

## JavaScript

You will learn more advanced html tags for creating graphical 'GUI' controls and the basics of JavaScript programming.

Pictures courtesy of James Tam

## Advanced HTML

- In the last section you learned how to use html to: format text, embed content and link to other pages.
- In this section you will learn how to use HTML to create graphical controls such as buttons and input fields.

## Creating GUI Controls

- **Format:**

```
<input type="<control type>" value="<Text description>"/>
```

- **Name of example:** 1emptyControl.htm

```
<input type="button" value="Press me"/>
```



- **Pressing the button does nothing! Why???**
- **You need to write the JavaScript instructions to indicate what happens next (details come later)**

## Types Of GUI Controls

GUI control	HTML	Appearance
Button	<code>&lt;input type="button" value="Press"/&gt;</code>	<input type="button" value="Press me"/>
Checkbox (select 1+)	<code>&lt;input type="checkbox" value="Checkbox one of many"/&gt;</code>	Checkbox opt 2 <input type="checkbox"/>
Password (input hidden)	<code>&lt;input type="password" value="def"/&gt;</code>	Password <input type="password" value="*****"/>
Radio (select exactly 1)	<code>&lt;input type="radio"/&gt;</code>	Radio1 <input checked="" type="radio"/>
Text (single line input)	<code>&lt;input type="Text" value="Type a line of text here"/&gt;</code>	<input type="text" value="Type a line of text here"/>

There's numerous places online where you can find information about how to use these controls

- Example:
- [https://msdn.microsoft.com/en-us/library/ie/ms535262\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/ie/ms535262(v=vs.85).aspx)

## Graphical Controls: More Complete Example

- **Name of example:** 2manyEmptyControls.htm

<pre>Button &lt;input type="button" value="Press me"/&gt;&lt;br&gt; &lt;br&gt; &lt;b&gt;Check all that apply&lt;/b&gt;&lt;br&gt; Checkbox opt 1&lt;input type="checkbox"/&gt;&lt;br&gt; Checkbox opt 2&lt;input type="checkbox"/&gt;&lt;br&gt; &lt;br&gt; Password &lt;input type="password" value="default"/&gt;&lt;br&gt; &lt;br&gt; &lt;b&gt;Select one of the following&lt;/b&gt;&lt;br&gt; Radio1 &lt;input type="radio"/&gt;&lt;br&gt; Radio2 &lt;input type="radio" checked="checked"/&gt;&lt;br&gt; &lt;br&gt; Text &lt;input type="Text" value="Type a line of text here"/&gt;</pre>	
---	--

## Adding JavaScript To A Webpage

- This is where you can augment a web page to make it interactive.
  - The JavaScript instructions must be enclosed with a 'script' tag at the start of your webpage.

```
<script>
```

```
  JavaScript program
```

```
</script>
```

- Similar to defining a 'sub' in VBA you need to define a 'function' in JavaScript.
- A function is a series of instructions that run when an event occurs e.g., the user clicks on a button

## Defining A JavaScript Function

- **Format:**

```
function <function name>()  
{  
    Instructions in body (indent 4 spaces);  
}
```

- **Example:**

```
function saySmartResponse()  
{  
    alert("Don't press that button");  
}
```

## Getting The GUI To React To User Input

- **Name of example:** 3controlsButton.htm

```
<script>  
function saySmartResponse() ←  
{  
    alert("Don't press that button");  
}  
</script>
```

```
<input type="button" value="Press me"  
onclick="saySmartResponse()"/>
```

**onclick**

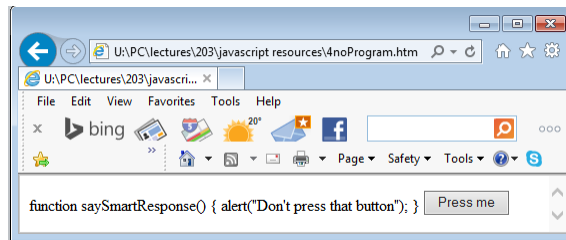
- When user clicks on button

## Common Mistake #1

- Forgetting the opening and closing script tags
- **Name of example:** 4noProgram.htm

```
function saySmartResponse()  
{  
    alert("Don't press that button");  
}  
<input type="button" value="Press me"  
    onclick="saySmartResponse()"/>
```

### Result



## Variables

- **Format:**  
`var <variable name> = <value>;`
- **Example:**  
`var num = 2;`

## Variables: Contrast Between Languages

- Contrast with VBA:

```
dim num as Long  
num = 2
```

- Unlike VBA the type of information to be stored in the variable doesn't have to be specified:

```
var num1 = 12;           Data specifies integer variable  
var num2 = 1.5;        Data specifies real number variable  
var text = "boo2!";    Data specifies String variable (text)
```

## A Program With Variables

- **Name of example:** 5variables.htm

```
<script>  
function buttonPress()  
{  
    var num = 2;  
    alert("Value = " + num);  
}  
</script>  
<input type="button" value="Press"  
onclick="buttonPress()"/><br>
```

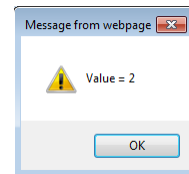
## Common Mistake #2

- A variable is just that – it can change as a program runs.

```
<script>
function buttonPress()
{
    var num = 2;
    alert("Value = " + num);
}
</script>
```

Vs.

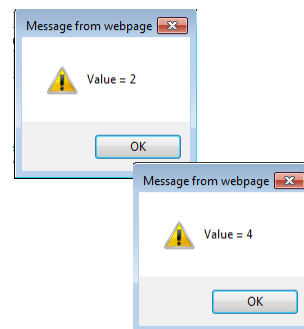
```
<script>
function buttonPress()
{
    alert("Value = 2");
}
</script>
```



## Changing The Value Of Variable

```
<script>
function buttonPress()
{
    var num = 2;
    alert("Value = " + num);

    num = num * 2;
    alert("Value = " + num);
}
</script>
```



## Naming Variables

- JavaScript language requirement (syntax ~ grammar)
  - Name can be alphanumeric but cannot start with a number  
num1 (OK) vs. 2num (Not OK)
  - Spaces are not allowed  
first name (Not OK) vs. firstName OR first\_name (OK)
  - Style requirement (~quality writing)
    - Variable names should be self-descriptive and lower case  
x, foo, AGE (Not OK) vs. firstName, age, height (OK)

## Operators

- The common operators that you will see in JavaScript are the same as other languages and applications (such as MS-Excel)

Operation	JavaScript operator
Division	/
Modulo (remainder)	%
Multiplication	*
Subtraction	-
Addition <sub>1</sub>	+

<sup>1</sup> Note:

Performs mathematical **addition** only if all the inputs are numeric

e.g.,  $12 + 12 = 24$

If one or more inputs are strings then a **concatenation** is performed

e.g., `alert("Value = " + num);`



## Documentation

- Single line

```
// Everything to the end of the line is not counted as a
// JavaScript instruction (used to explain details of the
// program to other programmers).
```

```
// Works just like prefacing text with the ' (VBA)
```

- Multiple line

- Start of documentation uses `/*`
- End of documentation uses `*/`
- Everything in between is not counted as a JavaScript instruction

```
/*
    Author: James Tam
    Tutorial: 888
*/
```

## Where To Document

- Immediately after the `<script>` “open tag”

```
<script>
// Single line documentation

/*
    Multi-line documentation
*/

function <function name>()
{
    // If needed additional documentation can be added here.
}
</script>
```

## Documentation: Common Mistake #3

- Take care not to nest multi-line comments
- This is an error!

```
/*  
    /* Author: James Tam */  
    Tutorial: 888  
*/
```

**Invalid statement (not JavaScript)**

## Minimum Documentation Needed For CPSC 203

```
// Name  
// Student ID  
// Tutorial #  
  
/*  
    A list of each program feature and an indication if the  
    feature was completed (yes or no).  
*/  
  
// (If applicable): explicit instructions on the name and  
// location of any data files required. For a  
// detailed example see: 17forLoop2.htm
```

## Multiple Controls

- If you need to identify one control from another then you can use the 'id' attribute when creating a control using html
- **Format:**  
`id = "<Descriptive string>"`
- **Example:**  
`<input type="text" id="text1"/>`
- When choosing a descriptive string to identify the control it's a good idea to apply the same conventions used as when you are naming variables e.g., no spaces, good and self-descriptive names

## Example Of Using Multiple Controls

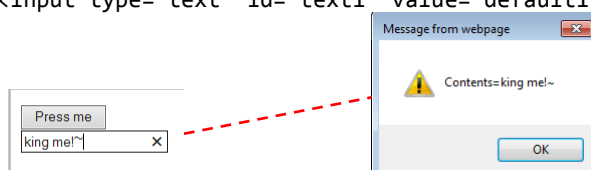
- **Name of example:** 6multipleControls.htm

```

<script>
function buttonPress()
{
    var text1 = document.getElementById("text1").value;
    alert("Contents=" + text1)
}
</script>
<input type="button" value="Press me"
onclick="buttonPress()"/><br>

<input type="text" id="text1" value="default1"/><br>

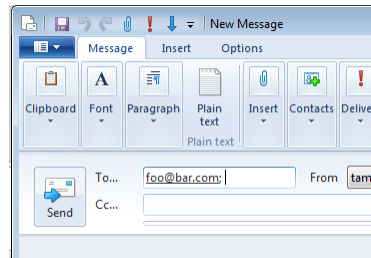
```



## Sending Email: Basic Version

- **Name of example:** 7sendingEmail.htm

```
<script>
function buttonPress()
{
    window.open("mailto:foo@bar.com");
}
</script>
<input type="button" value="Send mail" onclick="buttonPress()"/>
```

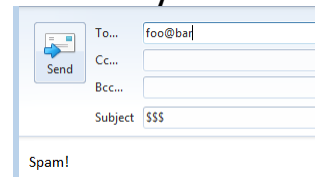


## Sending Mail: Fixed Subject And Body

- **Name of example:** 8sendingEmail.htm

```
<script>
function buttonPress()
{
    window.open("mailto:foo@bar" + "?subject=$$$
    &body=Spam!");
}
</script>

<input type="button" value="Send mail"
onclick="buttonPress()"/>
```



## Sending Mail: Fixed Subject And Body Is Variable

- **Name of example:** 9sendingEmail.htm

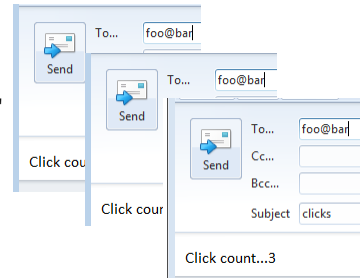
```

<script>
var count = 0
function buttonPress()
{
    count = count + 1;
    window.open("mailto:foo@bar" +
        "?subject=clicks&body=Click count..." + count);
}
</script>

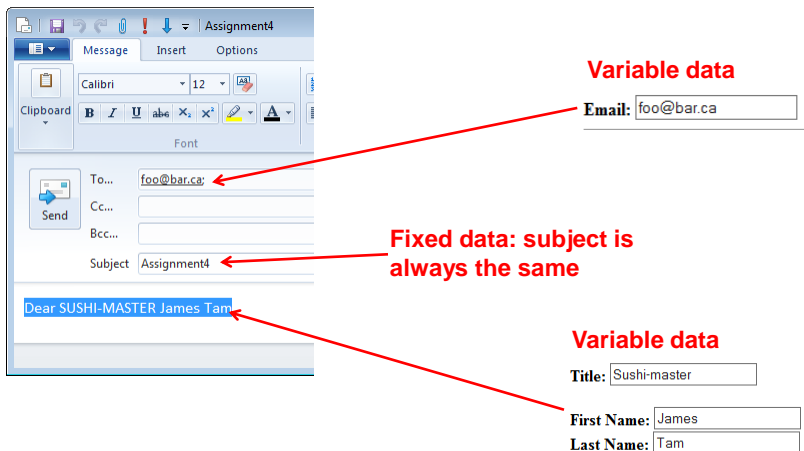
<input type="button" value="Send mail"
onclick="buttonPress()"/>

```

Creating 'count' outside of the function means that the previous count is retained



## Sending Mail: Assignment



**Example:** 6multipleControls.htm

- One example of accessing the data value of a control

## Accessing Data Of GUI Controls

- Name of example: 10getData.htm

```

<script>
function buttonPress()
{
    var login = document.getElementById("textControl1").value;
    var password =
        document.getElementById("passWordControl1").value;
    alert("Login name "+ login);
    alert("Secret password "+ password);
}
</script>

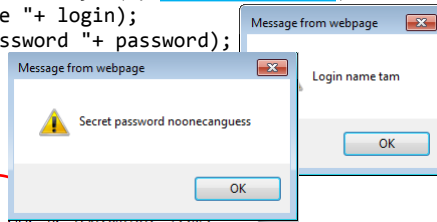
```

Show data

Login: tam

Password: ●●●●●●●●

Irrelevant: not used



```

<input type="button" value="Show data" onclick="buttonPress()"/><br>
Login: <input type="text" id="textControl1" /><br>
Password: <input type="password" id="passwordControl1"/><br>
<br>
Irrelevant: <input type="text" value="not used" id="textControl2" /><br>

```

## Acknowledgments

- The functional requirements of the email program were based on the assignment description produced by James Tam
- The different versions of the email program were based on a solution produced by Omar Addam

## Getting User Input

- Getting Input:

**Format:**

```
<variable> = prompt("<Prompting message>", "<default input>")
```

**Example** (assume we create a variable called 'age'):

```
age = prompt("Enter your age (zero or greater):", "e.g., 37");
```



## Getting JavaScript To **Run Automatically**

- Name of example:** 11getInputAutoRun.htm

```
<script>
function main()
{
    var age = -1;
    age = prompt("Enter your age (zero or greater):", "e.g., 37");
};
window.onload=main;
</script>
```

## Branching: IF

- VBA

- if then
- if then, else
- if then, elseif, else

- JavaScript

- if
- if, else
- if, else if, else

- When to use

- React when true
- React true & false
- At most only one if-case can be true (“select one of the following”)

## If: General Format

```
if (<Boolean expression>
{
    <body>; // Indent the body by an additional 4 spaces
}
```



## If: An Example

- **Name of example:** 12ifExample.htm

```
<script>
function main()
{
    var age = -1;
    age = prompt("Enter your age (zero or greater):","e.g., 37");
    if (age < 0)
    {
        alert("Age cannot be negative");
    }
};
window.onload=main;
</script>
```

## If-Else: General Format

```
if (<Boolean expression>)
{
    <body if>;
}
else
{
    <body else>;
}
```

## If-Else: An Example

- **Name of example:** 13ifElseExample.htm

```
<script>
function main()
{
    var age = -1;
    age = prompt("Enter your age (zero or greater):", "e.g., 37");
    if (age < 0)
    {
        alert("Age cannot be negative");
    }
    else
    {
        alert("Age verified as OK");
    }
};
window.onload=main;
</script>
```

## If, Else-If: General Format

```
if (<Boolean expression1>)
{
    <body if1>;
}
else if (<Boolean expression2>)
{
    <body if1>;
}
...
else // Not mandatory/optional
{
    <body else>;
}
```

## If, Else-If: An Example

- **Name of example:** 14ifElseIfExample.htm

```

window.onload=main;

function main()
{
    var age = -1;
    MAX_AGE = 122; // const MAX_AGE = 122 not supported by IE
    age = prompt("Enter your age (zero or greater):","e.g., 37");
    if (age < 0)
    {
        alert("Age cannot be negative");
    }
    else if (age < 6)
    {
        alert("Little kid");
    }
}

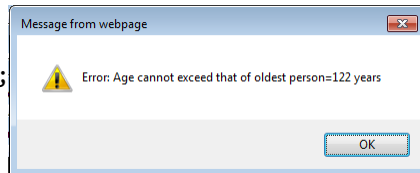
```

## If, Else-If: An Example (2)

```

    else if (age < 18)
    {
        alert("Grade schooler");
    }
    else if (age < 65)
    {
        alert("Non-senior");
    }
    else if (age <= MAX_AGE)
    {
        alert("Senior citizen");
    }
    else
    {
        alert("Error: Age cannot exceed that of oldest person="
            + MAX_AGE + " years");
    }
}

```



## Allowable **Operators** For Boolean Expressions

if (value **operator** value) then

JavaScript operator	Mathematical equivalent	Meaning	Example
<b>&lt;</b>	<	Less than	5 <b>&lt;</b> 3
<b>&gt;</b>	>	Greater than	5 <b>&gt;</b> 3
<b>==</b>	=	Equal to	5 <b>==</b> 3
<b>&lt;=</b>	≤	Less than or equal to	5 <b>&lt;=</b> 5
<b>&gt;=</b>	≥	Greater than or equal to	5 <b>&gt;=</b> 4
<b>!=</b>	≠	Not equal to	x <b>!=</b> 5

## Logical Operators

Logical operation	JavaScript	Example
AND	<b>&amp;&amp;</b>	if (x > 0 <b>&amp;&amp;</b> y > 0)
OR	<b>  </b>	if (x > 0 <b>  </b> y > 0)
NOT	<b>!</b>	if <b>!</b> (x > 0)

## Loops

- For
- While

### For Loops: General Format

```
for (<Set counter to initial>;  
    <Boolean Expression>;  
    <Increment or decrement counter>)  
{  
    <Body>; // Indent 4 spaces  
}
```

Note: the three statements in the form loop do not have to reside on three separate lines.

## For Loops: An Example

**Name of example:** 15forLoop1Example.htm

```
<script>
function main()
{
    var i = -1;
    var last = -1;
    last = prompt("Enter last value in number series: ", "");
    for (i = 0; i <= last; i = i + 1)
    {
        alert("i=" + i);
    }
};
window.onload=main;
</script>
```

## While Loops: General Format

```
while (Boolean Expression)
{
    <Body>; // Indent 4 spaces
}
```

## While Loops: An Example

- **Name of example:** 16whileLoop1Example.htm

```
<script>
function main()
{
    var i = -1;
    var last = -1;
    last = prompt("Enter last value in number series: ", "");
    i = 0;
    while (i <= last)
    {
        alert("i=" + i);
        i = i + 1;
    }
};
window.onload=main;
</script>
```

## Loops: A More Complex Example

- Learning concepts:
  - How JavaScript can change a document contents via document.write()
  - Using a loop to produce 'special effects'
    - Change the image to be added

## Loops: A More Complex Example (HTML Portion)

- **Name of example:** 17forLoop2Example.htm

```
</script>

```

This image  
cannot be  
clicked



```
<input type="image"
  src="pics/clickable.png"
  onclick="main()"
/><br>
```

Clickable  
image



## Loops: A More Complex Example (JavaScript Portion)

```
<script>
function main()
{
  var i = -1;
  funImage = ("<img src='pics/funGuy.jpg'>");
  fungiImage = ("<img src='pics/fungi.jpg'>");
  for (i = 1; i <= 4; i = i + 1)
  {
    prompt("Hit ok to continue", i);
    if (i % 2 == 0) // % Modulo operator
    {
      document.write(funImage); // even
    }
    else
    {
      document.write(fungiImage); // odd
    }
  }
}
```



4 spaces  
4 more spaces (4+4 = 8)



## Arrays

- Unlike 'simple' variables such as integers and real numbers an integer can hold multiple values e.g., a real number variable can store one student's grades while an array can store the grades for an entire class.
- **Format** (creating an array):  

```
var <array name> = new Array(<Number of elements>);
```
- **Example** (creating an array):  

```
var grades = new Array(4);
```

## Accessing Array Elements

- Use an 'index'
  - The first index begin at zero
  - The last index ends at (size of the array minus 1).
  - Example: an array of size 10 will have index values from 0 to (10-1 or 9)
- **Format** (assigning value to an element):  

```
<array name>[<index>] = value;
```
- **Example** (assigning value to elements):  

```
grades[0] = 4;  
grades[1] = 3;  
grades[2] = 2;  
grades[3] = 3;
```

## A More Realistic Example Program (HTML Portion)

- **Name of example:** 18array.htm

```

```

```
<input type="image" src="pics/clickable.png" onclick="main()"
/><br>
```

## A More Realistic Example Program (JavaScript Portion)

```
function main()
{
  var SIZE = 4;
  var pics = new Array(SIZE);
  var i = 1;
  for (i = 0; i < SIZE; i = i + 1)
  {
    pics[i] = "<img src='pics/' + i + '.jpg'"
  }

  for (i = 0; i < SIZE; i = i + 1)
  {
    prompt("Hit ok to continue", "");
    document.write(pics[i]);
  }
}
```



**i = 0**



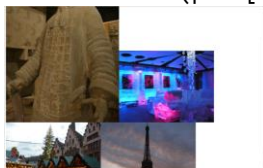
**i = 1**



**i = 2**



**i = 3**



## Backup Your Work Frequently!

- This is always a good idea but imperative when writing JavaScript programs.
- JavaScript will NOT give you helpful error messages!
  - Usually you get no error message at all.
  - Determining where the problem lies can be a great challenge!

## Example Program With An Error (Or Errors)

**Program name:** 19error.htm

```
function main()
{
    var const SIZE = 4;
    var pics = new Array(SIZE);
    var i = 1;
    for (i = 0; i < SIZE; i = i + 1)
    {
        pics[i] = "<img src='pics/" + i + ".jpg'>"
    }

    for (i = 0; i < SIZE; i = i + 1)
    {
        prompt("Hit ok to continue","");
        document.write(pics[i]);
    }
}
```

## Backups: Benefit

- If you accidentally introduce an error to your program and you cannot find it
- You can always:
  - Compare the backup version vs. the current version in order to more easily find the difference
  - Simply revert to the backup (if you cannot find the error)
- When you make backups always ensure that your backup is working prior to making a backup.
- You have been forewarned!
  - Failure to make a proper set of backups won't allow you to get an extension.

## If All Else Fails

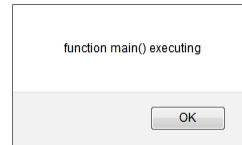
- What to do if you still didn't take this advice and you are desperately trying to find the error.
- First: there are no guarantees with this technique.
- Second: make liberal use of comments
  - "Comment out" the entire body of the function save for an alert (to tell you that the function is running properly).

## “Commenting Out Code” Previous Example

**Program name:** 20findingError.htm

```
function main()
{
    alert("function main() executing");
    /*
    var const SIZE = 4;
    var pics = new Array(SIZE);
    var i = 1;
    for (i = 0; i < SIZE; i = i + 1)
    {
        pics[i] = "<img src='pics/' + i + ".jpg'>"
    }

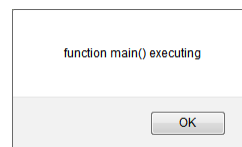
    for (i = 0; i < SIZE; i = i + 1)
    {
        prompt("Hit ok to continue","");
        document.write(pics[i]);
    }
    */
}
```



## “Commenting Out Code”: Gradually Move The Comments To Expose More Code

```
function main()
{
    alert("function main() executing");
    /*
    ↓ var const SIZE = 4;
    var pics = new Array(SIZE);
    var i = 1;
    for (i = 0; i < SIZE; i = i + 1)
    {
        pics[i] = "<img src='pics/' + i + ".jpg'>"
    }

    for (i = 0; i < SIZE; i = i + 1)
    {
        prompt("Hit ok to continue","");
        document.write(pics[i]);
    }
    */
}
```



**As you  
'uncomment' code  
take care that you  
don't introduce  
new errors e.g.,  
mismatch braces**

## Finding Errors

- This technique of “commenting out” JavaScript code will not work if the error is in your html tags.
- (Recall: documentation is used only in conjunction with programming instructions and html does not include programming instructions).
- Example: documentation will not help, what you were taught was commenting out JavaScript code (so BACKUP YOUR WORK)

```

<script>
function main()
{
    alert("function main() executing");
}
</script>

<input type="image" src="pics/clickable.png"
onclick="main" /><br>

```

## Strings

- A series of characters: alpha, numeric and other values that can be entered via the keyboard.
- A string has a length
- Examples:
  - var string1 = "a12!"; // 4 characters
  - var string2 = ""; // 0 characters
- The length method will return the length
- **Format:**

```
<string>.length;
```
- **Example:**

```
userInput.length == 0 // Assume userInput is a String
```

## Example Using The Length Method

- **Program name:** 21stringLength.htm

```
function main()
{
  var userInput = document.getElementById("input1").value;
  if (userInput.length == 0)
  {
    alert("Input field empty!");
  }
  else
  {
    alert(userInput);
  }
}

</script>
<input type="button" value="Press me" onclick="main()"/><br>
Enter some text and press the button <input type="input"
id="input1"/>
```

## Changing Capitalization

- This can be done with the methods:

- toUpperCase();
- toLowerCase();

- **Format:**

```
<string variable> = <string>.toUpperCase();
<string variable> = <string>.toLowerCase();
```

- **Example:**

```
userInput = userInput.toUpperCase(); //userInput is a string
```

## Example Of Changing Case

- **Program name:** 22changingCase.htm

```
function main()
{
  var userInput = document.getElementById("input1").value;
  if (userInput.length == 0)
  {
    alert("Input field empty!");
  }
  else
  {
    userInput = userInput.toUpperCase();
    alert(userInput); // All caps
    userInput = userInput.toLowerCase();
    alert(userInput); // All lower now
  }
}
</script>
<input type="button" value="Press me" onclick="main()"/><br>
Enter some text and press the button <input type="input"
id="input1"/>
```

## Strings Are Indexed

- Similar to arrays each character in a string is indexed from: 0 to (length – 1).
- Example:
  - var aString = "ab1";
  - Elements:** a b 1
  - Index:** 0 1 2
- The `indexOf()` method can be used to find the index of the first occurrence of a character.
- If the character is not found then the method returns -1.
- **Format:**

```
<variable> = <string>.indexOf(<character>);
```
- **Example:**

```
index = aString.indexOf("!");
```



## Example: Finding Location

- **Program name:** 23findingIndex.htm

```
<input type="button" value="Press me" onclick="main()"/><br>
Character to search for <input type="input" id="data"/><br>
Search string <input type="input" id="search"/>
```

```
function main()
{
    var searchCharacter =
        document.getElementById("data").value;
    var stringSearched =
        document.getElementById("search").value;
    if ((searchCharacter.length == 0) ||
        (stringSearched.length == 0))
    {
        alert("Unable to search. Either search string or
            character empty");
    }
}
```

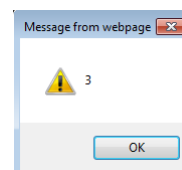
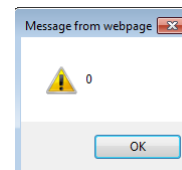
Press me  
 Character to search for x  
 Search string xyzlabx

## Example: Finding Location (2)

```
else
{
    // Index of character in input field
    var index = stringSearched.indexOf(searchCharacter);
    alert(index);

    // Index of exclamation mark '!'
    var index = stringSearched.indexOf("!");
    alert(index);
}
}
```

Press me  
 Character to search for x  
 Search string xyzlabx



## Additional Online Resources

- <http://www.w3schools.com/js/>
- [https://msdn.microsoft.com/en-us/library/ie/ms535262\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/ie/ms535262(v=vs.85).aspx)
- <http://trainingtools.com/online/javascript/index.htm>

## After This Section You Should Now Know

- How to use html to create a graphical controls such as buttons
- How to include JavaScript instructions via the `<script>` `</script>` tag
- How to define a 'sub-part' of a program using a function definition
- Getting a program to react to events (e.g., `onclick`,) by defining a function
- The display of output via `alert()`
- Defining variables with 'var'
- Variable naming conventions
- Common operators: mathematical (+, -, \*, /) and concatenation (+)

## After This Section You Should Now Know (2)

- Single and multi-line documentation
- How to use documentation and “commenting out” code in order to find and fix errors in a JavaScript program
- What should be included in the documentation for the programs that you write for this class
- In HTML: How to use the ‘id’ property to uniquely identify controls
- JavaScript: How to use the `document.getElementById("<string>")` method to access a graphical control
- Different variations when sending email using JavaScript instructions
- 

## After This Section You Should Now Know (3)

- How to get a JavaScript program to automatically run when the web page is loaded into a web browser via `window.onload()=<function name>`
- Getting input with a `prompt()`
- Branching/decision making mechanisms:
  - if
  - if-else
  - if, else if, else
- Defining variables with a stylistic convention (capitalization) and with a language defined approach ('const')
- Valid operators for Boolean expressions
- Valid logical operators

## After This Section You Should Now Know (4)

- Getting JavaScript instructions to repeat using looping mechanisms:
  - For
  - While
- Changing the contents of a webpage via `document.write()`
- How to create an array and how to access the elements
- Common string operations
  - Determining length
  - Converting case
  - Finding the location of the first instance of a character