## Practical 5: JavaScript(I)

## Objective:

## Do simple JavaScript debug using Chrome JavaScript console

## Use Variables – global vs local variables

## Apply JavaScript built-in function

## Define simple function

## Use Event handler

## PartA: Activity 1

## Create ex1.html with the following code:

<html>

<head>

<title>JavaScript - using variables</title>

**<script>**

**document.write("I like "+food);**

**</script>**

</head>

<body>

</body>

</html>

## Use Chrome browser to view ex1.html. Do you see anything?

## Oops! What happened? We will need to use the debugger.

## Right-click to get the following pop-up menu, select Inspect element:

## 

## Choose console:

## 

## Click at the line(s) on the right to go to the line with error:

## 

## food is not defined, so we must declare and assign a value to it. Add the following line before line 6:

## var food="ice-cream";

## Run the code again. What is the output when you view the web page through browser?

I like ice-cream

## Rearrange the code as below, such that the document.write statement is at the body.

<html>

<head>

<title>JavaScript - using variables</title>

**<script>**

**var food="ice-cream";**

**</script>**

</head>

<body>

**<script>**

**document.write("I like "+food);**

**</script>**

</body>

</html>

## Test it out! Does it make any difference?

## 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>

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

A blank screen and in the console a ReferenceError. The reason is because when a variable is in a function, the variable is a local variable, meaning it only works inside the function

## PartA: Activity 2

***Using function***

## Create another 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>

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

There is a popup that says 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>

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

There is a blank page. It is different as the code is in a function. As the function is not called, there is nothing that is executed and shown on the screen

## 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>

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

There are two alerts(popups) stacking on top of each other.

The usage of the 2 lines are used to call the function that is on the head

## 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>

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

As the variables on the function (num1, num2) can be changed when you call the function and add in the numbers, the result changes as well. I see two alerts saying 5 and 9 .

2 + 3 is 5

4 + 5 is 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>

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

There is no output as the function is returning a value and not outputting a popup (alert). To solve this, one can also connect the value to an element to show the value (although it would not be represented in an alert)

e.g <p id=”number”></p>

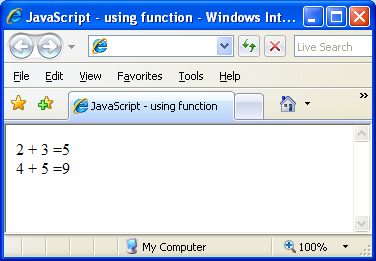
<script>

document.write(add(2,3));

</script>

This is in comparison to the other case with alert where it shows a popup

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



## PartB:

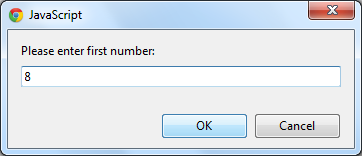
***Prompt Box***

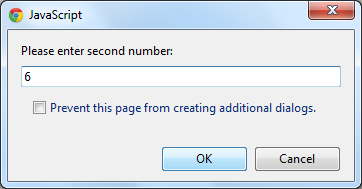
In Part A, The numbers to be added were hardcoded as 2 and 3 or 4 and 5. In this exercise, we are making the code more flexible to accept any 2 numbers.

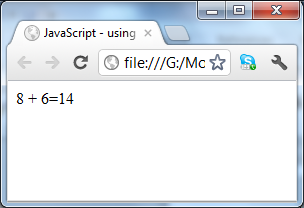
## Write a web page which

* 1. use prompt box to prompt user for 2 numbers
  2. Call the function **add** to add any 2 numbers and display the result

**Sample:**







Hints: If you do not get the correct answer for the sum, you might have forgotten the following that we mentioned in the lecture.

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**

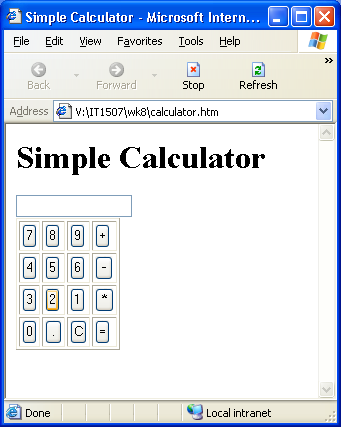
**Example:**

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

**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.

****

## 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:

## 

## 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.

## 

## *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;

## }