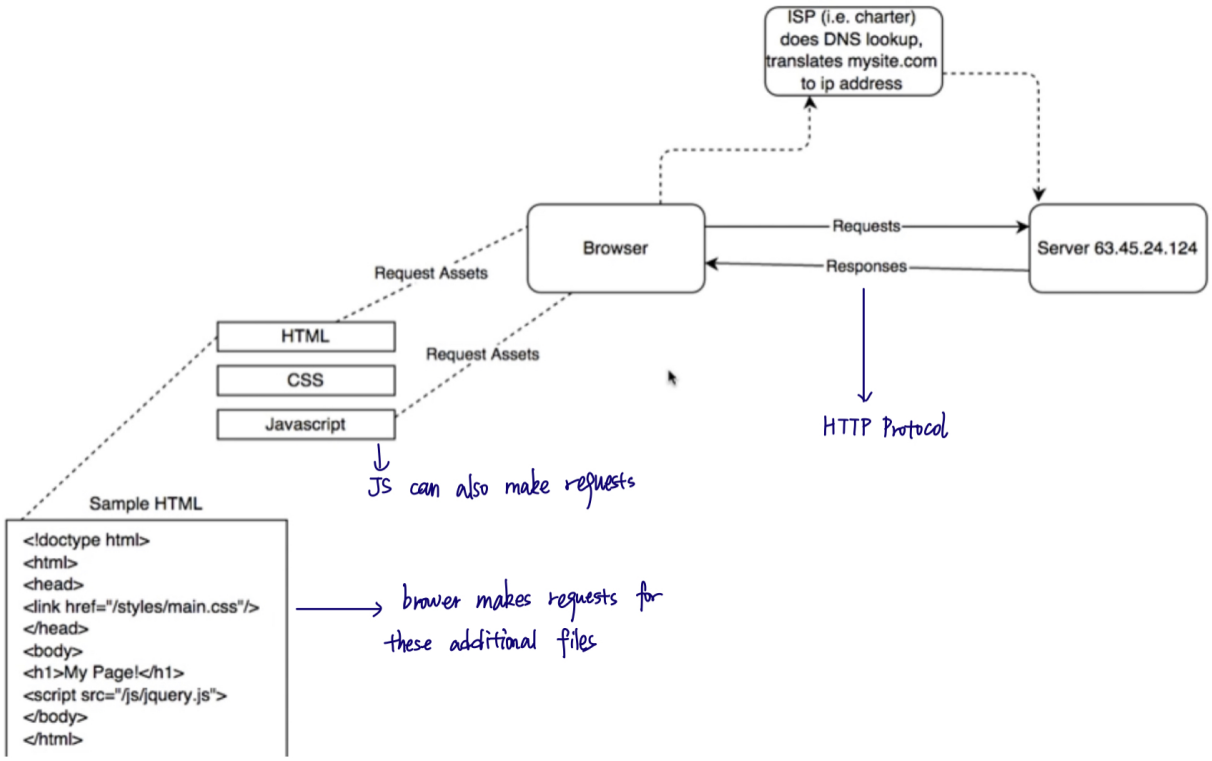
Part 0

Keep **browser's developer console** open for web development `F12`
(***Disable cache*** should be checked in the *Network* tab)



HTTP

The browser and the server communicate with each other using the **HTTP** protocol

- browser sends **HTTP requests,** server responds with **HTTP responses**

The **Network** tab in browser developer tools shows these requests and responses

HTTP (Hypertext Transfer Protocol)

- Follows the **client/server model**, with a client opening a connection to make a request, then waiting until it receives a response
- HTTP is **stateless** (each transaction is independent like flipping a coin)
  - FYI: Cookies and Sessions provide state
- It's an application layer protocol (above the transfer layer protocol (TCP))
- Clients will specify actions (GET/POST/DELETE/etc.)
- Servers define the responds via server status codes (e.g. 200: success, 404: not found error)
- HTTP requests and responses includes **headers** (custom information send along with the transaction)
  - e.g. **Content-Type** in the response let the browser know the type of the response and how to render its information

About web applications

Characteristics of tradition web applications:

- The browser is "dumb", it only fetches HTML data from the server, and all application logic is on the server

Running application logic on the browser:

- To achieve this, we can include a **script** tag that fetches a JS file in our HTML code.

- The JS file can send HTTP requests and make changes to the DOM, therefore enabling us to run application logic on the browser.

DOM (Document Object Model)

- We can think of HTML-pages as implicit element tree structures
- DOM is an API provided by browsers that enables **programmatic modification of the element trees of web pages**

```
// for example, we can append child to a <ul> tag with
list = document.getElementsByTagName('ul')[0]
newElement = document.createElement('li')
newElement.textContent = 'added text'
list.appendChild(newElement)
```

Event handlers and Callback functions

- Event handlers (Callback functions) are functions that are only invoked when certain criteria is met (e.g. when a button is clicked)
- The event handlers are invoked by the browser, instead of the application code

CSS

- `<link href="styles/main.css" />` fetches the CSS (Cascading Style Sheets) file, which make up the appearance of web pages.

Single Page App

- Single-page application (SPA) style of creating web applications (e.g. using `React`) is very popular in recent years
- SPA-style websites don't fetch all of their pages separately from the server, but instead of comprise only one HTML page fetched from the server, and the contents of it is manipulated with JS that executes in the browser.

SPA interacts with the user by dynamically rewriting the current page rather than loading entire new pages from a server

- We could achieve SPA using vanilla JS, but it's easier to work with **JS libraries** like **React, Redux, etc.**