UNIVERSITI MALAYSIA TERENGGANU
FACULTY OF OCEAN ENGINEERING TECHNOLOGY & INFORMATICS

Lab 3 (MP3)

CSM3103-K2-Front-End Development

Prepared for

DR RABIEI B MAMAT

Prepared by

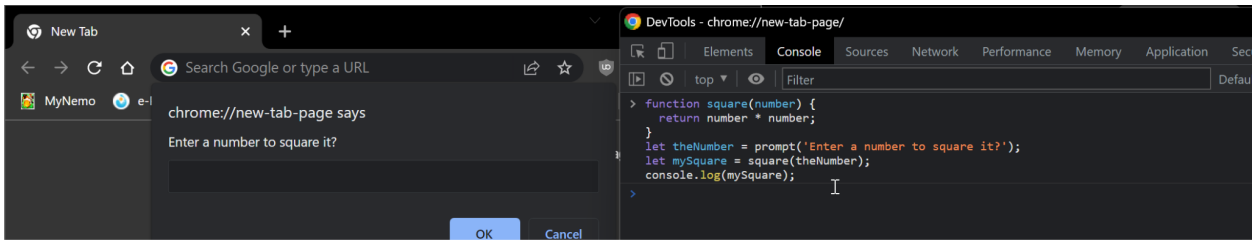
OMAR ISMAIL ABDJALEEL ALOMORY

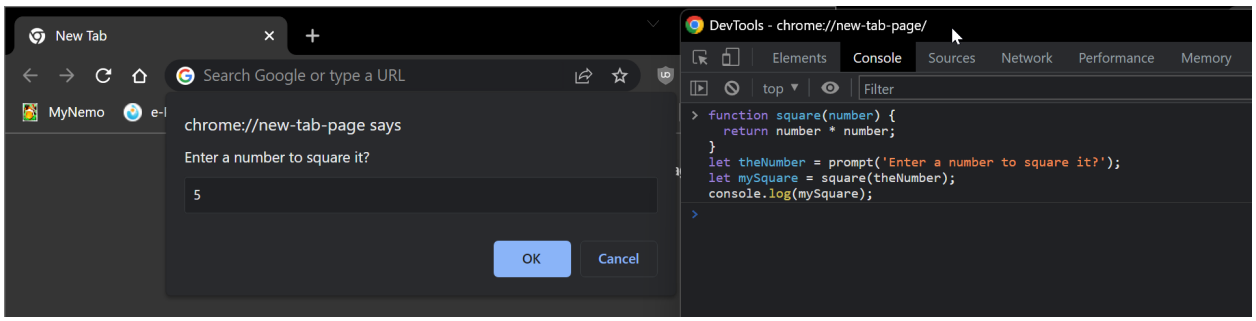
Task 1 – JavaScript Function

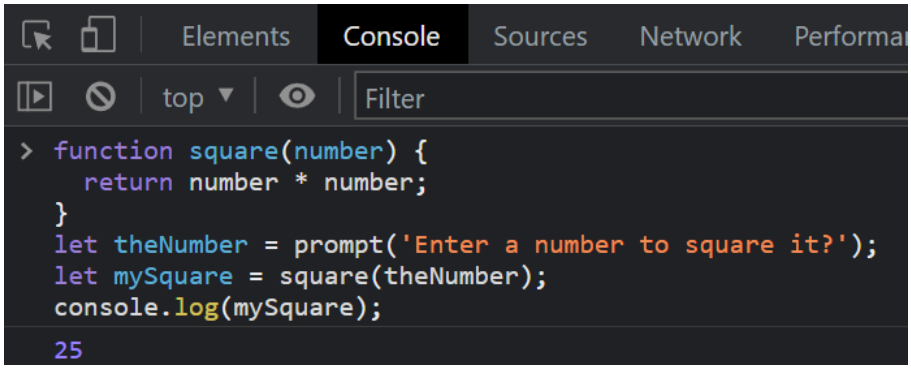
1. Write a function to find the square of a given number

```
// 1
function square(number) {
  return number * number;
}
let theNumber = prompt('Enter a number to square it?');
let mySquare = square(theNumber);
console.log(mySquare);
```

Output





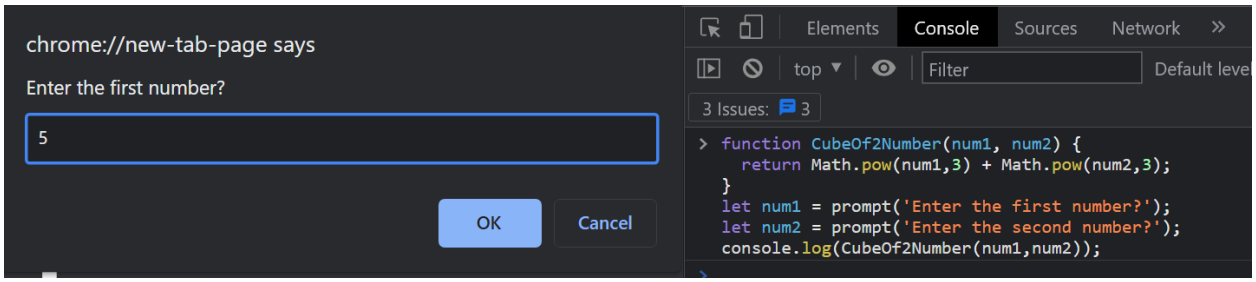


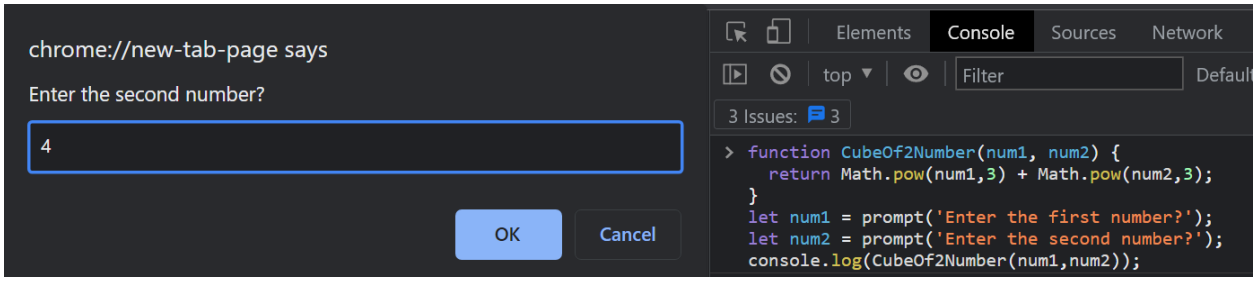
The square of 5 is 25.

2. Write a function to find sum of cubes of two numbers

```
// 2
function CubeOf2Number(num1, num2) {
  return Math.pow(num1,3) + Math.pow(num2,3);
}
let num1 = prompt('Enter the first number?');
let num2 = prompt('Enter the second number?');
console.log(CubeOf2Number(num1,num2));
//-----------------
```

Output





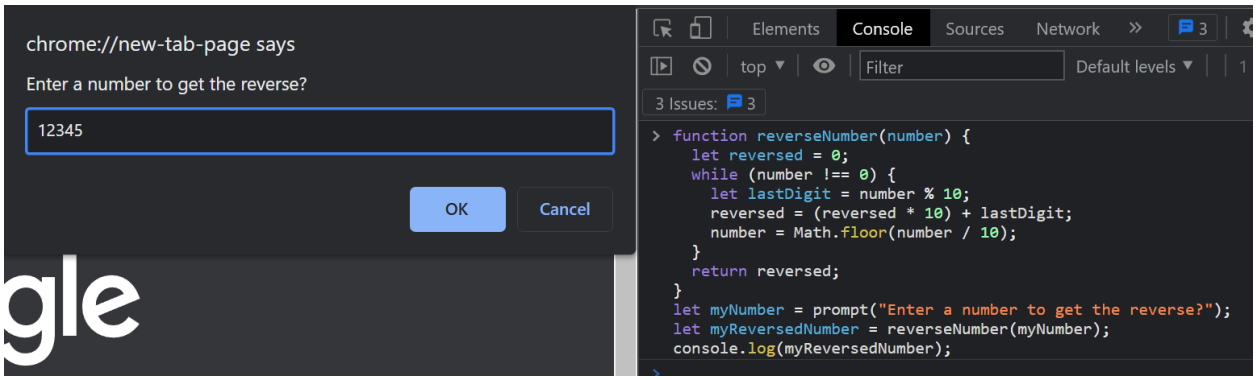

```
function CubeOf2Number(num1, num2) {
  return Math.pow(num1,3) + Math.pow(num2,3);
}
let num1 = prompt('Enter the first number?');
let num2 = prompt('Enter the second number?');
console.log(CubeOf2Number(num1,num2));
189
undefined
```

The cube of 5 and 4 are 125 and 64 which equal to 189

3. Write a function to reverse a number [Hint n =12345 output : 54321]

```
// 3
function reverseNumber(number) {
  let reversed = 0;
  while (number !== 0) {
    let lastDigit = number % 10;
    reversed = (reversed * 10) + lastDigit;
    number = Math.floor(number / 10);
  }
  return reversed;
}
let myNumber = prompt("Enter a number to get the reverse?");
let myReversedNumber = reverseNumber(myNumber);
console.log(myReversedNumber);
```

Output



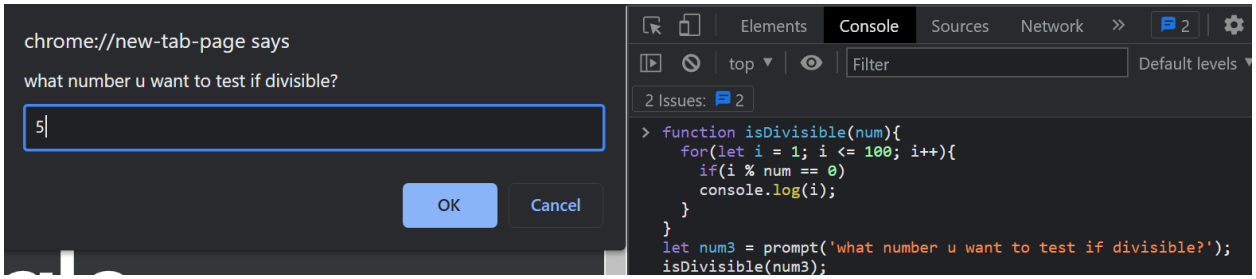

```
function reverseNumber(number) {
  let reversed = 0;
  while (number !== 0) {
    let lastDigit = number % 10;
    reversed = (reversed * 10) + lastDigit;
    number = Math.floor(number / 10);
  }
  return reversed;
}
let myNumber = prompt("Enter a number to get the reverse?");
let myReversedNumber = reverseNumber(myNumber);
console.log(myReversedNumber);
54321
```

Reverse of 12345 is 54321

4. Write a function to print all numbers between 1 and 100 which is divisible by given number z

```
function isDivisible(num){
  for(let i = 1; i <= 100; i++){
    if(i % num == 0)
    console.log(i);
  }
}
let num3 = prompt('what number u want to test if divisible?');
isDivisible(num3);
```

Output



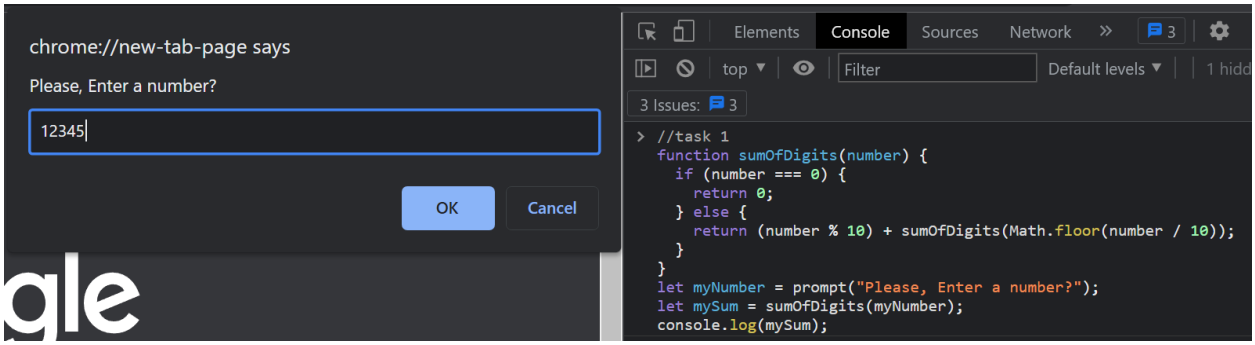

```
function isDivisible(num){
  for(let i = 1; i <= 100; i++){
    if(i % num == 0)
    console.log(i);
  }
}
let num3 = prompt('what number u want to test if divisible?');
isDivisible(num3);
5 VM37:4
10 VM37:4
15 VM37:4
20 VM37:4
25 VM37:4
30 VM37:4
35 VM37:4
40 VM37:4
45 VM37:4
50 VM37:4
55 VM37:4
60 VM37:4
65 VM37:4
70 VM37:4
75 VM37:4
80 VM37:4
85 VM37:4
90 VM37:4
95 VM37:4
100 VM37:4
```

Task 2 - JavaScript Recursion Function

1. Write a JavaScript function to find sum of digits of a number

```
//task 1
function sumOfDigits(number) {
  if (number === 0) {
    return 0;
  } else {
    return (number % 10) + sumOfDigits(Math.floor(number / 10));
  }
}
let myNumber = prompt("Please, Enter a number?");
let mySum = sumOfDigits(myNumber);
console.log(mySum);
```

Output



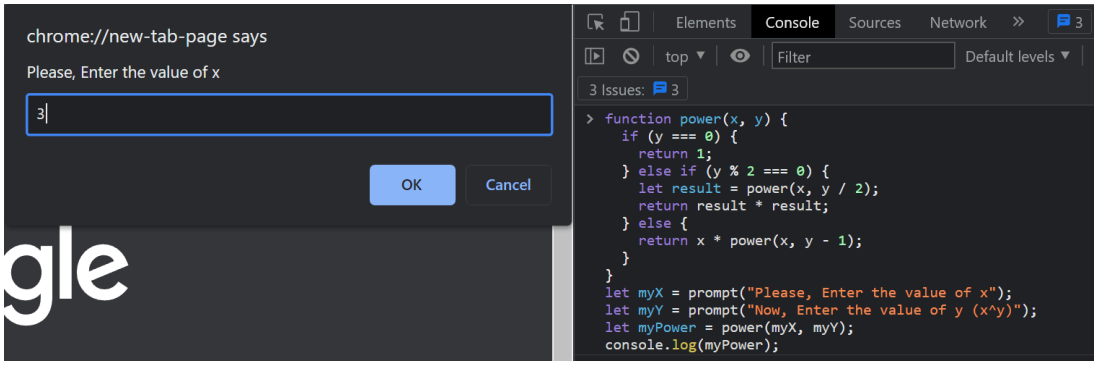

```
> //task 1
function sumOfDigits(number) {
  if (number === 0) {
    return 0;
  } else {
    return (number % 10) + sumOfDigits(Math.floor(number / 10));
  }
}
let myNumber = prompt("Please, Enter a number?");
let mySum = sumOfDigits(myNumber);
console.log(mySum);
15                                                    VM37:11
```

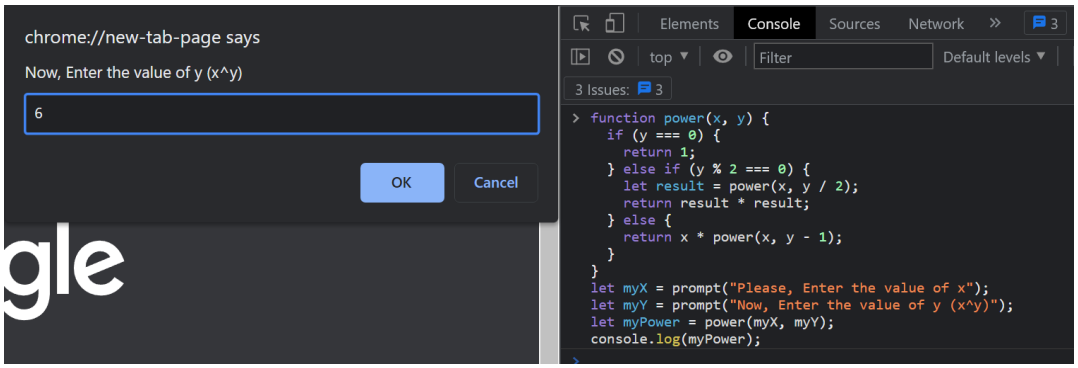
The sum of digits (12345) = 1 +2 +3 +4 +5 = 15

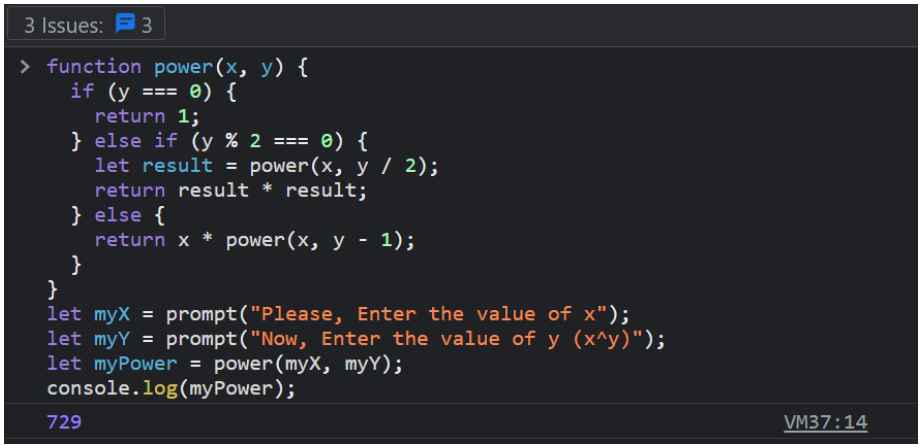
2. Write a JavaScript program to compute x raise to the power y using recursion

```
//task 2
function power(x, y) {
  if (y === 0) {
    return 1;
  } else if (y % 2 === 0) {
    let result = power(x, y / 2);
    return result * result;
  } else {
    return x * power(x, y - 1);
  }
}
let myX = prompt("Please, Enter the value of x");
let myY = prompt("Now, Enter the value of y (x^y)");
let myPower = power(myX, myY);
console.log(myPower);
```

Output







Here we are using the recursion to get the value of x raised to y, so if the input for x here is 3 and for y is 6 we get 729

Task 3 – JavaScript Object and Prototype

1. Write a JavaScript program to create object product

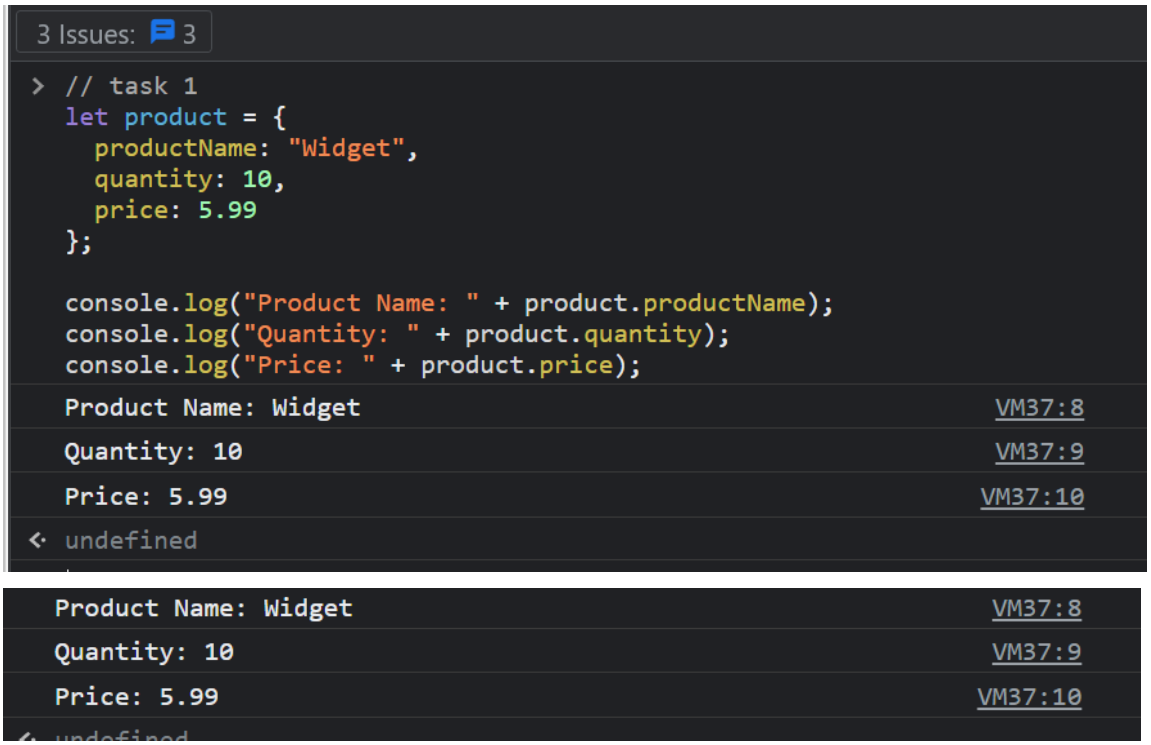
a. Add the property Product Name, Quantity and price.

```
// task 1
let product = {
  productName: "Widget",
  quantity: 10,
  price: 5.99
};
```

b. Access all the properties and display them.

```
console.log("Product Name: " + product.productName);
console.log("Quantity: " + product.quantity);
console.log("Price: " + product.price);
```

Output



2. Write a JavaScript program to create object book

a. Add the property book name, author name

```
function Book(bookName, authorName) {
  this.bookName = bookName;
  this.authorName = authorName;
}
```

b. Add the prototype property price .

```
Book.prototype.price = 10.99;
```

c. Display all the properties.

```
let myBook = new Book("The Great Gatsby", "F. Scott Fitzgerald");

console.log("Book Name: " + myBook.bookName);
console.log("Author Name: " + myBook.authorName);
console.log("Price: " + myBook.price);
```

Output

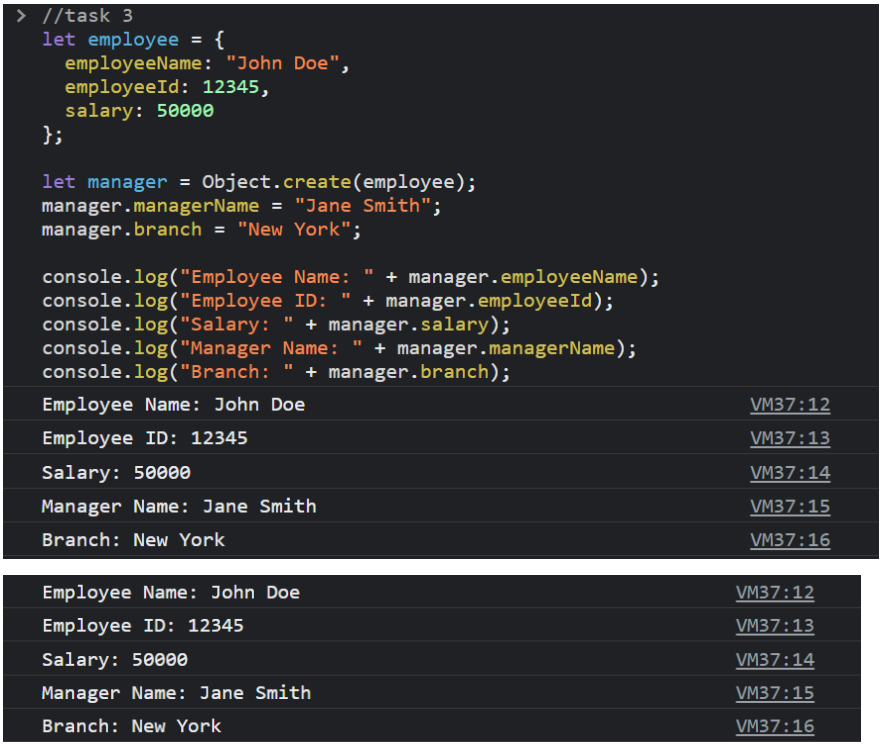
```
> // task 2
  function Book(bookName, authorName) {
    this.bookName = bookName;
    this.authorName = authorName;
  }

  Book.prototype.price = 10.99;

  let myBook = new Book("The Great Gatsby", "F. Scott Fitzgerald");

  console.log("Book Name: " + myBook.bookName);
  console.log("Author Name: " + myBook.authorName);
  console.log("Price: " + myBook.price);
  Book Name: The Great Gatsby                    VM37:11
  Author Name: F. Scott Fitzgerald               VM37:12
  Price: 10.99                                   VM37:13
< undefined
```


```
Book Name: The Great Gatsby                    VM37:11
Author Name: F. Scott Fitzgerald               VM37:12
Price: 10.99                                   VM37:13
< undefined
```

3. Write a JavaScript program to create Parent object employee (Property : Employee Name , Employee Id , Salary) and Child object Manager (Property :Manager Name , Branch). Inherit all the properties of employee and display all the properties.

```
//task 3
let employee = {
  employeeName: "John Doe",
  employeeId: 12345,
  salary: 50000
};

let manager = Object.create(employee);
manager.managerName = "Jane Smith";
manager.branch = "New York";

console.log("Employee Name: " + manager.employeeName);
console.log("Employee ID: " + manager.employeeId);
console.log("Salary: " + manager.salary);
console.log("Manager Name: " + manager.managerName);
console.log("Branch: " + manager.branch);
```

Output

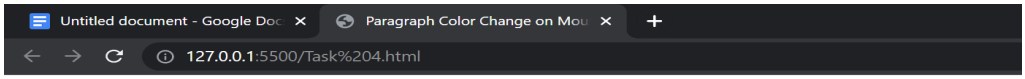


Employee Name: John Doe VM37:12
Employee ID: 12345 VM37:13
Salary: 50000 VM37:14
Manager Name: Jane Smith VM37:15
Branch: New York VM37:16

Here the manager object inherited all the attributes from the employee object

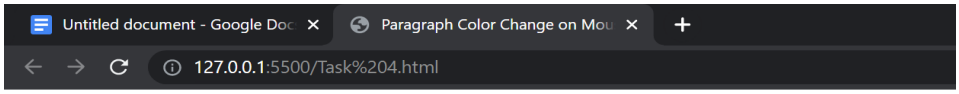
Task 4 – Event Manager

1. Create a HTML page with <p> paragraph. Change the paragraph color according to the following mouse events



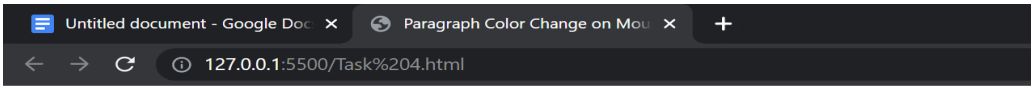
This is a paragraph. Hover over it or click on it to see the color change.

a. Onclick, yellow background



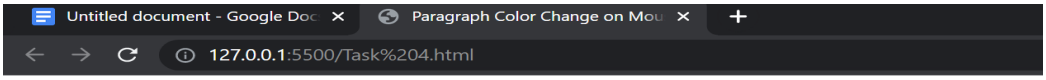
This is a paragraph. Hover over it or click on it to see the color change.

b. ondblclick, blue background



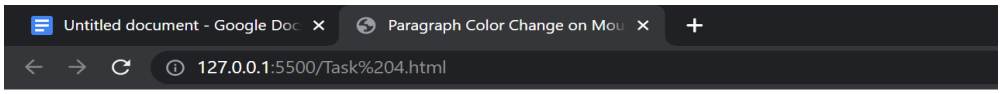
This is a paragraph. Hover over it or click on it to see the color change.

c. onmouseover , red background



This is a paragraph. Hover over it or click on it to see the color change.

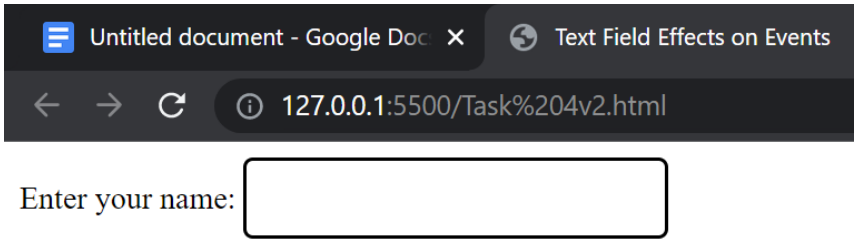
d. onmouseout, green background



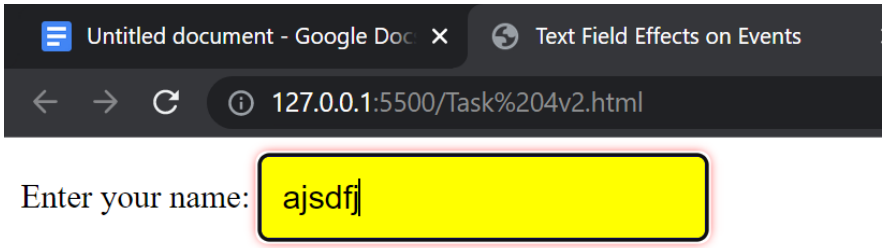
This is a paragraph. Hover over it or click on it to see the color change.

2. Create a HTML page with textfield. Show some effects on the textfield when the following events occurred:

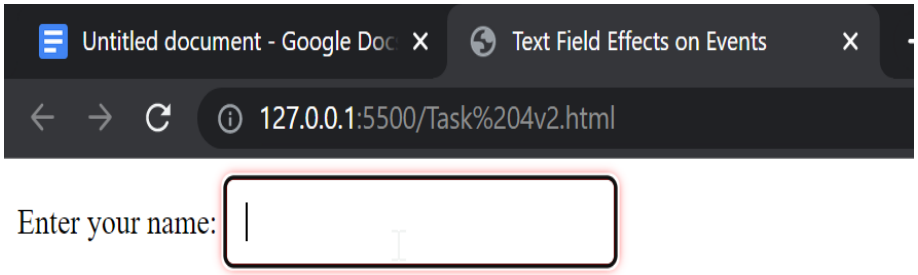
No event applied yet.



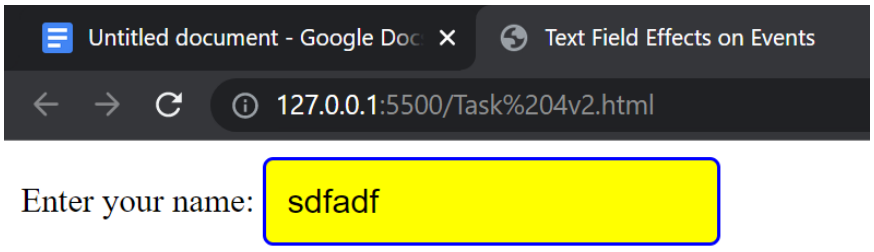
a. Onchange (background color of the text area would be yellow)



b. Onfocus (the border of the text area would be red shadow)



c. onblur (the border color would change to blue)



Task 5 Given the following HTML table

1	Ahmad Faisal	ahmadfaisal@gmail.com	0199088888
2.	Ismail Sabri	isabri@mail.com	0199076760
3	Fateh Yakin	ffateh@hotmail.com	0176067762

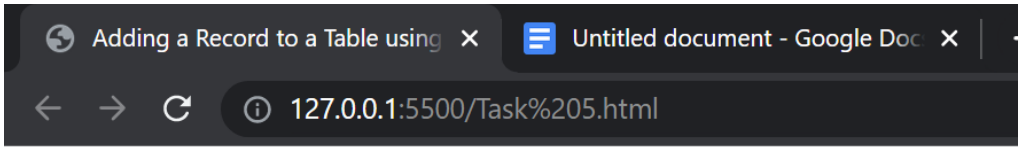
1. Using javascript add the following record into table

a. Name: Mukhriz Jamil Asoka

b. Email: mukriz@corp.jo

c. Phone: 651181187223

Given table.

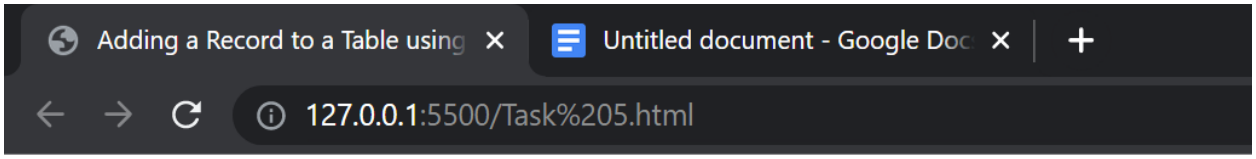


1	Ahmad Faisal	ahmadfaisal@gmail.com	0199088888
2	Ismail Sabri	isabri@mail.com	0199076760
3	Fateh Yakin	ffateh@hotmail.com	0176067762

Js code

```
// task 1
var table = document.querySelector("table");
var tbody = table.querySelector("tbody");

var newRow = document.createElement("tr");
var newNumber = document.createElement("td");
var newName = document.createElement("td");
var newEmail = document.createElement("td");
var newPhone = document.createElement("td");

newNumber.textContent = tbody.children.length + 1;
newName.textContent = "Mukhriz Jamil Asoka";
newEmail.textContent = "mukhriz@corp.jo";
newPhone.textContent = "651181187223";

newRow.appendChild(newNumber);
newRow.appendChild(newName);
newRow.appendChild(newEmail);
newRow.appendChild(newPhone);

tbody.appendChild(newRow);
```

Output



1	Ahmad Faisal	ahmadfaisal@gmail.com	0199088888
2	Ismail Sabri	isabri@mail.com	0199076760
3	Fateh Yakin	ffateh@hotmail.com	0176067762
4	Mukhriz Jamil Asoka	mukhriz@corp.jo	651181187223

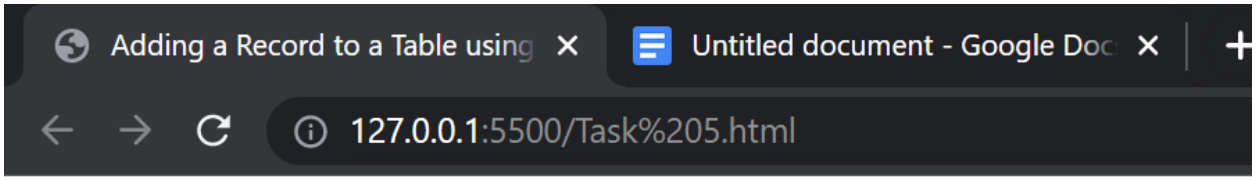
2. Using javascript add the table header as follow:

a. #, Name, Email, Phone

JS code

```
//Task 2
  var table = document.querySelector("table");
  var tbody = table.querySelector("tbody");

  // Create the table header row
  var thead = document.createElement("thead");
  var headerRow = document.createElement("tr");

  // Create and append the header cells
  var headerNumber = document.createElement("th");
  headerNumber.textContent = "#";
  headerRow.appendChild(headerNumber);

  var headerName = document.createElement("th");
  headerName.textContent = "Name";
  headerRow.appendChild(headerName);

  var headerEmail = document.createElement("th");
  headerEmail.textContent = "Email";
  headerRow.appendChild(headerEmail);

  var headerPhone = document.createElement("th");
  headerPhone.textContent = "Phone #";
  headerRow.appendChild(headerPhone);

  // Append the header row to the thead element
  thead.appendChild(headerRow);

  // Append the thead element to the table element
  table.appendChild(thead);
```

Output



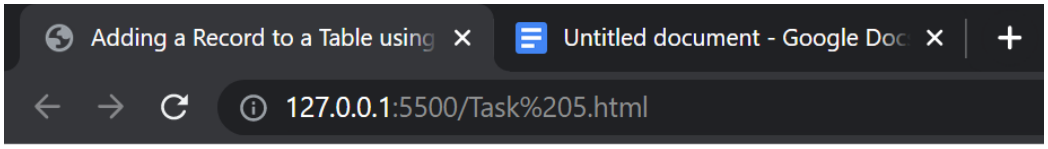
#	Name	Email	Phone #
1	Ahmad Faisal	ahmadfaisal@gmail.com	0199088888
2	Ismail Sabri	isabri@mail.com	0199076760
3	Fateh Yakin	ffateh@hotmail.com	0176067762
4	Mukhriz Jamil Asoka	mukhriz@corp.jo	651181187223

3. Using javascript, delete any row from table when clicked on that row

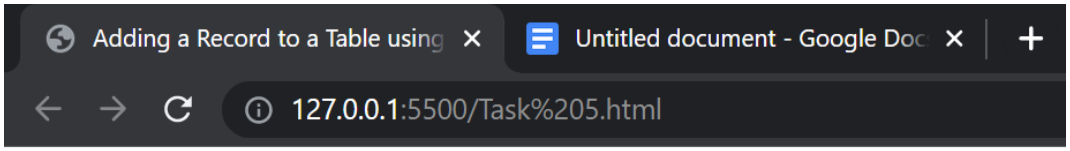
JS code

```
//task 3 deleting row when clicked
var table = document.querySelector("table");
var rows = table.querySelectorAll("tr");

for (var i = 0; i < rows.length; i++) {
  rows[i].addEventListener("click", function () {
    this.parentNode.removeChild(this);
  });
}
```

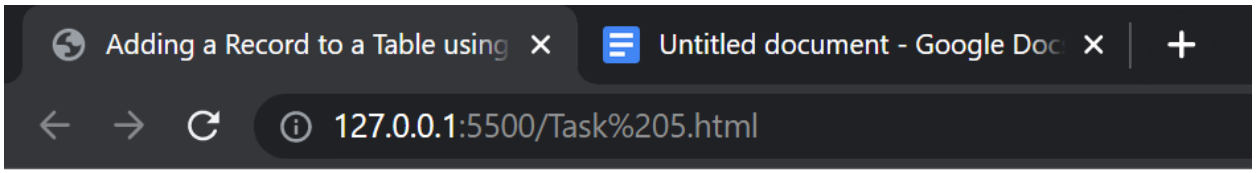
When we click a row it would be deteleted



#	Name	Email	Phone #
1	Ahmad Faisal	ahmadfaisal@gmail.com	0199088888
2	Ismail Sabri	isabri@mail.com	0199076760
3	Fateh Yakin	ffateh@hotmail.com	0176067762
4	Mukhriz Jamil Asoka	mukhriz@corp.jo	651181187223



#	Name	Email	Phone #
1	Ahmad Faisal	ahmadfaisal@gmail.com	0199088888
2	Ismail Sabri	isabri@mail.com	0199076760
3	Fateh Yakin	ffateh@hotmail.com	0176067762
4	Mukhriz Jamil Asoka	mukhriz@corp.jo	651181187223



#	Name	Email	Phone #
4	Mukhriz Jamil Asoka	mukhriz@corp.jo	651181187223

Task 6

Write a JavaScript program to move two small squares inside one big square in a random manner. User should be able to start and stop this animationusing button based events

Math.floor(Math.random() * Math.floor(max)) will give you a random number that is less than max value

Source code

HTML

```html
<div id="big-square">
    <div id="small-square1" class="small-square"></div>
    <div id="small-square2" class="small-square"></div>
</div>

<button id="start-btn">Start</button>
<button id="stop-btn">Stop</button>
```

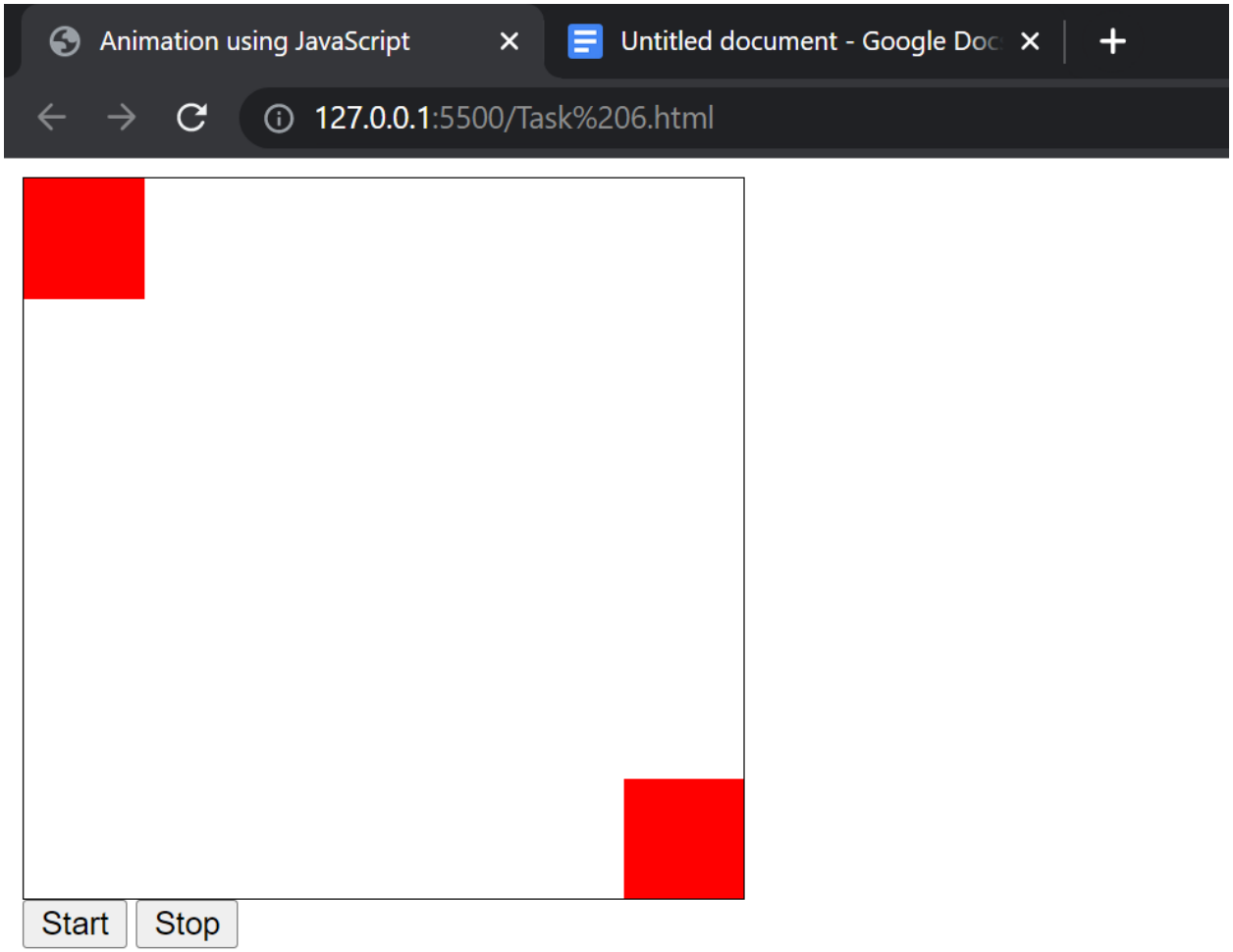
CSS

```css
#big-square {
    position: relative;
    width: 300px;
    height: 300px;
    border: 1px solid black;
}
.small-square {
    position: absolute;
    width: 50px;
    height: 50px;
    background-color: red;
}
#small-square1 {
    top: 0;
    left: 0;
}
#small-square2 {
    bottom: 0;
    right: 0;
}
```

JS

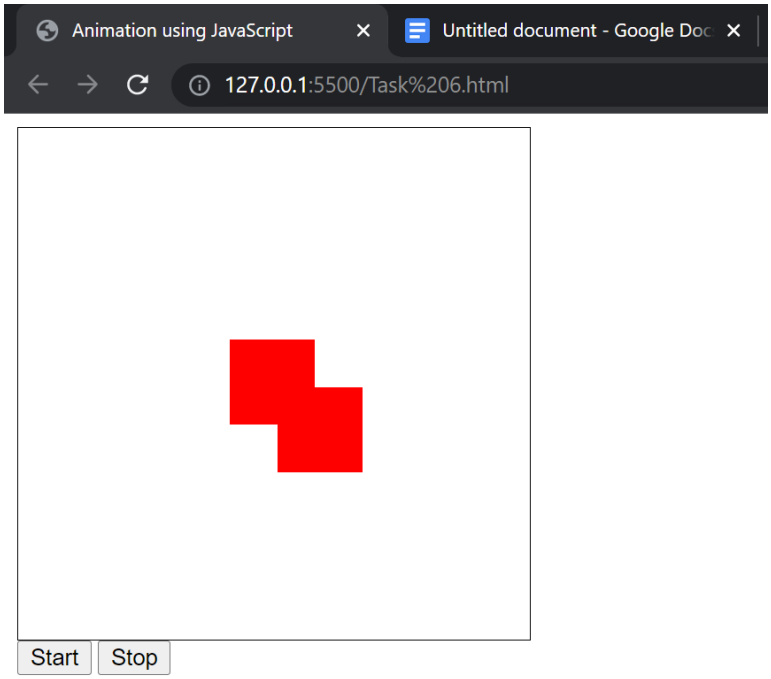
```
<script>
    const bigSquare = document.getElementById("big-square");
    const smallSquare1 = document.getElementById("small-square1");
    const smallSquare2 = document.getElementById("small-square2");
    const startBtn = document.getElementById("start-btn");
    const stopBtn = document.getElementById("stop-btn");

    let intervalId;

    function moveSquares() {
        const max = 200;
        const min = 0;
        const random1 = Math.floor(Math.random() * (max - min + 1)) + min;
        const random2 = Math.floor(Math.random() * (max - min + 1)) + min;
        smallSquare1.style.top = random1 + "px";
        smallSquare1.style.left = random1 + "px";
        smallSquare2.style.bottom = random2 + "px";
        smallSquare2.style.right = random2 + "px";
    }

    startBtn.addEventListener("click", function() {
        intervalId = setInterval(moveSquares, 1000);
    });

    stopBtn.addEventListener("click", function() {
        clearInterval(intervalId);
    });
</script>
```

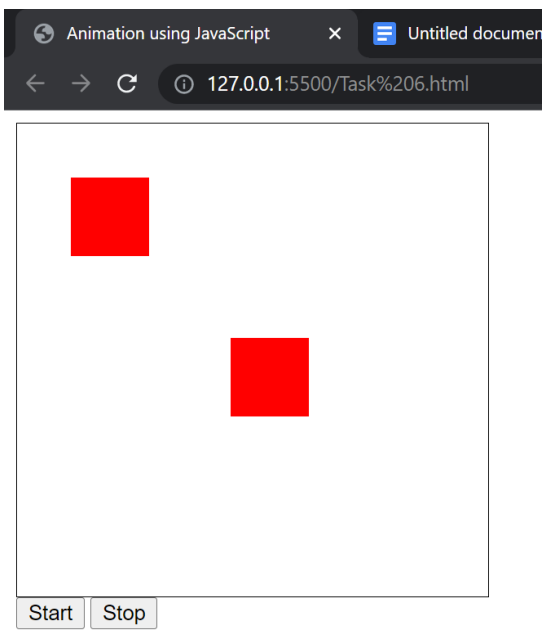
The two squares will initially start each in different corner as shown in the figure below

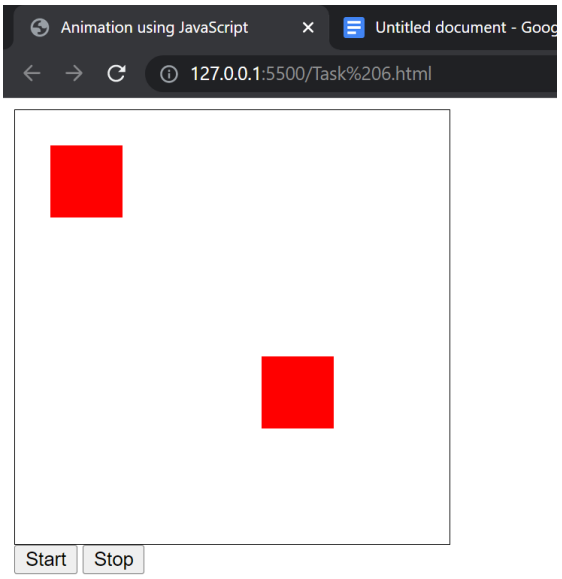


The small squares have the same css properties but the big square is different to contain them. The big square have relative position and the small squares have absolute position to allow overlapping between the small squares. We have two buttons, one to start the interval and the two squares will start moving in the big square randomly.

The values of movement obtained from the class Math and the method random in the class math. We also have two final integers which are max = 200 and min = 0 , so the random value will be multiplied by (max - min +1)+min. We will have two random values with the same concept and then assign these values to the small squares. As we mentioned earlier the small squats will start from different corners because we initialized the top-left square with top = 0 and left = 0, and the right-bottom square with right = 0 and bottom = 0. So when the obtained values from the random number are assigned to each top left or right bottom, they would move randomly in the bigger square.







Github : http://github.com/Alomory/CSM3103