**Quiz 3 Total**

Name: **Akash Rizvi**

Github link for quiz:[**https://github.com/AkashRizvi/AkashQuiz3**](https://github.com/AkashRizvi/AkashQuiz3)

**Marks 100**

**Section I – MCQs**

**Q1:**

Ans: b) Class

**Q2:**

Ans: a) A is the superclass and B is the subclass

**Q3:**

Ans: b) By reference

**Q4:**

Ans: d) inherit

**Q5:**

Ans: c) Inheritance

**Q6: What among the following is an appropriate when an event occurs when the user**

**clicks on an element?**

Ans: b) onchange

**Q7:**

Ans: b) document.getElementById ("name").value

**Q8:**

Ans: a) parts of a web page

**Q9:**

Ans: d) variable = new XMLHttpRequest();

**Q10:**

Ans: c) JSON.parse(undefined)

**Section – II**

**Q1. Write a JavaScript program to display the reading status (i.e. display book name, author**

**name and reading status) of the following books.**

**Ans:**

<body>

<div class="container mt-5">

<h1>Section - II</h1>

<h3>Q1: Reading Status</h3>

<script>

var library = [

{

author: 'Bill Gates',

title: 'The Road Ahead',

readingStatus: true

},

{

author: 'Steve Jobs',

title: 'Walter Isaacson',

readingStatus: true

},

{

author: 'Suzanne Collins',

title: 'Mockingjay: The Final Book of The Hunger Games',

readingStatus: false

}];

for (var i = 0; i < library.length; i++) {

var book = " " + library[i].title + " " + ' by ' + library[i].author + ".";

if (library[i].readingStatus) {

console.log("Already read <b> " + book + "");

document.write(`I have Already read <strong> ${book} </strong> <br>`);

} else {

console.log("You still need to read " + book);

document.write(`I still have to read <strong> ${book} </strong> <br>`);

}

}

console.log(library.length)

</script>

</div>

</body>

**Q** **2: Create a program that creates 3 Employee objects from an Employee class with ID,**

**Name, Department and Salary properties. Values of properties should be assigned**

**through getter and setter methods (ES6). Display all the properties of all 3**

**employees.**

**Ans:**

<body>

<h1>Section - II</h1>

<h2>Q2: Employee's Bio data</h2>

<script>

class employee {

setValues(id, name, dept, salary) {

this.id = id;

this.name = name;

this.dept = dept;

this.salary = salary;

}

getValues() {

document.write('<b>Employee id is ', this.id +'</b><br>');

document.write('Employee name is ', this.name +'<br>');

document.write('Employee dept is ', this.dept +'<br>');

document.write('Employee salary is ', this.salary +'<br>');

console.log('Employee id is', this.id);

console.log('Employee name is', this.name);

console.log('Employee dept is', this.dept);

console.log('Employee salary is', this.salary);

}

}

const object1 = new employee();

object1.setValues(17621, 'Akash', 'CS', 60000);

object1.getValues();

const object2 = new employee();

object2.setValues(2344, 'Rehan', 'IT', 33000);

object2.getValues();

const object3 = new employee();

object3.setValues(3209, 'Aliya', 'SE', 47000);

object3.getValues();

</script>

</body>

**Q** **3: At times, when we access a JavaScript object, we receive raw JSON string and we**

**need to convert it to an object ourselves to use it in our program.**

**Ans:**

import React from "react";

function JsonProblem() {

React.useEffect(() => {

fetch(

"https://mdn.github.io/learning-area/javascript/oojs/json/superheroes.json"

)

.then((res) => res.json())

.then((data) => console.log(JSON.stringify(data.members[2].powers)));

});

const result = () => {

fetch(

"https://mdn.github.io/learning-area/javascript/oojs/json/superheroes.json"

)

.then((res) => res.json())

.then((data) => console.log(JSON.parse(JSON.stringify(data.members))));

};

return (

<>

<h1>JSON Solution </h1>

<button onClick={result}>Parse Data</button>

</>

);

}

export default JsonProblem;

**Section – III**

**Q** **1: Differentiate arrow functions and regular functions.**

**Ans:**

|  |  |
| --- | --- |
| ***Arrow Functions*** | ***Regular Function*** |
| * in Arrow function you can skip return keyword and write in single line * In arrow function “this” equals to this of the outer function. * In arrow function arguments is not defined | * In regular function, you always must return any value * In regular function, “this” value is dynamic * In regular function, arguments will give you list of parameters passed in function |

**Q2: Write any three different ways to get an element from DOM?**

Different ways to access element from DOM are mentioned below.

1. Get HTML element by **Id**
2. Get HTML element by **Name**
3. Get HTML element by **class Name**

**Q3: Write a program that takes a number of greater than 5 digits from input field and shows sum of all odd numbers. For example, if the input is 196783, the sum would be 1 + 9 + 7 + 3 = 20.**

**Ans:**

<body>

<h1>Section - III</h1>

<h3>Q3: Write a program that takes a number of greater than 5 digits from input field and

shows sum of all odd numbers. For example, if the input is 196783, the sum would

be 1 + 9 + 7 + 3 = 20.</h3>

<script>

function reverse(n) {

let rev = 0;

while (n != 0) {

rev = (rev \* 10) + (n % 10);

n = Math.floor(n / 10);

}

// console.log(`rev = ${rev}`)

return rev;

}

function getSum(n) {

document.write(`<b>actual number = ${n} </b><br>`)

n = reverse(n);

let sumOdd = 0, c = 1;

while (n != 0) {

if (c % 2 == 1)

sumOdd += n % 10;

n = Math.floor(n / 10);

c++;

}

document.write("Sum of odd = " + sumOdd);

document.write("<br>");

}

let n = 345678;

getSum(n);

</script>

</body>

**Q4: Given an integer N, the task is to find the sum of interior angles of an N-sided polygon.**

**Ans:**

function sumOfInteriorAngles(n)

{

if (n < 3)

return 0;

else

return (n - 2) \* 180;

}

let n = 6;

document.write(sumOfInteriorAngles(n));

**Q** **5: Create a age calculator that takes date of birth of user and shows age in years,**

**months and days.**

**Ans:**

<body>

<div id="container" class="container mt-5">

<h1>Section - III</h1>

<form id="form" autocomplete="off">

<h3 class="text-primary mb-5">Q5: Calculate your data of birth here</h3>

<label for="date">

Date Of Birth:

<input type="text" maxlength="2" size="2" id="date" placeholder="Date" required />

<input type="text" maxlength="2" size="2" id="month" placeholder="Month" autocomplete="on" required />

<input type="text" maxlength="4" size="4" id="year" placeholder="Year" required />

</label>

<br />

<br />

<label for="date2">

Today's Date:

<input type="text" maxlength="2" size="2" id="date2" placeholder="Date" />

<input type="text" maxlength="2" size="2" id="month2" placeholder="Month" />

<input type="text" maxlength="4" size="4" id="year2" placeholder="Year" />

</label>

<br />

<br />

<button id="calbtn" class="btn btn-primary">Calculate</button>

<br />

<br />

<span id="age"></span>

<span id="months"></span>

<span id="days"></span>

</form>

</div>

<script>

var form = document.getElementById("form"),

bdate = document.getElementById("date"),

bmonth = document.getElementById("month"),

byear = document.getElementById("year"),

date = document.getElementById("date2"),

month = document.getElementById("month2"),

year = document.getElementById("year2"),

age = document.getElementById("age"),

days = document.getElementById("days"),

mons = document.getElementById("months"),

today = new Date();

// console.log(today);

year.value = today.getFullYear();

month.value = today.getMonth() + 1;

date.value = today.getDate();

form.addEventListener('submit', function (event) {

event.preventDefault();

var by = Number.parseFloat(byear.value),

bm = Number.parseFloat(bmonth.value),

bd = Number.parseFloat(bdate.value),

ty = Number.parseFloat(year.value),

tm = Number.parseFloat(month.value),

td = Number.parseFloat(date.value);

if (td < bd) {

days.innerHTML = (td - bd + 30) + ' days';

tm = tm - 1;

} else {

days.innerHTML = (td - bd) + ' days'

}

if (tm < bm) {

months.innerHTML = (tm - bm + 12) + ' months';

ty = ty - 1;

} else {

months.innerHTML = (tm - bm) + ' months'

}

age.innerHTML = "Age: " + (ty - by) + ' years';

});

</script>

</body>

**Section – IV**

**Q1:** **Showcase your solid knowledge of JavaScript, DOM, JSON and AJAX by creating a**

**simple webpage that displays information of pets from an API (URLs given below)….**

**Ans:**

<body>

<div class="container mt-5">

<button class="btn btn-primary" onclick="fetchData()">Fetch Data </button>

<script>

function fetchData() {

var url1 = 'https://learnwebcode.github.io/json-example/animals-1.json';

// var url2 = 'https://learnwebcode.github.io/json-example/animals-2.json';

// var url3 = 'https://learnwebcode.github.io/json-example/animals-3.json';

async function funcName(url) {

const response = await fetch(url);

var data = await response.json();

console.log("Data ", data)

}

funcName(url1);

}

</script>

</div>

</body>