J2EE Architecture

- The Java Platform, Enterprise Edition(J2EE) is a platform-independent,

Java-centric environment from Sun for building, developing, and deploying

web-based applications online. The platform consists of a set of services,

APIS, and protocols that provide functionality.

- Client: small, personal computer that accesses a server through a network.

- Server: a high-performance computer used to house and provide access to applications and data.

N-tiered systems: a client-server architecture in which presentation, application processing, and data management functions are physically separated.

- Transfer Protocols: A communications protocol that provides a structure for requests between client and server.

- HTTP - (HyperText Transfer Protocol): general means of communication for the world wide web. HTTP is based on request & response.

Get:

* Used to simply retrieve data
* No request body.
* Appends request parameters to the url (don’t use for sensitive info)… based on the browser url character limit, “R” in CRUD.

Post:

* Used typically to add data to server
* “C” in CRUD
* Safer -> can add sensitive data to required body & cannot bookmark required body.

Put: “u” in CRUD or replace

Delete: “D” in CRUD

TRACE-used for debugging

HEAD- same as GET but just returns response header.

OPTIONS-used to describe the communication options for the target resource.

PATCH – used to modify capabilities. Request only needs to contain the changes to the resource, not the resource.

Key terms:

* Idempotent: if an operation is idempotent, clients can make the same call repeatedly while producing the same result. HTTP verbs: GET, PUT, DELETE, HEAD, OPTIONS.
* Safe – client can perform the operation & not affect the state of the server.
* HTTP verbs: GET, HEAD, OPTIONS

🡪POST & PATCH are neither idempotent nor safe.

HTTP request – after opening a connection to the intended server, the client transmits a request.

HTTP response: the server gives a reply to a client in the form of a response complete with a body & status code.

Status codes:

1\*\* - information

2\*\* - success

3\*\* Redirected

4\*\* Client error

5\*\* Server error

HTML (HyperText Markup Language) building block of the web, it describes and defines the content of a webpage and the basic layout of the webpage.

- HyperText refers to links that connect web pages to one another, either within a single website or between websites.

- A markup language is a system for annotating a document in a way that is syntactically distinguishable from the text. Some common examples are HTML, XML, and XHTML.

- HTML consists of a series of elements, which you wrap different parts

of the content to make it appear a certain way.

- In markup language, an element may contain a data item or a chunk of text or an

image, or nothing at all.

- Tags begin or end an element in source code.

- Empty elements are elements with no content. For example: <hr> or <br>

Attributes are within the element. <img src="img\_girl.jpg" width="500" height="600">

- The <!DOCTYPE> declaration tag is used by the web browser to understand the version

of the HTML used in the document. All HTML needs to have a DOCTYPE declared. The DOCTYPE is not actually an element or HTML tag.

- Character encoding is a method of converting bytes into characters. To validate or display an HTML document properly, a program must choose a proper character encoding.

- Viewport: the user’s visible area on the web.

The body tag is where all content to be displayed in the window

- the <meta> tag is used to house metadata regarding the document.

HTML5 semantic tags are the key to an excellent and neat connection between the browser and your HTML code. HTML5 semantic tags are header, footer, nav, main, section, article, aside, address, figure.

CCS3: Cascading Style Sheets is simple mechanism for adding style (font, colors, spacing) to web documents.

# id, p for tag, !important, group (p,h1,li)

With an external style sheet, you can change the look of an entire website by changing just one file. The <link> element goes inside the <head> section

Internal styles are defined within the <style> element, inside the <head> section of an HTML page

An inline style may be used to apply a unique style for a single element

Box model: Margin –> border ->padding -> content.

Bootstrap

Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

How do i include it in my page: you can include it from a CDN, include link in head.

JavaScript is the programming language of HTML and the Web

**type** **coercion** is about *converting a value from one type to another*.

const numString = '3';  
const num = 3;

num == numString  
// true;

a **truthy** value is a value that is considered  true when encountered in a [Boolean](https://developer.mozilla.org/en-US/docs/Glossary/Boolean) context. All values are truthy unless they are defined as [falsy](https://developer.mozilla.org/en-US/docs/Glossary/Falsy" \o "falsy: A falsy value is a value that is considered false when encountered in a Boolean context.) (i.e., except for false, 0, "", null, undefined, and NaN)

Fixed values are called **literals**. Variable values are called **variables**.

Var scopes: Scope determines the accessibility (visibility) of variables

Variables declared outside any function have **Global Scope**. All scripts and functions on a web page can access it.

Variables declared inside a function have **Function Scope**

Variables declared with the **let keyword** can have Block Scope.

Hoisting: a variable can be declared after it has been used.

The **this** keyword refers to the object it belongs to.

**// TEMPLATE LITERALS: string interpolation.**  
console.log(`Three plus six is ${3 + 6}.`);

Arrow function: var addArrow = (a,b) => {return a + b};

A callback function is a function passed into another function as an argument, which is then invoked inside the outer function to complete some kind of action.

function processUserInput(callback) {

var name = prompt('Please enter your name.');

callback(name);

}

processUserInput(greeting);

An anonymous function is a function that was declared without any named identifier to refer to it

**var** anon = **function**() {

alert('I am anonymous');

}

anon();

A closure is an inner function that has access to the outer function’s variable. The closure has three scope chains: it has access to its own scope (variables defined between its curly brackets), it has access to the outer function’s variables, and it has access to the global variables.

function showName (firstName, lastName) {

var nameIntro = "Your name is ";

// this inner function has access to the outer function's variables, including the parameter

function makeFullName () {

return nameIntro + firstName + " " + lastName;

}

return makeFullName ();

}

showName ("Michael", "Jackson"); // Your name is Michael Jackson

An **IIFE**(Immediately Invoked Function Expression) is a [function](https://developer.mozilla.org/en-US/docs/Glossary/function) that runs as soon as it is defined.

(Function() {

Console.log(“IIFE”);

})();

var getUsername = function() {  
 return loggedIn && username;  
}

Guard: A && B will return B if A is true, otherwise it will return A.

Default: A || B will return A if A is true, otherwise it defaults to B

Ternary operator : voteable = (age < 18) ? "Too young":"Old enough"

== equal value.

=== equal value and equal type.

DOM: Document object model is an interface that allows a programming language to manipulate the content, structure, and style of a website.

Event handlers can be used to handle and verify, user input, user actions, and browser actions.

Event propagation is a way of defining the element order when an event occurs.

In *bubbling*the inner most element's event is handled first and then the outer: the <p> element's click event is handled first, then the <div> element's click event. In *capturing*the outer most element's event is handled first and then the inner.

With the addEventListener() method you can specify the propagation type by using the "useCapture" parameter. The default value is false, which will use the bubbling propagation. Event.stopPropagation();

jQuery is a lightweight, "write less, do more", JavaScript library.

The purpose of jQuery is to make it much easier to use JavaScript on your website

json-server --watch bookstore.json

ctr+c to stop server running.

JSON: javascript object notation, is a syntax for storing and exchanging data.

AJAX: Asynchronous JavaScript And XML

* NOT a programming language! It is a series of interactions between  
  JavaScript, the HTML DOM, and the XMLHttpRequest object from the browser
* It is "asynchronous" because AJAX can exchange information with a   
  server "behind the scenes" -- meaning, there is no need to reload   
  the page in order to display the content received from the response
* Requests will not have to wait for a response from a prior request  
  in order to be sent

About the XMLHttpRequest object

* Supported by most modern browsers \*cough\* dont use IE \*cough\*
* Used to exchange data with a server
* **Important Methods**
  + Var request = new XMLHttpRequest() --> creates new object
  + abort() --> cancels the current request
  + getAllResponseHeaders() & getResponseHeader() --> get response metadata
  + open(method, url, async, username, pwd) -> specify request;
    - method = HTTP verb
    - URL = location to send request
    - async = whether to execute request asynchronously or not; usually true
    - username and password = optional server credentials
  + send(content) -> send the request. content = request body; optional
  + setRequestHeader()--> set name/value pair in request header
* **Important Properties:**
  + onreadystatechange - defines a function to be executed when the ready state property changes
  + readyState - holds the XHR status
    - 0 - request not initialized
    - 1 - server connection established
    - 2 - request received
    - 3 - server is processing request
    - 4 - response is ready
  + responseText - response body as string
  + responseXML - response body as XML
  + status - HTTP response code (200)
  + statusText - HTTP response text (OK)
* SDLC
  + Fixed scope
  + Requirements
    - Analysis
      * Design
        + Implementation

Maintenance

* Agile
  + Individuals Interactions > process tools
  + Software > docs
  + Collaboration > contract
  + Change > plan
* Scrum
  + Variable scope
* UML
  + Unified Modeling Language
* Product Owner
* Triple Constraint
* Burn Down Chart
  + A **burn down** chart is a graphical representation of work left to do versus time. The outstanding work (or backlog) is often on the vertical axis, with time along the horizontal. That is, it is a run chart of outstanding work. It is useful for predicting when all of the work will be completed.
* Start, Stop, Continue
  + Start: start doing the new spring
  + Stop: stop anything that wasn’t working on the previous spring
  + Continue: continue on the things that worked on the previous spring
  + kudos