

Annexure-I

Two months summer Training Report on JavaScript for Web Development

Board Infinity

A Training Report

Submitted in partial fulfilment of the requirements for the award of degree
of

Masters in computer application

(SEMINAR ON SUMMER TRAINING CAP-735)



Submitted To

LOVELY PROFESSIONAL UNIVERSITY PHAGWARA, PUNJAB

From 01 July 23 To 25 Aug 23

SUBMITTED BY

Name of Student's : - Ekhlakh Ahmad

Registration Number : - 12209166

Course : - Master in Computer Application (MCA)

Signature of the Student's: - *Ekhlakh Ahmad*

Annexure-II

STUDENT DECLARATION

Ekhlakh Ahmad, Reg- 12209166 I hereby declare that I have completed my Two months summer training at Board Infinity platform from my 01 July 2023 to 25 Aug 2023 under the guidance of Akshay Mittal. I declare that I have worked full dedication during their 2 months of training and my learning outcomes fulfil the requirements of training for the award of degree of master's in computer application, Lovely Professional University, Phagwara.

25 Aug 2023

Name of Student's – Ekhlakh Ahmad

Registration no: 12209166

Ekhlakh Ahmad

ACKNOWLEDGEMENT-

I would like to express my gratitude toward my university as well as Board Infinity for providing me the golden opportunity to do this wonderful summer training regarding OOPS and many more in JavaScript, which also helped me in doing a lot of homework and learning. As a result, I came to know about so many new things.

So, I'm really thankful to them.

Moreover, I would like to thank my friend who helped me a lot whenever I stuck in some problem related to my course. I am thankful to have such a good support of them as they always have my back whenever I need.

I have taken efforts to in the project. However, it would not have been possible without the kind support and help of many individuals and organisations. I would like to extend my sincere thanks to all of them.

CERTIFICATE OF COMPLETION

THIS CERTIFICATE IS AWARDED TO

Ekhlakh Ahmad

for successfully completing Microlearning Program in
Javascript For Web Development

12 July, 2023

ISSUED DATE



CEO, Board Infinity
Sumesh Nair

BI22LPBI345426473

CERTIFICATE NO.

BOARD

Table of Contents

Introduction of the Company/Work: -	6
Company's Vision and Mission: -	6
Board Infinity offers a vast range of courses which include:	7
Various departments and their functions.....	8
Board Infinity Organization Chart: -	9
Introduction to Web Development: -	11
Fundamental of JS Programming.....	11
Conditionals & Loops in JS Programming.....	13
Functions & Methods ethos's in JS Programming.....	15
Array and Sorting in JS Programming.....	17
JS Map Reduce Filter	20
JavaScript AJAX.....	22
JavaScript DOM Manipulation: -	26
JS Manipulation Style: -	28
Conclusion: -	29

Introduction of the Company/Work: -

Company's Vision and Mission: -

Board Infinity is a full-stack career platform for students and jobseekers enabled by personalized learning paths, career coaches and access to opportunities. Our mission is to personalize your career journey, help you realize true potential and meet your career dreams.

Be it a career transition, your first job, campus placements preparation or any career guidance. Board Infinity is a one-stop solution to all your career needs. We connect career aspirants with industry experts for focused learning, guidance, mentoring and support. We also prepare you and connect to relevant opportunities and help you realize your career dreams.

Mission: - We intend to become a Full Stack Career platform solving various career problems (skilling, growth, transition, fulfillment, etc.) throughout our individual's career journey.

Board Infinity offers a vast range of courses which include:

Learning Path: -

Students can take Job-Assured Programs as part of this course category. It assists job seekers in advancing their careers by assisting them in acquiring in-demand skills through industry expert coaches. Data Science Courses, Digital Marketing Courses, Full Stack Developer Courses, Software Developer Courses, Sales Courses, AI Courses, and other specialties are among the subjects addressed.

Personalized Courses: -

This course offers one-on-one live coaching, lifetime course content, and personal mentoring from top language industry professionals. Depending on a learner's learning needs, the customized course structure offers three variations.

Micro-Learning: Short-term beginner-friendly courses such as Big Data, Power BI, Cyber Security, Python Programming, Machine Learning, Deep Learning, Affiliate Marketing, and Angular JS are examples of this course genre. Case studies and assignments are also part of this course style. Candidates are given a certificate of completion after completing the courses.

Free Courses: -

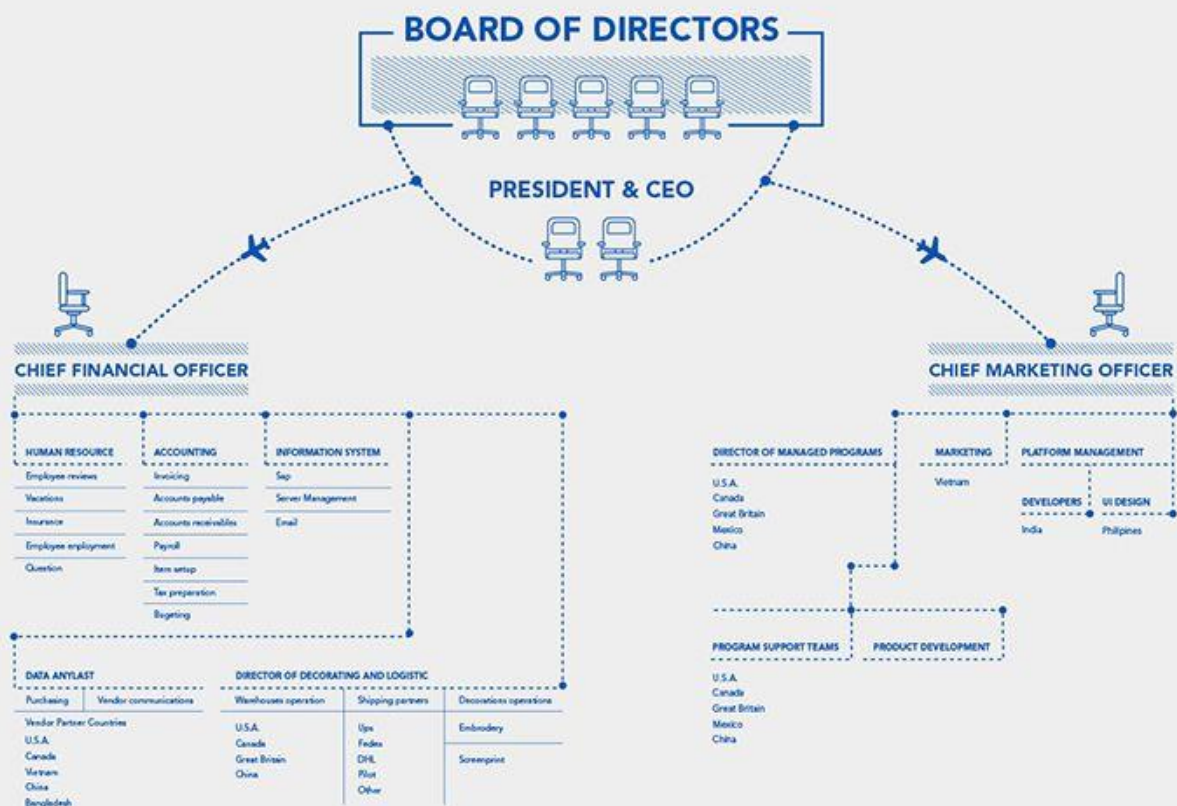
Board Infinity also offers a variety of free courses such as Resume Building, Python for Data Science, GD Preparation, Aptitude Test Preparation, Interview Preparation, Grooming and Etiquette Training, and Grooming and Etiquette Training to help students and early career professionals learn the fundamentals of any in-demand skill at their own pace and receive a certificate of completion.

Various departments and their functions

- Job Trends in This Decade
- Highest paying jobs of 2023 for Freshers
- Best Career option after Graduation
- 10 Best option After B. Com
- Booming Career paths for management graduates

Board Infinity Organization Chart: -

View Board Infinity org chart to access information on key employees and get valuable insights about Board Infinity organizational structure. The organizational chart highlights the reporting lines within the company, starting with the top decision makers: Viraj Malpekar, Assistant VP, Business Development, and Tejas Somani, Assistant Vice President



BOARD

Grooming and Etiquette Training



Introduction to Web Development: -

Fundamental of JS Programming

Variables and Data Types:

Variables are used to store data in a program. Variables has a name and a value. There is variable, declaration and initialization.

Once it's created, you can set its value.

In JavaScript, a variable is declared using the var keyword.

A variable in JavaScript can contain any data. A variable can at one moment be a string and number. Programming languages that allow such things are called “dynamically typed”, but variable is not bound to any of them.

1. Number

The number type represents both integer and floating-point numbers. You can perform *, /, +, - etc operations on numbers. Besides regular numbers, there are special numeric values like Infinity, -Infinity, and Nan (Not a number).

2. String

A string in JavaScript can be surrounded by single quotes, double quotes or backticks. Backticks allow you to embed variables inside a string.

3. Boolean

A Boolean has only two values: true or false. It is commonly used to store yes/no value.

4. Null

A special value that represents “nothing”, “empty” or “unknown value”.

5. Undefined

A special value which means ‘value is not assigned’.

6. Objects

All other types are called “primitive” because their values can contain only a single thing (be it a string or a number or whatever). In contrast, objects are used to store collections of data and more complex entities.

7. Arrays

An array is a single object that contains multiple values enclosed in square brackets and separated by commas.

an important part of programming is doing math and being able to compare values to make decisions in code. The outcome can be either true or false (Boolean type).

Conditionals & Loops in JS Programming

Conditionals

Conditionals are used to run a piece of code or another depending on the outcome of the comparisons made. JavaScript provides if and if-else statements to run code based on conditions. You can also chain if-else-if to run code based on multiple conditions. Logical operators like && (AND), || (OR) and ! (NOT) can also be used for making decisions.

Looping

Loops allow you to run a piece of code a certain number of times or until a certain condition is met. They're incredibly useful. They can be used to carry out actions on every item in an array or printing all the data inside an array or in searching etc.

Two of the most common loops in JavaScript are for and while.

A for loop contains three statements, separated by a semicolon (initialization, condition, updating or final expression). The loop first runs the initialization and then checks if the condition is true. If it is true, it runs the block of code inside and finally runs the updating code. After updating, it again checks the condition and performs these steps again and again until the condition becomes false.

A while loop repeats a block of code until a condition is met. For any update, you can do it inside the while loop.

Functions

Functions are blocks of code that you can reuse and call anytime you want to execute it. A function can take values, called arguments, and can return a value.

Objects

A JavaScript object is a collection of properties and methods/functions. It is a key-value pair collection where the value can be data or function. You can access the keys inside an object using the dot syntax. Objects are used to store relative information.

Arrays

Arrays in JavaScript are used to store a list of any kind of data. Each item in an array has an index/position (a number that can be used to retrieve an element from it). Arrays in JavaScript start at 0; so the last element has an index one less than the length of an array. Arrays, by default also have some properties/functions assigned to them, which gives more information about the array like the length etc.

Functions & Methods ethos's in JS Programming

JavaScript's console.log () function is used to print any variables that have been defined in the program or to display any other message to the user.

Parameters: It accepts a parameter, which may be an object, an array, or any message.

Return value: This function returns the value of the given parameter.

Here are some examples of JavaScript code that show how this function works:

A number as an argument: The console.log () function will display the number if a number is given as an argument.

String as an argument: If a string is supplied as an argument to the console.log () function, the function will show the string.

A char as an argument: If a char is given as an argument to the console.log (), then it will be displayed by the function.

A message as an argument: The function will display the message if the console.log () function is passed with the message.

A function as an argument: If a function is provided as an argument to the function `console.log ()`, it will display the value of the given function.

A number with a message as an argument: When a number is supplied to the `console.log ()` function, then it will display the number along with the given message.

A message with string as an argument: If the string is submitted to the `console.log ()` function, it will be displayed alongside the given message.

A message with char as an argument: If the char is submitted to the `console.log ()` function, it will be displayed alongside the given message.

Array and Sorting in JS Programming

Introduction

we can use the `split ()` and `Array. From ()` functions in JavaScript to convert string into a character array

String split () Function in JavaScript

`String. Split ()` is a built-in function in JavaScript that lets you split a string into an array of strings, with each item representing one word in the original string. The delimiter can be any character or sequence of characters that separates words in the input string, such as whitespace or punctuation. The `String. Split ()` method is used to break a string into an array of strings by splitting it on a given delimiter, which is specified as the argument to the method.

This is how it works:

1. A string is passed as the first argument to the `split ()` function.
2. The string is then split on whitespace (spaces, tabs, newlines) using a regular expression defined by the second argument.
3. If there are more than two matches for this regular expression, then only the first two matches are used in splitting the string; any additional matches are ignored and do not contribute to the result of that call to `split ()`.

The following example uses the `String.split ()` method to break up a sentence into words and store them in an array:

The array elements in an array, each separated by a specific separator, are concatenated together using the `join ()` method to produce a new string. The `join ()` method concatenates all the elements in an array (or an array-like object), separated by commas or a specified separator string, and returns a new string. If there is only one element in the array, the separator will not be used when returning the element.

Parameters: This method accept single parameter as mentioned above and described below:

separator: It is optional, meaning that it may or may not be used as a parameter. The comma is its default value (,). specifies a string to be used to divide up each pair of adjacent array members. If required, the separator is changed into a string. The array's elements are separated by commas if it is left out (","). All items are put together without any characters in between them if separator is an empty string.

Return Value: The collection of elements from the array is returned as a String. By employing the separator to link each element of the array, this function returns a string. The comma (,), which is this function's default separator, is used to combine the array items if no separator is specified. The elements are connected directly with no characters in between them if the separator is an empty string. This function returns an empty string if the array is empty.

Description

All array items' string conversions are combined into a single string. The string "null" or "undefined" is replaced with an empty string if an element is undefined or null.

Internally, `Array.prototype.toString()` with no arguments calls the `join` function. An array instance's function `toString()` { [native code] } behaviour will also be overridden if `join` is overridden.

The `join()` method iterates empty spaces as though they have the value `undefined` when applied to sparse arrays.

The generic `join()` method is used. Only the length and integer-keyed characteristics are required for this value.

JS Map Reduce Filter

Introduction

Map is a collection of key data items, Same as Object. A map is a data structure that stores key-value pairs and allows you to look up the value associated with a given key. Each key belongs to a data value, and the values are stored in an array. You can access a value by using its associated key as a string, or by using array syntax to index it.

Important:

The keys are not necessarily unique, so you can have multiple maps with the same key.

Map is an associative array, which means that it can be indexed by either the key or the value. The values in a map can be of any type, including other maps or sets. Unlike arrays, there is no limit on how many items can be stored in a map, if you have enough memory available to accommodate all of them. Maps also allow more than one item with the same key in them - this is called "chaining".

Parameter:

N -- Iterable object whose values are stored as key, value pair, If the parameter is not specified then a new map is created is Empty.

Properties:

map.set(key, value) - set the value for the key in the map object.

map.get(key) - returns the value associated with the key, undefined if key doesn't exist in map.

map.has(key) - returns true if value associated with the key exists, false otherwise.

map.delete(key) - removes the value by the key.

map.clear() - removes all elements from the map.

map.size - returns the count of the elements .

map.values() - returns a new iterator object that contain values for each element in insertion order.

map.keys() - returns a new iterator that contain the keys for element in insertion order.

JavaScript AJAX

Introduction

With the use of Ajax technology, developers may send asynchronous HTTP queries without having to reload the website completely. Developers have been utilizing the jQuery library for years to make the procedure less time-consuming than it would be in pure JavaScript. More control over the Ajax calls we wish to make is sometimes necessary. For instance, we need to describe what should happen if an Ajax call fails or if we must make an Ajax request, but the outcome is only necessary if it is obtained quickly.

The settings Parameter

You can specify many different options to bend \$.ajax () to your needs. In the list below, you can find their names and their description:

dataType: The type of data you're expecting from the server.

error: A callback function to be executed when the request fails.

global: A Boolean indicating whether to trigger a global Ajax request handler. The default is true.

jsonpCallback: String containing the callback function name for a JSONP request.

mimeType: String containing a mime type to override the XMLHttpRequest mime type.

password: A password to be used with XMLHttpRequest in response to an HTTP access authentication request.

timeout: A number value in milliseconds for the request timeout.

type: A type of HTTP request, e.g., POST, PUT and GET. The default is GET.

url: A string containing the URL to which the request is sent.

accepts: The content type sent in the request header tells the server what kind of response it will accept in return.

async: By default, all requests are sent asynchronously. Set it to false to make it synchronous.

beforeSend: A callback function to be executed before the Ajax request is sent.

cache: A boolean indicating browser cache. The default is true.

complete: A callback function to be executed when the request finishes.

contentType: A string containing a type of content when sending MIME content to the server. Default is "application/x-www-form-urlencoded; charset=UTF-8"

crossDomain: A boolean value indicating whether a request is a cross-domain.

data: Data to be sent to the server. It can be a JSON object, string, or array.

Async, Promises and callback.

Introduction

To indicate that a function is an asynchronous function, we use the `async` keyword with the function. The promise that the `async` function returns.

Await Keyword in JavaScript:

The `async` function uses the `await` keyword to wait for the asynchronous action.

Advantages of Async Functions:

Compared to calling a callback or a promise, the code is easier to read.

Handling errors is easier.

Testing is simpler.

Note: The more recent version of JavaScript added the terms `async` and `await` (ES8). The usage of `async/await` may not be supported by some older browsers. Visit [JavaScript async/await browser support](#) for additional information.

Callback function in JavaScript. I know that when we put a function inside another function as an argument that is called Callback function. But I am confused how it actually works & what's the most use cases of it. Is it part of asynchronous javascript ? Someone please explain this in details with real examples. Is the `addEventListener` a callback function & does it works asynchronously ?

There are lots of ways to write and create asynchronous code in JavaScript, some of them are

Callbacks: -

- Promises
- Callbacks

Callbacks are nothing but a convention name used for JavaScript functions. This is how asynchronous was handled for a long time and now is the old-fashioned classical approach. In a callback, we pass in a function inside an asynchronous call that will execute the call back when the main asynchronous call is done, hence the name being callback

The main disadvantage of this approach occurs when you have multiple chained asynchronous tasks, which requires you to define callback functions within callback functions within callback functions... This is called callback hell, it looks something like this ->

Promises: -

Promises were introduced in ES[^](2015), A promise represents the result a asynchronous code would output in the future. They provide a clearer way of writing code and representing sequential steps in asynchronous tasks because even in parallel operations you do need to maintain an order of higher sequence, you can set the value of a variable before using it in some other function. They allow us to write asynchronous code in a way that resembles synchronous code.

JavaScript DOM Manipulation: -

The DOM is a W3C (World Wide Web Consortium) standard. The DOM defines a standard for accessing documents:

The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document.

The W3C DOM standard is separated into 3 different parts:

Core DOM - standard model for all document types

XML DOM - standard model for XML documents

HTML DOM - standard model for HTML documents

Levels of DOM:

Level 0: Provides a low-level set of interfaces.

Level 1: DOM level 1 can be described in two parts: CORE and HTML.

CORE provides low-level interfaces that can be used to represent any structured document.

HTML provides high-level interfaces that can be used to represent HTML documents.

Level 2: consists of six specifications: CORE2, VIEWS, EVENTS, STYLE, TRAVERSAL, and RANGE.

CORE2: extends the functionality of CORE specified by DOM level 1.

VIEWS: views allows programs to dynamically access and manipulate the content of the document.

EVENTS: Events are scripts that are either executed by the browser when the user reacts to the web page.

STYLE: allows programs to dynamically access and manipulate the content of style sheets.

TRAVERSAL: This allows programs to dynamically traverse the document.

RANGE: This allows programs to dynamically identify a range of content in the document.

Level 3: consists of five different specifications: CORE3, LOAD and SAVE, VALIDATION, EVENTS, and XPATH.

CORE3: extends the functionality of CORE specified by DOM level 2.

LOAD and SAVE: This allows the program to dynamically load the content of the XML document into the DOM document and save the DOM Document into an XML document by serialization.

VALIDATION: This allows the program to dynamically update the content and structure of the document while ensuring the document remains valid.

EVENTS: extends the functionality of Events specified by DOM Level 2.

XPATH: XPATH is a path language that can be used to access the DOM tree.

JS Manipulation Style: -

In order to change or modify an element in the DOM, you need to select that specific element. Thus, JavaScript has six methods to select an element from a document in dom manipulation in javascript. `getElementById`: returns an element whose id matches a passed string.

JavaScript manipulation of styles involves dynamically changing the visual presentation of HTML elements on a web page. This capability allows developers to create interactive and visually appealing user interfaces. Through JavaScript, properties such as colors, sizes, positions, and visibility can be altered to respond to user interactions or specific events. This process entails selecting the desired element using methods like `getElementById` or `querySelector`, and then accessing its style property to modify specific style attributes.

For instance, to change the background color of a button when it's clicked, you can attach an event listener to the button element. Inside the event handler function, the `style.backgroundColor` property can be updated to the desired color. Additionally, animations and transitions can be achieved by manipulating style properties over predefined time intervals, giving rise to fluid and engaging user experiences.

Moreover, JavaScript can enable the creation of responsive designs by adjusting styles based on viewport dimensions. By accessing the `window.innerWidth` property, developers can determine the width of the viewport and apply style changes accordingly, ensuring that content is displayed optimally on different devices and screen sizes.

In summary, JavaScript manipulation of styles provides the means to transform static web pages into dynamic, interactive interfaces. This approach empowers developers to adapt and enhance the user experience by altering visual elements based on user input, events, or dynamic conditions.

Conclusion: -

My experience with the summer training program offered by Board Infinity has been truly transformative. Over the course of 2 Months, I had the opportunity to delve into various aspects of JavaScript for Web Development, honing my skills and expanding my knowledge base.

Throughout the program, I had the privilege of working on projects that pushed my boundaries and allowed me to apply theoretical concepts to real-world scenarios. These hands-on experiences not only strengthened my problem-solving abilities but also instilled in me a deeper appreciation for the nuances of JavaScript for Web Development.

I am particularly proud of Web Development, which challenged me to collaborate with a diverse team and deliver innovative solutions within tight deadlines. The guidance provided by the trainers and mentors at Board Infinity played a pivotal role in my success, offering insights that I will carry with me throughout my career journey.

This training experience has not only equipped me with practical skills but has also fostered personal growth. It has taught me the value of adaptability, perseverance, and continuous learning. I am immensely grateful to Board Infinity for the support, knowledge, and exposure they provided during this enriching journey.

As I move forward, I am excited to leverage the expertise gained from this program to contribute meaningfully to JavaScript for Web Development. The lessons learned and the connections established will undoubtedly influence my trajectory positively.