

## Practical: JavaScript(II)

### Objective:

- Use Variables - global vs local variables
- Define simple function
- Use Event handler

### Part A: Activity 1

1. Create **ex1.html** with the following code:

```
<html>
<head>
<title>JavaScript - using variables</title>
<script>
  var food="ice-cream";
</script>
</head>
<body>
<script>
  document.write("I like "+food);
</script>
</body>
</html>
```

2. Test it out!
3. Now, modify the code to the following:

```
<html>
<head><title>JavaScript - using variables</title>
<script>
  function eat() {
    var food="ice-cream";
  }
</script>
</head>
<body>
<script>
  document.write("I like "+food);
</script>
</body>
</html>
```

4. What do you get when you view the web page through browser? Explain why it is different from the previous output.

## Part A: Activity 2

### *Using function*

1. Create another AddTwoNumbers.html file with the following content:

```
<html>
<head>
<title>JavaScript - using function</title>
<script>
  var num1=1;
  var num2=2;
  alert(num1+num2);
</script>
</head>
<body>
</body>
</html>
```

2. What is the output when you view the web page through browser?

3. Modify the code to the following:

```
<html>
<head>
<title>JavaScript</title>
<script>
  function add() {
    var num1=1;
    var num2=2;
    alert(num1+num2);
  }
</script>
</head>
<body>
</body>
</html>
```

## IT1301 UX Design

4. What do you get when you view the web page through browser? Explain why it is different from the previous output.

5. Now, modify the code to the following:

```
<html>
<head>
<title>JavaScript - using function</title>
<script type="text/javascript">
    function add() {
        var num1=1;
        var num2=2;
        alert(num1+num2);
    }
</script>
</head>
<body>
    <script>
        add();
        add();
    </script>
</body>
</html>
```

6. What do you get when you view the web page through browser? What is the usage of the 2 lines of JavaScript added?

7. Modify the code to the following:

```
<html>
<head>
<title>JavaScript - using function</title>
<script>
    function add(num1, num2){
        alert(num1+num2);
    }
</script>
</head>
<body>
<script>
    add(2,3);
    add(4,5);
</script>
</body>
</html>
```

8. What do you get when you view the web page through browser? Briefly describe how the function **add** has been changed.

9. Modify the code to the following:

```
<html>
<head>
<title>JavaScript - using function</title>
<script type="text/javascript">
    function add(num1, num2){
        var result=num1+num2;
        return result;
    }
</script>
</head>
<body>
<script>
    add(2,3);
    add(4,5);
</script>
</body>
</html>
```

10. When you view the web page through the browser, there is no output.  
Why?



11. Modify the JavaScript without changing the function, such that the following output will be displayed.



```
2 + 3 =5
4 + 5 =9
```

## **Part B:**

### ***Functions***

1. Write a JavaScript program called sumTriple. This function will compute the sum of the two given integers. If the two values are same, then returns triple their sum.

The function will take in 2 numbers as integers. Use if/else to check if the numbers are the same, if yes, return sum \*3, otherwise return the sum only.

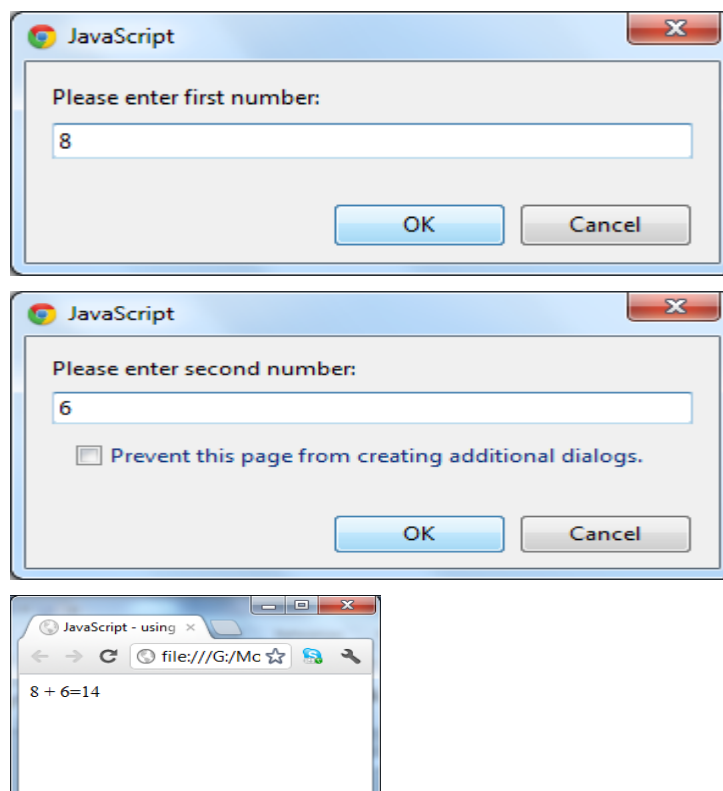
Use the integers – 10, 20 and 20, 20.

### ***Built-in Functions, Prompt Box***

In Part A, the numbers to be added were hardcoded as 2 and 3 or 4 and 5. In this exercise, we are making the code more flexible to accept any 2 numbers.

2. Write a web page which
  - a. Uses the prompt box to prompt user for 2 numbers
  - b. Call the function **add** to add any 2 numbers and display the result

#### **Sample:**



Hints:

JavaScript is loosely typed. By default, user inputs will be treated as string.

Adding 2 strings means concatenate the strings, e.g. "12"+"23" will form a string "1223". To treat "12" as a number twelve, we need to make use of built-in function **parseInt** or **parseFloat** or **Number**

**Example:**

**parseInt("12")+parseInt("23") will give you the result 35.**

3. Write a JavaScript program to get the current date.

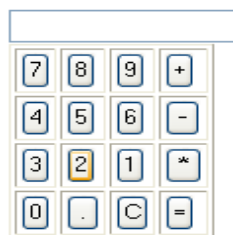
- a. Expected Outputs: dd-mm-yyyy, dd/mm/yyyy
- b. Use getDate(), getMonth() and getFullYear()

## Challenge Level 1:

1. You are to create a simple calculator as shown. Whenever a number button or operator button is clicked, the text will be displayed. When “=” is clicked, the result will be displayed. “C” is to clear the display.

You do not create it from scratch, **calculator.htm** is given to you as part of the practical files (check blackboard). All functions are given; you are to figure out how to call the appropriate function when each of the buttons is clicked.

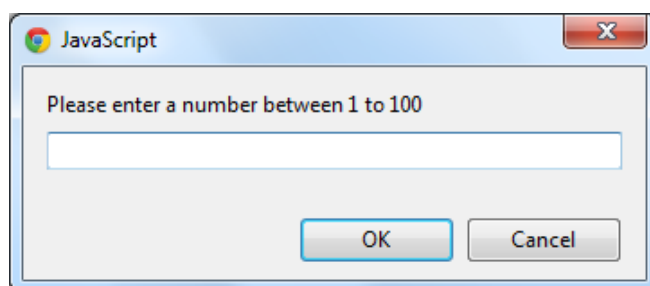
### **Simple Calculator**



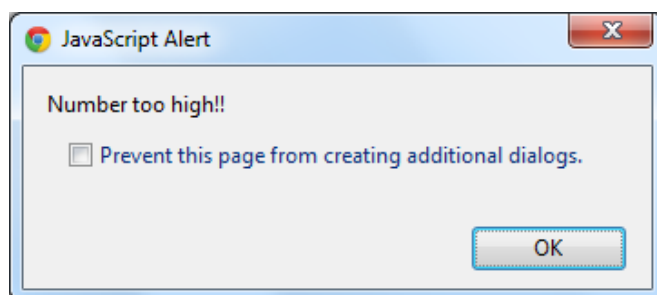
## Challenge Level 2:

Create a **super simple High Low Game** using the JavaScript knowledge that you have learnt. (Partial code is available from blackboard)

Your script should prompt as:



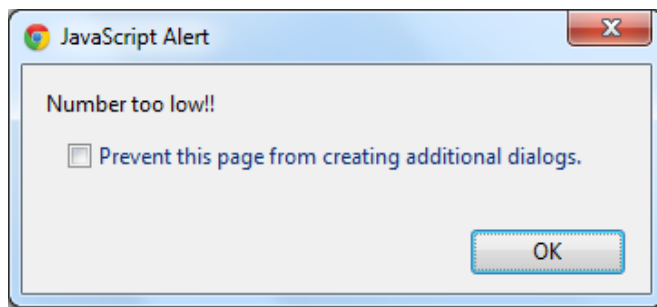
If number entered is too high:



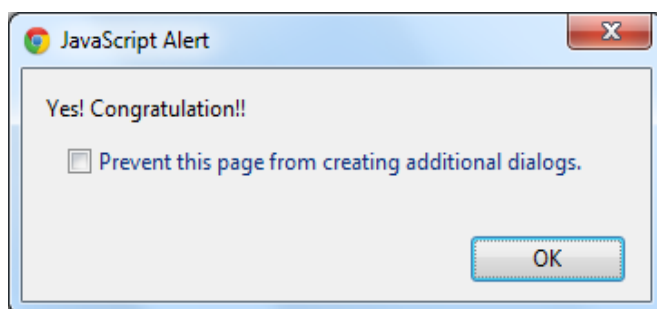


## IT1301 UX Design

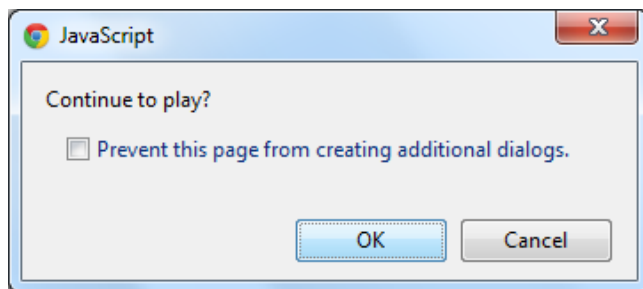
If number entered is too low:



If you got the correct answer:



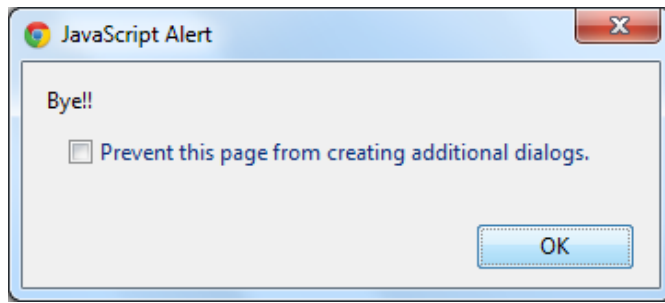
You may further enhance it to prompt this:



If you click ok, your program should generate another number and let you play again.

If you click cancel, your program should say bye and close the window after the ok is clicked.

## IT1301 UX Design



**Hint:** You may use the following user defined function to generate lucky number between 1 and 100.

```
function generateNumber() {  
    luckyNo=Math.floor(Math.random()*100)+1;  
}
```