Assignment Day 5 | 20th July 2020

Question 1:

Write a program which does the following things:

1. Takes a positive number from the user.

Answer:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>Day 5</title>
<script>
function check_number() {
var num_value= document.getElementById("num_value").value;
if (num_value >=0 ) {
result = num_value + " is a Positive Number.";
}
else {
alert("Enter athe Positive Number")
}
document.getElementById("demo").innerHTML = result;
}
</script>
```

```
</head>
<body>
<h3> Positive Number Checker </h3>
<br/><p class="num">Enter a Number</p>
<input type="text" id="num_value" name="num_value" size="5">
<br><br>
<button onclick="check_number()">Check Number</button>
<br><br>
<p id="demo"></p>
</div>
</body>
</html>
```

2.Makes an array of numbers till the number given by user.

Answer:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title></title>
<script>var inputArray = [];
var size = 5; //Maximum Array size
for(var i=0; i<size; i++) {
//Taking Input from user
inputArray[i] = prompt('Enter Element ' + (i+1));

}

//Print the array in the console.

console.log(inputArray);</script>

</head>

<body>

</body>

</html>
```

3. Use higher order function to filter the odd numbers

Answer:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title></title>

<script>

function countBy(arr, fn) {

return arr.filter(function(acc, nums) {

// console.log(nums);

let oddCount = 0

console.log(nums, fn(nums))

if(fn(nums) === "odd"){

oddCount++;

acc['odd'] = oddCount;
}
return acc
}, {}, 0)
}

function odd(n) {
if (n % 2 === 0) return "odd";
}
var nums = [1, 2, 3, 4, 5];
console.log(countBy(nums, odd));
</script>
</head>
<body>
</body>
</html>
```

Question 2:

Create a Class User with properties as name, age & email.

He can login and logout

Create another class Moderator which has all the features of User, plus additional functionality to Add coins and remove coins.

Create one more class Admin which has all the features of Moderator plus additional features like add a Course and delete a course for a user.

Answer:

```
<html>
<body>
```

```
<script>
  class User
{
   constructor(name, age, email)
   {
      this.name = name;
      this.age = age;
      this.email = email;
      this.coins = 0;
      this.courses = [];
   }
   logIn()
   {
      console.log(`${this.name} have logged in.`);
   }
   logOut()
   {
      console.log(`${this.name} have logged out.`);
   }
}

class Moderator extends User
{
   constructor(name, age, email)
   {
      super(name, age, email);
   }
```

```
    addCoins(user)
    {
        let coin = parseInt(prompt("Enter the coins to be added."));
        if(coin > 0)
        {
            user.coins += coin;
        }
        console.log(user);
        return user;
    }
    removeCoins(user)
    {
        let coin = parseInt(prompt("Enter the coins to be removed."));
        if(user.coins > 0)
        {
            user.coins -= coin;
            if(user.coins < 0)
            {
                user.coins = 0;
            }
        }
        console.log(user);
        return user;
    }
}

class Admin extends Moderator
```

```
{

    constructor(name, age, email)

    {

        super(name, age, email);

    }

    addCourses(user, course)

    {

        user.courses.push(course);

        console.log(user);

        return user;

    }

    removeCourse(user, course)

    {

        user.courses = user.courses.filter(element => element != course


    console.log(user);

        return user;

    }

}


let user1 = new User("Syed Aame Ali", 20, "xyz123@gmail.com");

user1.logIn();

user1.logOut();


let moderator1 = new Moderator("Adil", 22, "xyz123@gmail.com");

moderator1.logIn();

user1 = moderator1.addCoins(user1);
```

```
user1 = moderator1.removeCoins(user1);

moderator1.logOut();


let admin1 = new Admin("LetUpgrade", 25, "LetsUpgrade@gmail.com");

admin1.logIn();

user1 = admin1.addCourses(user1, "Python");

user1 = admin1.addCourses(user1, "JavaScript");

user1 = admin1.removeCourse(user1, "Python");

admin1.logOut();

</script>

</body>

</html>
```

Question 3:

Using promises/async await/fetch get the random todos from the json placeholder api. And log all the completed todos to the console. API Endpoint : https://jsonplaceholder.typicode.com/todos.

Answer:

```
<html>

<body>

<script>

  let jsonObject = [];

let url = "https://jsonplaceholder.typicode.com/todos";

fetch(url)

.then(response => response.json())

.then(data => console.log(data))

.catch(error => console.log(error));
```

```
async function completedToDos()
{
    const response = await fetch(url);56
    let data = await response.json();
    console.log(data);

    for(let i = 0; i < 200; i++)
    {
        jsonObject = JSON.stringify(data[i]);
        if(jsonObject.includes("true"))
        {
            console.log(jsonObject);
        }
    }
}
completedToDos();
let str = jsonObject.forEach(element => {
    JSON.stringify(element);
});
let data = JSON.stringify(jsonObject);
console.log(data)
console.log(str);
</script>
</body>
</html>
```