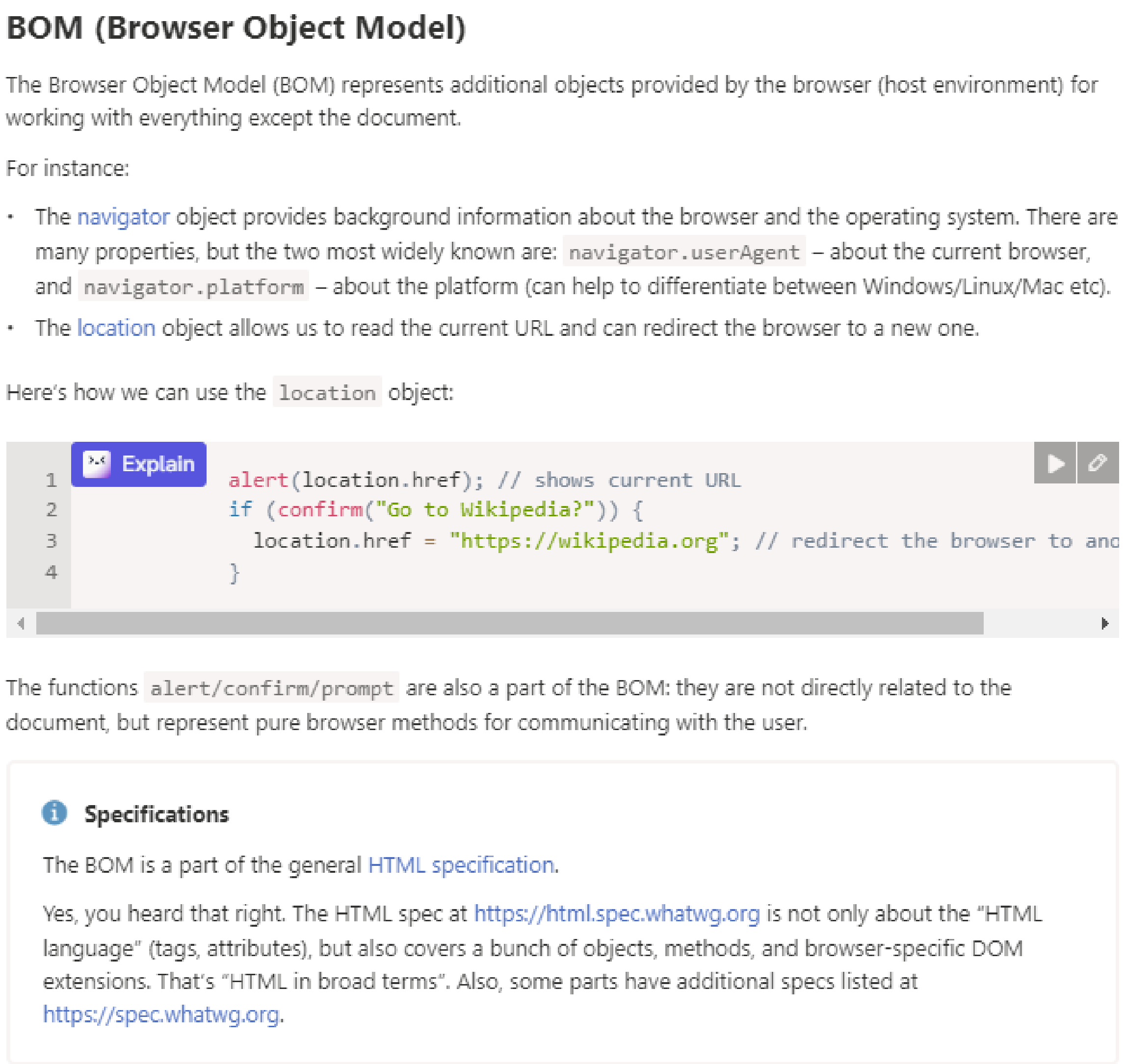
FRONTEND DEVELOPER
Are you a DR. Frontend?
CSS FRAMEWORKS
(any one)
• Tailwind
• Bootstrap
• Bulma...
VSC
• Git
GitHub (any one)
GitLab
TOOLING
• Linting/Hinting (eslint)
• Formatting (prettier)
• Minifier
JAVASCRIPT
• Syntax
• DOM
• Fetch API/Ajax (XHR)
• ES6+ & Modules
• FP
• prototype, OOP...
PACKAGE MANAGERS
• NPM
• YARN
(any one)
JS LIBRARIES/ FRAMEWORKS
(any one)
• React
• Angular
• Vue...
CSS PRE/POST PROCESSORS
• Sass
• Less
• PostCSS
(any one)
BUNDLERS
• Webpack
• Parcel
• esbuild...
(any one)
CSS
• Inheritance
• Cascade
• Specificity
• Box Model
• Grid/Flexbox
• Media Queries
• Animation
SSR
• Next.js (React)
• Universal (Angular)
• Nuxt.js (Vue)
HML
• Semantic HTML
• Element Attributes
• ARIA and Roles
TESTING
• Cypress
• Jest
• Enzyme...
(any one)
WWW BASIC MECHANICS
KEEP LEARNING
• PWA
• Web Components
• Web Assembly

Introduction to Markup Languages and Web Technologies

The BROWSER OBJECT MODEL **BOM**

https://javascript.info/browser-environment



BOM (Browser Object Model)

The Browser Object Model (BOM) represents additional objects provided by the browser (host environment) for working with everything except the document.

For instance:

- The navigator object provides background information about the browser and the operating system. There are many properties, but the two most widely known are: `navigator.userAgent` – about the current browser, and `navigator.platform` – about the platform (can help to differentiate between Windows/Linux/Mac etc).
- The location object allows us to read the current URL and can redirect the browser to a new one.

Here's how we can use the `location` object:

Explain

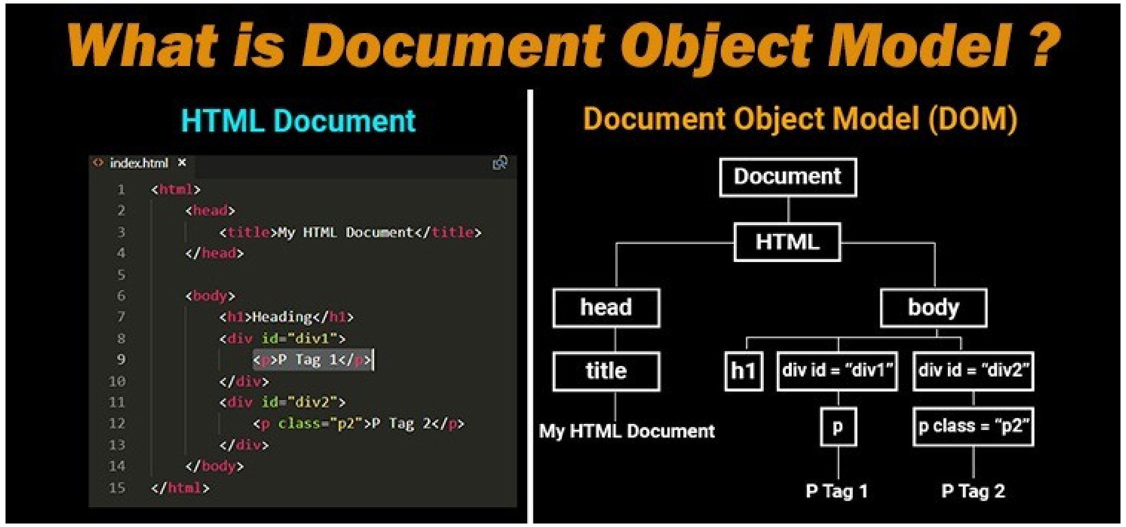
```
alert(location.href); // shows current URL
if (confirm("Go to Wikipedia?")) {
  location.href = "https://wikipedia.org"; // redirect the browser to ano
}
```

The functions `alert/confirm/prompt` are also a part of the BOM: they are not directly related to the document, but represent pure browser methods for communicating with the user.

Specifications

The BOM is a part of the general HTML specification.

Yes, you heard that right. The HTML spec at https://html.spec.whatwg.org is not only about the "HTML language" (tags, attributes), but also covers a bunch of objects, methods, and browser-specific DOM extensions. That's "HTML in broad terms". Also, some parts have additional specs listed at https://spec.whatwg.org.



```
<html>
  <head>
    <title>My HTML Document</title>
  </head>

  <body>
    <h1>Heading</h1>
    <div id="div1">
      <p>P Tag 1</p>
    </div>
    <div id="div2">
      <p class="p2">P Tag 2</p>
    </div>
  </body>
</html>
```

THE BOM Exposes the DOM NODES as API end point hooks.

JavaScript provides APIs to connect to these hook points to enable programmatic changes to the html document contents.

Lab Workbook: HTML and the BOM

Understanding the HTML DOM and BOM

https://coda.io/@peter-sigurdson/10-use-cases-of-building-css-and-flexbox-into-a-web-page

- These Lab Exercises emphasize the seamless collaboration between DOM and BOM for a dynamic and interactive user experience.
- Encourage students to explore further with tools and resources for hands-on learning.
- This presentation can serve as an introductory guide for new developers or as a refresher for seasoned professionals on how the HTML DOM and BOM collaborate to present and interact with web content dynamically.
- Each Lab Exercise is designed to build upon the previous concepts, culminating in a comprehensive understanding of web page rendering and manipulation.

Introduction to HTML DOM and BOM

Understanding the HTML DOM and BOM

- **Definition of DOM**: The Document Object Model (DOM) is a programming interface for HTML documents. It represents the page so that programs can change the document structure, style, and content.
- **Definition of BOM**: The Browser Object Model (BOM) allows JavaScript to 'talk' to the browser, controlling things like window size and the URL displayed.
- **Interplay Between DOM and BOM**: While the DOM focuses on the content, the BOM focuses on the browser's features, yet both are essential in web content presentation.

Architecture of the HTML DOM

DOM as a Node Tree:

Every element, attribute, and piece of text in the markup is part of the DOM as a node in a tree structure.

Node Types:

Discussion of different node types - Element, Text, Attribute

Accessing Nodes: How JavaScript accesses nodes using methods like getElementById(), getElementsByClassName(), querySelector(), and querySelectorAll().

Selectors and Style Manipulation

Role of CSS Selectors:

Explaining how selectors target HTML elements to apply styles.

Selector Types: Overview of basic selectors (e.g., type, class, id), combinators, and pseudo-classes.

Styling via JavaScript:

How JavaScript uses these selectors to dynamically change styles, such as element.style.color = 'blue';.

BOM and Browser Interactions

Browser Object Model (BOM) in Action

Window Object: Central to the BOM, the window object represents the browser's window, with methods to manipulate browser history, location, and more.

Interacting with DOM: How BOM methods like window.open() or window.close() can affect the DOM by opening or closing tabs that contain DOM trees.

Responsive Design: Utilizing BOM properties like window.innerWidth and window.innerHeight for responsive design adjustments in real-time.

HTML Nodes and Element Relationships

Navigating Node Relationships in the DOM

Parent-Child Relationship:

Understanding the hierarchy and how parent nodes contain child nodes, which can be other elements or text.

Siblings: How elements at the same level in the hierarchy can be accessed as next or previous siblings.

Navigation with JavaScript:

Demonstrating DOM traversal using properties like parentNode, childNodes, nextSibling, and previousSibling.

Dynamic Content with DOM and BOM

Dynamic Web Content Creation

DOM Manipulation:

Creating, removing, or modifying HTML elements and content dynamically, such as using document.createElement() and node.appendChild().

Event Handling: How BOM events trigger DOM updates, e.g., a click event altering an element's class to change its style.

Real-World Example: A brief case study of a dynamic web application, showing how DOM and BOM work together to render and manage interactive content.

The origin story of HTTP

I was there. I saw how it started. This is the origin story of HTTP:

It was a bright and sunny day at the European Organization for Nuclear Research (CERN) in Switzerland. Dr. Tim Berners-Lee was standing in front of a group of computer science students, eagerly discussing the latest project he had been working on.

"As you all know, the Internet is a vast and constantly growing network of interconnected computers," he began. "But how do we access and share information on this network? That's where the HTTP protocol comes in."

Dr. Berners-Lee explained that HTTP, or Hypertext Transfer Protocol, was a set of rules for transmitting data over the internet. He had developed it in 1989 as a way for computers to communicate with each other and access information stored on servers.

"But here's the thing," he continued. "HTTP is a stateless protocol, which means that it doesn't store any information about previous requests. Each request is treated independently from the others."

This concept puzzled some of the students, so Dr. Berners-Lee provided an example to help clarify.

"Imagine you're at a coffee shop ordering a drink," he said. "The barista doesn't remember your order from the last time you came in, right? They treat each request for a drink as a separate transaction."

The students nodded in understanding.

"It's the same with HTTP," Dr. Berners-Lee continued. "Each request for information is treated as a separate transaction, and the server doesn't store any information about previous requests."

The students nodded, now fully grasping the concept of the stateless nature of HTTP.

"So that's how HTTP works," Dr. Berners-Lee concluded. "It's a simple, yet powerful protocol that allows us to access and share information on the internet."

The students asked: “Won’t the stateless nature of HTTP be a problem in the future for web commerce”?

But Dr. Berners-Lee was in a hurry to get in some Alpine Skiing and brushed off the concern by saying: “Nah. It’ll be fine”. But it was not fine. The stateless nature of HTTP has required use to put in place many work-arounds such as URL rewriting which you will soon be seeing.

The HTML Document Object Model (DOM) containment schema.

- The DOM is a way of representing and interacting with HTML documents in a structured manner. It is essentially a tree-like structure, with the document itself at the root, and all the elements within the document as nodes within that tree.
- Each element in the DOM has a parent element, which is the element that contains it, as well as any number of child elements, which are the elements contained within it. This creates a hierarchy of elements, with the root element at the top and the child elements nested within it.
- **One of the key features of the DOM is that it allows for the manipulation of HTML documents. This means that you can use the DOM to add, delete, or modify elements within the document, as well as change the attributes of those elements. JavaScript has a number of APIs that address the BOM to do exactly this.**
- To access and manipulate the DOM, you can use a variety of programming languages, including JavaScript. With JavaScript, you can use the DOM to create dynamic and interactive web pages that can respond to user input and change based on certain conditions.
- So, in summary, the DOM is a way of representing and interacting with HTML documents in a structured manner, allowing for the manipulation of those documents and the creation of dynamic and interactive web pages. I hope this has been a helpful introduction to the HTML DOM containment schema.

An HTML Document is a graph of connected NODEs.

- Each NODE has data: addressable and mutable by CSS
- Each NODE has STATE and EVENTS: addressable and mutable by JavaScript.

Who owns and controls the Internet and the HTML standard

- The Internet is not owned by any one entity, but rather it is a global network of interconnected computer networks. It is a decentralized system, which means that there is no central authority that controls it. Instead, the Internet is controlled by a variety of organizations and individuals that work together to ensure that it is a functioning network.
- The HTML standard, which is the standardized system for creating and displaying web content, is developed and maintained by the World Wide Web Consortium (W3C). The W3C is an international organization that works to develop open standards for the web. It is made up of member organizations and individual members, and it operates through a consensus-based process to develop standards.

Building Web Applications

To make a website, you will work in 3 domains:

The programming language and database (server side)

Scripting Language: JavaScript

Markup Languages: HTML and CSS

This course focuses on the client-side. The HTML, JavaScript and CSS live in the Web Browser.

In more advanced web application development courses, you will write code and databases that live on the SERVER.

(JavaScript can also run on the Server in a format called NodeJS).

In this course you will be using your web server to host simple text files. These files will contain your HTML, JavaScript and CSS code.
The web browser will emit a URL to your server and the server will respond by sending HTML documents. JavaScript and CSS files which the browser will render. This is why we call HTTP are request, response. Model. The HTTP request response model has no memory. Browser makes the request to the server. Server sends back the contents to the browser. And then forgets the whole thing ever happened.

Development environment setup

We will be working with the Chromium Web Browser to take advantage of the Chromium Debugging Protocol that let’s us visualize and make dynamic changes to the Browser Object Model.

The Browser Object Model (BOM) refers to the interface between web browsers and scripts (such as JavaScript).

It allows scripts to "talk to" the browser and interact with the content of web pages.

The BOM consists of a set of objects that represent different aspects of the browser and the web page. For example, the "window" object represents the current browser window, and the "document" object represents the web page that is currently loaded in the window.

The BOM also provides a number of methods and properties that allow scripts to manipulate the content of a web page.

For example, a script can use the "getElementById" method to access a specific element on the page, and then use the "innerHTML" property to change the content of that element.
Web developers use the BOM to create interactive and dynamic web pages. By using scripts to manipulate the content of a web page, developers can create engaging and user-friendly experiences for their users.

Development environment setup

Download and install the 14 Day Trial Version of MMAP. MMAP works on Windows 11.
If you are pre-Windows 11, you can also use XAMPP.

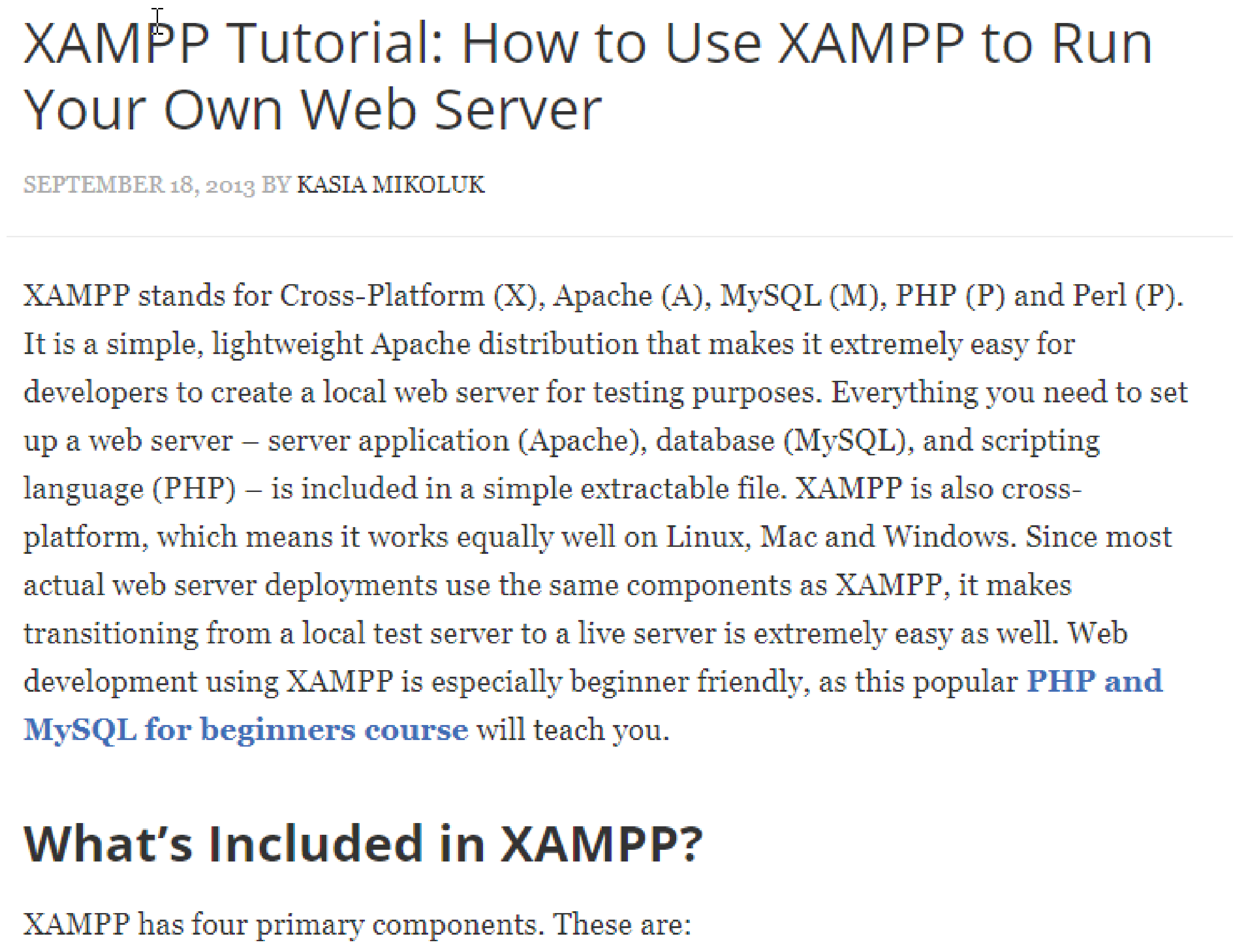
An HTTP server is a computer program that listens for incoming requests from clients and sends back responses. It is designed to handle the HTTP protocol, which is the standard protocol for transmitting data on the World Wide Web.
When a client, such as a web browser, sends an HTTP request to a server, the server processes the request and sends back an HTTP response. The response typically includes a status code, which indicates whether the request was successful, and a message body, which is the data that the server is sending back to the client.
There are many different types of HTTP servers, ranging from simple programs that can only handle a few requests at a time to more complex servers that can handle thousands of requests simultaneously. The specific implementation of an HTTP server can vary, but most servers follow a similar set of steps when processing requests:

1. Listen for incoming requests on a specific port (e.g., port 80 for HTTP).
2. Parse the request to determine the requested resource and any other relevant parameters.
3. Check if the requested resource is available on the server.
4. If the resource is available, create an HTTP response with the appropriate status code (e.g., 200 OK) and message body (e.g., the content of the requested resource). If the resource is not available, create an HTTP response with an appropriate error status code (e.g., 404 Not Found).
5. Send the HTTP response back to the client.

HTTP servers are an essential part of the World Wide Web and play a critical role in how we access and interact with online content.

https://blog.udemy.com/xampp-tutorial/

XAMPP gives us the ability to create dynamic, server-side programming for our web pages



XAMPP Tutorial: How to Use XAMPP to Run Your Own Web Server

SEPTEMBER 18, 2013 BY KASIA MIKOLUK

XAMPP stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing purposes. Everything you need to set up a web server – server application (Apache), database (MySQL), and scripting language (PHP) – is included in a simple extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server is extremely easy as well. Web development using XAMPP is especially beginner friendly, as this popular **PHP and MySQL for beginners course** will teach you.

What's Included in XAMPP?

XAMPP has four primary components. These are:

What is a Language

As preparation for our investigation of Markup Languages, let’s refine our View of what a Language is.

- Claude Shannon, who is known as the "father of information theory" focused on the concept of information and how it can be transmitted, stored, and processed.
- In Shannon's view, information can be thought of as a sequence of symbols or messages that are transmitted from one place to another through a communication channel. He developed mathematical theories and principles that describe how information can be transmitted and received in an efficient and reliable way, and how it can be quantified and measured.
- In this context, language can be thought of as a system of symbols or signals that are used to transmit information. For example, the English language uses a set of 26 letters, along with punctuation marks and other symbols, to represent and transmit information through written or spoken communication. Similarly, programming languages like CSS, HTML, and XML use a set of symbols and rules to represent and transmit instructions to computers.
- Language consists of 2 domains: Syntactic and Symantec. (We will see these concepts reflects in HTML tag nomenclature in a few minutes).
- Language is the process of exchange of information between parties using symbols in a prescribed order. Send and receiver must agree on these protocols and have a shared common context. This is referenced to as the Symantec Web which is where the term “Web” originated.

The semantic web – The topology of meaning.

Language is the system of symbols, rules, and conventions that we use to communicate and exchange information with one another.

In order for effective communication to take place, both the sender and receiver must understand and agree upon the protocols and conventions of the language being used, and they must have a shared context or understanding of the information being transmitted.

This shared understanding is known as the semantic web, or the web of meaning, which is where the term 'web' as we know it originated.

In the context of programming and the internet, the semantic web refers to the interconnected network of information and meaning that is shared and communicated through the use of language and technology.

Syntax and Semantics

- Language is a system of symbols and rules that we use to communicate and convey information. Within the study of language, there are two main domains that are important to understand: syntactic and semantec.
- Syntax refers to the structure and form of a language. It is the set of rules that govern how words and phrases can be combined to create meaning. In programming languages like HTML, syntax is important because it determines how the code is written and how it is interpreted by the computer.
- Semantics, on the other hand, refers to the meaning of words and phrases within a language. It is the study of how the words and symbols of a language are used to convey meaning. In HTML, semantics are important because they determine the purpose and function of the tags and elements that are used to structure and format the content of a web page.
- We will see these concepts reflected in the nomenclature of HTML tags as we begin to work with them. Understanding the syntax and semantics of a language is essential for writing clear and meaningful code, and for creating effective and well-designed websites.

What is a Markup Language

Carefully structured and arranged set of descriptors (tags) which inform the Browser’s Rendering Engine HOW the content should be displayed.

What are the constituent compounds of a markup language:

- A markup language consists of a set of elements that define the structure and content of a document. These elements are usually represented by tags, which are enclosed in angle brackets and are used to mark the start and end of an element.
- The content of an element is the text or other data that is contained between the start and end tags.
- Some common examples of elements in markup languages include headings, paragraphs, lists, links, and images.
- Markup languages also often include attributes, which are additional pieces of information that can be associated with an element and are used to provide additional detail or specify the behavior of the element.
- Markup languages are used to create documents that can be displayed or shared in a variety of formats, including web pages, ebooks, and print documents.
- JSON and Latex are other command ML languages you may encounter.

The study of Mark Up Languages

HTML : A schema for structuring the presentation of data

CSS : Used to change the details of HTML DOM elements (color, visible)

JavaScript: Used to manipulate to change the details of HTML Dom Elements

Using Node.js to experiment with JavaScript

Tools needed: Node.js **Visual Studio Code**

We can run JavaScript inside the Browser because the Browser contains Chromium – The JavaScript Interpreter.

We can run JavaScript at the command line using Node.js:

1. Easier for practicing basic JavaScript Code formulations like loops.
2. We can write JavaScript server side programs to process form data.

Lesson Plan : Day 01:

1. Introduce the concept of HTML as a directed graph of nodes, where each node represents an element in the document and the edges represent the relationships between the elements.
2. Use Chromium Dev Tools to visualize the BOM.
3. Discuss the containment hierarchy of objects in the HTML DOM Document Object MODEL. Each OBJECT is NODE which is an HTML SCRIPT TAG. Nodes have : Data, Events (they can be the source of or target of an EVENT. onClick() for example.
4. Introduce the concept of data attributes, which are attributes of nodes, and which set the display characteristics of page elements.
5. Explain how HTML Nodes can be modified by CSS Scripts.
6. Provide exercises for students to practice using data attributes and events in HTML and CSS. (Later we will do this with JavaScript).

Chromium Dev Tools

Chrome DevTools is a set of web development tools built into the Google Chrome browser. It allows developers to debug, inspect, and modify the content and behavior of web pages.

One of the main ways that DevTools works with the Browser Object Model (BOM) is by providing a console that allows developers to interact with the BOM in real-time. The console is a command-line interface that lets developers execute JavaScript code and see the results immediately.

For example, a developer can use the console to access and modify the properties of the "window" object, which represents the current browser window. They can also use the console to call methods of the "document" object, which represents the web page currently loaded in the window.

In addition to the console, DevTools provides a number of other tools that allow developers to interact with the BOM.
These tools include the Elements panel, which lets developers view and modify the HTML and CSS of a web page; the Network panel, which lets developers see how a web page is loading and interacting with the network; and the Sources panel, which lets developers view and debug JavaScript code.

Overall, DevTools provides a powerful set of tools that allow developers to work with the BOM

Describe the containment hierarchy of the HTML Document in the browser.

In the browser, an HTML document is represented as a containment hierarchy of objects, where each object corresponds to an element in the document and represents the properties and behavior of that element. When we study JavaScript, we will learn the JavaScript Classes that correspond to each Object in the DOM.

At the top of the hierarchy is the Document object, which represents the entire document.

The Document object contains a head property that refers to the Head object, which represents the <head> element of the document, and a body property that refers to the Body object, which represents the <body> element of the document.

The Body object contains a collection of Node objects, each of which represents an element in the body of the document.

The Node objects are organized into a tree structure, with each node having a parent node (except for the root node, which is the Body object) and possibly one or more child nodes.

The containment hierarchy of the HTML document in the browser is used to represent the structure and relationships of the elements in the document, and it is used by the browser to render the document and respond to user input and other events.

The Document Object Model (DOM) is a programming interface.

In JavaScript, the Document Object Model (DOM) is a programming interface for HTML and XML documents. It represents the structure of a document as a tree of objects, with each object representing a part of the document (such as an element, an attribute, or a piece of text).

JavaScript classes are used to define the properties and behaviors of objects in the DOM. For example, you might use a Button class to define a button object in the DOM, with properties such as the button's text label and the actions that should be taken when the button is clicked.

You can use JavaScript to manipulate the DOM by creating, modifying, and deleting objects in the DOM tree. This is often used to change the appearance or behavior of a webpage in response to user input or other events.

Cascading Style Sheets

Animations and transitions: CSS can be used to create smooth animations and transitions between different states of an element. This can be used to draw the attention of the user or to enhance the user experience. For example, you can use the transition property to smoothly change the size or color of an element when it is hovered over, or the animation property to create more complex animations.

Responsive design: CSS can be used to create responsive designs that adapt to the size and orientation of the screen. This is important for creating websites that work well on a wide range of devices, from desktop computers to mobile phones. You can use media queries to apply different styles based on the size of the screen, and the flexbox and grid layout systems to create flexible and responsive layouts.

Typography: CSS provides a wide range of options for styling text, including font family, size, color, and alignment. You can use these options to create beautiful and effective typography for your website or application.

Backgrounds and gradients: CSS allows you to specify backgrounds for elements using colors, images, or gradients. You can use gradients to create smooth transitions between colors, and use the background-size and background-position properties to control how the background is displayed.

Box model: The CSS box model defines how the dimensions and spacing of elements are calculated, and how they interact with each other. You can use the box model to create complex layouts and control the size and position of elements on the page.

CSS works with Rules and Selectors.

A CSS selector is a pattern that is used to select the element or elements on a web page that you want to style with CSS. There are many different types of CSS selectors, each with its own specific syntax.

CSS selectors can be made in a variety of ways. Some common types of selectors include:

Element selectors: These selectors select elements based on their name, such as p for paragraphs or h1 for headings.

Class selectors: These selectors select elements based on their class attribute, using a period (.) followed by the class name. For example, .highlight would select all elements with a class of "highlight".

ID selectors: These selectors select elements based on their ID attribute, using a hash symbol (#) followed by the ID name. For example, #main would select the element with an ID of "main".

Attribute selectors: These selectors select elements based on their attributes and attribute values. For example, a[href="http://example.com"] would select all links with an href attribute of "http://example.com".

We use CSS selectors to apply styles to specific elements on a web page. For example, if we wanted to change the color of all the paragraphs on a page to red, we could use the element selector p and set the color property to red:

```
p {
  color: red;
}
```

We use CSS selectors because they allow us to target specific elements on a web page and apply styles to them in a very precise and controlled way. This helps us create consistent and visually appealing designs, and makes it easier to maintain and update our styles as needed.

There are three main types of CSS selectors:

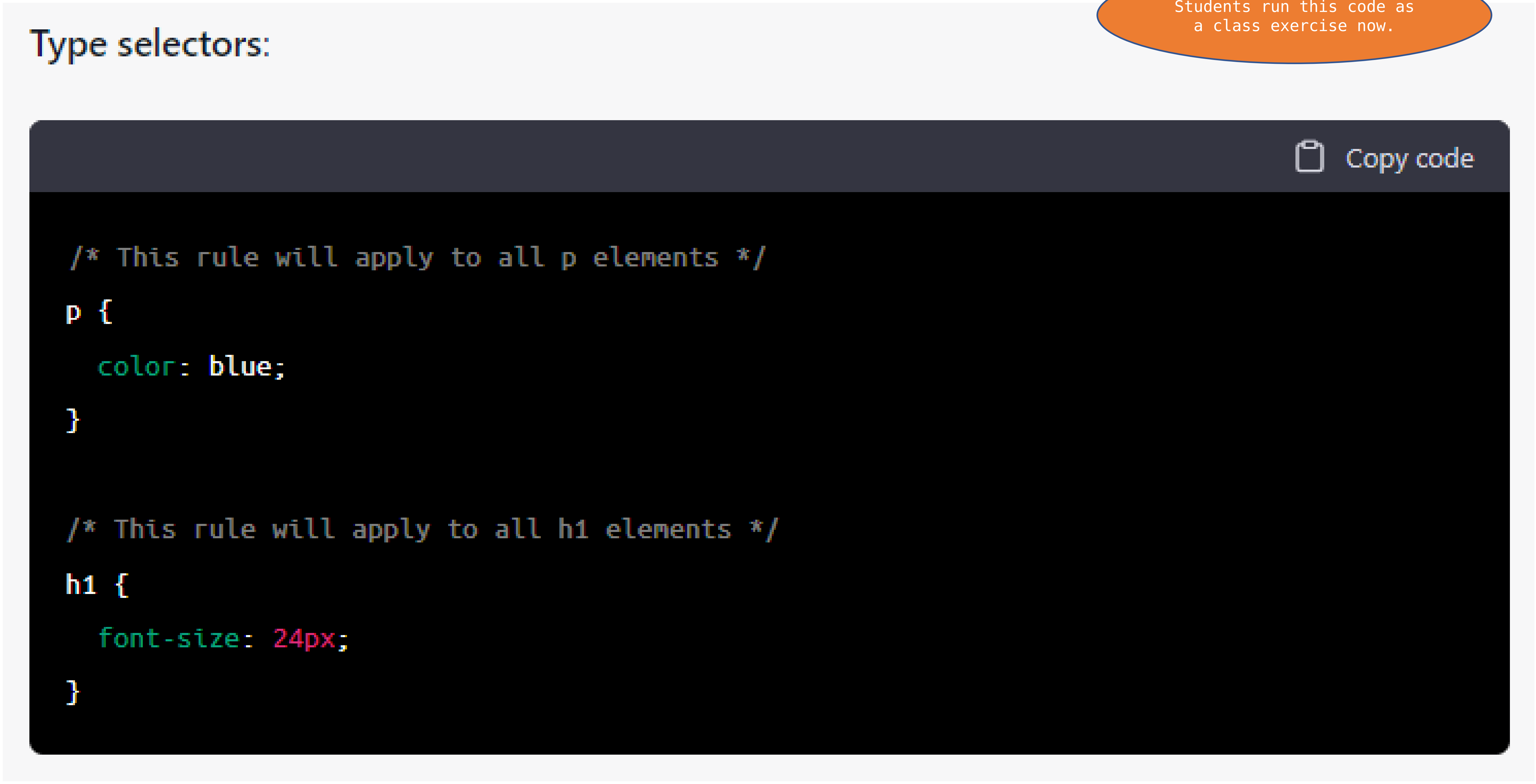
Type selectors: These selectors match elements based on the element name. For example, the type selector p would match all p elements in the HTML document.

Class selectors: These selectors match elements based on the class attribute. Class selectors are denoted by a period (.) followed by the class name. For example, the class selector .example would match all elements with a class of "example".

ID selectors: These selectors match elements based on the ID attribute. ID selectors are denoted by a hash symbol (#) followed by the ID name. For example, the ID selector #main-content would match the element with an ID of "main-content".

There are many other types of CSS selectors as well, such as attribute selectors, pseudo-class selectors, and pseudo-element selectors.

Students run this code as a class exercise now.

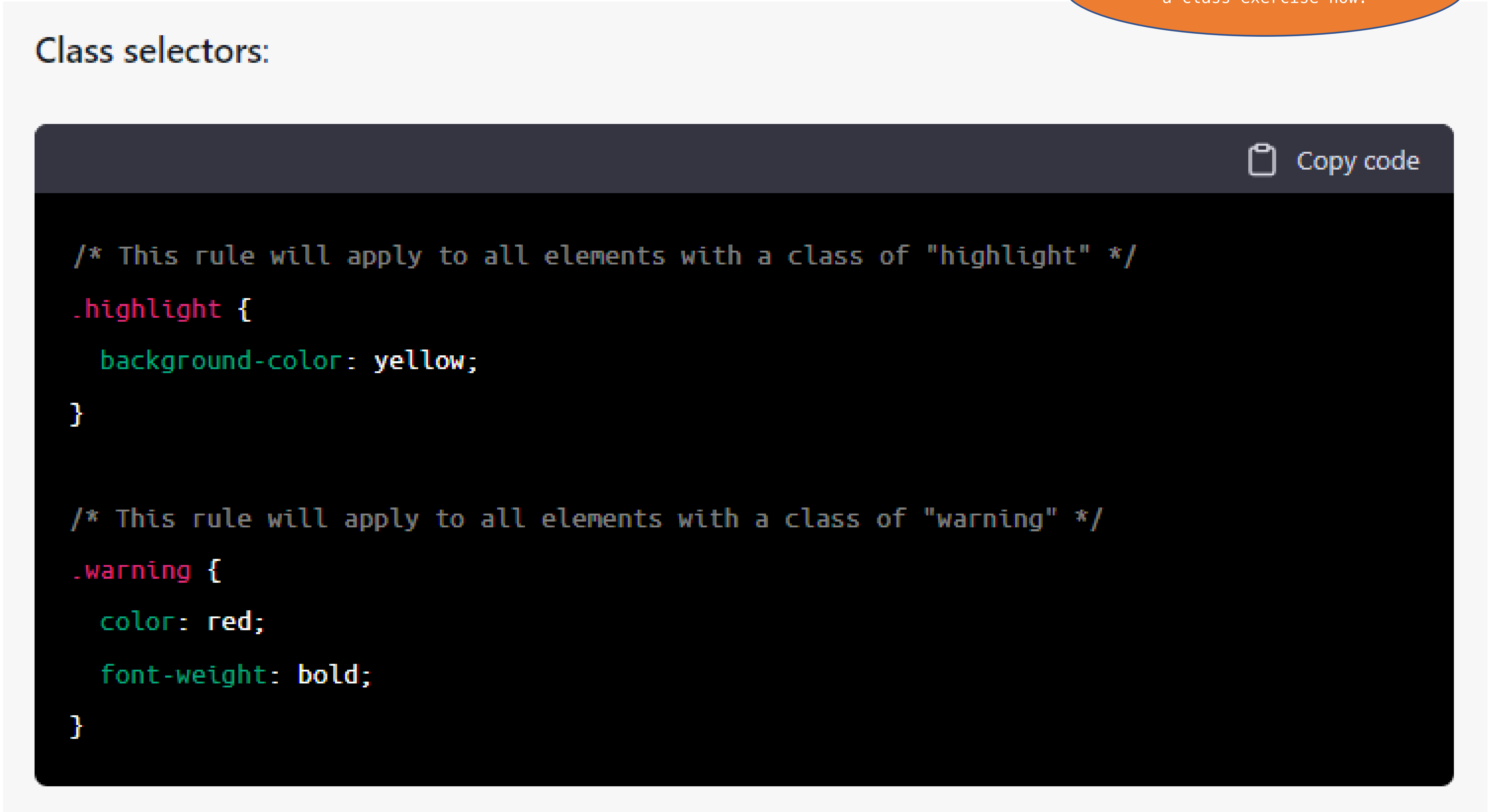


Type selectors:

```css
/* This rule will apply to all p elements */
p {
  color: blue;
}

/* This rule will apply to all h1 elements */
h1 {
  font-size: 24px;
}
```

Students run this code as a class exercise now.



Class selectors:

Copy code

```css
/* This rule will apply to all elements with a class of "highlight" */
.highlight {
  background-color: yellow;
}

/* This rule will apply to all elements with a class of "warning" */
.warning {
  color: red;
  font-weight: bold;
}
```

they work only if you've marked up your document appropriately, so using them generally involves a little forethought and planning.

For example, say a document contains a number of warnings. You want each warning to appear in boldface text so that it will stand out. However, you don't know which elements these warnings will be. Some warnings could be entire paragraphs, while others could be a single item within a lengthy list or a small section of text. So, you can't define a rule using element selectors of any kind. Suppose you tried this route:

```
p {
  font-weight: bold;
  color: red;
}
```

All paragraphs would be red and bold, not just those that contain warnings. You need a way to select only the text that contains warnings, or more precisely, a way to select only those elements that are warnings. How do you do it? You apply styles to parts of the document that have been marked in a certain way, independent of the elements involved, by using class selectors.

Class Selectors

The most common way to apply styles without worrying about the elements involved is to use *class selectors*. Before you can use them, however, you need to modify your actual document markup so that the class selectors will work. Enter the `class` attribute:

```
<p class="warning">When handling plutonium, care must be taken to avoid
the formation of a critical mass.</p>
<p>With plutonium, <span class="warning">the possibility of implosion is
very real, and must be avoided at all costs</span>. This can be accomplished
by keeping the various masses separate.</p>
```

To associate the styles of a class selector with an element, you must assign a `class` attribute the appropriate value. In the previous code block, a `class` value of `warning` was assigned to two elements: the first paragraph and the `span` element in the second paragraph.

All you need now is a way to apply styles to these classed elements. In HTML documents, you can use a compact notation where the name of a `class` is preceded by a period (`.`) and can be joined with an element selector:

```
.warning {font-weight: bold;}
```

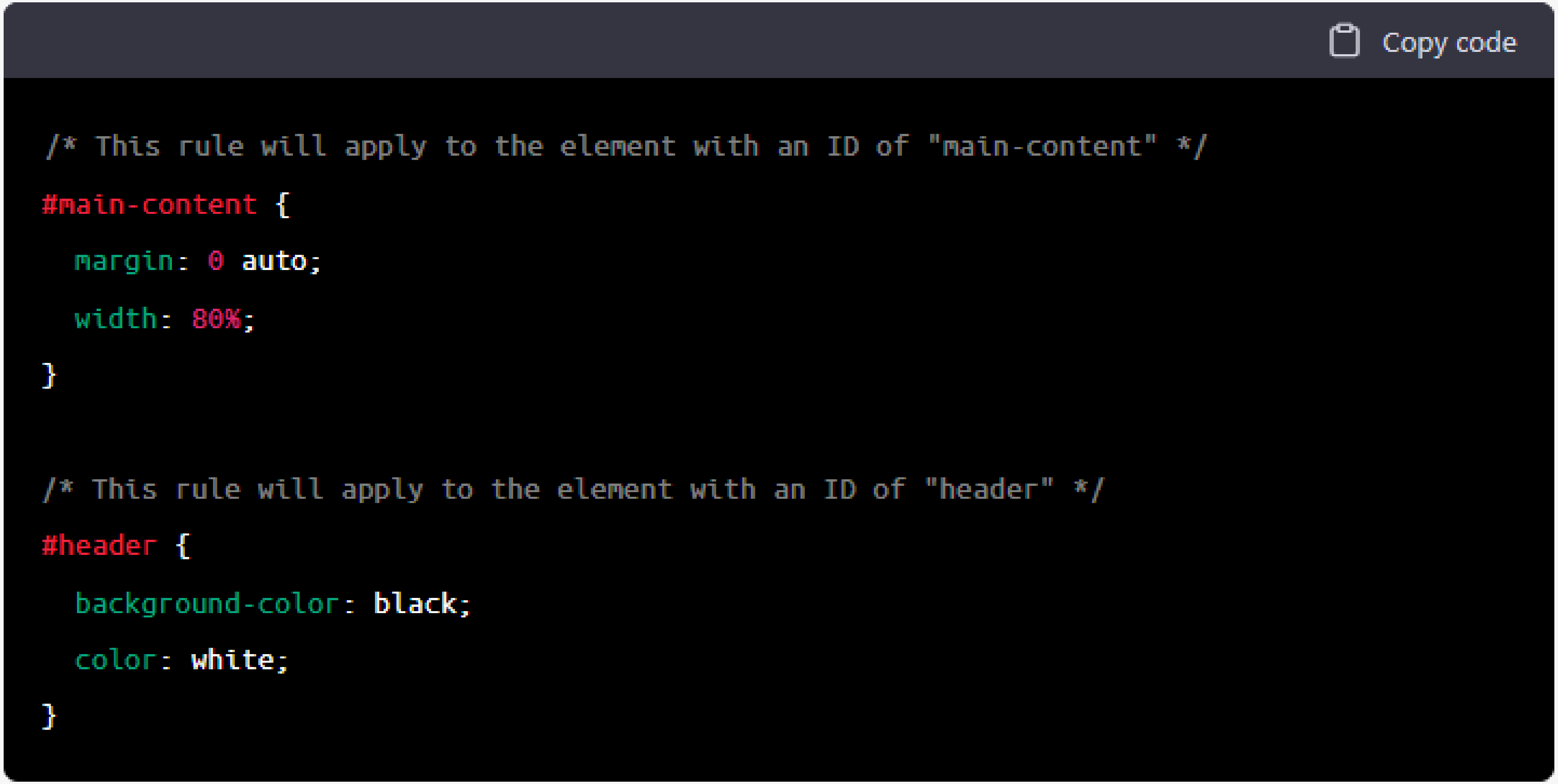
When combined with the example markup shown earlier, this simple rule has the effect shown in Figure 2-6. That is, the declaration `font-weight: bold` will be applied to every element (thanks to the presence of the implicit universal selector) that carries a `class` attribute with a value of `warning`.

NOTE

The universal selector, represented by `*`, is implied when an ID, class, attribute selector, pseudo-class or pseudo-element selector is written without being attached to an element selector.

ID selectors:

Students run this code as a class exercise now.

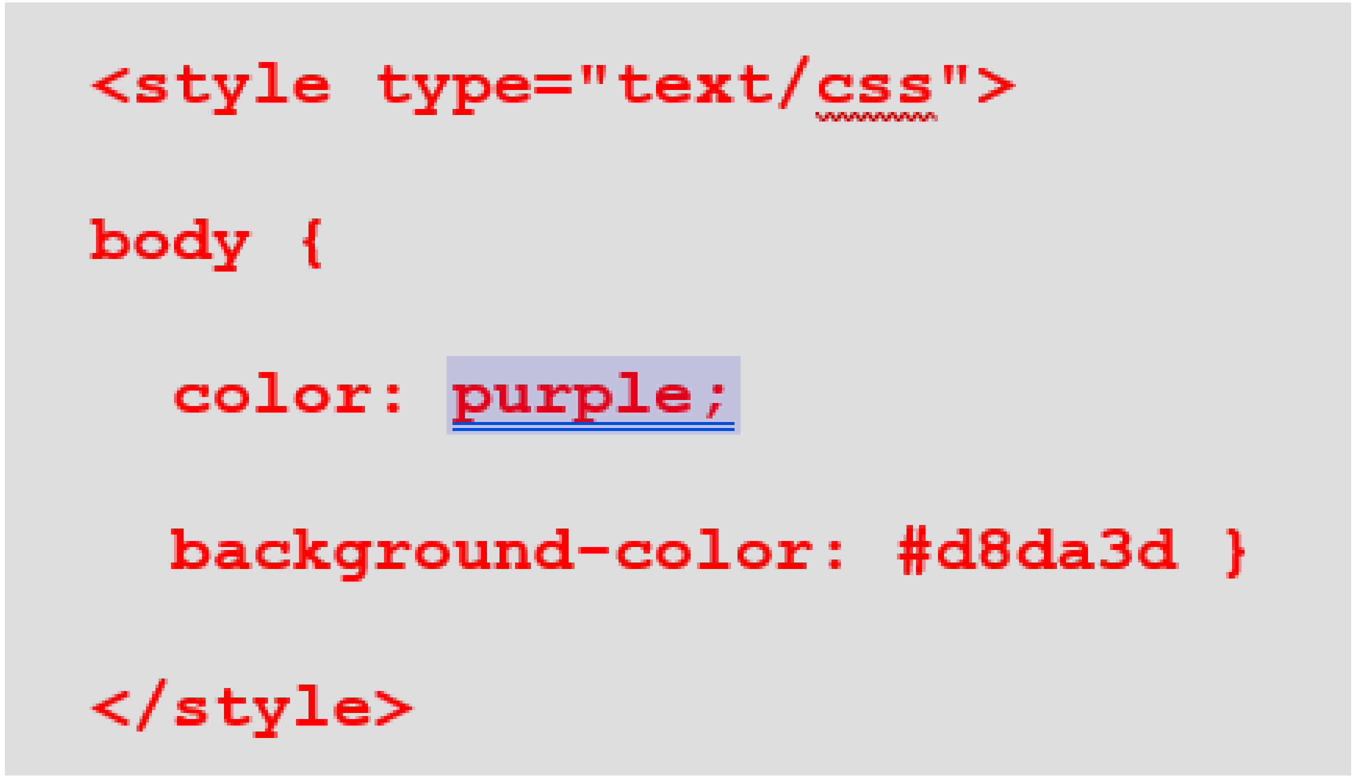


Copy code

```css
/* This rule will apply to the element with an ID of "main-content" */
#main-content {
  margin: 0 auto;
  width: 80%;
}

/* This rule will apply to the element with an ID of "header" */
#header {
  background-color: black;
  color: white;
}
```

What will be the effect of this CSS rule?



```
<style type="text/css">

body {

   color: purple;

   background-color: #d8da3d }

</style>
```

A CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document.

A style rule is made of three parts:

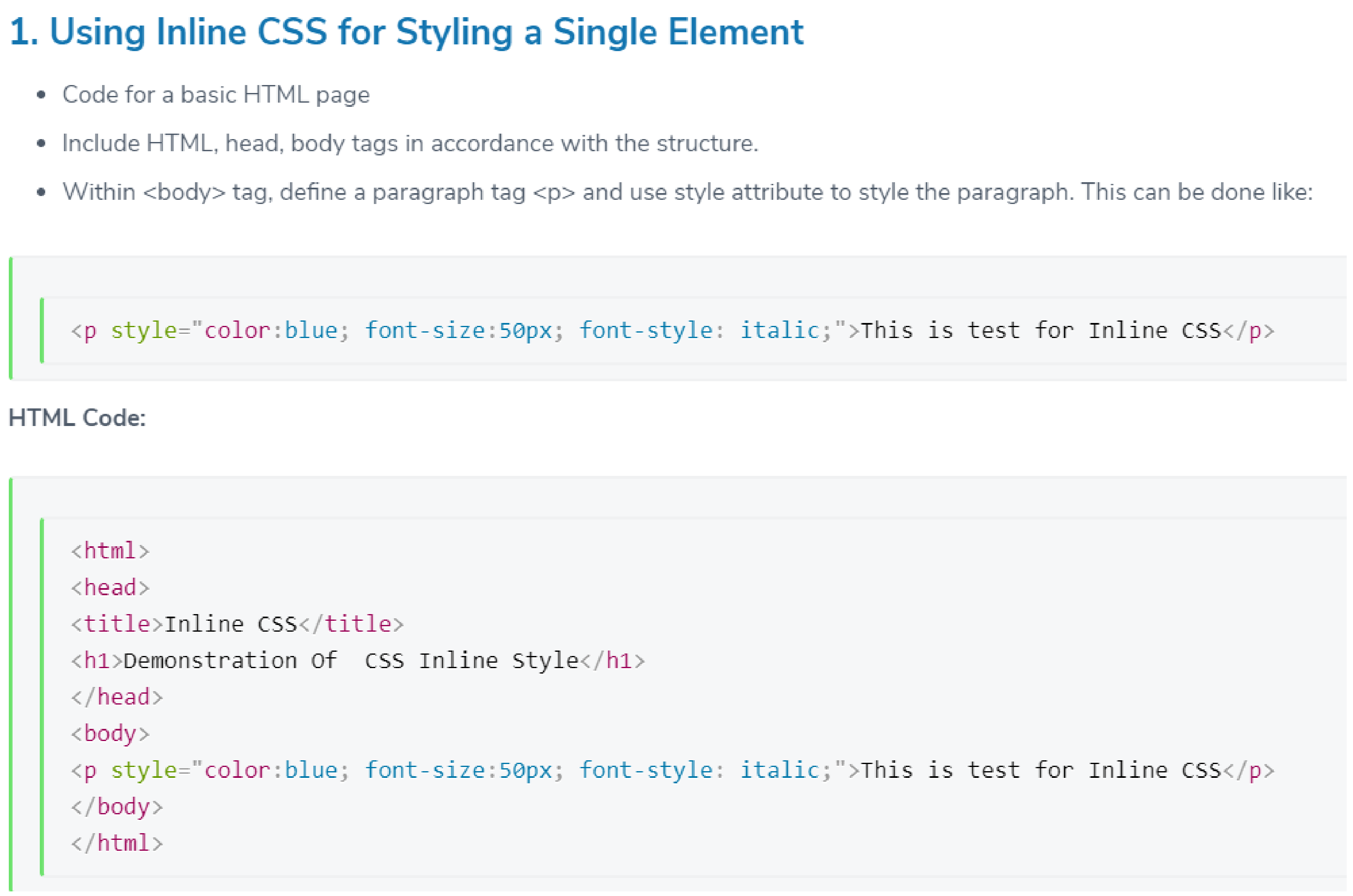
Selector: A selector is a Class, ID, or HTML tag at which a style will be applied.

Property: A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border, etc.

Value: Values are assigned to properties. For example, color property can have the value either red or #F1F1F1 etc.

CSS can be inlined with the NODE it is applied to

Students run this code as a class exercise now.



1. Using Inline CSS for Styling a Single Element

- Code for a basic HTML page
- Include HTML, head, body tags in accordance with the structure.
- Within <body> tag, define a paragraph tag <p> and use style attribute to style the paragraph. This can be done like:

```
<p style="color:blue; font-size:50px; font-style: italic;">This is test for Inline CSS</p>
```

HTML Code:

```
<html>
<head>
<title>Inline CSS</title>
<h1>Demonstration Of  CSS Inline Style</h1>
</head>
<body>
<p style="color:blue; font-size:50px; font-style: italic;">This is test for Inline CSS</p>
</body>
</html>
```

CSS works with the HTML <script> tag

- A standard way to use CSS is to apply selectors to the NODES in the HTML DOM
- You put your CSS RULES either in the:

  Same file as the HTML code, in

  ```
  <script>
  </script>
  ```

  OR

  You can put the CSS in a separate file and reference it as follows:

  ```
  <script src=“filename.css”/>
  <script src=“http://remoteserver.com/filename.css”/>
  ```

CSS works with the HTML <script> tag

- You can import several CSS libraries into one DOM.
- At work, the company will probably have standard CSS design rules you must use for branding purposes. For example, there is a specific shade of IBM Blue that all IBM branding must display. The graphics department creates these. Customarily, you will point to a URL in src= so that when changes are make, those changes will be picked up automatically without you needed to manually replace all those files.
- There could be conflicts. If this happens, generally the first rule loaded is the one that gets applied. This is brower dependent. At work, be conscientious and be sure to study and know the contents of your scripts.

CSS flex and CSS grid as two of the most popular layout tools

Flex is a one-dimensional layout tool, while grid is a two-dimensional layout tool.

Flex is best suited for laying out items in a single direction (either horizontally or vertically), while **grid is better for creating more complex layouts with multiple rows and columns.**

CSS grid is the layout tool that allows you to create flexible and responsive layouts using a grid system.

CSS Exercises: Access the Code Along Document and run CSS Recipes 1, 2, and 3

https://docs.google.com/document/d/1QizIxaVYHfPR8cVjV381RxeTO4lUieX4ZW22g0Kg9nw/edit?usp=sharing

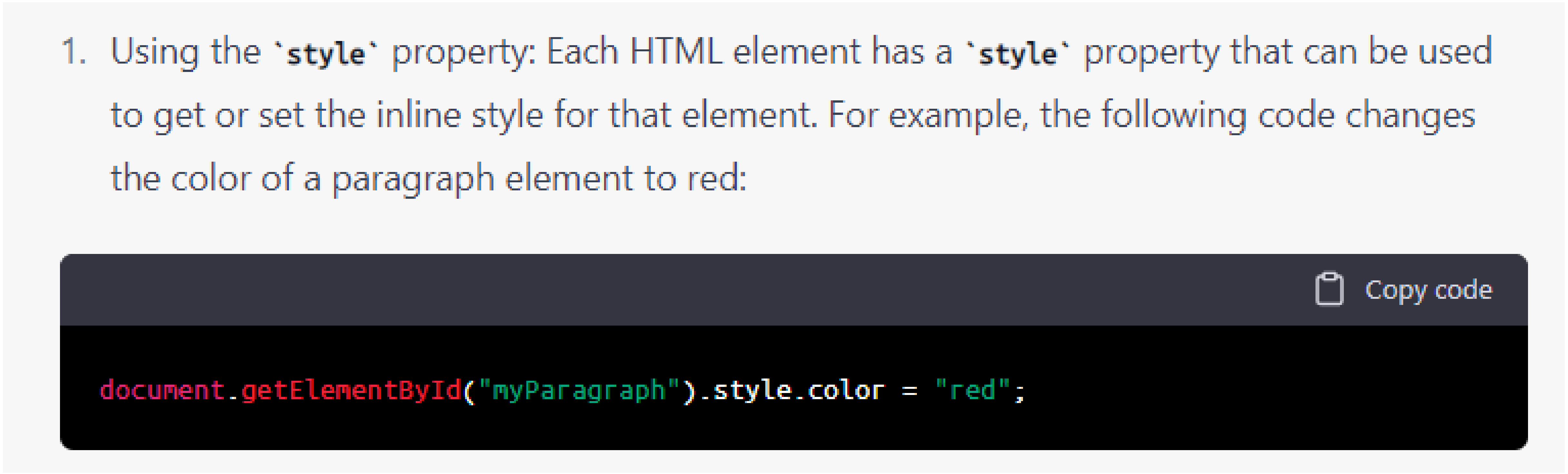
Output: For each exercise, View the output in your local development web server and use Chromium Dev Tools to observe the BOM.

CSS Exercises 2 : Access the Lab Notebook: “CSS - Tutorial and Lab Exercises Notebook”

Output: For each exercise, View the output in your local development web server and use Chromium Dev Tools to observe the BOM.

Creating Dynamic Websites with JavaScript

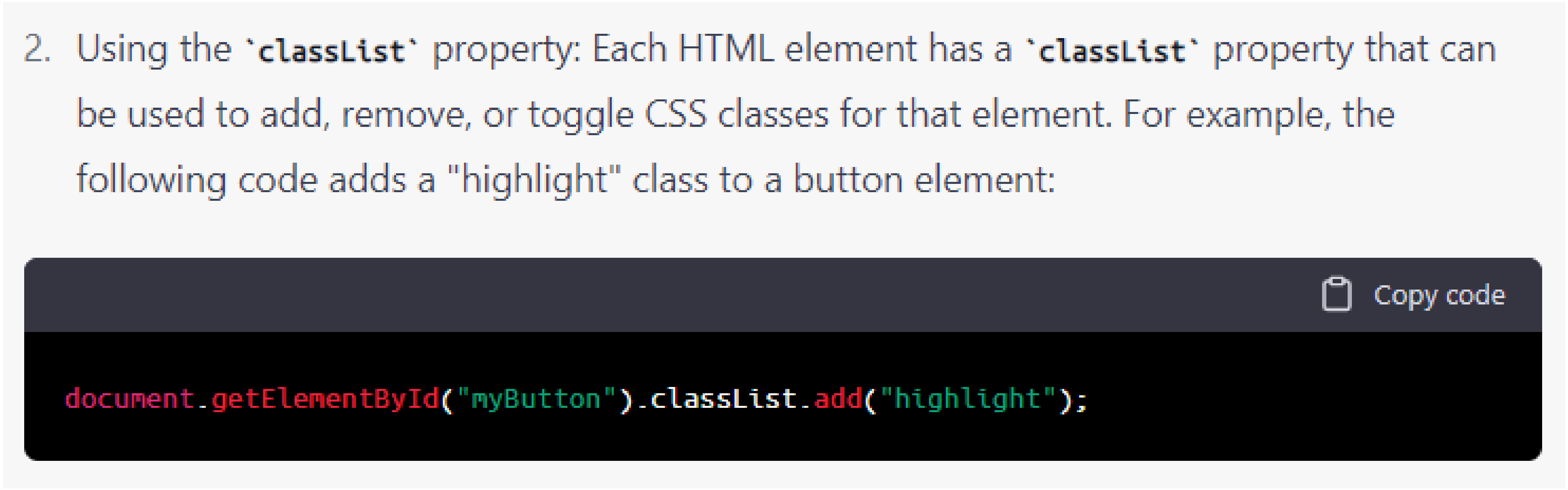
JavaScript can be used to apply CSS to an HTML document



1. Using the `style` property: Each HTML element has a `style` property that can be used to get or set the inline style for that element. For example, the following code changes the color of a paragraph element to red:

```
document.getElementById("myParagraph").style.color = "red";
```

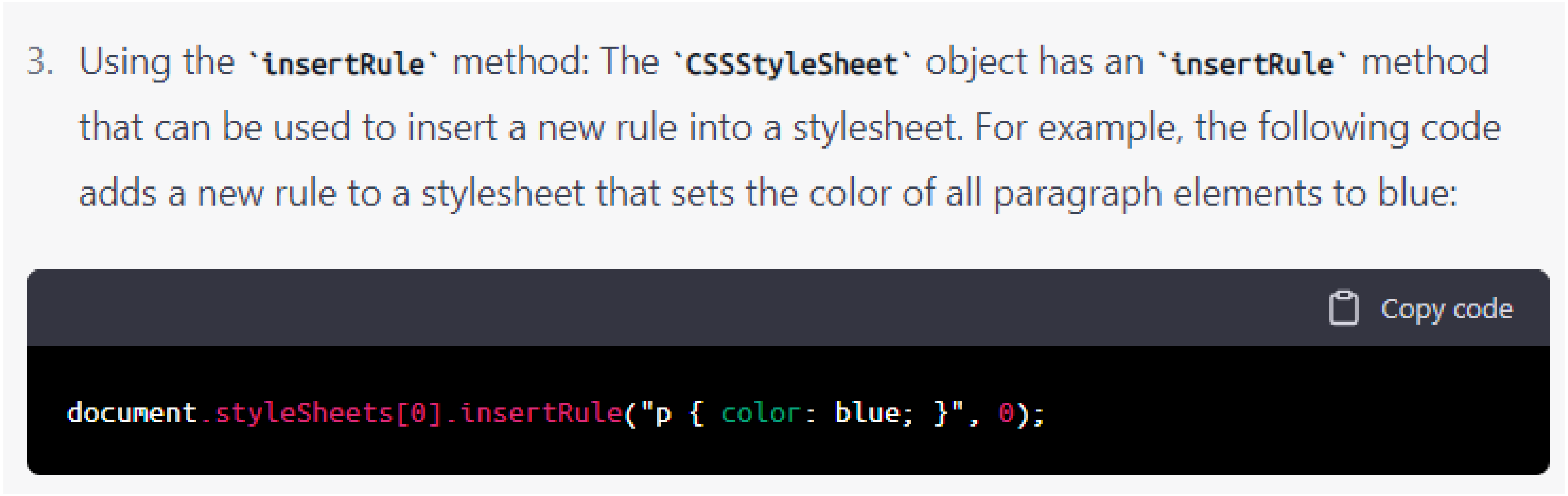
JavaScript can be used to apply CSS to an HTML document



2. Using the `classList` property: Each HTML element has a `classList` property that can be used to add, remove, or toggle CSS classes for that element. For example, the following code adds a "highlight" class to a button element:

```
document.getElementById("myButton").classList.add("highlight");
```

JavaScript can be used to apply CSS to an HTML document

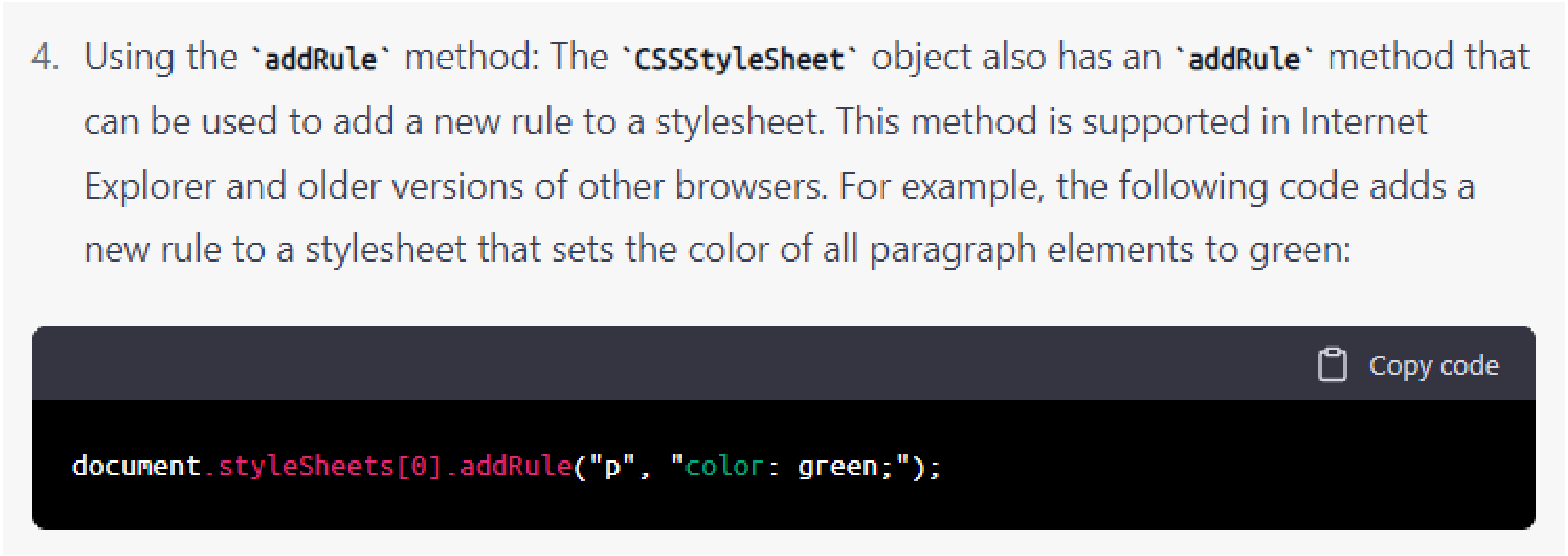


3. Using the `insertRule` method: The `CSSStyleSheet` object has an `insertRule` method that can be used to insert a new rule into a stylesheet. For example, the following code adds a new rule to a stylesheet that sets the color of all paragraph elements to blue:

Copy code

```
document.styleSheets[0].insertRule("p { color: blue; }", 0);
```

JavaScript can be used to apply CSS to an HTML document



4. Using the `addRule` method: The `CSSStyleSheet` object also has an `addRule` method that can be used to add a new rule to a stylesheet. This method is supported in Internet Explorer and older versions of other browsers. For example, the following code adds a new rule to a stylesheet that sets the color of all paragraph elements to green:

```
document.styleSheets[0].addRule("p", "color: green;");
```

HTML Coding Activity 1

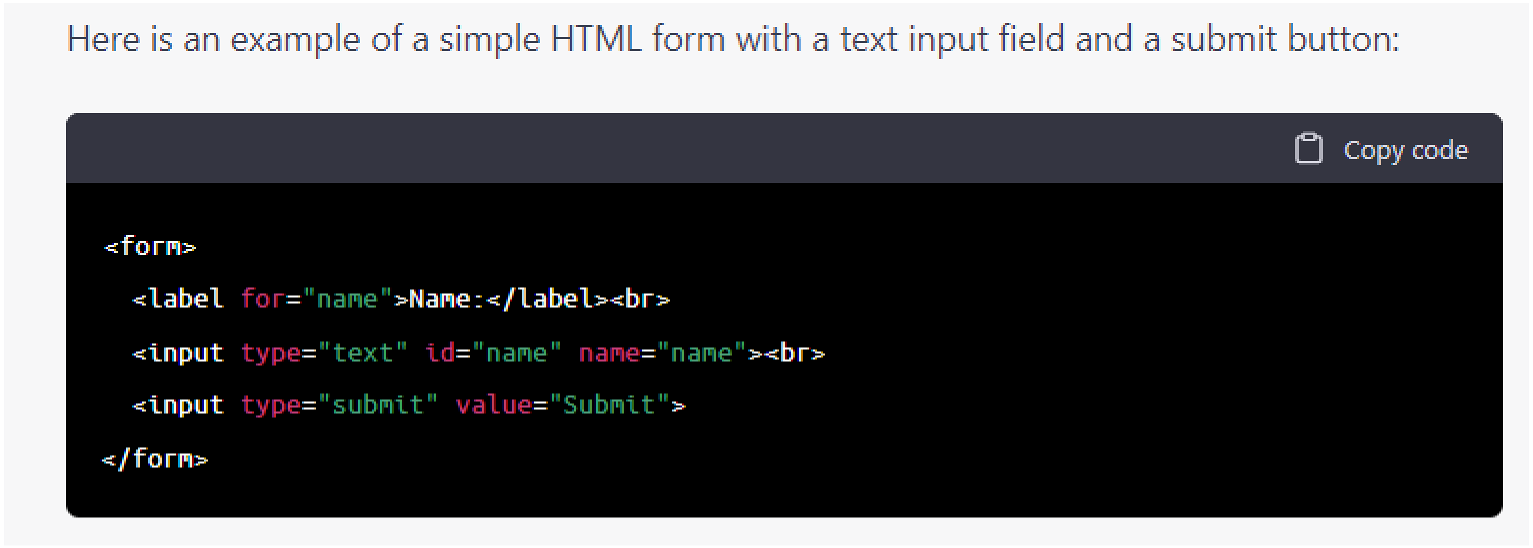
W3Schools.com is an excellent reference for HTML and other markup languages.

Client – Server Interaction

An HTML form is a way to collect input from a user on a web page. Forms are created using the form element in HTML and are typically used to send data to a server for processing (e.g., submitting a search query, filling out a survey, etc.).

A form consists of one or more input fields, which can be text boxes, checkboxes, radio buttons, etc. Users can enter data into these fields and submit the form by clicking a button. The data is then sent to the server, where it can be processed and stored.

Forms are an important part of the World Wide Web and are widely used to allow users to interact with websites and applications.



Here is an example of a simple HTML form with a text input field and a submit button:

```html
<form>
  <label for="name">Name:</label><br>
  <input type="text" id="name" name="name"><br>
  <input type="submit" value="Submit">
</form>
```

But, you are now wondering:

What will happen with this information?

It will be passed to the SERVER via a POST or GET request.

The mechanisms for transmitting form data to a server.

As you may know, forms are a common mechanism for collecting input from users on web pages.

When a user submits a form, the data is typically sent to a server for processing.

There are two main ways in which this data can be transmitted to the server: via a POST request or a GET request.

A POST request is a type of HTTP request that is used to send data to a server for processing. The data is included in the body of the request and is not visible to the user. POST requests are typically used when the data being sent to the server is sensitive or large, as they are not limited by the size of the URL. Post field name/value pairs are embedded in the TCP PIPE that exists between client and server. You could use a tool like Wireshark to sniff the data in the packets, but POST is much more secure than GET.

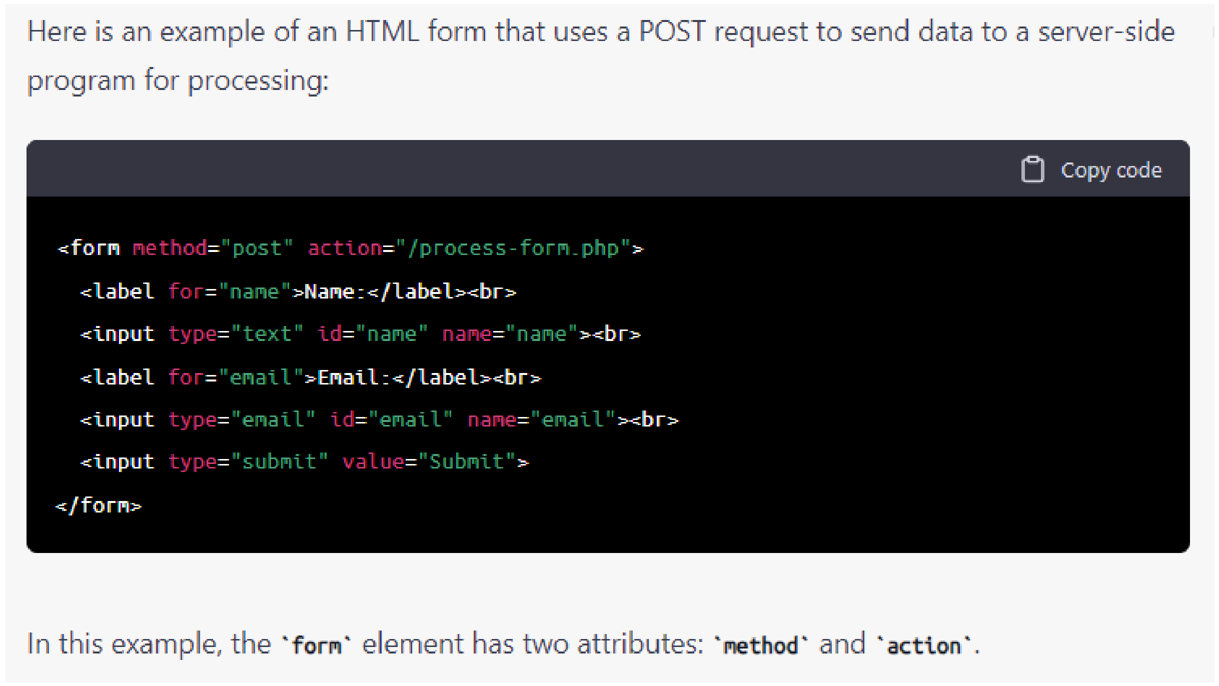
A GET request is another type of HTTP request that is used to retrieve data from a server. When a user submits a form using a GET request, the data is included in the URL as a query string. GET requests are typically used when the data being sent to the server is not sensitive and is relatively small. Get field name/value pairs are encoded on the URL.

It is also important to note that when data is transmitted to a server, it is often passed through

A POST request is a type of HTTP request that is used to send data to a server for processing.

- The data is included in the body of the request and is not visible to the user. This means that the data is not visible in the URL or as part of the request headers, which makes it more secure than a GET request.
- POST requests are typically used when the data being sent to the server is sensitive or large. This is because POST requests are not limited by the size of the URL, which is a common limitation of GET requests. Instead, the data is transmitted in the body of the request, which allows for more data to be sent at once.
- When sending data via a POST request, the data is usually sent as name/value pairs. These pairs are embedded in the Transmission Control Protocol (TCP) connection that exists between the client (e.g., a web browser) and the server. This data can be accessed by the server and used for various purposes, such as storing information in a database or processing a search query.
- While it is possible to "sniff" the data in the packets transmitted during a POST request (for example, using a tool like Wireshark), POST requests are generally considered to be more secure than GET requests. This is because the data is not visible in the URL, which makes it less vulnerable to tampering or interception.

Writing Programs to service Forms


Here is an example of an HTML form that uses a POST request to send data to a server-side program for processing:

```
<form method="post" action="/process-form.php">
  <label for="name">Name:</label><br>
  <input type="text" id="name" name="name"><br>
  <label for="email">Email:</label><br>
  <input type="email" id="email" name="email"><br>
  <input type="submit" value="Submit">
</form>
```

In this example, the `form` element has two attributes: `method` and `action`.

In this example, the form element has two attributes: method and action.

The method attribute specifies the request method that will be used to transmit the form data to the server. In this case, the value is "post", which indicates that a POST request will be used.

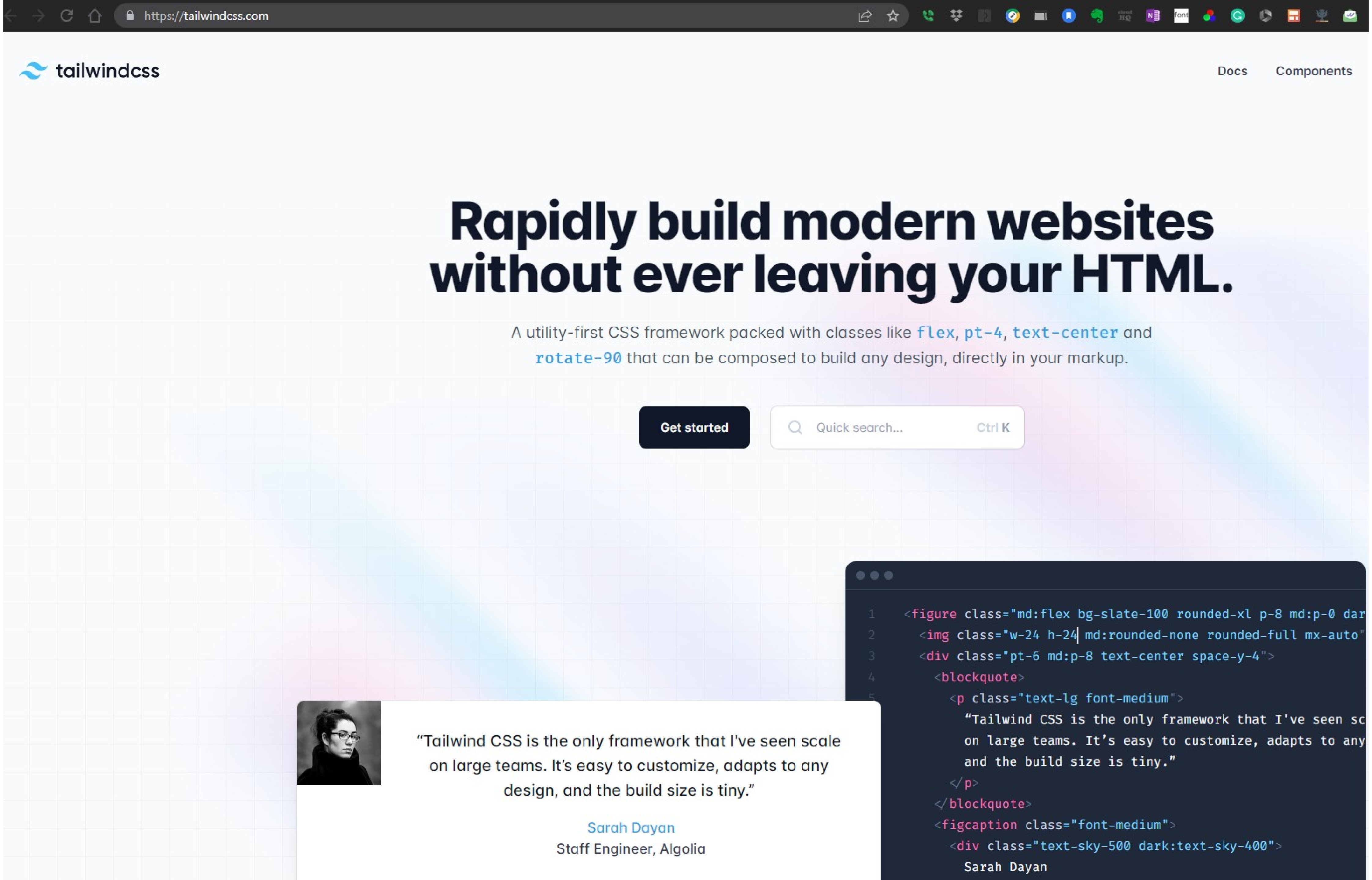
The action attribute specifies the server-side program that will process the form inputs. In this case, the value is "/process-form.php", which is the URL of the program on the server.

The form also includes two input fields: a text field for the user's name and an email field for the user's email address. When the user submits the form, the data entered in these fields will be sent to the server-side program specified in the action attribute via a POST request.

Client apps
Server Apps: program runs on the server: receives form data and runs and send back HTML to client

Client will always be a Markup Language: HTML

Server can run any programming language: receive input : generate output in the form of HTML sent back to the server



https://tailwindcss.com

tailwindcss

Docs Components

Rapidly build modern websites without ever leaving your HTML.

A utility-first CSS framework packed with classes like `flex`, `pt-4`, `text-center` and `rotate-90` that can be composed to build any design, directly in your markup.

Get started

Quick search... Ctrl K

"Tailwind CSS is the only framework that I've seen scale on large teams. It's easy to customize, adapts to any design, and the build size is tiny."

Sarah Dayan
Staff Engineer, Algolia

```
<figure class="md:flex bg-slate-100 rounded-xl p-8 md:p-0 dar
  <img class="w-24 h-24 md:rounded-none rounded-full mx-auto"
  <div class="pt-6 md:p-8 text-center space-y-4">
    <blockquote>
      <p class="text-lg font-medium">
        "Tailwind CSS is the only framework that I've seen sc
        on large teams. It's easy to customize, adapts to any
        and the build size is tiny."
      </p>
    </blockquote>
    <figcaption class="font-medium">
      <div class="text-sky-500 dark:text-sky-400">
        Sarah Dayan
```

JavaScript is a programming language that enables us to add interactivity and dynamic functionality to websites.

JavaScript is a scripting language, meaning that it runs inside the execution context of a services-providing platform and allows us to extend and customize the operation of that platform. For example, you can think of Visual Basic for Applications, which lives inside Excel and allows programming Excel applications.

JavaScript can manipulate the data values and event handling of objects in the browser object model (BOM), such as buttons and fields, and set data values such as color and opacity. It can also programmatically apply CSS rules to selector-specified DOM nodes, allowing us to control the appearance and layout of web pages.

In this way, JavaScript enables user interaction with your Web Sites and makes them more user-friendly and engaging.

JavaScript:

JavaScript is a programming language

JavaScript can be run in 2 contexts

1. Inside the Web browser: Because Chromium has built into it a JavaScript Interpreter called Chromium.
2. At the command line: Node.js

Ways to Store Data
Data Storage:

Simple Variables

Complex Data Structures

Flow of Control
Ways to implement Algorithms

Branching if / then
Looping
Reusable code blocks (methods)

JavaScript is what lets our Web Pages be responsive to Events

- Events:
- onBlur, onFocus
- onMouseOver
- onClick

Console Object Model / **Browser** Object Model

- BOM -> Created by Chrome or other Web browsers, BOM provides JavaScript Code with “Hooks” to access the internal details of the HTML DOM object Model living in the Browser.

- COM: Console Object Model: created by Node.js: COM gives us HOOKS to elements available at the command line (file system, CPU process, sensors on the motherboard).