

# CPSC 1045 – LAB #2

---

## Objective(s):

1. Learn JavaScript basics
  - a. Values, Types, Operators
  - b. Expressions and Statements
2. Create a webapp using HTML/JavaScript
  - a. Use Values, Types and Operators in your page
  - b. Use Expressions and Statements in your page
  - c. Use buttons, built-in functions, and HTML element access (document.getElementById) in your page
  - d. Use IF statements and switch statement

## Resource(s):

- <http://www.w3schools.com/default.asp>
- Web Browser and a computing device (campus or personal)
- Lecture and Lab notes
- Prior Labs and Worksheets
- Appendix A of this document

## Grading Scheme:

- Tasks I, and II - **15 Points**
- Task III – **85 Points**
  - 30 Points for following directions in each webapp (15 points each) – submission directions, using templates provided, following instructions outlined for each webapp
  - 25 Points for correct Guessing Game webapp
  - 30 Points for correct What Province Are You From webapp
  - 30 Points for correct My Next Birthday webapp (**bonus**)

## Assigned Task(s):

- I. Evaluate IF Statement ( **7 Points** )
  - a. Given the following code, answer the questions below ( **2 Points** )

```
if (word != "foo") {  
    console.log ("You ain't got foo!<BR>");  
}
```

1. What will be displayed on the console if the value of the variable **word** is “bar” when the IF statement is executed?
2. What will be displayed on the console if the value of the variable **word** is “foo” when the IF statement is executed?

b. Given the following code, answer the questions below ( 2 Points )

```
if (num1 % num2 == 0) {  
    console.log(num2 + " does divide into " + num1 + " evenly");  
}  
else {  
    console.log(num2 + " does not divide into " + num1 + " evenly");  
}  
console.log("ok");
```

1. What will be displayed on the console if the value of the variable **num1** is 12, and the value of the variable **num2** is 5, when the IF statement is executed?
2. What will be displayed on the console if the value of the variable **num1** is 23, and the value of the variable **num2** is 23, when the IF statement is executed?

c. Given the following code, and assume x, y, z are valid numbers, answer the questions below ( 3 Points )

```
if ( !isNaN(x) && !isNaN(y) && !isNaN(z) ){  
    if ( x < 0 ){  
        alert("Snowbird");  
    } else if ( Math.pow(x,2)+Math.pow(y,2)+Math.pow(z,2) <= 1){  
        alert("goose egg");  
    } else if ( Math.pow(x,2)+Math.pow(y,2)+Math.pow(z,2) > 1 ){  
        alert("Blue Angel");  
    } else {  
        alert("Maverick");  
    }  
}  
alert("I am done!");
```

1. Will the line alert("Maverick") ever execute? Why or why not?
2. When will the line alert("goose egg") execute?
3. For a single pass of the code snippet, what is the maximum and minimum number of alert statements that will be executed?

II. Write/Correct IF Statement ( 8 Points )

- a. At Langara, "roll call" grades are assigned around midterm to help students to assess their performance. A common rule-of-thumb is to consider a grade of C or better to be Satisfactory, while C- or worse is Unsatisfactory. Write a code segment which prompts the user for their average, and then displays a message on the console assessing their performance. If their average is 73 or higher, they should receive a Satisfactory rating, otherwise Unsatisfactory. ( 2 Points )

- b. Add an else statement to display "You're not John!" to the following code's if statement **( 2 Points )**

```
<!DOCTYPE html>
<html>
<body>

<p id="demo">Display the result here.</p>

<script>
var firstName = "Greg";

if (firstName === "John") {
    document.getElementById("demo").innerHTML = "Hello John!";
}
</script>

</body>
</html>
```

- c. Write an if/else statement with the following condition: **( 2 Points )**

If the variable age is greater than 18, output "Old enough", otherwise output "Too young"

```
<!DOCTYPE html>
<html>
<body>

<p id="demo">Display the result here.</p>

<script>
var age = 25;

// Add the if/else statement here
</script>

</body>
</html>
```

- d. The if/else statement below does not work. What is the fix? **( 2 Points )**

```
<!DOCTYPE html>
<html>
<body>

<p id="demo">Display the result here.</p>
```

```

<script>
var greeting;
var hour = new Date().getHours();

if (hour < 18) {
    greeting = "Good day";
} else {
    greeting = "Good evening";
}
document.getElementById("demo").innerHTML = greeting;
</script>
</body>
</html>

```

### III. Create a webapp using HTML/JavaScript ( 85 Points )

In this lab, you are going to write three short webapps, and they are:

- Guessing Game
- What Province Are You From
- My Next Birthday (Day of the Week) (**bonus 30 Points**)

Each lab will have its own set of instructions to follow.

#### Guessing Game Webapp

In this webapp, you will prompt the user for a number between 1 and 10, and compare the user's number to a random number you generate (using the Math.random function) to see if they match. If they match, you will alert the user (using the alert function) with the message "Great Job – Your guess is correct." If the numbers do not match, you will alert the user with the message "Sorry – Your guess is not a match!" If the user does not enter a valid number or a number in the range (1 through 10), your webapp will alert the user with the message "Sorry – invalid number!"

Here are the steps to complete this webapp:

- Download Lab2-Template.html from D2L  
Open this html file, and then save it as **FirstName-LastName-SID- LAB2-GuessingGame.HTML**
- Add the necessary JavaScript code in this html file to create this webapp. See Appendix B to review a possible program flow for this webapp.

Your code may include, but not limited to, the following – NOTE – DO NOT USE ANY LOOPS, as it is not necessary:

- alert function
- prompt function
- if-else
- isNaN
- Math.random – Note - this returns a number not a string
- document.write

- Number function – Note – input from prompt function is always considered a string
- c. Before your webapp completes, use `document.write` to display on the webapp's page the user's guess, and the computer's random number. Your webapp output will have a heading (CPSC 1045 – Lab 2 – Guessing Game), followed by two lines (Computer Number: ?? and Your Number: ??), and then on the next line the same message you alerted the user – whether there was match, no match or invalid number. Sample outputs for this webapp are shown below.

## **CPSC 1045 - Lab #2 - Guessing Game WebApp**

Computer Number: 2  
User Number: 2  
Great Job - Your guess is correct

## **CPSC 1045 - Lab #2 - Guessing Game WebApp**

Computer Number: 5  
User Number: 7  
Sorry - Your guess is not a match!

## **CPSC 1045 - Lab #2 - Guessing Game WebApp**

Computer Number: 5  
User Number: abc  
Sorry - invalid number!

### **What Province Are You From Webapp**

In this webapp, you will prompt the user for a 3-digit area code, show on the webapp's page which western Canadian province the area code is from with a message such as "You are from 'province name'!" If the area code is not a western Canadian province area code or the input is not a valid 3-digit string, show on the webapp's page the message "Sorry – your area code: " + user input + "is not from western Canada!" The following are the western Canadian provinces' area codes:

Alberta – 403, 587, 780  
British Columbia – 236, 250, 604, 778  
Manitoba – 204, 431  
Saskatchewan – 306, 639  
Yukon - 867

Here are the steps to complete this webapp:

- Download Lab2-Template.html from D2L  
Open this html file, and then save it as **FirstName-LastName-SID- LAB2-WhatProvince.HTML**
- Add the necessary JavaScript code in this html file to create this webapp.

Your code may include, but not limited to, the following – NOTE – DO NOT USE ANY LOOPS, as it is not necessary:

- prompt function
- if-else if or switch

- .length function
- document.write

Sample outputs for this webapp are shown below.

## CPSC 1045 - Lab #2 - What Province WebApp

Your Area Code: 403  
You are from Alberta!

## CPSC 1045 - Lab #2 - What Province WebApp

Your Area Code: 999  
Sorry - your area code: 999 is not from Western Canada!

### My Next Birthday Webapp (Bonus Only)

In this webapp, you will prompt the user for a number for a birthdate in form of mmddyyyy (post 1970 and less than the current date), and show on the webapp's page the message "Your next birthday: " + the date of next birthdate + " is on 'day of the week'!" If the birthdate provided is not a valid date, show on the webapp's page the message "Sorry – invalid date!"

Here are the steps to complete this webapp:

- Download Lab2-Template.html from D2L  
Open this html file, and then save it as **FirstName-LastName-SID- LAB2-MyBirthday.HTML**
- Add the necessary JavaScript code in this html file to create this webapp.

Your code may include, but not limited to, the following – NOTE – DO NOT USE ANY LOOPS, as it is not necessary:

- JavaScript Date Object and Methods (functions) -  
[http://www.w3schools.com/js/js\\_date\\_methods.asp](http://www.w3schools.com/js/js_date_methods.asp)  
[http://www.w3schools.com/jsref/jsref\\_obj\\_date.asp](http://www.w3schools.com/jsref/jsref_obj_date.asp)  
[http://www.w3schools.com/Js/js\\_date\\_formats.asp](http://www.w3schools.com/Js/js_date_formats.asp)
- prompt function
- if-else
- switch
- .toString
- .substr
- document.write

Sample outputs for this webapp are shown below.

## **CPSC 1045 - Lab #2 - My Next Birthday WebApp**

Your Birthdate: 3/5/3333/33  
Sorry - invalid date!

## **CPSC 1045 - Lab #2 - My Next Birthday WebApp**

Your Birthdate: 01/01/1980  
Your next birthday: 01/01/2017 is on Sunday!

## **CPSC 1045 - Lab #2 - My Next Birthday WebApp**

Your Birthdate: 1/1/1980  
Sorry - invalid date!

## **CPSC 1045 - Lab #2 - My Next Birthday WebApp**

Your Birthdate: 01/01/1966  
Sorry - invalid date!

### **Deliverable(s):**

Submit one zipped file called **FirstName-LastName-SID-LAB2.zip** to D2L under the folder called **Lab #2**. This zip file will contain the following files:

**FirstName-LastName-SID- LAB2.docx**

This document will contain your answers to Parts I, and II above.

**FirstName-LastName-SID- LAB2-GuessingGame.html**

**FirstName-LastName-SID- LAB2-WhatProvince.html**

**FirstName-LastName-SID- LAB2-MyBirthday.html (submit if you complete it)**

This document will contain the Webapp HTML files from Part III above.

Note – replace SID in filenames above with your 9 digit student ID number – starts with “10”.

### **Due Date:**

**See D2L. No late submissions accepted/graded.**

# Appendix A

---

## Control Statements

So far our programs have been list of instructions executed in sequential order. This applies to both the HTML files and the JavaScript you wrote in the last labs. Conditional statements are what we call control structures, as they control the statements to be executed. It's also the first thing that causes our program not to run strictly from beginning to end.

### if, else if, and else statement

The first condition is always an if, followed by a Boolean expression

```
if (condition1){  
    //Runs if condition 1 is true  
}  
else if (condition 2){  
    //Runs if condition 2 is true  
}  
else{  
    //Only runs if all the previous conditions are false  
}
```

A second conditional statement is the switch statement. The switch statement is useful if you want to compare a single value against multiple values.

```
switch(expression){  
    case value1: //Code block 1  
        break;  
    case value2: //Code block 2  
        break;  
    default: // Code block run by default  
}
```

## Inputs: Prompt Function

In this lab we will use the **prompt** statement to get input from the user. In modern web development the prompt statement is rarely used, except for two situations. One is for debugging and the second is for learning. At the moment it is perfect for us!

Prompt brings up a **modal dialog** box that does not go away until the user enters some input. If you understood the above sentence, good for you and you can answer the following question.

What is a **modal dialog** box?

To explain modal dialog boxes, we need to a simple code example and utilize our knowledge of expression evaluation.



```
var userInput = '';

userInput = prompt('Please enter a sentence');
```

How do we interpret the statement?

```
userInput = prompt('Please enter a sentence');
```

We break it down like any other statement. Here we have a function call, where the function is named **prompt**. (We will cover functions in more depth in future labs.) For now, a function is a separate piece of code that executes a particular task (prompting), even though we don't know exactly how it is doing it.

To actually call the function, we first must evaluate the arguments, which could be complicated expressions. Since it's just a string we don't have to do much. But we can imagine more complex evaluations can also go here. Next we evaluate the function call.

```
prompt('Please enter a sentence');
```

What does **prompt** do?

Prompt brings up a dialog that presents the user with the string, in this case "Please enter a sentence" and then waits for input. Your program does not continue running, until the user types in some input and press enter.

The prompt statement returns the value that the user typed in to your code so you can do further work with it. In our case, we store it in the variable `userInput`.

Note: **prompt** *always* returns to you an answer that has a value type of **String**.

Example:

```
var userResponse1 = '';

userResponse1 = prompt('Do you wish to continue yes/no');

switch(userResponse1)
{
    case 'yes' :
        console.log('You wish to continue');
        break;
    case 'no' :
        console.log('you do not wish to continue');
        break;
    default :
        console.log('I do not understand');
}
```

The above example asks if the user wishes to continue, and waits for their response. Since there are three possibilities for the response, we have 3 blocks of code. They can enter "yes", "no", or something else entirely.

Remember: prompt always returns a value type of String. However, in many programs we need the input as a number (or something else) so we'll need to convert it. See **Number** function below in the second line.

Example:

```
var userResponse1 = prompt('Enter a number:');
var enteredNumber = Number(userResponse1);

if(!isNaN(enteredNumber)){
  if(enteredNumber > 5){
    console.log('Your number is bigger than 5!');
  } else if (enteredNumber < 4){
    console.log('Your number is less than 4');
  } else{
    console.log('Your number is 4 or 5');
  }
}
else {
  console.log('You did not enter a number!!!'); // Data Validation!
}
```

When we convert to a number, it will sometimes fail. If the user did not enter a number, the **Number** function will return or evaluate to **NaN**. So we have to check if the user actually entered a number that JavaScript understands. If it is a number, then we use it to execute some logic, in this case we check if it's 4 or 5.

## Example: Simple Prompt program

promptExample.html:

```
<!DOCTYPE html>
<html>
<head>
  <title>Example using prompt</title>
  <meta charset="utf-8" >
</head>
<body>
  <h1> Prompt and if Example!</h1>
  <section id="outputSection"></section>
  <script src="promptExample.js"></script>
</body>
</html>
```

### **promptExample.js:**

```
//Get a string
var userInput = prompt("Please enter a number between 1 and 5");
var inputNum = Number(userInput); // Convert the string to a number
var outputSection = document.getElementById("outputSection");

if(isNaN(inputNum)){
    outputSection.innerHTML = "You did not enter a number!"
} else{
    if(inputNum >=1 && inputNum <=5){
        outputSection.innerHTML = "Your number was " + inputNum;
    } else {
        outputSection.innerHTML = "Number out of range";
    }
}
```

# Appendix B

---

## Guessing Game Program Flow

The following outlines a possible program flow for this webapp.

1. Initialize a variable called guessNumber to 0
2. Initialize a variable called guessInput to "" (blank string)
3. Initialize a variable called randomNumber to 0
4. Call Math.random to obtain a random number, and assign it to your variable randomNumber
5. Prompt the user to enter a number between 1 and 10 (inclusive) – include in the prompt a hint on what the user should enter, i.e. "Enter a number (valid range: 1 to 10)?"

Assign user's input to your variable guessInput

6. Write to HTML page using document.write heading
7. Write to HTML page using document.write Computer Number: + randomNumber
8. Write to HTML page using document.write User Number: + guessInput
9. Check if guessInput is a valid number using isNaN

If so,

alert the user with a message of invalid number  
write to the webapp's page the same message

If not (else),

convert guessInput to a number using the Number function and assign it to your variable guessNumber  
compare guessNumber to randomNumber

if so,

alert the user with a message of match  
write to the webapp's page the same message

if not,

alert the user with a message of no match  
write to the webapp's page the same message

10. End of webapp – no code necessary for this because once last JavaScript statement gets executed in this webapp, your webapp will end