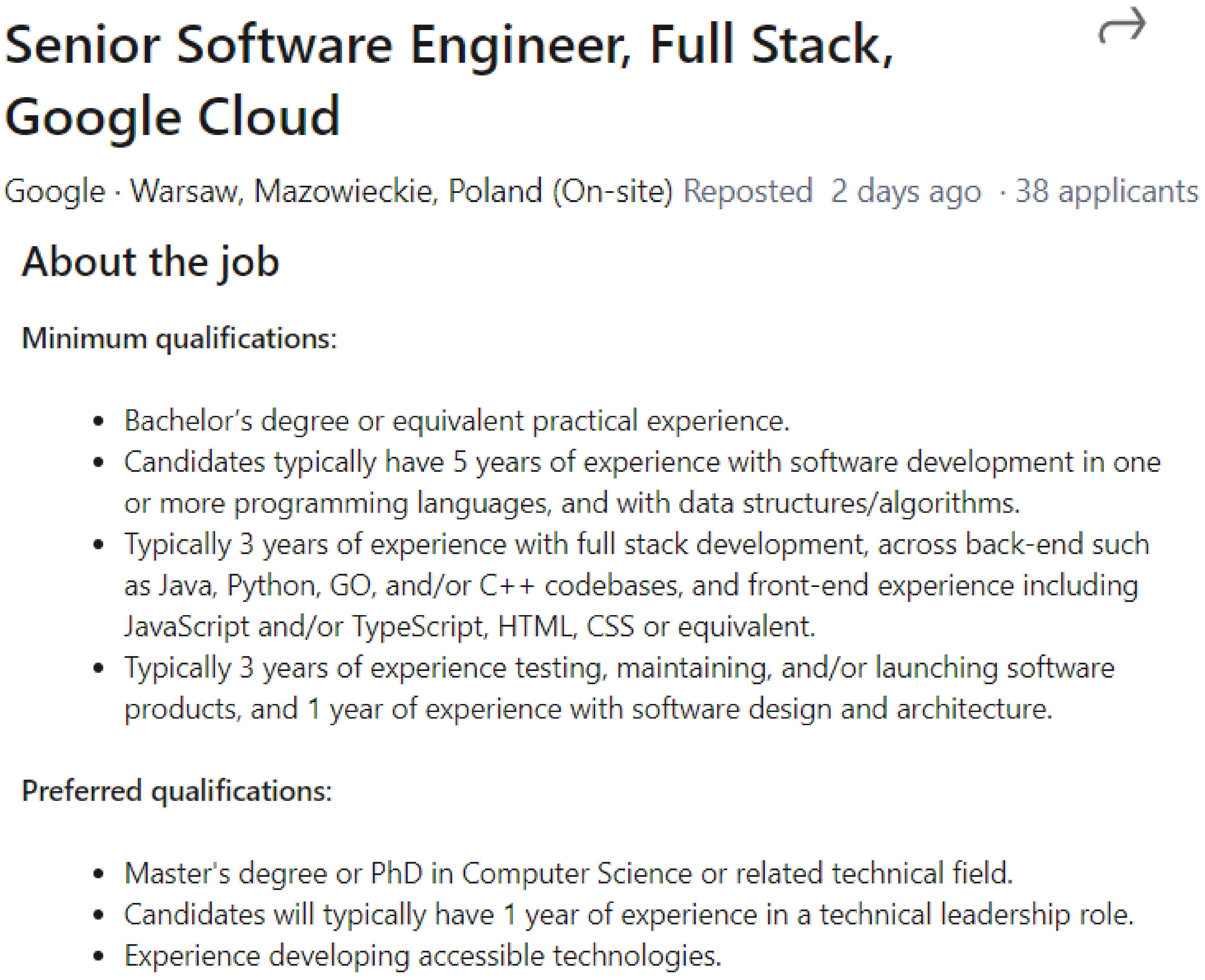


Responsibilities

- Optimize Machine Learning models for Google Tensor and make the process repeatable and automated as much as possible.
- Design and implement new ways to gather useful performance metrics from hardware and/or software stack.
- Design and implement tools that can correlate performance data at a ML graph level and/or logical hardware level.
- Show users how to use our tools to analyze, debug and improve latency, accuracy, and power through codelabs, documentation, and tutorials.
- Propose new ways of authoring ML models that are optimized for inference hardware.

K2

Full Stack Development Careers



Senior Software Engineer, Full Stack, Google Cloud

Google · Warsaw, Mazowieckie, Poland (On-site) Reposted 2 days ago · 38 applicants

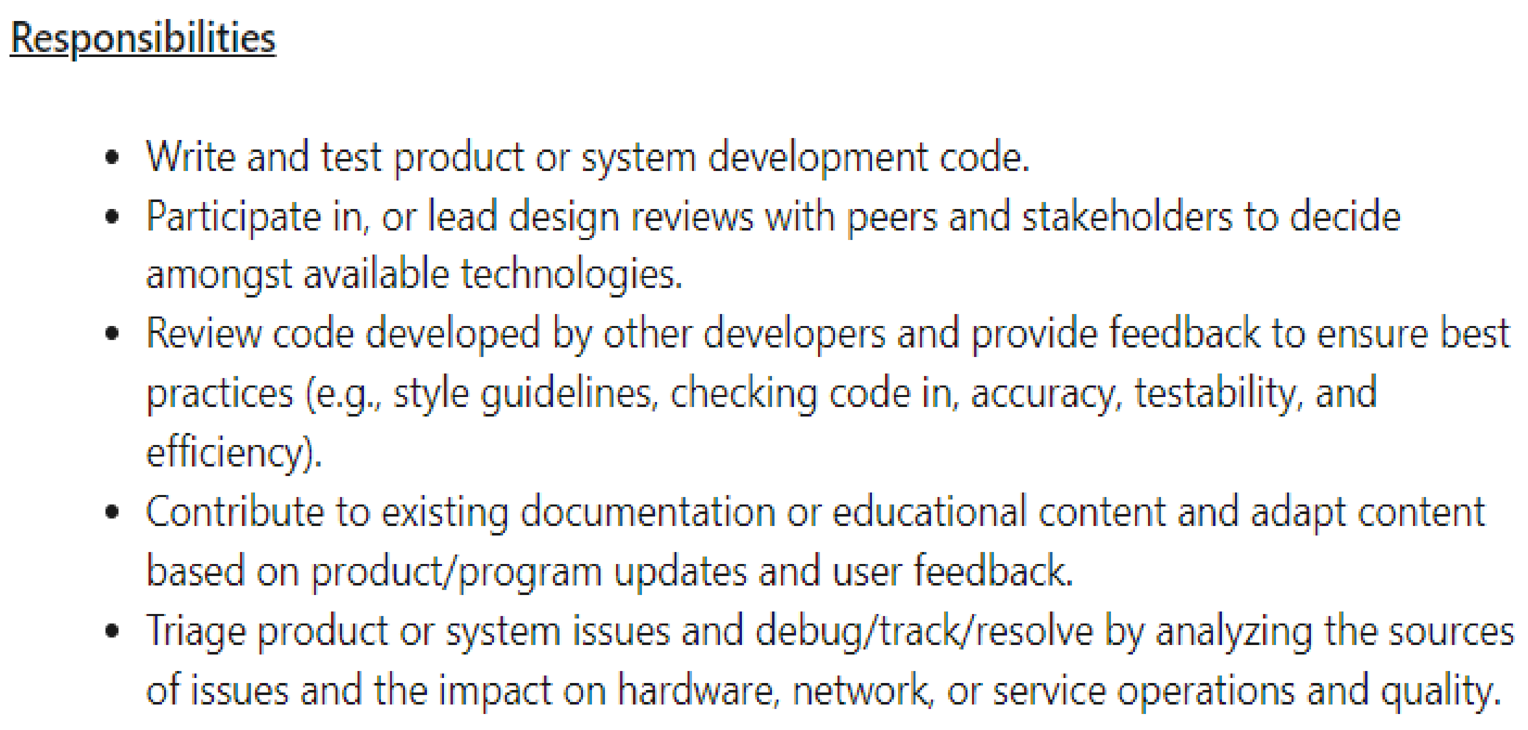
About the job

Minimum qualifications:

- Bachelor's degree or equivalent practical experience.
- Candidates typically have 5 years of experience with software development in one or more programming languages, and with data structures/algorithms.
- Typically 3 years of experience with full stack development, across back-end such as Java, Python, GO, and/or C++ codebases, and front-end experience including JavaScript and/or TypeScript, HTML, CSS or equivalent.
- Typically 3 years of experience testing, maintaining, and/or launching software products, and 1 year of experience with software design and architecture.

Preferred qualifications:

- Master's degree or PhD in Computer Science or related technical field.
- Candidates will typically have 1 year of experience in a technical leadership role.
- Experience developing accessible technologies.



Responsibilities

- Write and test product or system development code.
- Participate in, or lead design reviews with peers and stakeholders to decide amongst available technologies.
- Review code developed by other developers and provide feedback to ensure best practices (e.g., style guidelines, checking code in, accuracy, testability, and efficiency).
- Contribute to existing documentation or educational content and adapt content based on product/program updates and user feedback.
- Triage product or system issues and debug/track/resolve by analyzing the sources of issues and the impact on hardware, network, or service operations and quality.

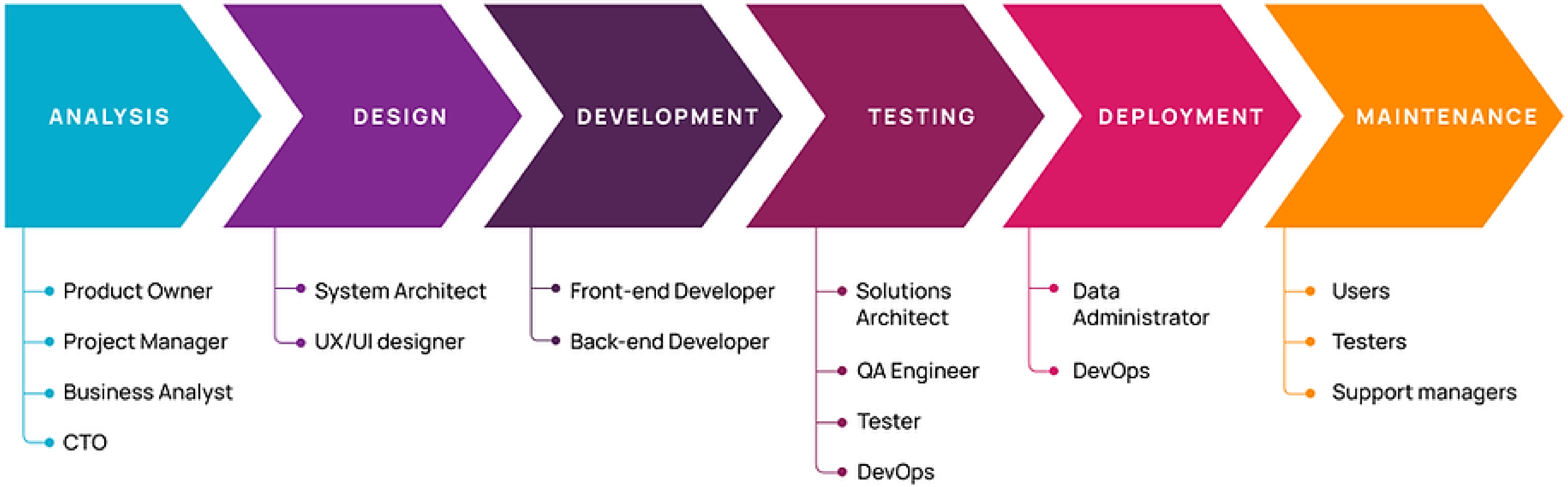
K2

Software Development Life-Cycle?

K2

Software Development Life-Cycle

6 Phases of the Software Development Life Cycle



ANALYSIS
- Product Owner
- Project Manager
- Business Analyst
- CTO

DESIGN
- System Architect
- UX/UI designer

DEVELOPMENT
- Front-end Developer
- Back-end Developer

TESTING
- Solutions Architect
- QA Engineer
- Tester
- DevOps

DEPLOYMENT
- Data Administrator
- DevOps

MAINTENANCE
- Users
- Testers
- Support managers

K2

Questions?

SHIV NADAR
UNIVERSITY
CHENNAI

Introduction – Web Technology

- **Web?**

K2

Introduction – Web Technology

- **Web**
  - **Collection of Machines**
  - **Connected via the Internet**
  - **HTTP**
- **Web Client?**
- **Web Browser?**
- **Web Server?**

K2

Introduction – Web Technology

- **Web**
  - **Collection of Machines**
  - **Connected via the Internet**
  - **HTTP**
- **Web Client**
  - **Machines that Request Information**
- **Web Browser?**
- **Web Server?**

K2

Introduction – Web Technology

- Web
  - Collection of Machines
  - Connected via the Internet
  - HTTP
- Web Client
  - Machines that Request Information
- Web Browser
  - End User Software
  - Render Web Pages
- Web Server?

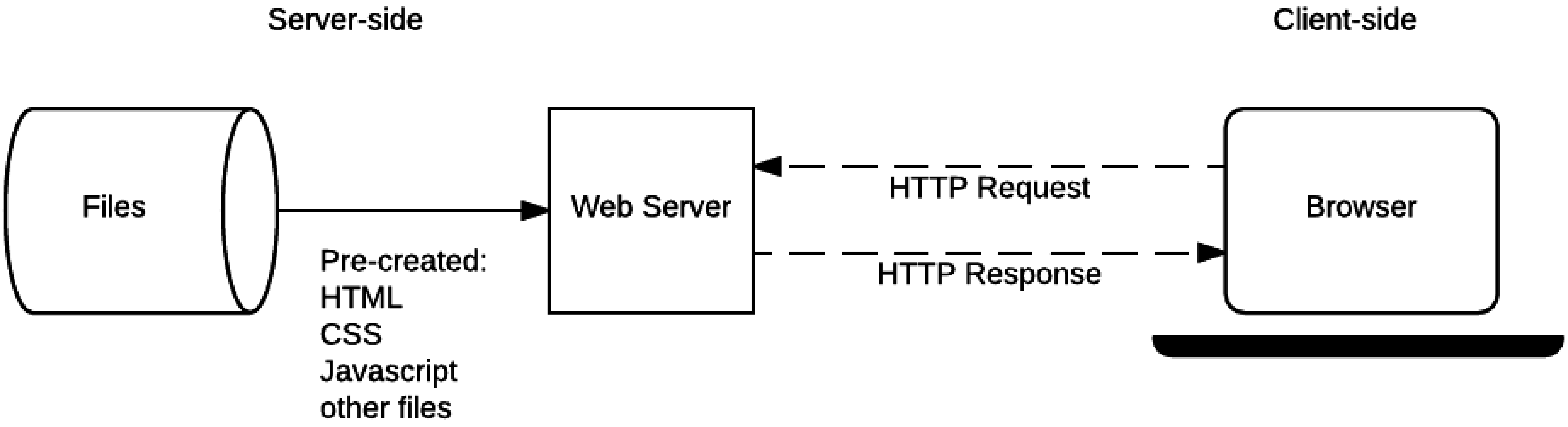
K2

Introduction – Web Technology

- **Web**
  - **Collection of Machines**
  - **Connected via the Internet**
  - **HTTP**
- **Web Client**
  - **Machines that Request Information**
- **Web Browser**
  - **End User Software**
  - **Render Web Pages**
- **Web Server**
  - **Machines that Provide Information**

K2

Introduction – Web Technology



K2

Introduction – Web Technology

- **Web**
  - **Web 1.0**
  - **Web 2.0**
  - **Web 3.0**
  - **Web 4.0**
  - **Web 5.0**

K2

Introduction – Web Technology

- **Web**
  - **Web 1.0 - Read – Only**
  - **Web 2.0 - Social - Read/Write**
  - **Web 3.0 - Semantic – Read/Write/Execute**
  - **Web 4.0 - Mobile**
  - **Web 5.0 - Intelligent/ Emotional Symbiotic**

K2

Introduction – Web Technology

- **Technology?**
- **Technology vs Engineering?**

K2

Introduction – Web Technology

- **Technology**
  - **Focuses more on application.**
  - **Adapts to changes in the industry**
- **Technology vs Engineering?**

K2

Introduction – Web Technology

- **Technology**
  - **Focuses more on application.**
  - **Adapts to changes in the industry**
- **Technology vs Engineering**
  - **Technology – Focus more on Application, Analysis, and Evaluation**
  - **Engineering – Focus more on Design, Analysis, and Evaluation.**

K2

Summary

- **Class Timings**
- **Mark Split-up**
- **Prerequisites**
- **Course Outcomes**
- **Full Stack Career Examples**
- **Software Development Life Cycle**
- **Web**
- **Web Client, Browser, Web Server**
- **Web 1.0 to 5.0**
- **Technology**
- **Technology vs Engineering**

K2

References

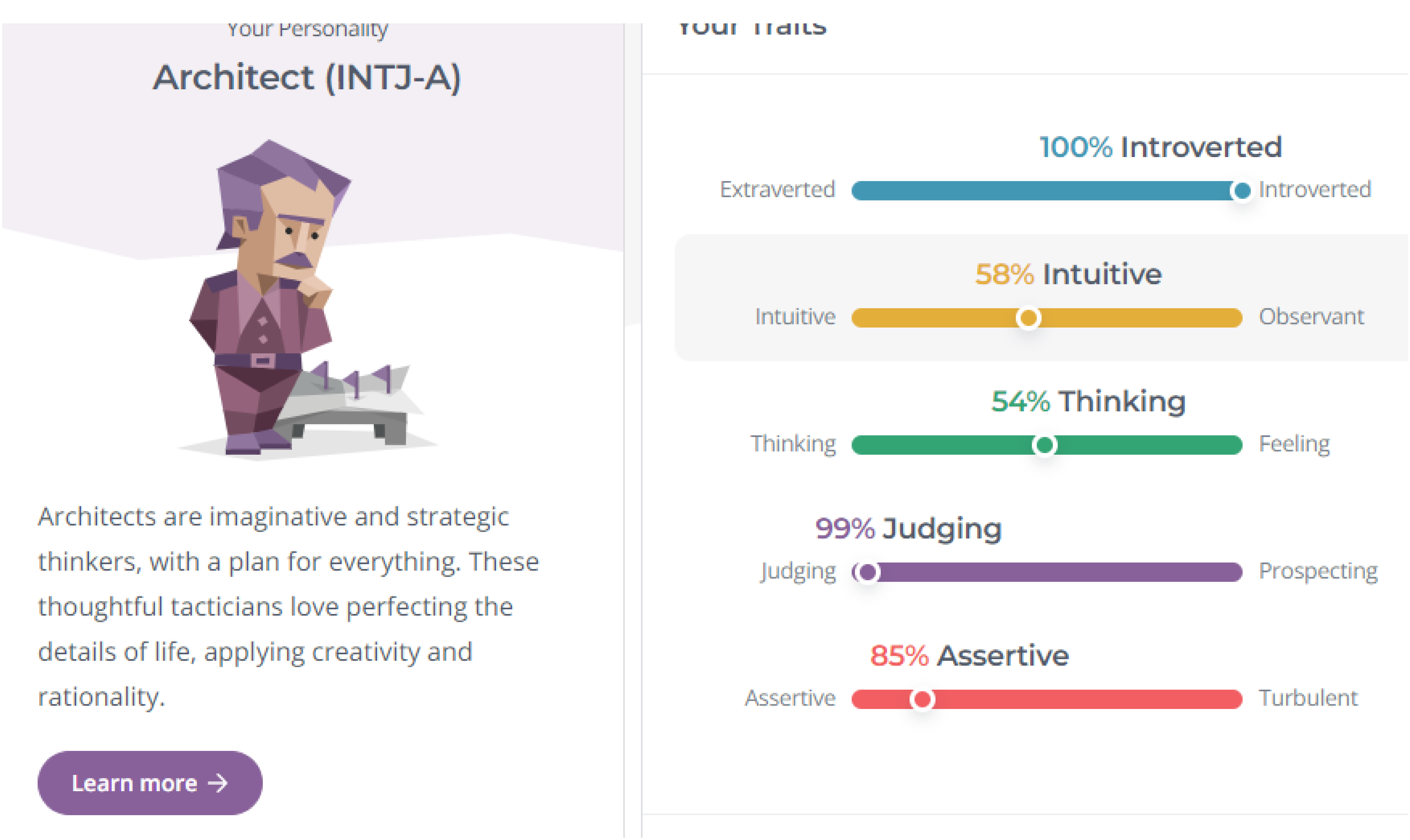
1. **Paul J Deitel, Harvey M Deitel, and Abbey Deitel, "Internet and the world wide web: How to program," Pearson, Fifth Edition, 2012**

THANK YOU

SHIV NADAR
UNIVERSITY
CHENNAI

Lab - 0

1. https://www.16personalities.com/
2. Fill the quiz with as extreme values as possible.
3. Screenshot
    1. Your MBTI Person.
    2. Your Result Overview.
4. Create a HTML document with
    1. Write a heading and a paragraph about your MBTI
    2. Add the two screenshot images
    3. Write a heading and paragraph on the function stack relating to your personality
    4. At the last line of your HTML page,
        1. Add “I am an _____.” Fill the blank with your MBTI (E.g. INTJ). If any of the MBTI letter falls between 40 to 60%, Fill the above blank with an ‘x’. (E.g. INXJ).



Your Personality

Architect (INTJ-A)

Architects are imaginative and strategic thinkers, with a plan for everything. These thoughtful tacticians love perfecting the details of life, applying creativity and rationality.

Learn more →

Your Traits

100% Introverted

Extraverted — Introverted

58% Intuitive

Intuitive — Observant

54% Thinking

Thinking — Feeling

99% Judging

Judging — Prospecting

85% Assertive

Assertive — Turbulent

Web Technology

SHIV NADAR
UNIVERSITY
CHENNAI

Hypertext Markup Language (HTML)

Department of Computer Science and Engineering
Shiv Nadar University

Last Class Summary

- Class Timings
- Mark Split-up
- Prerequisites
- Course Outcomes
- Full Stack Career Examples
- Software Development Life Cycle
- Web
- Web Client, Browser, Web Server
- Web 1.0 to 5.0
- Technology
- Technology vs Engineering

K2

Introduction – Web Technology

- **Client-side Programming**
  - **HTML**
  - **CSS**
  - **Javascript**
- **Server-side Programming**
- **Web Services**

K1

HTML5, CSS3, and Javascript

- **HTML5**
- **CSS3**
- **Javascript**
- **Browsers**

K1

HTML5, CSS3, and Javascript

- **HTML5 - Markup Language**
  - **Content and Structure**
- **CSS3**
- **Javascript**
- **Browsers**

K2

HTML5, CSS3, and Javascript

- **HTML5 - Markup Language**
  - **Content and Structure**
- **CSS3 – Presentation**
  - **Advantage – Swap Styles**
- **Javascript**
- **Browsers**

K2

HTML5, CSS3, and Javascript

- **HTML5 - Markup Language**
  - **Content and Structure**
- **CSS3 – Presentation**
  - **Advantage – Swap Styles**
- **Javascript – Dynamic web pages**
  - **Events**
- **Browsers**

HTML5, CSS3, and Javascript

- **HTML5 - Markup Language**
  - **Content and Structure**
- **CSS3 – Presentation**
  - **Advantage – Swap Styles**
- **Javascript – Dynamic web pages**
  - **Events**
- **Browsers**
  - **Chrome, Firefox, Safari, Opera, Microsoft Internet Explorer 9/ Edge**

K2

HTML

- **HTML**
  - **Hypertext Markup Language**
  - **Hypertext**

K2

HTML

- **HTML**
    - **Hypertext Markup Language**
        - **Hypertext**
            - **Highlighted Links**
            - **Easy Navigation within or Across Webpages**
        - **Markup**

K2

HTML

- **HTML**
    - **Hypertext Markup Language**
        - **Hypertext**
            - **Highlighted Links**
            - **Easy Navigation within or Across Webpages**
        - **Markup**
            - **Mark Text (Bold, Italic, Underline)**
        - **Language**

K2

HTML

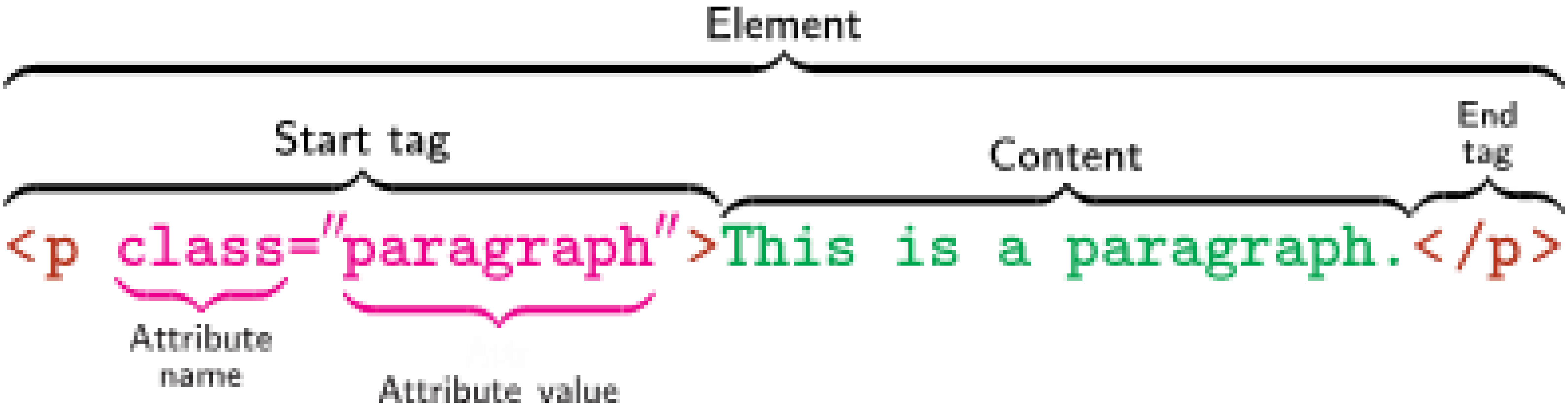
- **Language**

	English	C	HTML
Alphabet	A-Z, a-z	A-Z, a-z	A-Z, a-z
Numbers	0-9	0-9, 0 and 1, 0-7, 0-F	0-9
Words	Words	Tokens	Nodes
Sentences	Grammar + Words = Sentences	Syntax + Tokens = Statements	Syntax + Nodes = Elements
Paragraph	Paragraph	Block	
Chapter/Book	Chapter/Book	Program	Web Page
Library	Library	Library	Website

Questions?

SHIV NADAR
UNIVERSITY
CHENNAI

HTML - Element



```
<p class="paragraph">This is a paragraph.</p>
```

K2

HTML - Tags

- **Keywords enclosed in angular brackets (<>)**
- **Total Tags in HTML 5.2 - 142 tags**
- **115 Tags are compactible across HTML versions**
- **Forgetting to close tags – Syntax error**
- **Tags are not case sensitive**
- **Most tags have <tag_name> content <\tag_name>**
- **Best Practice 1 – Use HTML tags with the same case (preferably lower case)**

K1

HTML – Tag Attributes

- **Keywords present in Tags**
- **Specific set of values**
- **Additional Characteristics to Elements**
- **Eg: height, width for an image element**
- **Not mandatory for most tags**
- **Found in start tag**

Best Practices

1. **Use HTML tags with the same case (preferably lower case)**

K1

Summary

- **Overview – Client-side Programming**
- **HTML, CSS, Javascript, Browsers**
- **Hypertext**
- **Markup**
- **Language**
- **HTML Elements**
- **HTML Tags**
- **HTML Attributes**
- **Best Practices**

K1

References

1. **Paul J Deitel, Harvey M Deitel, and Abbey Deitel, "Internet and the world wide web: How to program," Pearson, Fifth Edition, 2012**

THANK YOU

SHIV NADAR
UNIVERSITY
CHENNAI

Web Technology

SHIV NADAR
UNIVERSITY
CHENNAI

HTML - Element

Department of Computer Science and Engineering
Shiv Nadar University

Last Class Summary

- **Overview – Client-side Programming**
- **HTML, CSS, Javascript, Browsers**
- **Hypertext**
- **Markup**
- **Language**
- **HTML Elements**
- **HTML Tags**
- **HTML Attributes**
- **Best Practices**

K1

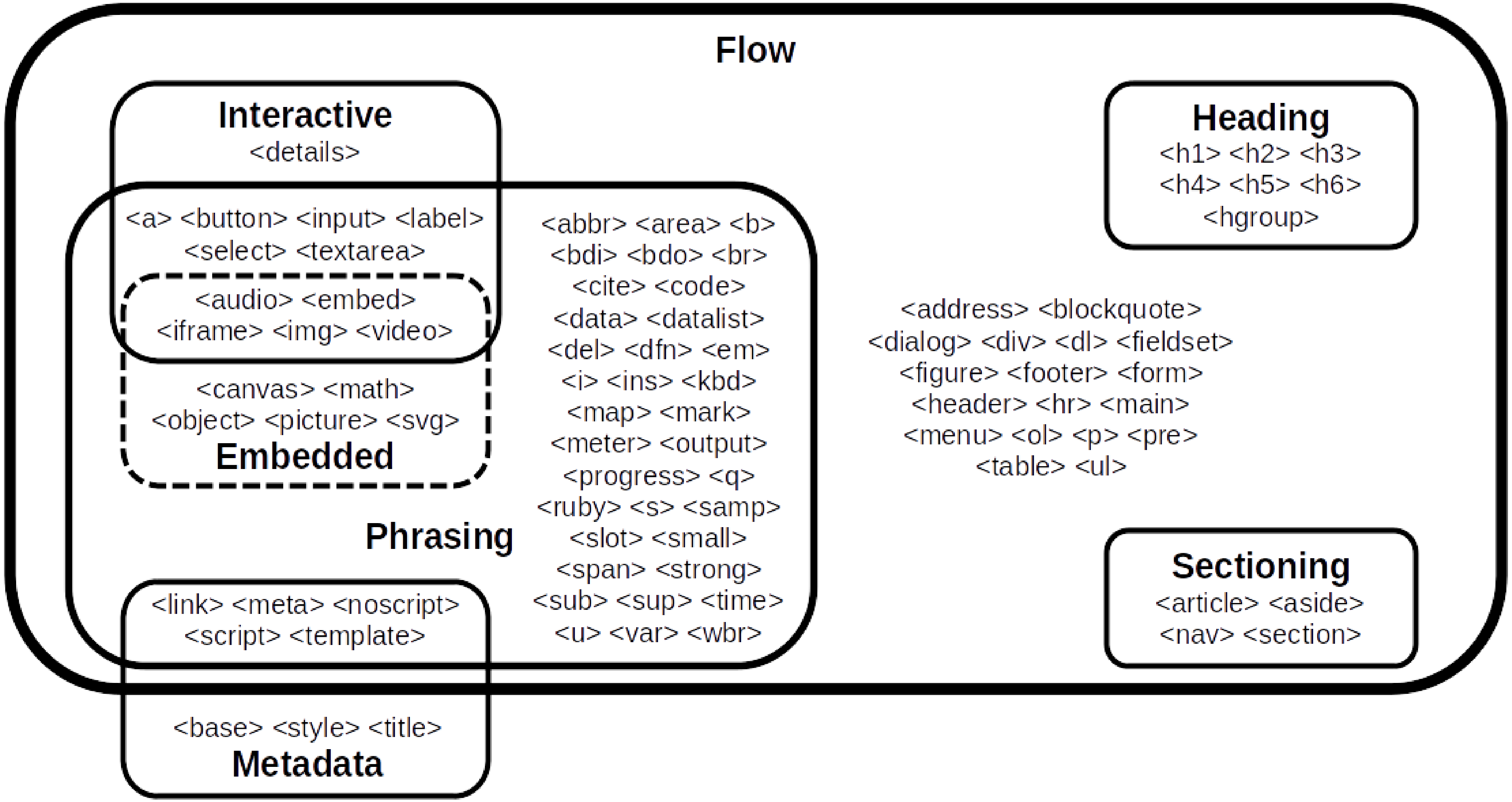
HTML Editing

- **HTML Files – Text Editor**
- **File extension - .html or .htm**
- **Best Practice - Filename to be based on functionality**
  - **Eg: Main web page (Homepage) - index.html**
- **HTML errors – Not fatal – They do not crash the program**

K2

HTML Elements

HTML element content categories



Flow

Interactive

<details>

<a> <button> <input> <label>
<select> <textarea>

<audio> <embed>
<iframe> <img> <video>

<canvas> <math>
<object> <picture> <svg>

Embedded

Phrasing

<abbr> <area> <b>
<bdi> <bdo>
<cite> <code>
<data> <datalist>
<del> <dfn> <em>
<i> <ins> <kbd>
<map> <mark>
<meter> <output>
<progress> <q>
<ruby> <s> <samp>
<slot> <small>
<span> <strong>
<sub> <sup> <time>
<u> <var> <wbr>

<link> <meta> <noscript>
<script> <template>

<base> <style> <title>

Metadata

Heading

<h1> <h2> <h3>
<h4> <h5> <h6>
<hgroup>

<address> <blockquote>
<dialog> <div> <dl> <fieldset>
<figure> <footer> <form>
<header> <hr> <main>
<menu> <ol> <p> <pre>
<table> <ul>

Sectioning

<article> <aside>
<nav> <section>

<body> <caption> <col> <colgroup> <dd> <dt> <figcaption> <head>
<html> <legend> <li> <optgroup> <option> <rp> <rt> <source>
<summary> <tbody> <td> <tfoot> <th> <thead> <tr> <track>

K1

HTML !DOCTYPE

- **Syntax for HTML5**

<!DOCTYPE html>

- **Indicates browser – HTML file**
- **Syntax may vary on version**
- **Empty element**
  - **No content**
  - **No ending tag**

HTML <html>

- **Syntax**

```
<html>
      content
</html>
```

- **<html> - root element**
- **Contains all other elements – except !DOCTYPE**

K2

HTML <head>

- **<head> element – between <html> and <body>**
- **<head> - metadata**
- **Information generally not rendered**
- **Metadata tags -** <title>, <style>, <meta>, <link>, <script>, <base>
- **Best Practice – When opening a tag, have a "tab space" for each line until you close the tag. It helps in the readability of your HTML document.**

K2

HTML <title>

- **<title> - Set the title of the webpage.**
- **Generally used in every HTML document.**
- **Must be placed inside <head>.**
- **One <title> per webpage.**

K2

HTML <body>, <p>, comments

- **<body> - document content**
- **Text, Paragraph, Formatting, Image, Hyperlinks, Tables, Lists, Frames**

- **Comment syntax - <!-- content -->**
- **Best Practice - Give a comment describing a web page as a first line of your HTML document.**

- **<p> - Paragraph**
- **Browser renders a new line before and after <p> element**

K2

HTML Text Formatting

- **<strong> or <b>** - Bold text
- **<u>** - Underline
- **<em> or <i>** - Emphasized text (Italic)
- **<mark>** - Marked text
- **<small>** - Smaller text
- **<del>** - Deleted text represented as strikethrough
- **<ins>** - Inserted text represented as underline
- **<sub>** - Subscript text
- **<sup>** - Superscript text

- **<i>** and **<b>** are deprecated

K2

HTML Headers

- Default text formatting
- **<h1> to <h6> elements**
- **h1** – Largest
- **h6** - Smallest
- Browser renders and decides the size
- Can vary significantly between browsers

K2

HTML Hyperlinks

- **<a> element (Anchor)**
- **Mandatory attribute - href**
- **Syntax:**
  - **<a href="address">content</a>**
- **All links – underlined**
- **An unvisited link – blue**
- **A visited link – purple**
- **An active link - red**

HTML Internal Linking

- **Linking to a location within a webpage**

1. **Set location to go**
   - **Syntax: <a name=“value”></a>**
2. **Refer using href as usual.**
   - **Syntax: <a href=“#value”>content</a>**

K2

HTML Images, Special Character, <hr>, <br>

- <img> element
- Mandatory attribute - src

- Special characters like math characters
  - Added in code form
  - Syntax: &code;

- <br> element
- Line break.

- <hr> element
- Adds a horizontal line/rule
- Both <hr> and <br> - No closing tag

K2

HTML Lists

- **List - <li> element – Closing tag optional**
- **Unordered List - <ul> element - Bullets**
- **Ordered List - <ol> element – Numbers**
- **Each <ul> or <ol> elements consist of ‘n’ number of <li> elements**
- **<ul> and <ol> elements can be nested**
- **Newline after every closed list**

K2

HTML Tables

- <table> - table element
  - <caption> - caption element
  - <tr> - row element
    - <td> - data element
  - <colgroup> - column group element – styling group of columns
    - <col> - column to be styled
  - <thead> - table header element
    - <th> - table head element
  - <tbody> - table body element
    - <td> - data element
  - <tfoot> - table foot element
    - <td> - data element

K2

HTML iFrames

- **Group multiple HTML files**
- **<iframe> element – inline frame**
- **Styling – Attributes/ CSS files**
- **Syntax:**

```
<iframe src=“HTML file" title=“Title for the HTML file"></iframe>
```

K2

HTML Example

SHIV NADAR UNIVERSITY CHENNAI

K3

Best Practices

1. **Use HTML tags with the same case (preferably lower case).**
2. **Filename to be based on functionality.**
3. **When opening a tag, have a "tab space" for each line until you close the tag. It helps in the readability of your HTML document.**
4. **Give a comment describing a web page as a first line of your HTML document.**

Summary

- **Editing HTML files**
- **List of HTML Element**
- **!DOCTYPE HTML**
- **<html>, <head>, <title>, <body>, <p>, Comment Elements**
- **Text Formatting**
- **HTML Headers, and Hyperlinks**
- **HTML Images, Special Character, <hr>, <br>**
- **HTML Lists, Tables, iFrames**
- **Best practices**

K1

References

1. **Paul J Deitel, Harvey M Deitel, and Abbey Deitel, "Internet and the world wide web: How to program," Pearson, Fifth Edition, 2012.**

THANK YOU

SHIV NADAR UNIVERSITY CHENNAI

Web Technology

SHIV NADAR
UNIVERSITY
CHENNAI

CSS – Cascading Style Sheet

Department of Computer Science and Engineering
Shiv Nadar University

Last Class Summary

- **Editing HTML files**
- **List of HTML Element**
- **!DOCTYPE HTML**
- **<html>, <head>, <title>, <body>, <p>, Comment Elements**
- **Text Formatting**
- **HTML Headers, and Hyperlinks**
- **HTML Images, Special Character, <hr>, <br>**
- **HTML Lists, Tables, iFrames**
- **Best practices**

K1

CSS Introduction

- **CSS – Cascading Style Sheets**
- **CSS - Presentation**
- **Separate Design elements from Logic**
- **Cascade – Change one style to another**

- **Three ways**
  - **Inline - HTML tag attributes**
  - **Internal - HTML <style> element**
  - **External - CSS separate files.**

K2

CSS Introduction

Every keystroke counts!

- Smaller files load more quickly
- Save disk space
- Example *some.html*

–**Original:** 27.2K

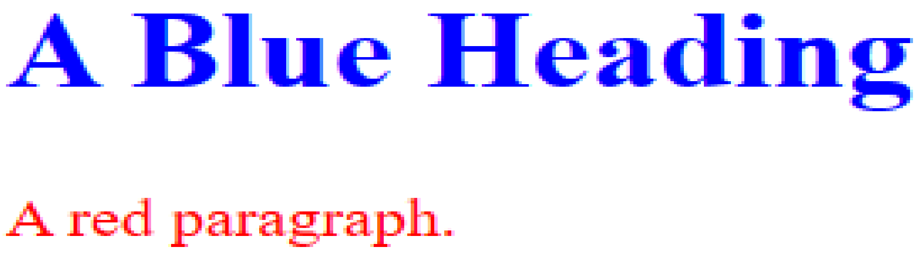
– **Embedded Style Sheet:** 26.2K

– **External Style Sheet:** 25.6K

K2

Inline - CSS Tag Attributes

- **Style attribute in tags**
- **Applied in start tag**



A Blue Heading

A red paragraph.

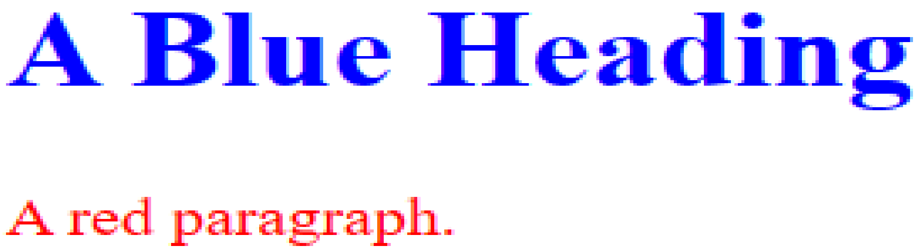
K2

Inline - CSS Tag Attributes

- **Style attribute in tags**
- **Applied in start tag**

```
<h1 style="color:blue;">A Blue Heading</h1>

<p style="color:red;">A red paragraph.</p>
```



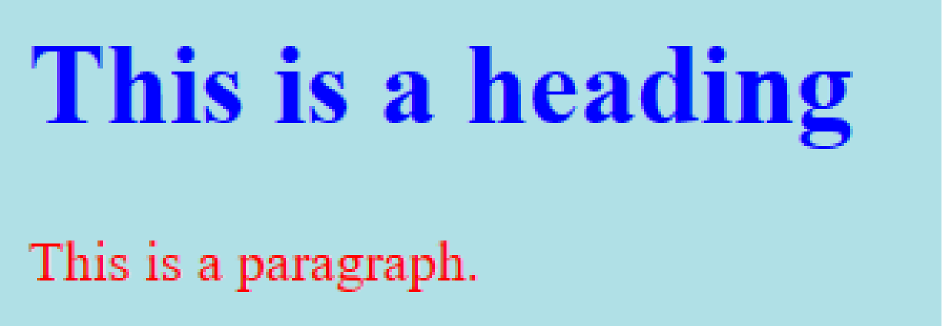
A Blue Heading

A red paragraph.

K3

Internal - CSS Style Element

- **<style> element.**
- **Defined in <head> element**



This is a heading

This is a paragraph.

K2

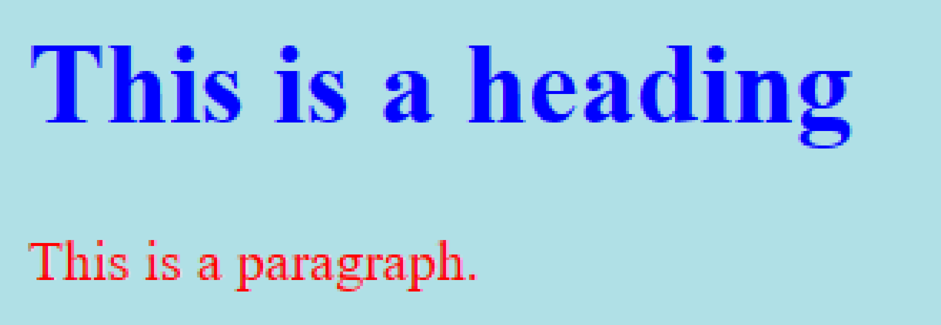
Internal - CSS Style Element

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>

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

</body>
</html>
```

- **Whole – selector string**
- **body – selector**
- **background-color – property names**
- **background-color: powderblue; - declaration**
- **{} – declaration block**



This is a heading

This is a paragraph.

K3

Internal - CSS Style Element

- **Language**

	English	CSS	HTML
Alphabet	A-Z, a-z	A-Z, a-z	A-Z, a-z
Numbers	0-9	0-9, 0-F	0-9, 0-F
Words	Words	Tokens	Nodes
Sentences	Grammar + Words = Sentences	Syntax + Tokens = Selector String	Syntax + Nodes = Elements
Paragraph	Paragraph		
Chapter/Book	Chapter/Book		Web Page
Library	Library		Website

K2

External - CSS Separate Files

- **Separate style file ending with .css**
- **Add using <link> element**
- **rel and href – mandatory attributes**
- **Other attributes – title and type**
- **Defined in <head> element**

K2

External - CSS Separate Files

```css
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

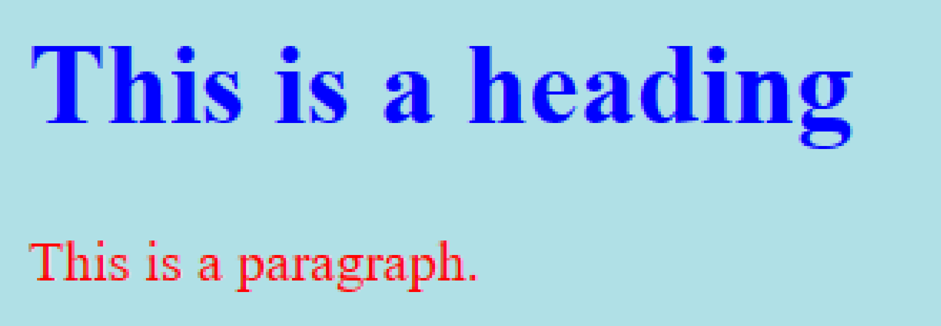
K2

External - CSS Separate Files

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href=“sample.css">
</head>
<body>

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

</body>
</html>
```



This is a heading

This is a paragraph.

K3

Multiple Style Sheets

- **Inline, Internal, External Combination at different sequence**
  - **The latest style will be applied.**
- **Inline, Internal, External Combination on a same element**
  - **Inline**
  - **Internal and External (Whichever first)**
  - **Default**

K2

Selector Strings

- **Single element type**

  Syntax:

  - **element_name{ property_name: value, ..}**
- **Multiple element type**

  Syntax:

  - **element_name1, element_name2, .. , element_name_n, { property_name: value, ..}**
- **All element type**

  Syntax:

  ***{ property_name: value, ..}**
- **Elements by id**

  Syntax:

  - **#id_name{ property_name: value, ..}**

K2

Selector Strings

- **Single element type**

  Syntax:

  - **.class_name{ property_name: value, ..}**

- **Multiple element type**

  Syntax:

  - **element_ name.class_name{ property_name: value, ..}**

- **CSS Comments**

  Syntax:

  - **/* */**

K2

CSS Property

- **Total of more than 200 property names**
- **https://www.dofactory.com/css/properties**

- **<div> element – division/ section in HTML**
- **Browser renders a new line before and after a <div> element**
- **Can style any content.**
- **<p> element – style text**

K2

Best Practices

1. **Use HTML tags with the same case (preferably lowercase).**
2. **Filename to be based on functionality.**
3. **When opening a tag, have a "tab space" for each line until you close the tag. It helps in the readability of your HTML document.**
4. **Give a comment describing a web page as the first line of your HTML document.**
5. **Keep HTML and CSS files separate**

K1

Summary

- **CSS Introduction**
- **Inline style**
- **Internal Style**
- **External Style**
- **Multiple Style**
- **Selector Strings**
- **CSS Property**
- **Best practices**

References

1. **Paul J Deitel, Harvey M Deitel, and Abbey Deitel, "Internet and the world wide web: How to program," Pearson, Fifth Edition, 2012.**

THANK YOU

SHIV NADAR
UNIVERSITY
CHENNAI

Web Technology

SHIV NADAR
UNIVERSITY
CHENNAI

Javascript

Department of Computer Science and Engineering
Shiv Nadar University

Last Class Summary

- **CSS Introduction**
- **Inline style**
- **Internal Style**
- **External Style**
- **Multiple Style**
- **Selector Strings**
- **CSS Property**
- **Best practices**

K1

Introduction

- **Most Popular Programming Language**
- **Webpages – Dynamic and Interactive**
- **Javascript?**

K2

Introduction

- **Most Popular Programming Language**
- **Webpages – Dynamic and Interactive**
- **Javascript**
  - **Java?**

K2

Introduction

- **Most Popular Programming Language**
- **Webpages – Dynamic and Interactive**
- **Javascript**
  - **Java**
    - **Indonesia – Java Coffee Bean**
  - **Javascript is different from Java**
    - **Why – Javascript?**

K2

Introduction

- **Most Popular Programming Language**
- **Webpages – Dynamic and Interactive**
- **Javascript**
  - **Java**
    - **Indonesia – Java Coffee Bean**
  - **Javascript is different from Java**
    - **Javascript**
      - **Borrowed Concepts from Java**
        - **Object Oriented Programming**
      - **Java was popular (steal some popularity)**
      - **Netscape/Mosaic browser supported Java Applets**

K2

Introduction

- **Most Popular Programming Language**
- **Webpages – Dynamic and Interactive**
- **Javascript**
- **Language?**

K2

Introduction

Language

	English	Javascript	HTML
Alphabet	A-Z, a-z	A-Z, a-z	A-Z, a-z
Numbers	0-9	0-9, 0 and 1, 0-7, 0-F	0-9
Words	Words	Tokens	Nodes
Sentences	Grammar + Words = Sentences	Syntax + Tokens = Statements	Syntax + Nodes = Elements
Paragraph	Paragraph	Block	
Chapter/Book	Chapter/Book	Program	Web Page
Library	Library	Library	Website

Introduction

- **Javascript – History**
    - **Reading Assignment -** [JavaScript – Wikipedia]

K2

Introduction

- **Javascript – History**
  - **Reading Assignment -** JavaScript – Wikipedia
- **High-Level Language**

K2

Introduction

- **Javascript – History**
  - **Reading Assignment -** JavaScript – Wikipedia
- **High-Level Language**
- **European Computer Manufacturer's Association (ECMA) script Standard**

K2

Introduction

- **Javascript – History**
    - **Reading Assignment -** JavaScript – Wikipedia
- **High-Level Language**
- **European Computer Manufacturer's Association (ECMA) script Standard**
- **Dynamically Typed**

K2

Introduction

- **Javascript – History**
  - **Reading Assignment -** JavaScript – Wikipedia
- **High-Level Language**
- **European Computer Manufacturer's Association (ECMA) script Standard**
- **Dynamically Typed**
- **Object Oriented Programming Language**

K2

Introduction

- **Javascript – History**
  - **Reading Assignment -** JavaScript – Wikipedia
- **High-Level Language**
- **European Computer Manufacturer's Association (ECMA) script Standard**
- **Dynamically Typed**
- **Object Oriented Programming Language**
- **Interpreted/ Just-in-time Compiled**

K2

Introduction

- **Javascript – History**
  - **Reading Assignment -** JavaScript – Wikipedia
- **High-Level Language**
- **European Computer Manufacturer's Association (ECMA) script Standard**
- **Dynamically Typed**
- **Object Oriented Programming Language**
- **Interpreted/ Just-in-time Compiled**

- **Types of Programming?**

Introduction

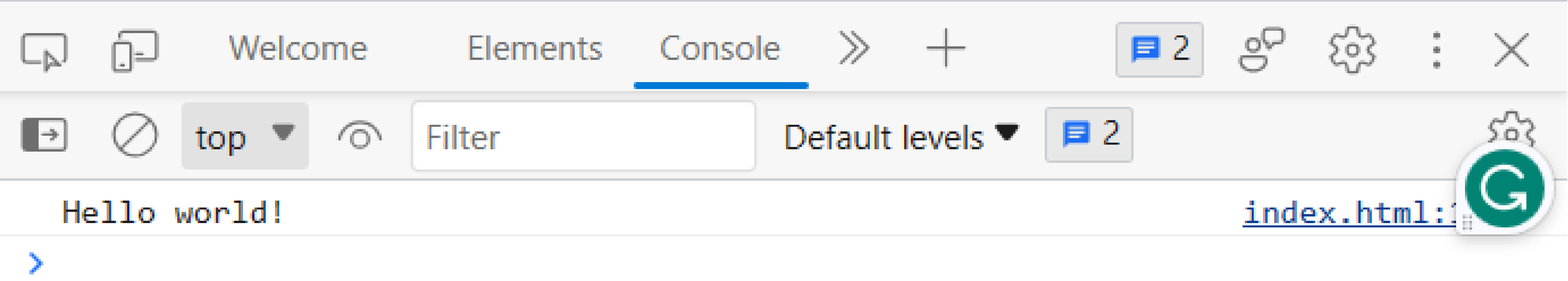
- **Javascript – History**
  - **Reading Assignment -** JavaScript – Wikipedia
- **High-Level Language**
- **European Computer Manufacturer's Association (ECMA) script Standard**
- **Dynamically Typed**
- **Object Oriented Programming Language**
- **Interpreted/ Just-in-time Compiled**

- **Types of Programming**
  - **Inline Programming**
  - **External Programming**
- **<script> tag**

K2

Introduction – Example 1

- **The Hello World Program - Inline Programming**
- **<script> element**
- **Print in Console (Syntax:** console.log (“”); **)**



```
Hello world!                                                index.html:1
>
```

K2

Introduction – Example 1

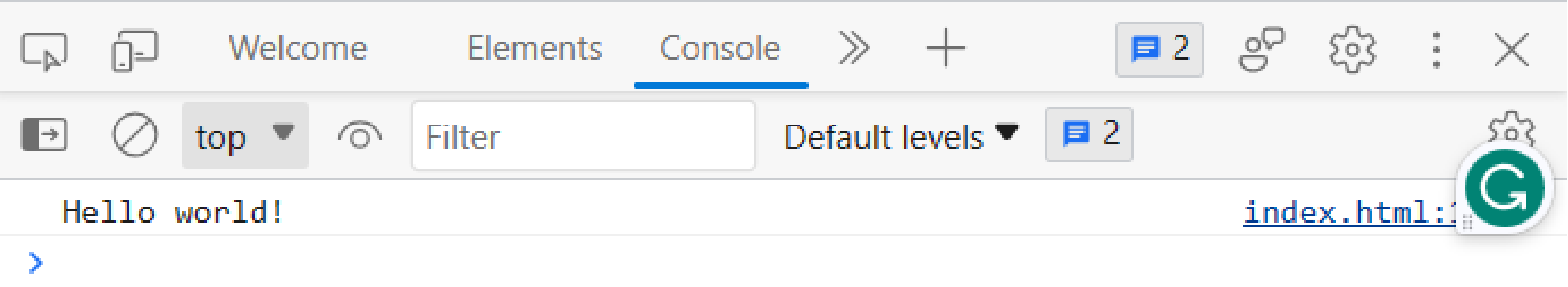
- **The Hello World Program - Inline Programming**

```
<!DOCType html>
<html>
    <head>
        <title> Inline Javascript </title>
    </head>
    <body>
        <script>
            console.log("Hello world!");
        </script>
    </body>
</html>
```

K3

Introduction – Example 2

- **The Hello World Program - External Programming (save as .js)**
- **<script> element. Include the Javascript file in src attribute of <script> element.**
- **Print in Console (Syntax:** `console.log (“”);` **)**

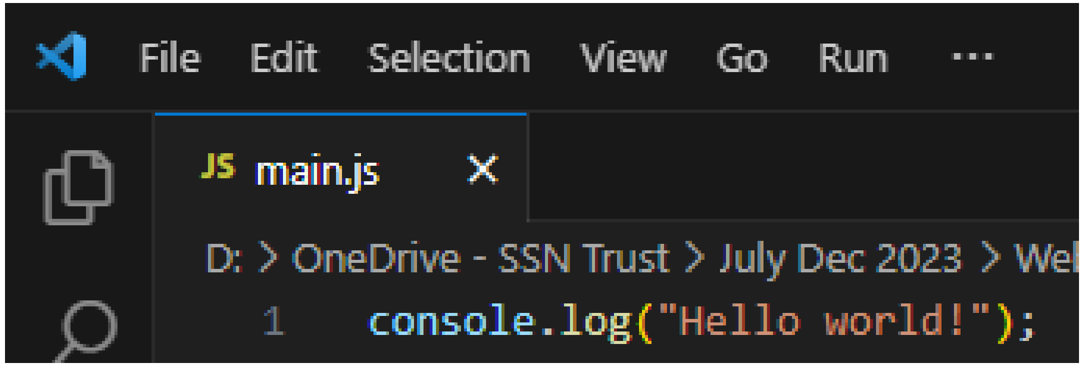


```
Hello world!                                                    index.html:1
```

Introduction – Example 2

- **The Hello World Program - External Programming**
- **Script content - empty**

```
<!DOCType html>
<html>
    <head>
        <title> External Javascript </title>
    </head>
    <body>
        <script src="main.js">
        </script>
    </body>
</html>
```

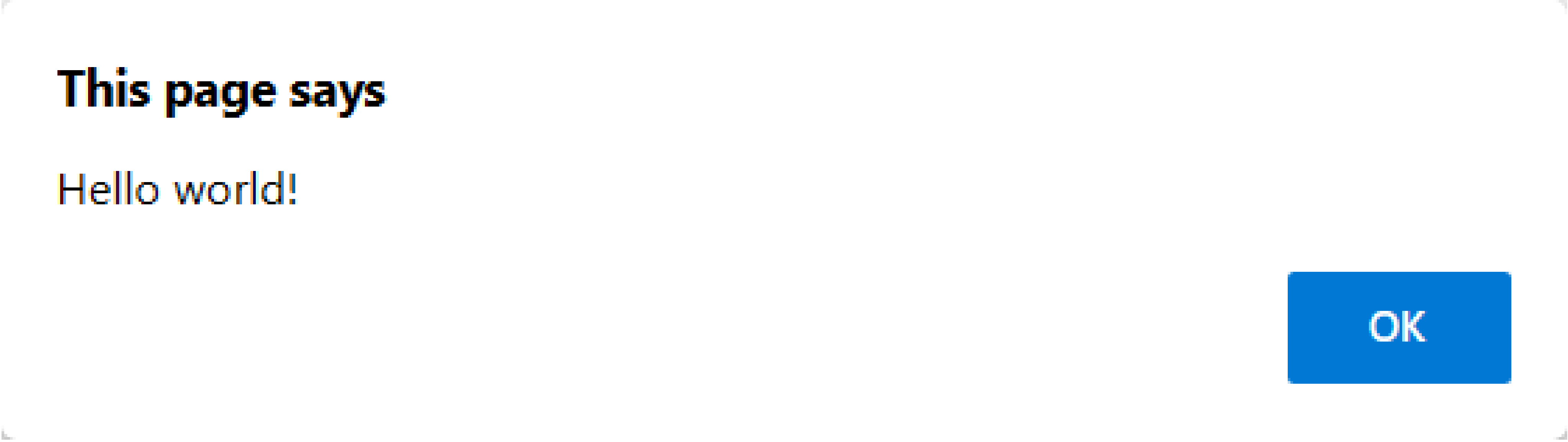


```
console.log("Hello world!");
```

K3

Introduction – Example 3

- **The Hello World Program - External Programming (save as .js)**
- **Pop an alert box. (Syntax:** `alert(“”);` **)**



This page says

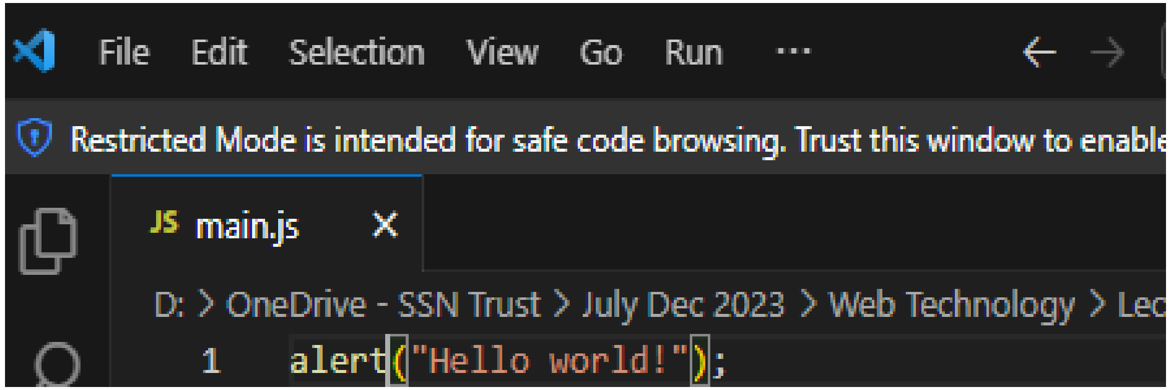
Hello world!

K2

Introduction – Example 3

- **The Hello World Program - External Programming**
- **Script content - empty**

```
<!DOCType html>
<html>
    <head>
        <title> External Javascript </title>
    </head>
    <body>
        <script src="main.js">
        </script>
    </body>
</html>
```

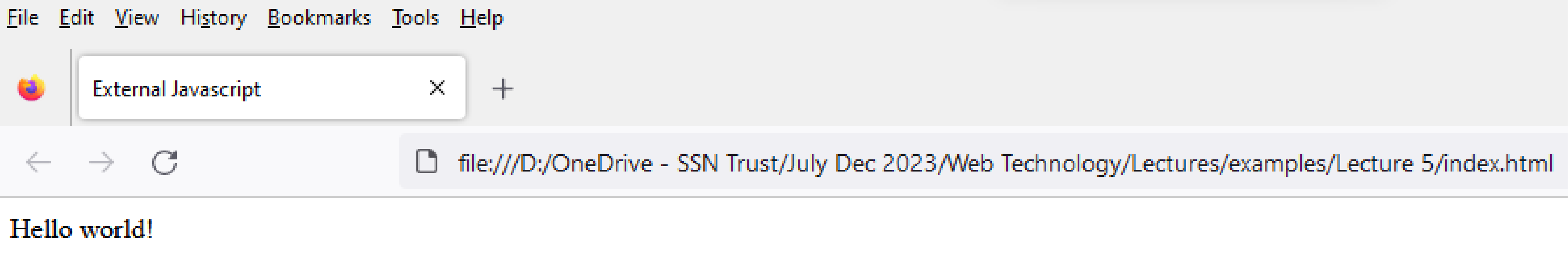

File
Edit
Selection
View
Go
Run
...
Restricted Mode is intended for safe code browsing. Trust this window to enable
main.js
D: > OneDrive - SSN Trust > July Dec 2023 > Web Technology > Lec

```
alert("Hello world!");
```

K3

Introduction – Example 4

- **The Hello World Program - External Programming (save as .js)**
- **Pop an alert box. (Syntax:** `document.write(“”);` **)**


File
Edit
View
History
Bookmarks
Tools
Help
External Javascript
file:///D:/OneDrive - SSN Trust/July Dec 2023/Web Technology/Lectures/examples/Lecture 5/index.html

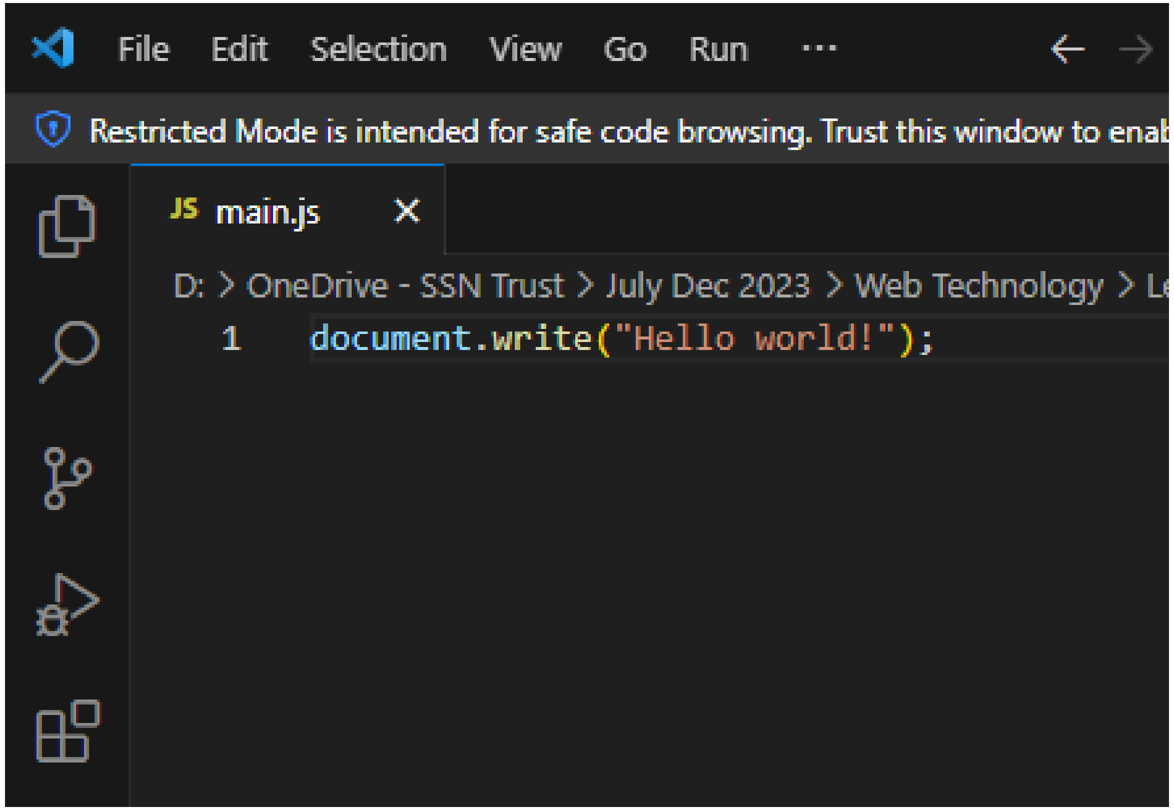
```
Hello world!
```

K2

Introduction – Example 4

- **The Hello World Program - External Programming**
- **Script content - empty**

```
<!DOCType html>
<html>
    <head>
        <title> External Javascript
</title>
    </head>
    <body>
        <script src="main.js">
        </script>
    </body>
</html>
```


File
Edit
Selection
View
Go
Run
...
main.js
D: > OneDrive - SSN Trust > July Dec 2023 > Web Technology > Le

Restricted Mode is intended for safe code browsing. Trust this window to enab

```
document.write("Hello world!");
```

K3

Best Practices

1. Use HTML tags with the same case (preferably lowercase).
2. Filename to be based on functionality.
3. When opening a tag, have a "tab space" for each line until you close the tag. It helps in the readability of your HTML document.
4. Give a comment describing a web page as the first line of your HTML document.
5. Keep HTML, CSS, and Javascript files separate

K1

Summary

- **Javascript Statistics**
- **Java vs Javascript**
- **Language**
- **Introduction**
- **Types of Programming**
- **Inline – Hello World - Inline**
- **External – Hello World – Inline**
- **External – Hello World - Alert**
- **Keywords**
- **Best Practices**

K1

References

1. **Paul J Deitel, Harvey M Deitel, and Abbey Deitel, "Internet and the world wide web: How to program," Pearson, Fifth Edition, 2012.**

THANK YOU

SHIV NADAR
UNIVERSITY
CHENNAI

Web Technology

SHIV NADAR UNIVERSITY CHENNAI

DOM, Manipulation and Event Handling

Department of Computer Science and Engineering
Shiv Nadar University

Last Class Summary

- **Keywords Updated**
- **Form Validation Introduction**
- **HTML Form Validation**
- **Introduction DOM**
- **Javascript Form Validation**
- **Best Practices**

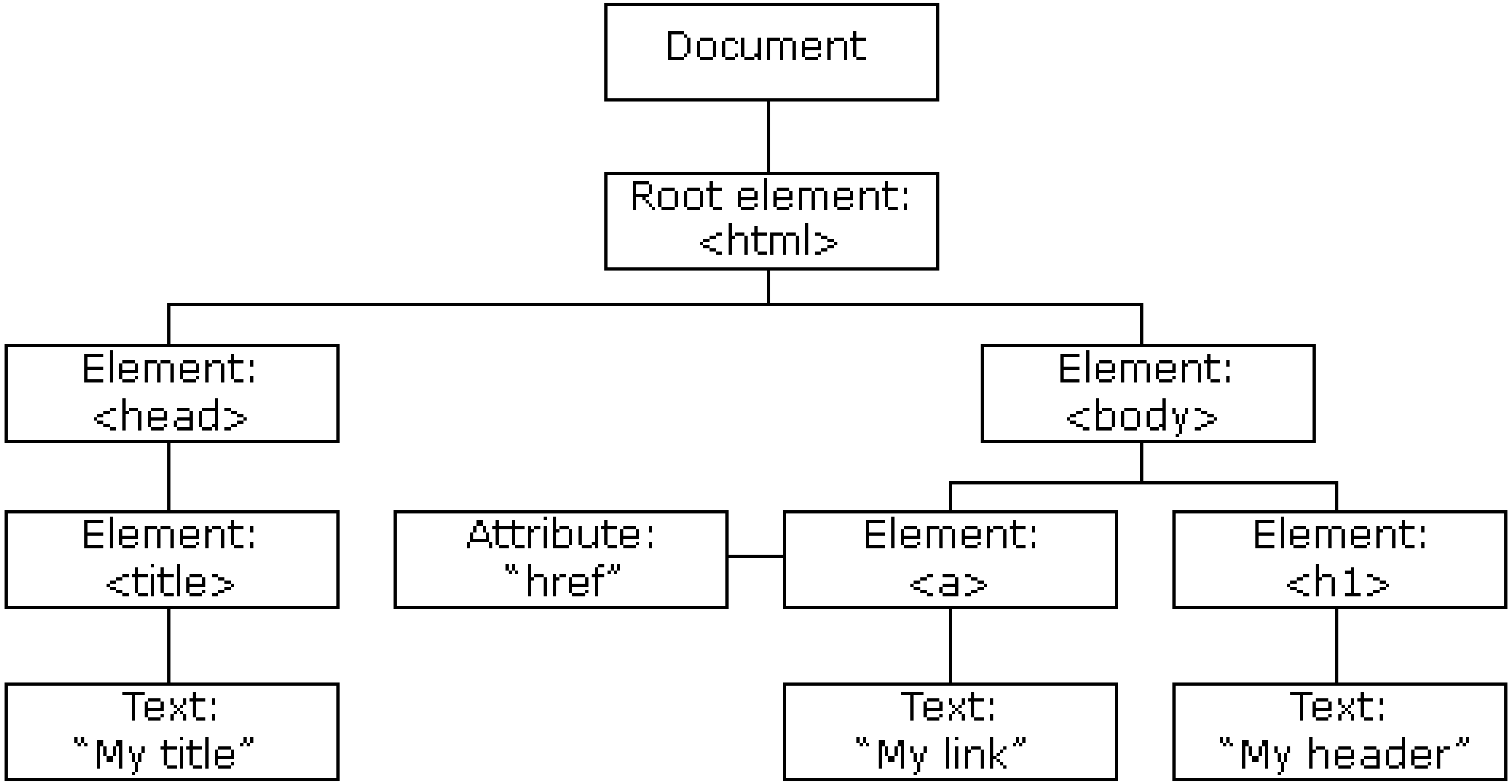
K1

Keywords - 48

var	let	const	typeof	if	else
switch	for	do	while	in	of
continue	break	null	true	false	function
return	new	delete	this		

K1

Introduction to Document Object Model (DOM)



K2

DOM

- **DOM – Standard for Accessing Documents**

K1

DOM

- **DOM – Standard for Accessing Documents**
- **Three Parts**
  - **Core DOM**
  - **XML DOM**
  - **HTML DOM**

DOM

- **DOM – Standard for Accessing Documents**
- **Three Parts**
  - **Core DOM**
  - **XML DOM**
  - **HTML DOM**
- **HTML DOM - get, change, add, or delete HTML elements.**

K2

DOM

- **DOM – Standard for Accessing Documents**
- **Three Parts**
  - **Core DOM**
  - **XML DOM**
  - **HTML DOM**
- **HTML DOM - get, change, add, or delete HTML elements.**
  - **Dynamic HTML**

K2

DOM

- **DOM – Standard for Accessing Documents**
- **Three Parts**
  - **Core DOM**
  - **XML DOM**
  - **HTML DOM**
- **HTML DOM - get, change, add, or delete HTML elements.**
  - **Dynamic HTML**
- **HTML elements – Objects?**

K2

DOM

- **DOM – Standard for Accessing Documents**
- **Three Parts**
  - **Core DOM**
  - **XML DOM**
  - **HTML DOM**
- **HTML DOM - get, change, add, or delete HTML elements.**
  - **Dynamic HTML**
- **HTML elements – Objects**
  - **Properties**
  - **Methods**

K2

DOM

- DOM – Standard for Accessing Documents
- Three Parts
  - Core DOM
  - XML DOM
  - HTML DOM
- HTML DOM - get, change, add, or delete HTML elements.
  - Dynamic HTML
- HTML elements – Objects
  - Properties
  - Methods
- Events for HTML
- Javascripts

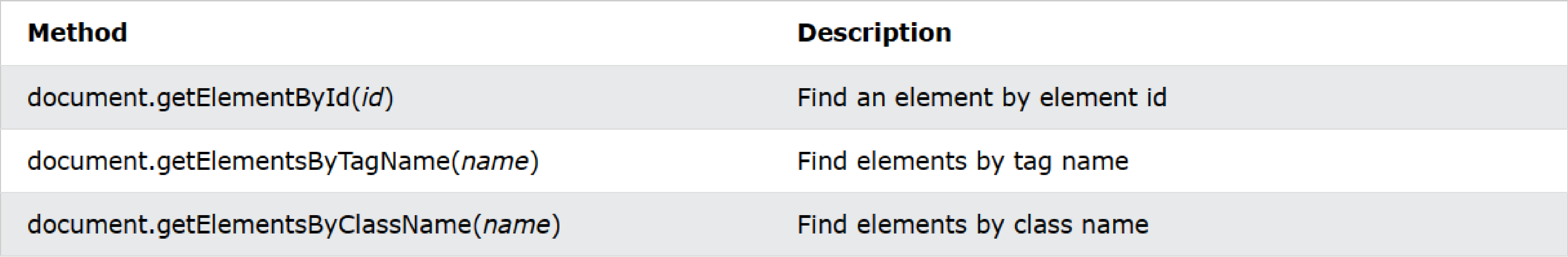
K2

DOM - DHTML

- **Add new HTML elements, attributes, and CSS styles**
- **Change HTML elements, attributes, and CSS styles**
- **Remove existing HTML elements, attributes, and CSS styles**
- **React to HTML events**
- **Create new HTML events**
- **Remove existing HTML events**

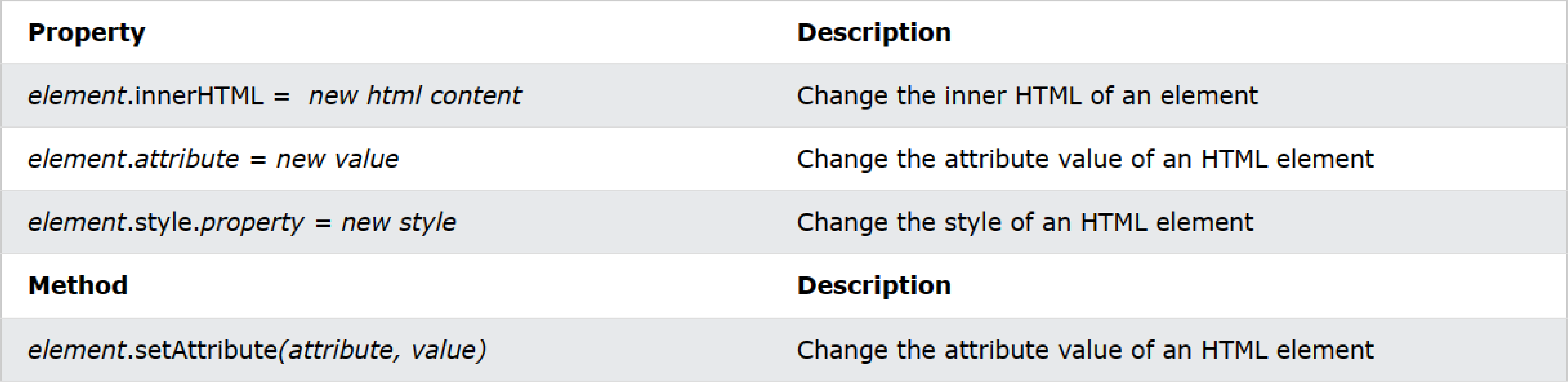
K1

DOM – Finding HTML Elements



Method	Description
document.getElementById(*id*)	Find an element by element id
document.getElementsByTagName(*name*)	Find elements by tag name
document.getElementsByClassName(*name*)	Find elements by class name

DOM – Changing HTML Elements



Property	Description
element.innerHTML = *new html content*	Change the inner HTML of an element
element.attribute = *new value*	Change the attribute value of an HTML element
element.style.*property* = *new style*	Change the style of an HTML element
Method	**Description**
element.setAttribute*(attribute, value)*	Change the attribute value of an HTML element

K2

DOM – Tree Traversal

- **Traverse 3 directions**
  - **Upward**
  - **Downward**
  - **Sideways**

K2

DOM – Tree Traversal

- **Traverse 3 directions**
  - **Upward**
  - **Downward**
  - **Sideways**
- **document.querySelector("element or attribute");**
- **document.querySelectorAll("element or attribute");**

K2

DOM – Tree Traversal

- **Traverse 3 directions**
  - **Upward**
  - **Downward**
  - **Sideways**
- **document.querySelector(“element or attribute”);**
  - **First element or attribute**
  - **Eg: document.querySelector(“p”);**
- **document.querySelectorAll(“element or attribute”);**

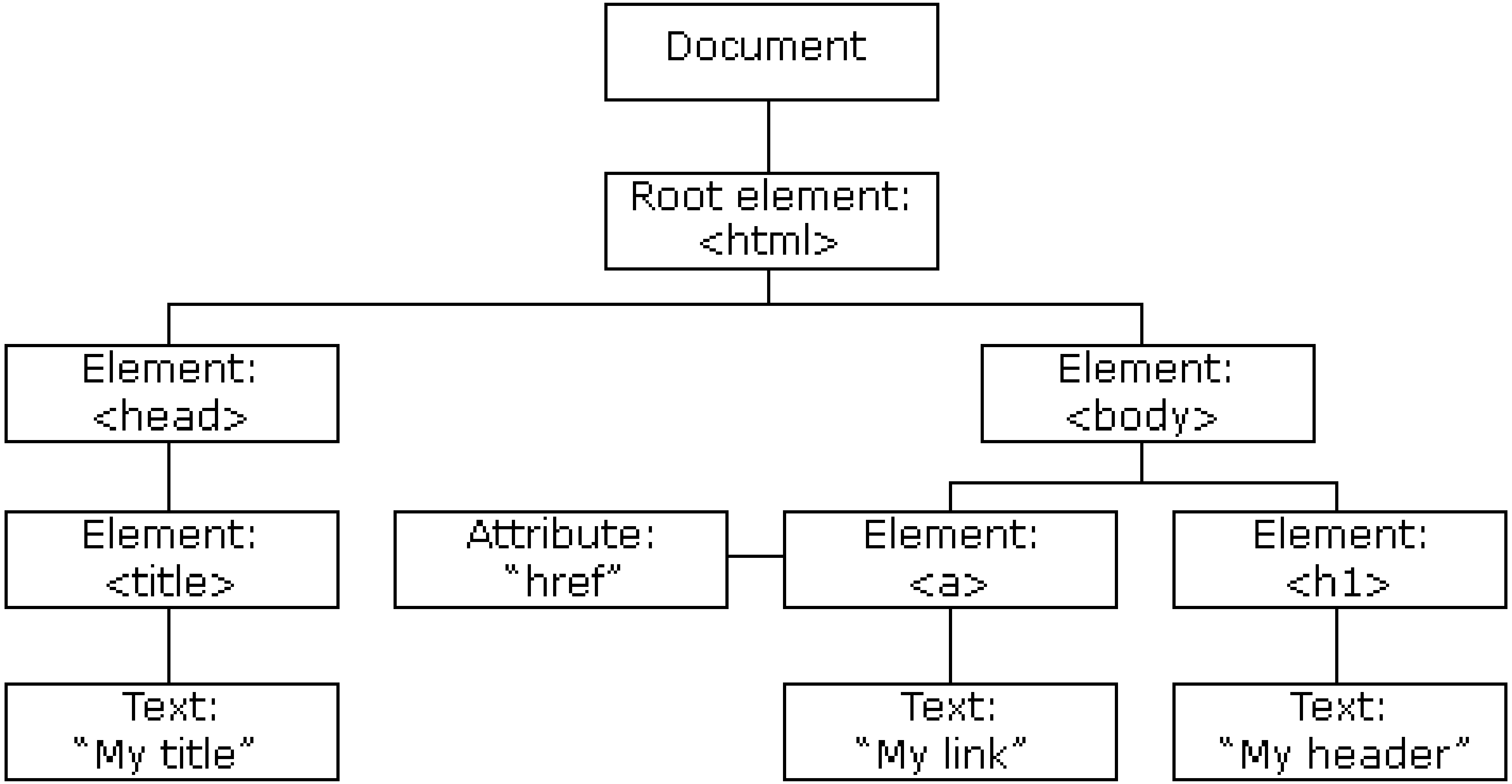
K2

DOM – Tree Traversal

- Traverse 3 directions
  - Upward
  - Downward
  - Sideways
- document.querySelector(“element or attribute”);
  - First element or attribute
  - Eg: document.querySelector(“p”);
- document.querySelectorAll(“element or attribute”);
  - Selects all elements or attributes
  - Returns a NodeList.
  - Eg: document.querySelectorAll(“p”);

K2

Introduction to Document Object Model (DOM)



K2

DOM – Downward Traversal

- **firstElementChild**
- **children**
- **lastElementChild**
- **childNodes, firstChild, lastChild**

DOM – Upward Traversal

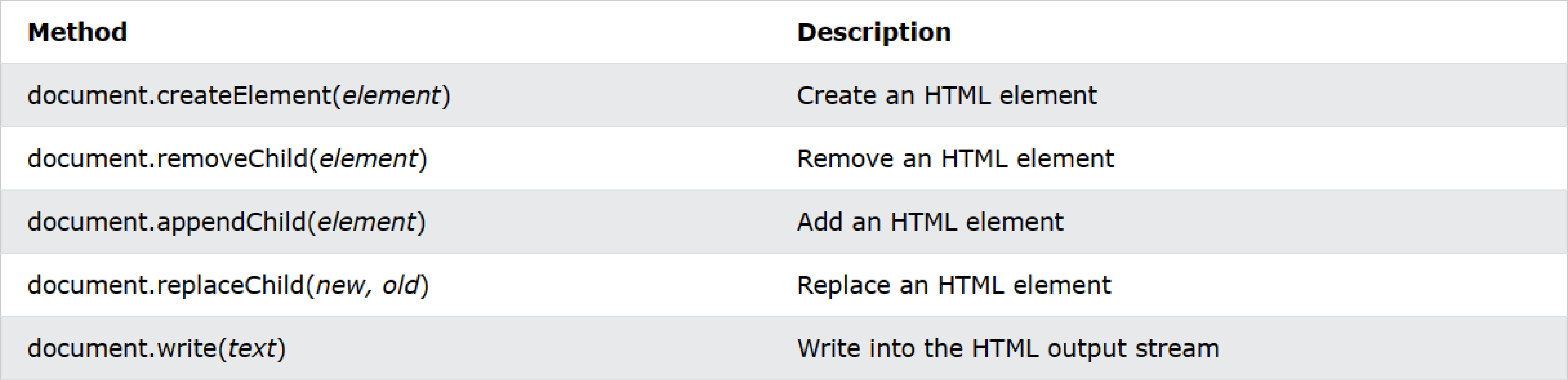
- **parentElement**
- **parentNode**

DOM – Upward Traversal

- **nextElementSibling, previousElementSibling**
- **nextSibling, previousSibling**

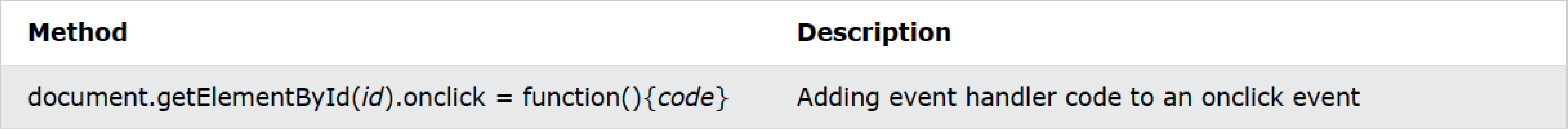
K2

DOM – Adding and Deleting Elements



Method	Description
document.createElement(*element*)	Create an HTML element
document.removeChild(*element*)	Remove an HTML element
document.appendChild(*element*)	Add an HTML element
document.replaceChild(*new, old*)	Replace an HTML element
document.write(*text*)	Write into the HTML output stream

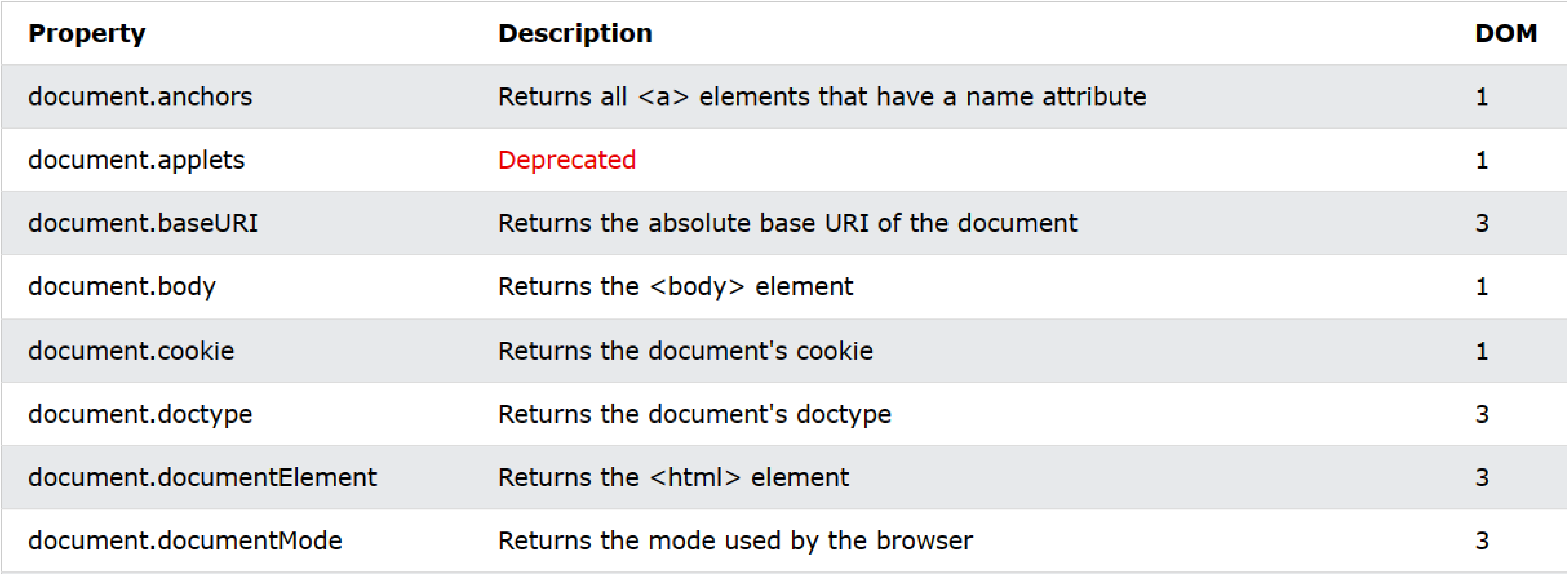
DOM –HTML Event Handling



Method	Description
document.getElementById(*id*).onclick = function(){*code*}	Adding event handler code to an onclick event

K2

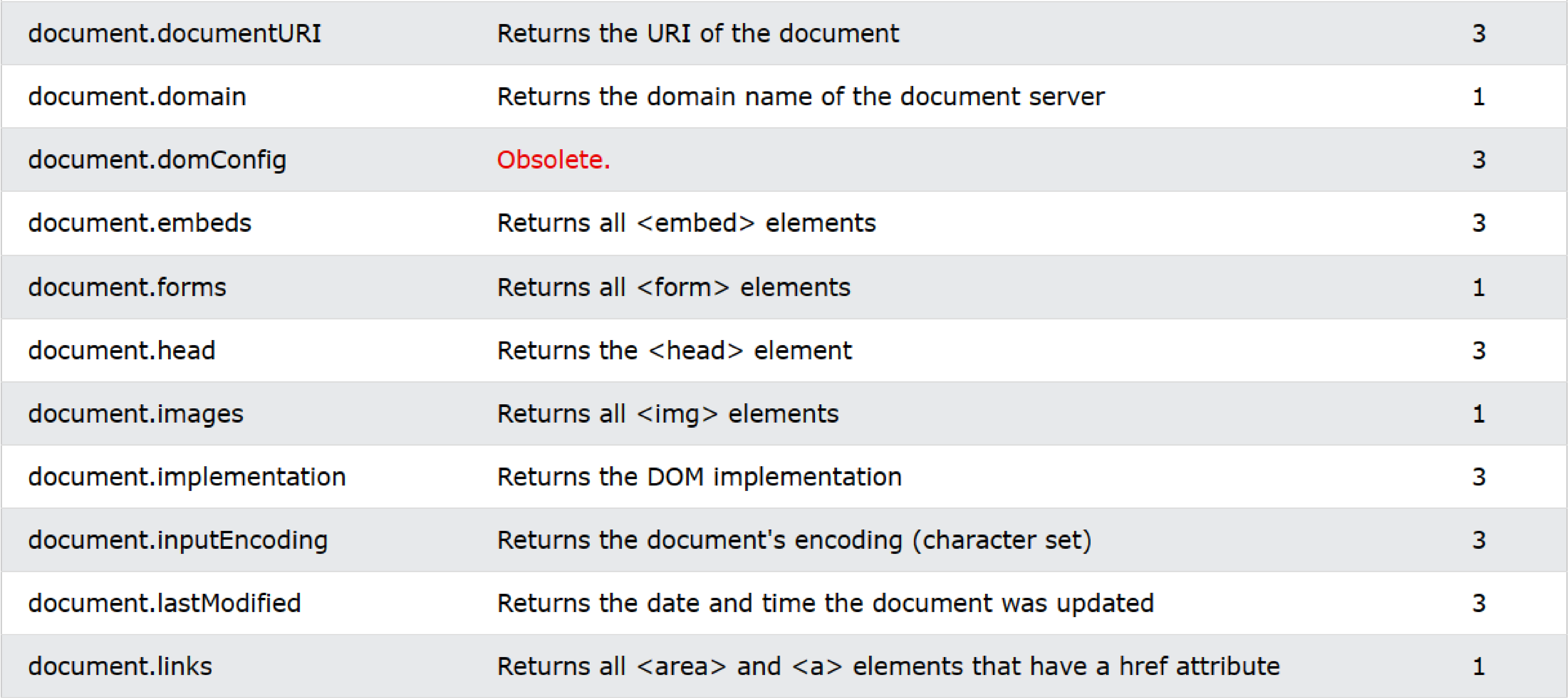
DOM – Finding HTML Objects



Property	Description	DOM
document.anchors	Returns all <a> elements that have a name attribute	1
document.applets	Deprecated	1
document.baseURI	Returns the absolute base URI of the document	3
document.body	Returns the <body> element	1
document.cookie	Returns the document's cookie	1
document.doctype	Returns the document's doctype	3
document.documentElement	Returns the <html> element	3
document.documentMode	Returns the mode used by the browser	3

K2

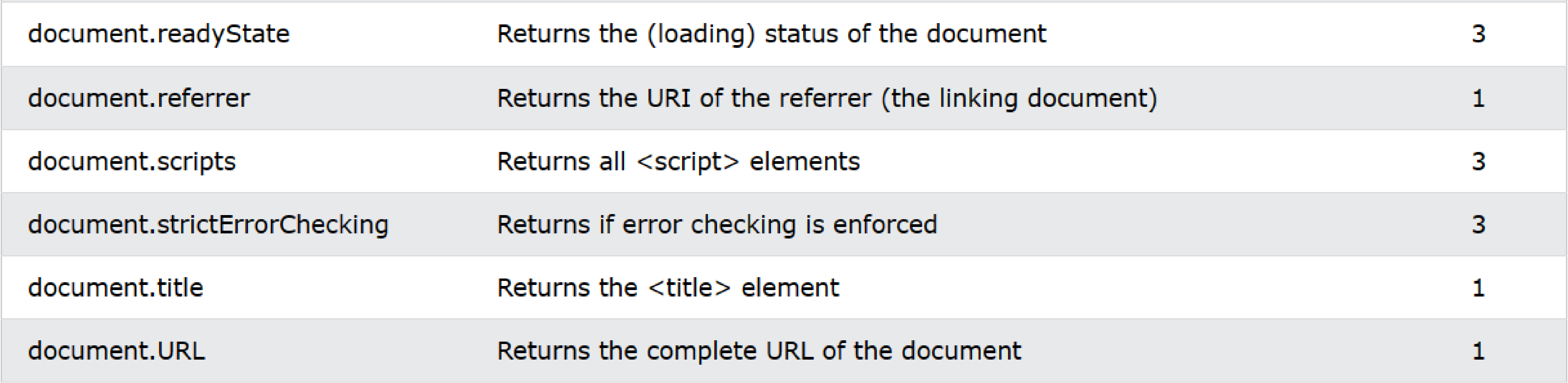
DOM – Finding HTML Objects



document.documentURI	Returns the URI of the document	3
document.domain	Returns the domain name of the document server	1
document.domConfig	Obsolete.	3
document.embeds	Returns all <embed> elements	3
document.forms	Returns all <form> elements	1
document.head	Returns the <head> element	3
document.images	Returns all <img> elements	1
document.implementation	Returns the DOM implementation	3
document.inputEncoding	Returns the document's encoding (character set)	3
document.lastModified	Returns the date and time the document was updated	3
document.links	Returns all <area> and <a> elements that have a href attribute	1

K2

DOM – Finding HTML Objects



document.readyState	Returns the (loading) status of the document	3
document.referrer	Returns the URI of the referrer (the linking document)	1
document.scripts	Returns all <script> elements	3
document.strictErrorChecking	Returns if error checking is enforced	3
document.title	Returns the <title> element	1
document.URL	Returns the complete URL of the document	1

K2

DOM – HTML Events

- **Event Listener**
- **Event Handler**

K2

DOM – HTML Events

- **Event Listener**
  - **Some events – mouse click, scrolling, key press, load, animations, online, and so on**
- **Event Handler**

K2

DOM – HTML Events

- **Event Listener**
  - **Some events – mouse click, scrolling, key press, load, animations, online, and so on**
- **Event Handler**
  - **Javascript – Responds to event**

DOM – HTML Events

- **Event Listener**
  - **Some events – mouse click, scrolling, key press, load, animations, online, and so on**
- **Event Handler**
  - **Javascript – Responds to event**
- **HTML – Events can be given Element attributes**

K2

DOM – HTML Events

- **Event Listener**
  - **Some events – mouse click, scrolling, key press, load, animations, online, and so on**
- **Event Handler**
  - **Javascript – Responds to event**
- **HTML – Events can be given Element attributes**
- **Javascript – Events can be added to element objects**
  - **Syntax: element.addEventListener(event, function());**
  - **Syntax: element.removeEventListener (event, function());**

DOM – HTML Events

- Event Listener
  - Some events – mouse click, scrolling, key press, load, animations, online, and so on
- Event Handler
  - Javascript – Responds to event
- HTML – Events can be given Element attributes
- Javascript – Events can be added to element objects
  - Syntax: element.addEventListener(event, function());
  - Syntax: element.removeEventListener (event, function());
- List of all possible events
  - Source: HTML DOM Event Object (w3schools.com)

K2

Best Practices

1. **Use HTML tags with the same case (preferably lowercase).**
2. **Filename to be based on functionality.**
3. **When opening a tag, have a "tab space" for each line until you close the tag. It helps in the readability of your HTML document.**
4. **Give a comment describing a web page as the first line of your HTML document.**
5. **Keep HTML, CSS, and Javascript files separate.**
6. **Give meaningful names to identifiers.**
7. **Identifiers should not be too long or too short (min 3 character length).**
8. **Add comments on the outermost nested control structure.**

K1

Summary

- **DOM**
- **DHTML**
- **DOM – Finding HTML Elements**
- **DOM – Changing HTML Elements**
- **DOM – Tree Traversal**
- **DOM – Adding and Deleting HTML Elements**
- **DOM – HTML Event Handling**
- **DOM – Finding HTML Objects**
- **DOM – HTML Events**
- **Best Practices**

References

1. **Paul J Deitel, Harvey M Deitel, and Abbey Deitel, "Internet and the world wide web: How to program," Pearson, Fifth Edition, 2012.**

THANK YOU

SHIV NADAR
UNIVERSITY
CHENNAI

Web Technology

SHIV NADAR
UNIVERSITY
CHENNAI

AJAX Introduction – Requests and Response

Department of Computer Science and Engineering
Shiv Nadar University

Last Class Summary

- **DOM**
- **DHTML**
- **DOM – Finding HTML Elements**
- **DOM – Changing HTML Elements**
- **DOM – Adding and Deleting HTML Elements**
- **DOM – HTML Event Handling**
- **DOM – Finding HTML Objects**
- **DOM – HTML Events**
- **Best Practices**

K1

AJAX – Asynchronous Javascript And XML

- **Asynchronous?**

K2

AJAX – Asynchronous Javascript And XML

- **Asynchronous**
  - **Send request – Server**
  - **Proceeds to next request before response**
  - **Response Handled – Background**

AJAX – Asynchronous Javascript And XML

- Asynchronous
  - Send request – Server
  - Proceeds to next request before response
  - Response Handled – Background
- Features
  - Update Pages without Reloading

AJAX – Asynchronous Javascript And XML

- **Asynchronous**
  - **Send request – Server**
  - **Proceeds to next request before response**
  - **Response Handled – Background**
- **Features**
  - **Update Pages without Reloading**
  - **Request Data – After page load**
  - **Respond Data – After page load**

AJAX – Asynchronous Javascript And XML

- **Asynchronous**
  - **Send request – Server**
  - **Proceeds to next request before response**
  - **Response Handled – Background**
- **Features**
  - **Update Pages without Reloading**
  - **Request Data – After page load**
  - **Respond Data – After page load**
  - **Send Data to Server – Background**

K2

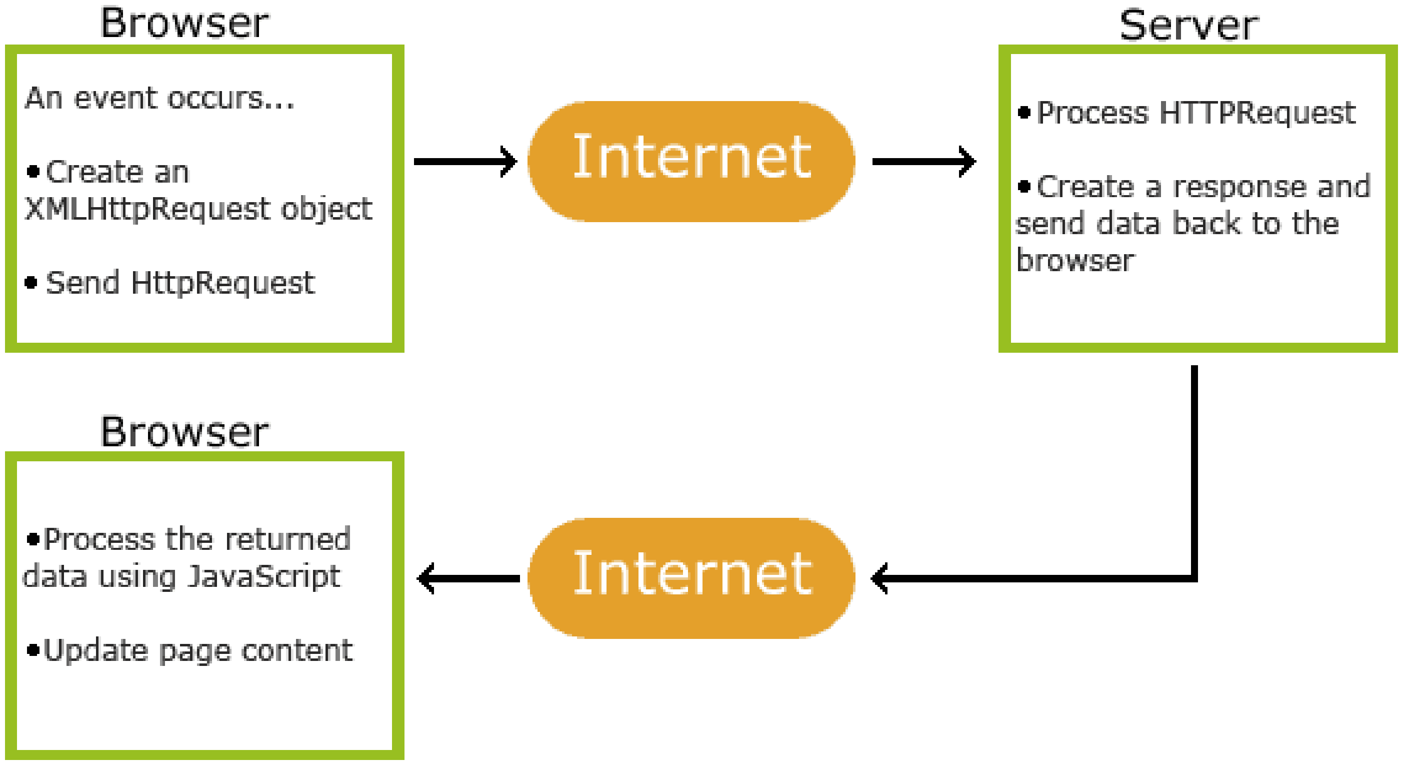
AJAX Working

- **AJAX Utilizes**
  - **Browser Built-in XMLHttpRequest object – send request to server**
  - **HTML DOM and Javascript – display data**

K2

AJAX Working

- **AJAX Utilizes**
  - **Browser Built-in XMLHttpRequest object – send request to server**
  - **HTML DOM and Javascript – display data**



Browser

An event occurs...

- Create an XMLHttpRequest object
- Send HttpRequest

Internet

Server

- Process HTTPRequest
- Create a response and send data back to the browser

Internet

Browser

- Process the returned data using JavaScript
- Update page content

K2

AJAX XMLHttpRequest

- **XMLHttpRequest - Object**

K2

AJAX XMLhttpRequest

- **XMLHttpRequest – Object**
    - **Methods – Background**

K2

AJAX XMLHttpRequest

- **XMLHttpRequest – Object**
  - **Methods – Background**
- **Syntax: identifier_name = new XMLHttpRequest();**

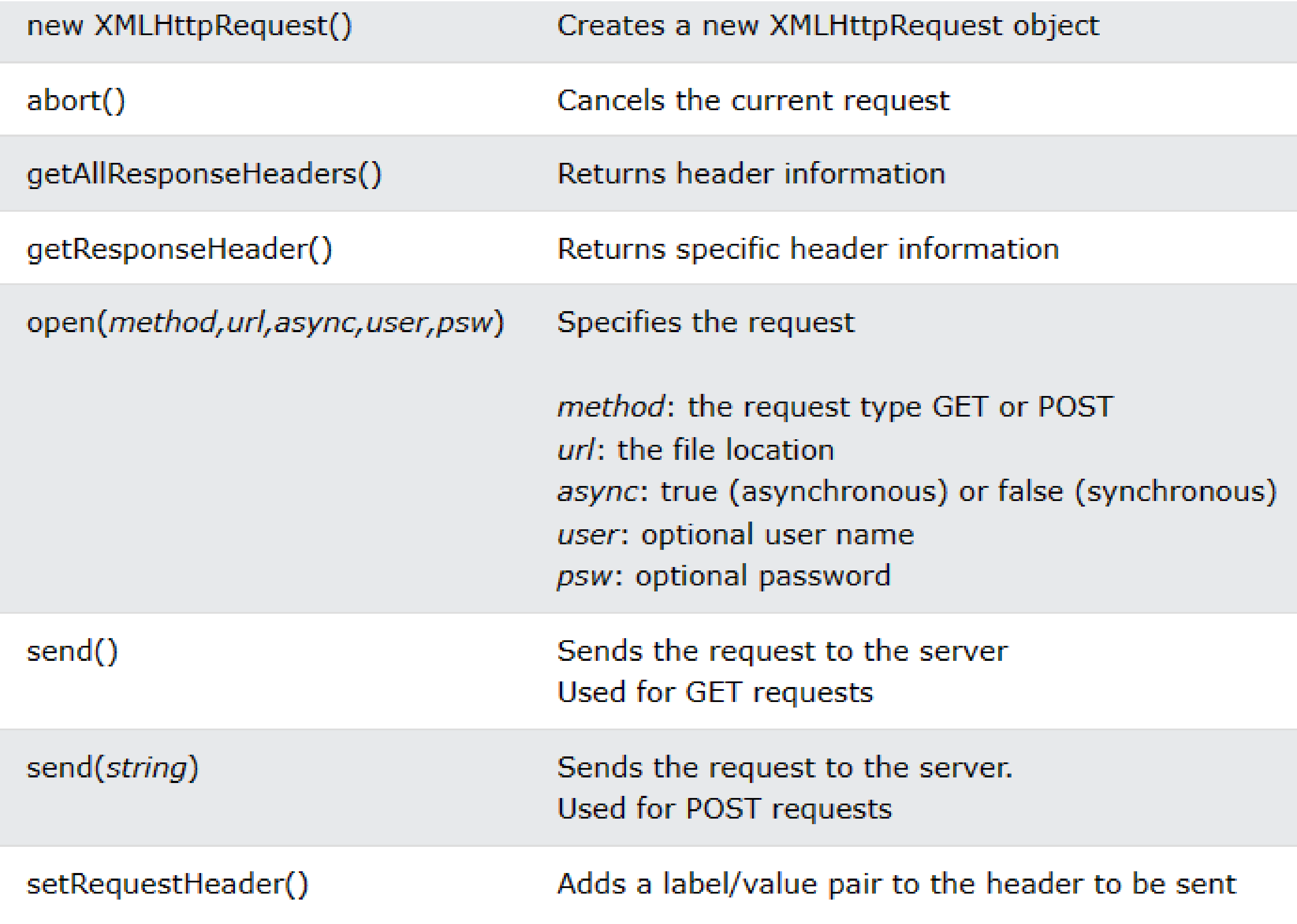
K2

AJAX XMLHttpRequest

- **XMLHttpRequest – Object**
  - **Methods – Background**
- **Syntax: identifier_name = new XMLHttpRequest();**
- **Note: AJAX – Cross domain not allowed.**

K2

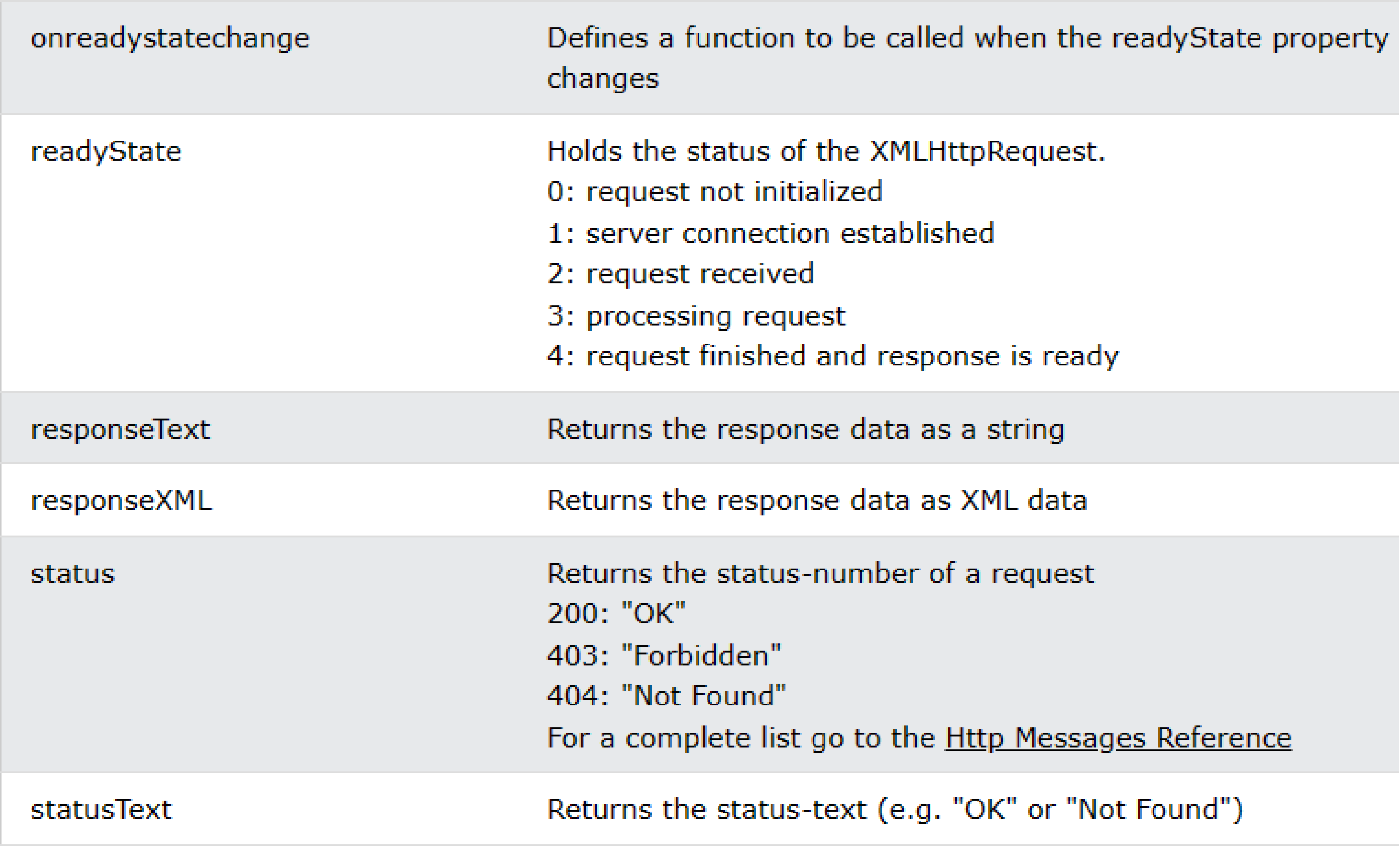
AJAX XMLHttpRequest Methods



new XMLHttpRequest()	Creates a new XMLHttpRequest object
abort()	Cancels the current request
getAllResponseHeaders()	Returns header information
getResponseHeader()	Returns specific header information
open(*method*,*url*,*async*,*user*,*psw*)	Specifies the request *method*: the request type GET or POST *url*: the file location *async*: true (asynchronous) or false (synchronous) *user*: optional user name *psw*: optional password
send()	Sends the request to the server Used for GET requests
send(*string*)	Sends the request to the server. Used for POST requests
setRequestHeader()	Adds a label/value pair to the header to be sent

K2

AJAX XMLHttpRequest Methods



onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
responseText	Returns the response data as a string
responseXML	Returns the response data as XML data
status	Returns the status-number of a request 200: "OK" 403: "Forbidden" 404: "Not Found" For a complete list go to the Http Messages Reference
statusText	Returns the status-text (e.g. "OK" or "Not Found")

K2

AJAX Request and Response

ajax_info.txt

```
<h1>AJAX</h1>
<p>AJAX is not a programming language.</p>
<p>AJAX is a technique for accessing web servers from a web page.</p>
<p>AJAX stands for Asynchronous JavaScript And XML.</p>
```

K2

AJAX Request and Response

```
function loadDoc() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            document.getElementById("demo").innerHTML =
            this.responseText;
       }
    };
    xhttp.open("GET", "ajax_info.txt", true);
    xhttp.send();
}
```

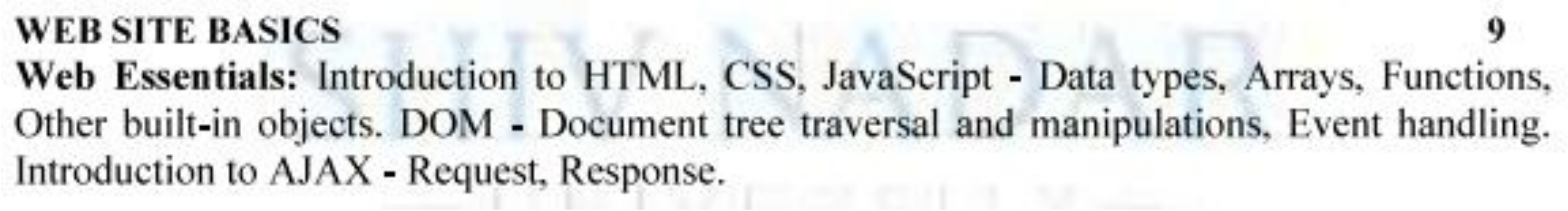
K2

Summary

- **AJAX**
  - **Asynchronous**
  - **Features of AJAX**
- **AJAX Working**
- **AJAX XMLhttp**
- **AJAX Request**
- **AJAX Response**
- **Best Practices**

K1

Unit – 1 – Syllabus



WEB SITE BASICS **9**

Web Essentials: Introduction to HTML, CSS, JavaScript - Data types, Arrays, Functions, Other built-in objects. DOM - Document tree traversal and manipulations, Event handling. Introduction to AJAX - Request, Response.

K1

References

1. **Paul J Deitel, Harvey M Deitel, and Abbey Deitel, "Internet and the world wide web: How to program," Pearson, Fifth Edition, 2012.**

THANK YOU

SHIV NADAR
UNIVERSITY
CHENNAI