<https://www.facebook.com/100063208241831/videos/161306924410589>

Day 3 Revisit

Objects, Arrays

Control Statements

ES6 Features (let, const, class)

Hoisting (variable and function declaration will be moved to top automatically)

Template Literal (Template String /String Template) -- ` Template Literals usually contains back tick symbol and variables “a =” ${a}, ‘b =’ ${b}`;

Functions (types of functions) [1.IIFE 2. Arrow Functions 3. Functional Expressions ]

Truthy & Falsy

Pass By Value and Pass by reference

For in & for of

Class syntax

Agenda

JS DOM manipulation

Selecting elements

Modifying elements (adding, removing, updating )

Events (Event Handling, Event Listeners, Call back methods, types of Events)

Asynchronous JS (AJAX)

DOM – Document Object Model

DOM is a tree like data structure with single root (document)

Document – html – Head && Body

Document[0].element[0] = head

Document[0].element[1] = body

Document[0].element[0].element[0] = title

Document[0].element[1].element[0] = h1

Internet is working as a Client / Server Concepts. It uses Request & Response objects.

Request object will be created by the browser (with the help of end user)

Google.com -🡪 <https://www.google.com> (full fledged URL) -- > creates Request Object

Types of webpages

1. Static Web Page (The contents will not change with respect to time and user)
2. Dynamic Web Page (The contents will change with respect to time and user)

Static Web Pages --- WebPages about a Church, Temple, Place (Using HTML alone)

Dynamic Web Pages – Facebook, twitter, linkedIn, Gmail, Outlook, News web pages (HTML + Server Side Coding)

Ways of Submitting form Data

1. Get method (Default) [v.fast, less secured, data will be added in the URL itself, not used for sending secured/confidential data like passwords , size limitations]
2. POST method [ more secured, data will be added to the Request object body not to the URL, recommended for sending secured/confidential data. No Size restrictions]

Validations

1. Client Side Validation [Checking for empty data, improper format, invalid number etc.,]
2. Server Side Validation [ entered proper/ valid data] Login

**Server side validation** is performed by a web server, after input has been sent to the server.

**Client side validation** is performed by a web browser, before input is sent to a web server.

Events (Actions performed on the Web Page)

Types of Events

1. Keyboard Events [Keypressup, keypressdown, focus, blur]
2. Mouse Events [onclick, onmouseover ]
3. Touch Events [single tap, double tap, zoom]
4. Form Events [form load, onsubmit,]
5. Document Events [onload, onunload]

Event – EventListener – EventHandler

Event is an action (performed in web page)

Event Handler is a method (which handles particular event) – It always starts with on and all lower case.

EventHandlers – Default Methods for Each Events

EvenetListeners – Custom Event Handlers

JSON – JavaScript Object Notation

JS Object

Emp = { id: 100, fName:”abc”, lName:”xyz”};

JSON

Emp = {“id”:100,”fName”:”abc”,”lName”:”xyz”};

JSON is a platform/language/architecture independent way of representing data

JSON format is widely used in data transfer and data exchange.

XML was initially used for the same data transfer/data exchange purpose.

XML consumes more space than JSON.

|  |  |
| --- | --- |
| XML Representation | JSON Representation |
| <employees>  <employee>  <id> 100</id>  <fname>ABC</fname>  <lname>XYZ</lname>  </employee>  <employee>  <id> 101</id>  <fname>MNO</fname>  <lname>PQR</lname>  </employee>  </employees> | Emps= [  {“id” :100,”fname”:”ABC”,”lname”:”XYZ”},  {“id” :101,”fname”:”MNO”,”lname”:”PQR”}  ]; |

AJAX – Asynchronous JavaScript and XML

Asynchronous – Doing multiple jobs at the same time. (Parallel programming)

All programs executes statements line by line (one line at a time)

Synchronous Program (Blocking operations)

1. Print some data in console (2 ms)
2. Insert a new record in the database (750 ms)
3. Open a file and read the 10th line of that file (400 ms)
4. Get input from user and display it ( 1500 ms)

Total time = (2+750+400+1500) = 2652 ms (overall time)

Asynchronous (Non-blocking operations)

Total time = 1500 ms

AJAX is used in Google

Client – Server Concepts [Request Object & Response Object] – http Request Object, http Response Object

Asyn – xmlhttprequest object (XMLHttpRequest)

AJAX Example

1. Download and Extract Eclipse IDE (zip file ) (<https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2022-06/R/eclipse-jee-2022-06-R-win32-x86_64.zip> )
2. Download and extract Tomcat Web Server (8.5.x) (<https://tomcat.apache.org/download-80.cgi> )
3. Open Eclipse IDE (Set the Workspace or accept the default workspace)
4. Create a new Dynamic Web Project (File 🡪 New 🡪 Others 🡪 Web (Dynamic Web Project)
5. Enter the name of the project, and select the Run time environment [apache tomcat]
6. Click on Finish button
7. Create a HTML file under src/main/webapp (table1.html) [File🡪New🡪HTML]
8. Copy & paste the code from <https://www.javatpoint.com/ajax-example> for table1.html
9. Create a JSP file under src/main/webapp (calculate.jsp) [File🡪 New🡪 JSP]
10. Copy & paste the code from [https://www.javatpoint.com/ajax-example for index.jsp](https://www.javatpoint.com/ajax-example%20for%20index.jsp)
11. Right click your project and select Run As 🡪 Run on Server
12. Select tomcat server to run this project
13. Enter table1.html in the address bar
14. Type any number in the text box and press “show table” button