Frontend Technologies

Web Development

Entering in Web World

TRAINING AGENDA

A High-Level Overview of Web Development
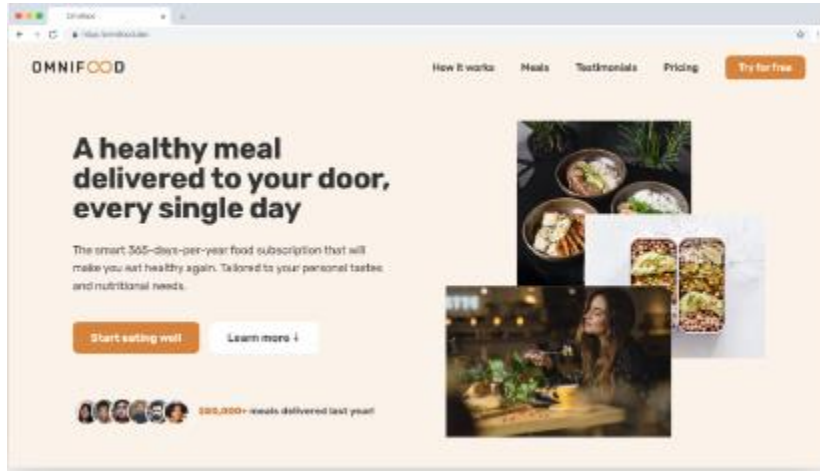
Introduction to HTML, CSS & JavaScript

HTML Elements – Block and Text

Overview of Web Design and Website Personalities

Responsive Design Principles

HTML APIs –Storage, Socket, Canvas, Drag/Drop, Audio/Video etc
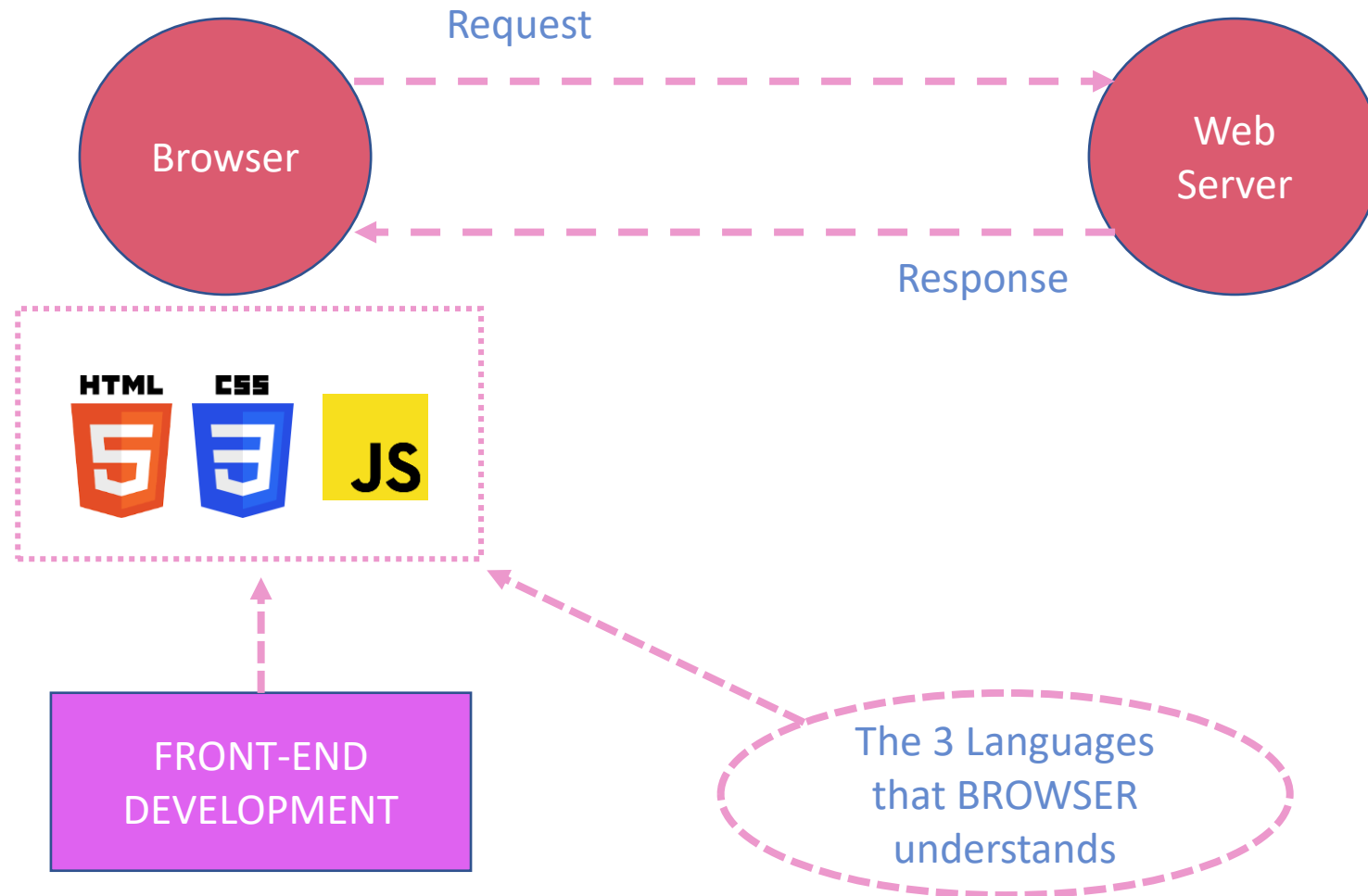
FRONTEND VS BACKEND DEVELOPMENT
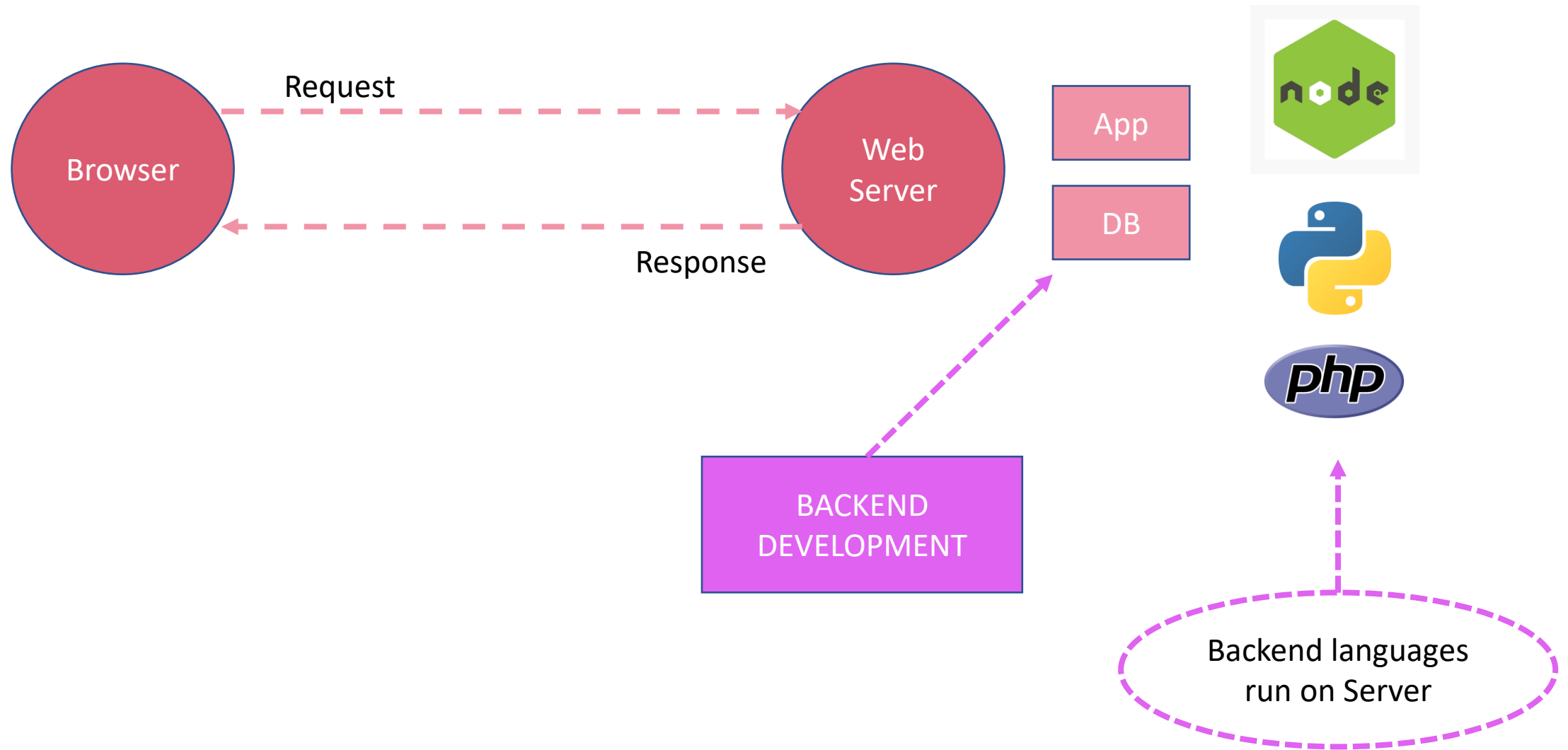
index.html

styles.css

script.js

Images / links

FRONTEND VS BACKEND DEVELOPMENT

Request

Browser

Web
Server

Response

HTML CSS JS

FRONT-END
DEVELOPMENT

The 3 Languages
that BROWSER
understands

FRONTEND VS BACKEND DEVELOPMENT

THE 3 LANGUAGES OF FRONT-END

ADJECTIVES

p { color : 'red' }
means the paragraph
text is *red*

NOUNS

<p></p>
means
"paragraph"

VERBS

p.hide()
means hide the
paragraph

CSS

HTML

JS

FRONTEND
DEVELOPMENT
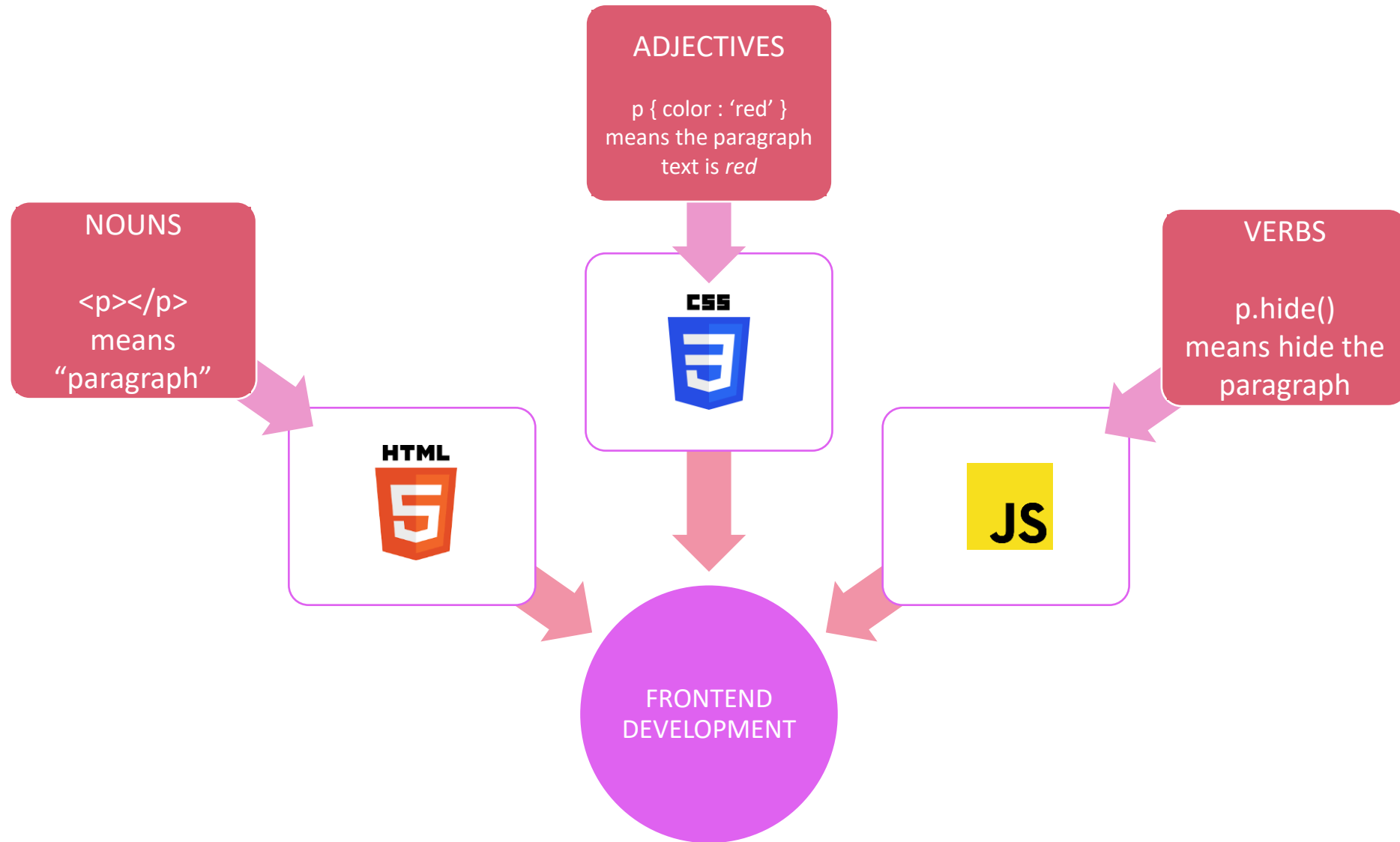
INTRODUCTION TO HTML

HyperText Markup Language

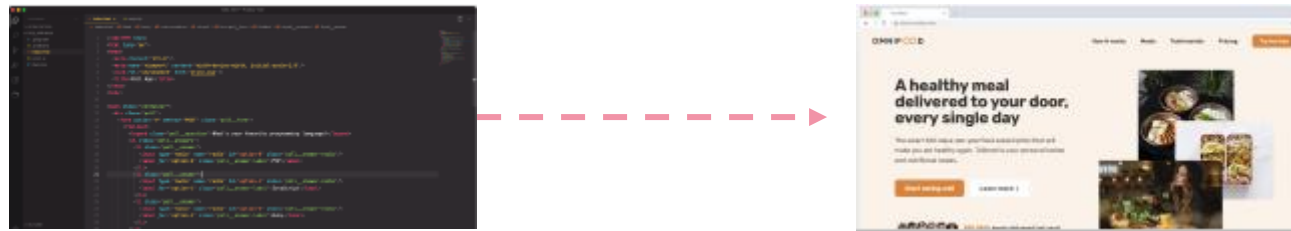HTML is a markup language that web developers use to structure and describe the content of a webpage (not a programming language)

HTML consists of elements that describe different types of content: paragraphs, links, headings, images, video, etc.

Web browsers understand HTML and render HTML code as websites

PROTOCOLS

A protocol is a set of rules that define how communication occurs in a network

Internet Protocol (IP)	Ensures the successful delivery of packets from source to destination
Transmission Control Protocol (TCP)	TCP is a connection-oriented reliable protocol. Provides reliable transmission of packets.
User Datagram Protocol (UDP)	UDP is a connectionless and unreliable protocol
Hypertext Transfer Protocol (HTTP)	Helping applications to communicate with the users
File Transfer Protocol (FTP)	A standard network protocol provided by TCP/IP, and it is used to transfer files from one server to another
Secure Shell (SSH)	Network protocol that uses cryptography in order to secure network services over unsecured networks

Hypertext Transfer Protocol (HTTP)

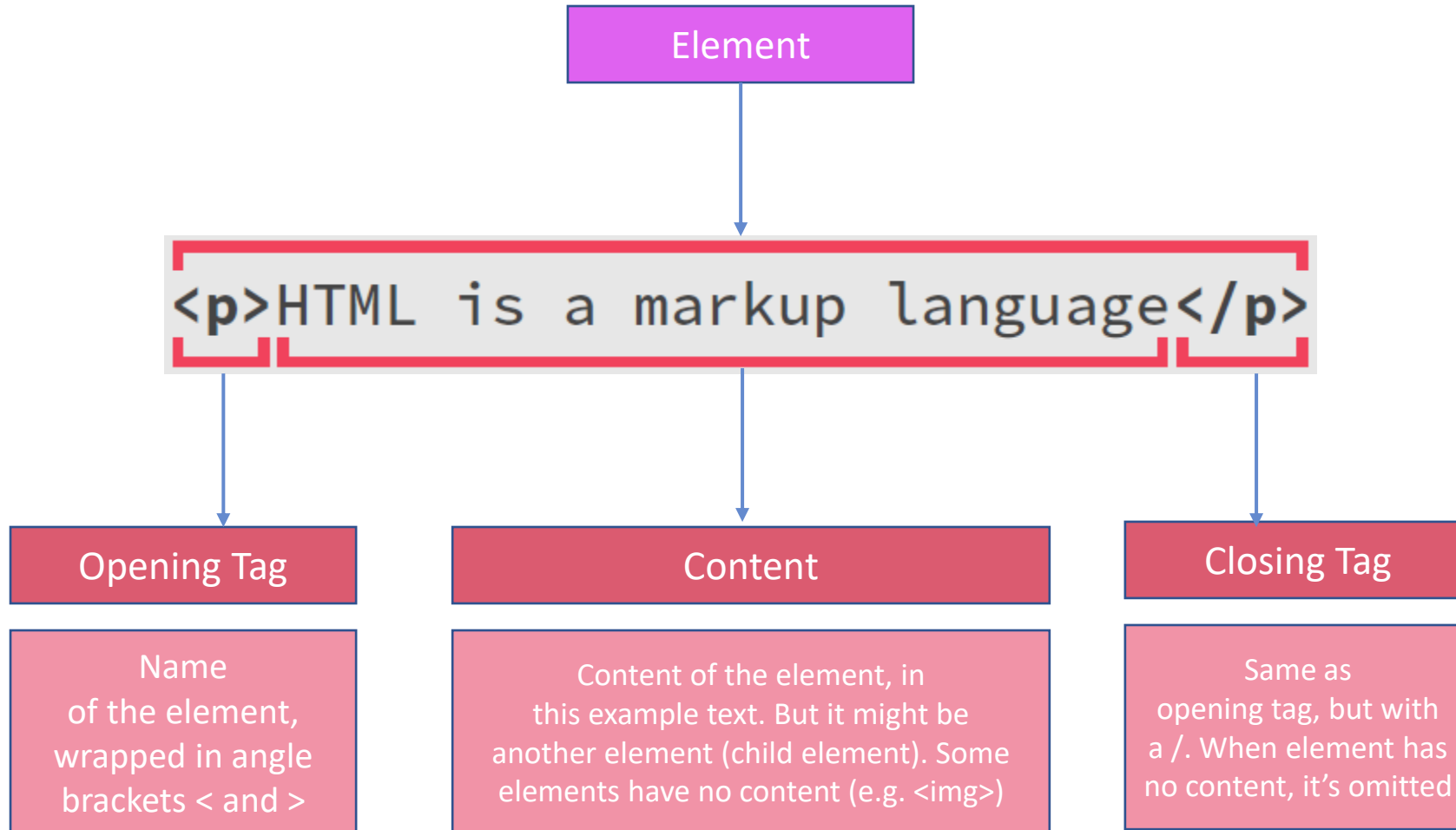The Hypertext Transfer Protocol (HTTP) is the foundation of the World Wide Web (WWW) and is used to load web pages using hypertext links

It allows the user to see a user-friendly interface designed to transfer information between networked devices

HTTP is a connectionless and stateless protocol. A client and server know each other only during communication

HTTP is also media independent. Any data can be sent through HTTP.

ANATOMY OF AN HTML ELEMENT

Element

`<p>HTML is a markup language</p>`

Opening Tag

Name
of the element,
wrapped in angle
brackets < and >

Content

Content of the element, in
this example text. But it might be
another element (child element). Some
elements have no content (e.g. <img>)

Closing Tag

Same as
opening tag, but with
a /. When element has
no content, it's omitted

HTML Elements

title	specifies the label that appears in the browser window's *title bar*.
meta	Provide information about the web page.
style	Writing internal stylesheet for your page.
script	Adding actions to your app by including JavaScript.
body	Includes various other elements to display data on webpage

Elements To Watch For...

blockquote

ul

div

li

p

ins/del

header

nav

footer

sup/sub

b

abbr

aside

pre

br

time

u

mark

span
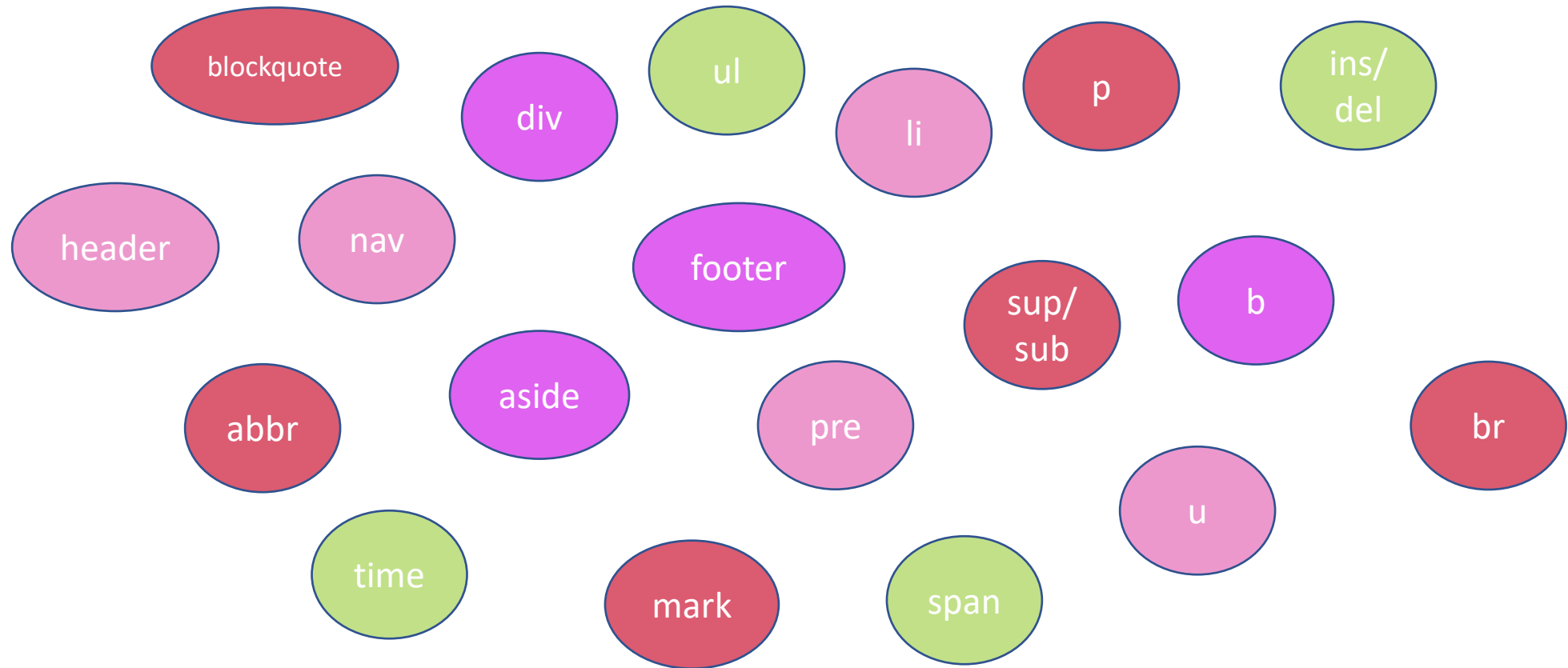
CHARACTER REFERENCES

Display a character that would otherwise be difficult to display.

Character	Character Reference	Description
<	<	less than
>	>	greater than
≤	≤	less than or equal
½	½	one-half
¼	¼	one-fourth
&	&	ampersand
"	"	quote
'	'	apostrophe
space		nonbreaking space
←	← ←	left arrow
•	·	bullet
✓	✓	check mark
©	©	copyright

THE TRIBUTE PAGE

You must include the following parts in your completed project:

- You must include a title or heading on the tribute page.
- You should include an image on the tribute page.
- You included image must have a caption.
- You should include a timeline of the life of the tribute in the form of a list

The Input (Form Input) Element

Accepting The User Inputs

<input> Elements

The <input> HTML element is used to create interactive controls for web-based forms in order to accept data from the user

Type	Description
button	A push button with no default behavior displaying the value of the value attribute, empty by default
checkbox	A check box allowing single values to be selected/deselected
color	A control for specifying a color; opening a color picker when active in supporting browsers
date	A control for entering a date (year, month, and day, with no time)
email	A field for editing an email address
file	A control that lets the user select a file
number	A control for entering a number
password	A single-line text field whose value is obscured

More On <input> Elements

Type	Description
radio	A radio button, allowing a single value to be selected out of multiple choices with the same name value
range	A control for entering a number whose exact value is not important
search	A single-line text field for entering search strings
submit	A button that submits the form
tel	A control for entering a telephone number
text	The default value. A single-line text field
time	A control for entering a time value with no time zone
url	A field for entering a URL

CSS

Let's make it better looking
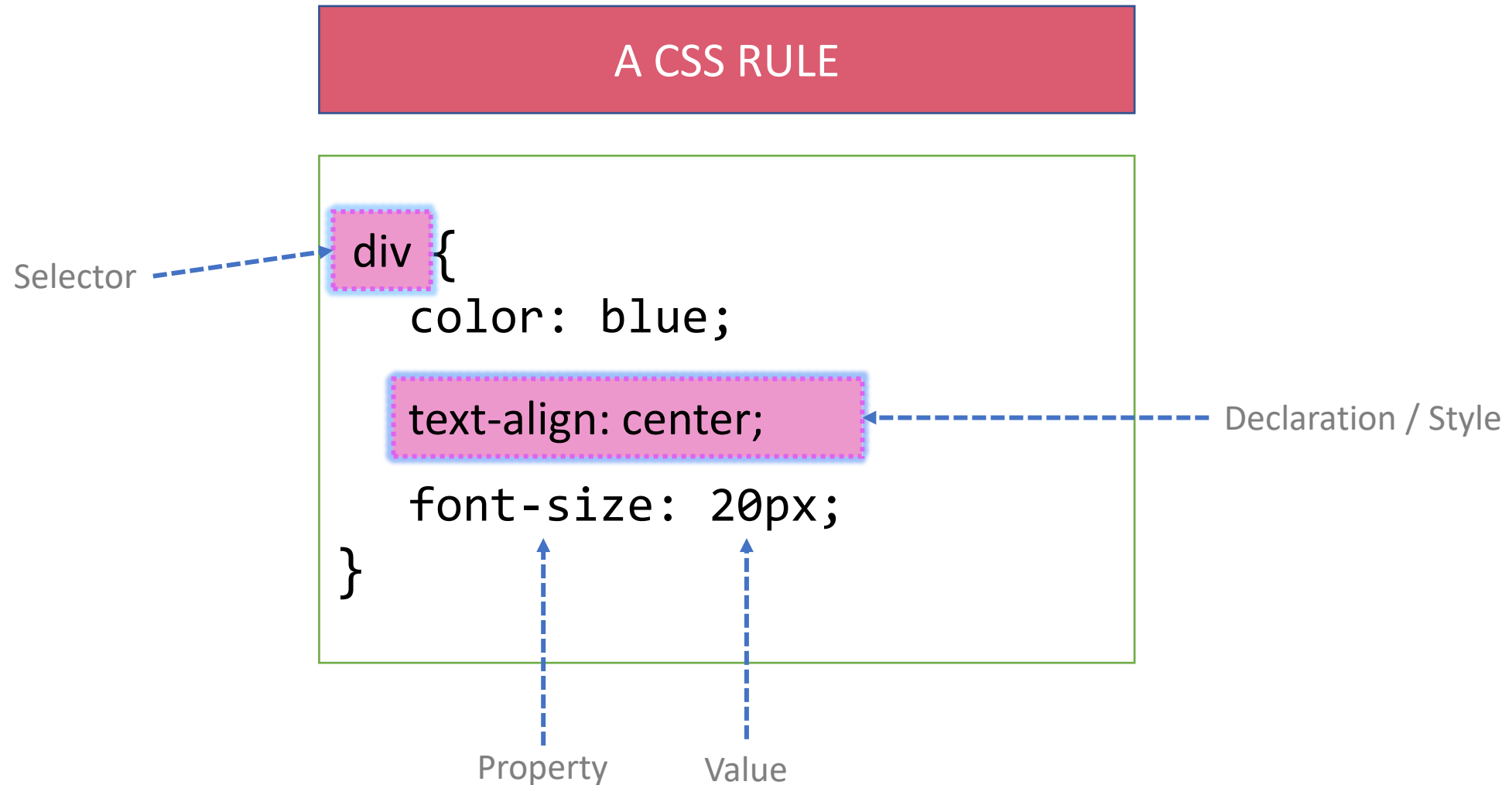
INTRODUCTION TO CSS

Cascading Style Sheets

CSS describes the visual style and presentation of the content written in HTML

CSS consists of countless properties that developers use to format the content: properties about font, text, spacing, layout, etc.

Web browsers understand HTML and render HTML code as websites

HOW WE SELECT AND STYLE ELEMENTS

A CSS RULE

Selector

```
div {
    color: blue;

    text-align: center;

    font-size: 20px;
}
```

Declaration / Style

Property   Value

CSS SELECTORS

Selector Type	Syntax	Description	Example
Universal	*	Selects all elements	* will match all the elements
Type	elementName	Selects all elements that have the given node name	input will match any <input> element.
Class	.classname	Selects all elements that have the given class attribute	.index will match any element that has a class of "index"
ID	#id	Selects an element based on the value of its id attribute	#toc will match the element that has the ID "toc"
Attribute	[attr]	Selects all elements that have the given attribute	[autoplay] will match all elements that have the autoplay attribute

MORE CSS SELECTORS

Selector Type	Syntax	Description	Example
Grouping Selector List	,	The , selector is a grouping method that selects all the matching nodes	div, span will match both <span> and <div> elements
Descendant combinator	" "	The " " (space) combinator selects nodes that are descendants of the first element	div span will match all <span> elements that are inside a <div> element
Child combinator	>	The > combinator selects nodes that are direct children of the first element	ul > li will match all elements that are nested directly inside a element.
General sibling combinator	~	The ~ combinator selects siblings. This means that the second element follows the first (though not necessarily immediately), and both share the same parent	p ~ span will match all <span> elements that follow a <p>, immediately or not.
Adjacent sibling combinator	+	The + combinator matches the second element only if it immediately follows the first element	h2 + p will match all <p> elements that immediately follow an <h2> element

MORE CSS SELECTORS

Selector Type	Syntax	Description	Example
Pseudo classes	:	The : pseudo allow the selection of elements based on state information that is not contained in the document tree	a:visited will match all <a> elements that have been visited by the user
Pseudo elements	::	The :: pseudo represent entities that are not included in HTML	p::first-line will match the first line of all <p> elements

Task

THE GOOD VIBES FORM

Forms are always an essential part of any project and you will be working with a lot of forms in most of the applications so why not practice it earlier and test your knowledge. Once you get familiar with the input field or basic tags in HTML to create a form make a project using all those tags. How to use a text field, checkbox, radio button, date, and other important elements in a single form. You will be learning how to give a proper structure to a webpage while creating a form. Knowledge of HTML/HTML5 is good enough but you can use a bit of CSS to make this project look better

HTML5 APIs'

Entering In The World Of Web Programming

AGENDA

History Management

Canvas

Web Storage

Audio / Video

Drag & Drop

WebSocket

Server-Sent Events

WEB STORAGE

The Web Storage API provides mechanisms by which browsers can store key/value pairs, in a much more intuitive fashion

Web Storage Type

sessionStorage maintains a separate storage area for each given origin that's available for the duration of the page session (as long as the browser is open, including page reloads and restores)

localStorage does the same thing, but persists even when the browser is closed and reopened.

sessionStorage vs. localStorage

sessionStorage	**localStorage**
Stores data only for a session, meaning that the data is stored until the browser (or tab) is closed.	Stores data with no expiration date, and gets cleared only through JavaScript, or clearing the Browser cache / Locally Stored Data.
Data is never transferred to the server.	
	Storage limit is the maximum amongst the two.
Storage limit is larger than a cookie (at most 5MB).	

HISTORY MANAGEMENT

The DOM Window object provides access to the browser's session history through the History object.

The Window History object methods and properties let you navigate back and forth through the user's history, and manipulate the contents of the history stack.

Moving backward and forward through the user's history is done using the back(), forward(), and go() methods.

The popstate event of the Window interface is fired when the active history entry changes while the user navigates the session history.

WORKING WITH CANVAS

The Canvas API provides a means for drawing graphics via JavaScript and the HTML <canvas> element.

Canvas can be used for animation, game graphics, data visualization, photo manipulation, and real-time video processing.

<canvas id="canvas"></canvas>

The HTMLCanvasElement.getContext() method gets that element's context—the thing onto which the drawing will be rendered.

DRAG & DROP

HTML Drag and Drop interfaces enable applications to use drag-and-drop features in browsers.

The user may select draggable elements with a mouse, drag those elements to a droppable element, and drop them by releasing the mouse button

You can customize which elements can become draggable, the type of feedback the draggable elements produce, and the droppable elements.

HTML drag-and-drop uses the DOM event model and drag events inherited from mouse events

DRAG & DROP EVENTS

Event	Event Handler	Fires when…
drag	ondrag	a dragged item (element or text selection) is dragged.
dragend	ondragend	a drag operation ends (such as releasing a mouse button or hitting the Esc key)
dragenter	ondragenter	a dragged item enters a valid drop target.
dragleave	ondragleave	a dragged item leaves a valid drop target
dragover	ondragover	a dragged item is being dragged over a valid drop target, every few hundred milliseconds
dragstart	ondragstart	the user starts dragging an item.
drop	ondrop	an item is dropped on a valid drop target.

AUDIO / VIDEO API

HTML5 comes with elements for embedding rich media in documents — <video> and <audio> — which in turn come with their own APIs for controlling playback, seeking, etc.

HTMLMediaElement API provides features to allow you to control video and audio players programmatically

AUDIO / VIDEO API ATTRIBUTES

Attribute	Description
autoplay	Can tell the browser to play the video or audio automatically
controls	You can use some simple JavaScript to write your own controls or you can tell the browser to provide them automatically
poster (Video only)	The poster attribute points to an image that the browser will use while the video is downloading, or until the user tells the video to play
height, width	tell the browser the size in pixels of the video/audio
loop	loops the media playback
src	this attribute points to the file to be displayed

WEB SOCKETS

The WebSocket API makes it possible to open a two-way interactive communication session between the user's browser and a server.

With this API, you can send messages to a server and receive event-driven responses without having to poll the server for a reply.

Client ⟷ Server
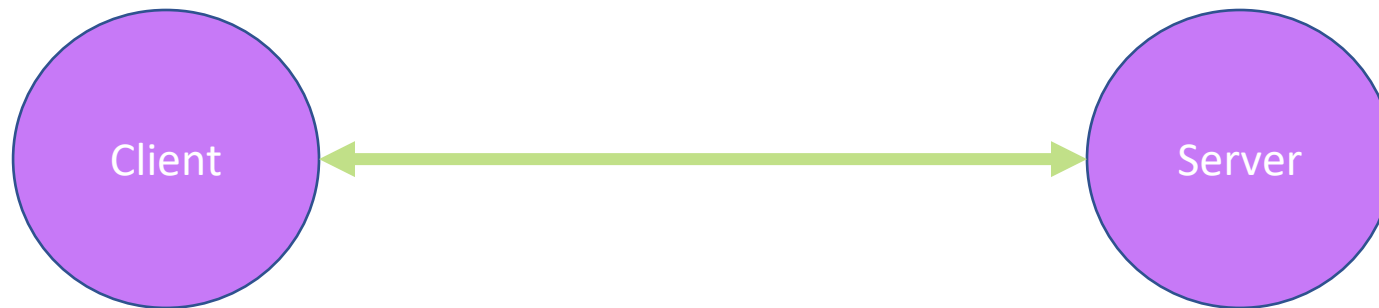
Bi-directional Continuous Communication Channel

WEB SOCKETS - EVENTS

| Event Name	Fires when ...
close	a connection with a WebSocket is closed. Also available via the onclose property
error	a connection with a WebSocket has been closed because of an error, such as when some data couldn't be sent. Also available via the onerror property
message	data is received through a WebSocket. Also available via the onmessage property
open	a connection with a WebSocket is opened. Also available via the onopen property

SERVER SENT EVENTS

Traditionally, a web page has to send a request to the server to receive new data

With server-sent events, it's possible for a server to send new data to a web page at any time, by pushing messages to the web page.

Incoming messages can be treated as Events + data inside the web page.

Client

Server

One-way connection
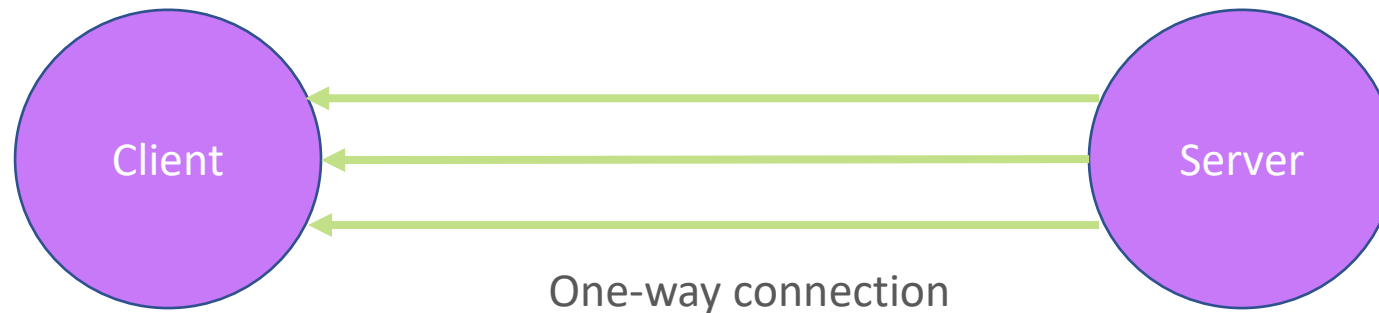
REFERENCES

READING MATERIAL

- https://developer.mozilla.org/en-US/docs/Web/HTML
- https://html.com
- https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps

VIDEO LINKS

- https://www.youtube.com/playlist?list=PLWPirh4EWFpH2Pj1lQ4wMfPgdEjiDrTGA
- https://www.youtube.com/playlist?list=PLu0W_9lII9agiCUZYRsvtGTXdxkzPyItg
- https://www.youtube.com/watch?v=1Rs2ND1ryYc&t=2s