Introduction to Javascript

Dr. Nadine ZBib

What it is and what it does?

What it is?

- It is **NOT** Java
- It is **NOT** Server-side programming
  - Users can see code
- It is a client-side programming tool
- It is embedded within an HTML page
- JavaScript **is** case-sensitive

What it does?

- Allows interactive computing **at the client level**
- Supported by IE, Netscape, firefox etc.
- Dynamically changes HTML
- Reacts to events
- Read and write HTML elements
- Validates data

Where to put Javascript

- You can have more than one <script> element
- Javascript code can go almost anywhere
- There is a common pattern: placing the code in head, or after all HTML has rendered
- javascript is case sensitive

```
<!DOCTYPE html>
<html>

<head>

    . . . load JavaScript libraries here . . .

</head>

<body>

    . . . your JavaScript code typically goes at the end of body . . .

</body>

</html>
```

How Javascript looks like

JAVASCRIPT IN THE SAME FILE

```
<script>
function surprise() {
    alert("Hello!");
}
</script>
```

JAVASCRIPT IN ANOTHER FILE

```
<script src="mycode.js"></script>
```

In *mycode.js*:

```
function surprise() {
    alert("Hello!");
}
```

Javascript Example

```
<html>
<body>
<script>
  document.write("<h1>Hello World! This is
  Me</h1>");
</script>
</body>
</html>
```

Hello World! This is Me

Begins with *<script>* **and ends with** *</script>*

Simple Interactions

- There are 3 Javascript popups:
  - alert() *Show a message*
  - confirm() *Make a decision*
  - prompt() *Ask for input from user*

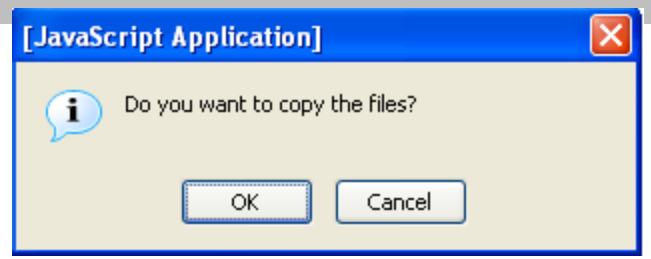
alert() Example (*alert()*)

```
<html>
<body>
<script type="text/javascript">
  alert("Hello World");
</script>
</body>
</html>
```

Click here to view a sample alert code running (and check code)

Exercise: confirm.html (*confirm()*)

```
<html>
<body>
<script type="text/javascript">
  confirm("Do you want to copy the files?");
</script>
</body>
</html>
```



Note the difference between ALERT and CONFIRM

Click here to view a sample confirm code running (and check code)

How to put comments?

```
<html>
<body>
<script>
  // This is single line comments
  /* this is multi-line comments especially if
  you like to babble along */
  document.write("Hello Me");
</script>
</body>
</html>
```

Learn to comment your scripts

Variables in Javascript

- A variable is like a box
- You can make a variable and put something in it e.g

```
var totalCost = 7000;
```

- Later, you can take it out of the box and use it
- You can change what is stored in the box any time

prompt()

- For getting input from the user, you can use prompt(), e.g:

```
var user_name; // Create a variable
user_name=prompt("What is your name?");
```

- You don't have to create a variable before you use it
- However, it is good habit to get into

Exercise: prompt.html (*prompt()*)

```
<html>
   <head>
      <title>Example of prompt()</title>
      <script>
      var user_name;
      user_name=prompt("What is your name?");
      document.write("Welcome to my page "
       + user_name + "!" );
       </script>
   </head>
</html>
```

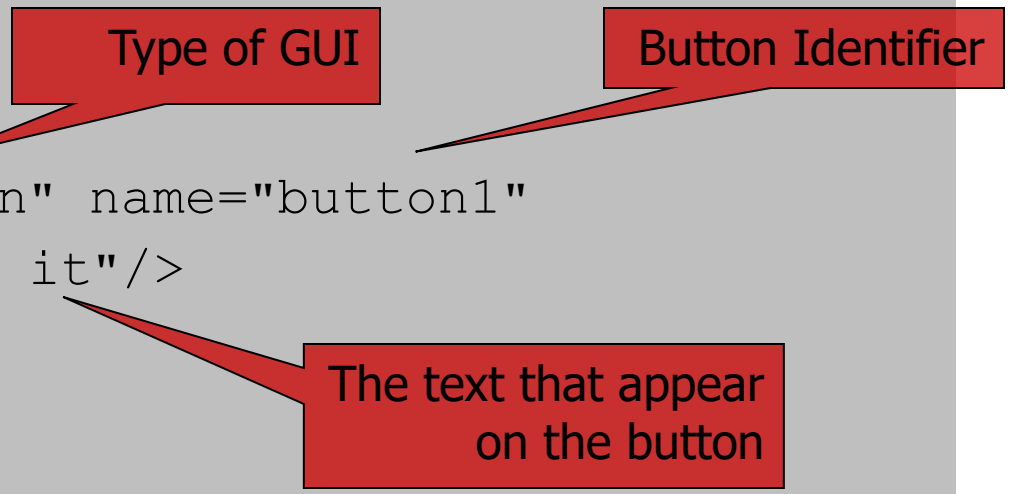
Click here to view a sample prompt code running (and check code)

Creating an interactive page

- Use of event programming.
- An event is anything that happens to an object or element of an object.
  - Click on a button: click event
  - Select some text: select event
  - Mouse hovering over: mouseover event
- When an event happen, we need to have an event handler.

Let us create a simple button

```
<html>
<body>
<h1> Hey you are in luck. Do you want your sTMA
  grade? </h1>

<form>
  <input type="button" name="button1"
  value="Yes! I want it"/>
</form>

</body>
</html>
```



Add event programming - body.

```
<body>
<h1> Hey you are in luck. Do you want your sTMA
  grade? </h1>

<form>
  <input type="button" name="button1"
  value="Yes! I want it "
  onclick="displayMyMessage()"/>
</form>

</BODY>
</HTML>
```

Event Handler: onclick="displayMyMessage() "

Add event programming - head.

```
<HTML>
<HEAD>
<TITLE>Interactive Web Page</TITLE>
<script type="text/javascript">
      function displayMyMessage(){
         alert("Sorry! Can't help you today!");
      }
</script>
</HEAD>
```

displayMyMessage: This is a javaScript function declared inside the “head” section

Declaring functions in head

- Most event programming handlers activate some form of javascript functions that is declared in the “head”
- We’ll start to do that immediately.
- Good Practice.
- You declare the function, and you write a series of instructions of “what to do”
- In this case, we are just “displaying” a message with a “alert” box.

Exercise: javascript.html

- Make a page that has two buttons.
- It also has two javascript functions declared.
- If you press on one button, one of the functions will be called (activated). If you press the other button, the other function will be called instead.

```
<head>
<title>Interactive Web Page</title>
<script type="text/javascript">
function displayMyMessage1(){


}


function displayMyMessage2(){


}
</script>
</HEAD>
```

Data Types

- **Number**
- String
- Boolean (true/false)

- JavaScript has only one type of number
- Can be written with or without a decimal place

```
var number1 = 34.289;
var number2 = 100;
```

- Can use scientific notation

```
var big_number = 123e5;    //12300000
var small_number = 123e-5; //0.00123
```

Data Types

- Number
- **String**
- Boolean (true/false)

- A string simply means text
- You can use single or double quotes

```
var name = "David";
var title = 'Professor';
```

- You can use quotes inside a string, as long as they don’t match the quotes surrounding the string

```
var message = "It's alright";
```

Data Types

- Number
- String
- **Boolean (true/false)**

- A Boolean value can only be true or false

```
var condition1 = true;
var condition2 = false;
```

- Do not confuse Boolean values with String values

```
var myBool = true;      //Boolean type
var myString = "true"; //String type
```

Data types : more comments

- There are other data types (file/date/…)
- A variable type can change
  - If you do this

```
var storage = "David";
```

  - …then this

```
storage = 98;
```

  - the type of the variable is immediately changed
- There is a function *typeof* to find out what data type is stored in a variable

```
<!doctype html>
<html>
<head>
  <title>Variable Type Example</title>
</head>
<body>
  <script>
  alert( '"John" is type: ' + typeof "John" + "\n\n"
        + "3.14 is type: " + typeof 3.14 + "\n\n"
        + "false is type: " + typeof false ) ;
  </script>
</body>
</html>
```

Processing HTML Forms

Event Handlers

```
<HEAD>
<SCRIPT LANGUAGE=JavaScript><!--

function name(){
   statement;
   .
   .
   .
   statement;
}

//--></SCRIPT>

</HEAD>
<BODY>

<FORM>
 <INPUT TYPE=button VALUE="Click Here" onclick="name()">
</FORM>

</BODY>
```

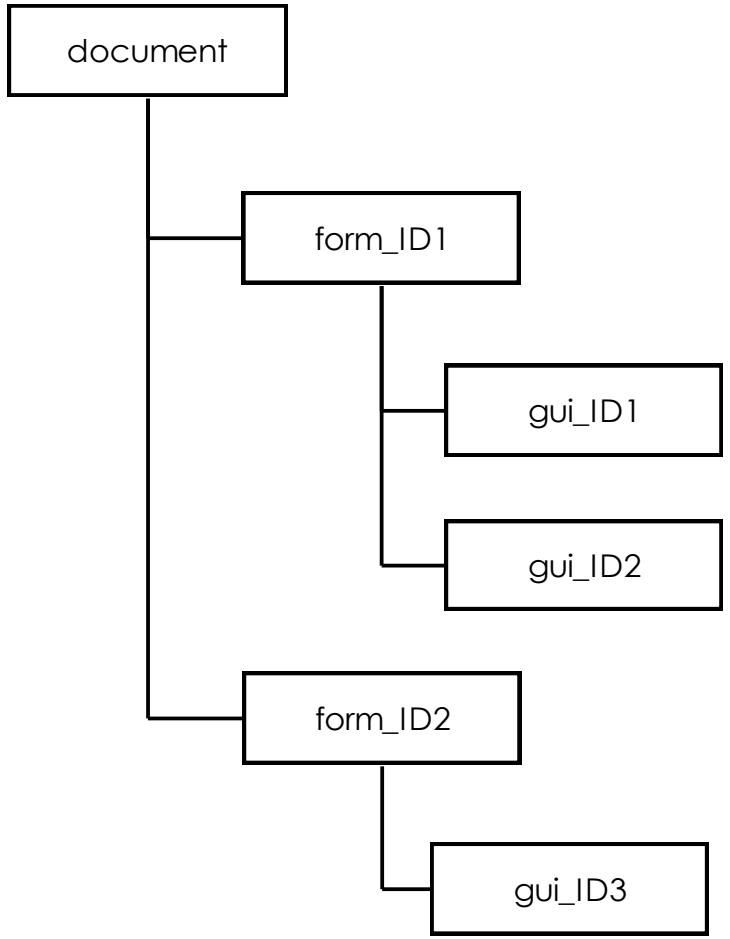
1) event handler detects button click
2) event handler calls function by name
3) function causes something to happen

Begins with *<script type="text/javascript">* **and ends with** *</script>*

Document Object

- In order for us to do event programming we need to understand document object in java
- It allows us to display text in the text area
- It allows us to change “values” on buttons
- In general it gives you more control of the graphic user interface (gui) objects in the forms

Generic Representation



Some properties of *document*

- For example if you want to control the color of a background, you can simply say document.bgcolor="#FF0000"
- if you want to display a message you perform document.write("Hello World"); or document.write("<h1> Hello World </h1>");
- bgcolor and write are some of the properties that you can use.

Example: Button and Response

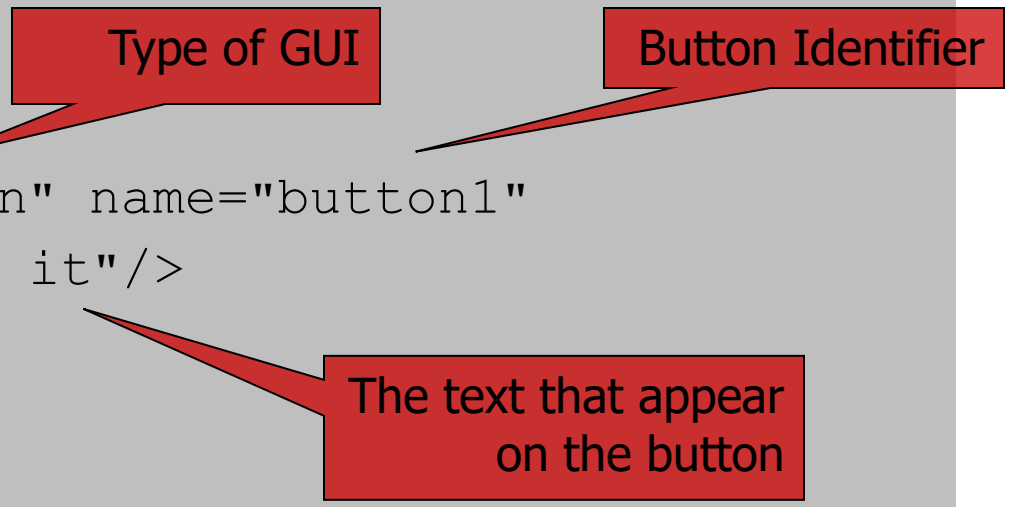
- Make a button and a message box.
- Need to have a form to “hold” this GUI objects
- You write form instructions in the body.
- You write the javascript function in the head.

```
<FORM name=simpleForm>
<input type="button" name="button1"
   value="RED" onclick="displayText()">
<BR>
<TEXTAREA name="response" rows=3
   cols=27> </textarea>
<INPUT type="reset" value="Reset Form">
```

```
function displayText() {
document.simpleForm.response.value="You click the red button!";
}
```

Recall from last lecture

```
<html>
<body>
<h1> Hey you are in luck. Do you want your sTMA
  grade? </h1>

<form>
  <input type="button" name="button1"
  value="Yes! I want it"/>
</form>

</body>
</html>
```



Exercise: javascript3.html

- Write a html page that has two buttons, a text area and a reset button.
- When you click button A, you display in the text area that you click on button A.
- When you click button B, you display in the text area that you click on button B.
- The reset button clears the display area.

Variable Definitions

- Like most programming you need to be able to use variables to help you process data i.e. to store it temporary, for calculation etc.
- Variables in javascript is very simple to define.

  var age;

  var counter;
- You can put these variables anywhere inside the “javascript” portion of the page.

Assignment and arithmetic operations

Working with numbers:

```
var age;
age = 7;
age = 7+1;

// shortcut will work too //
var age=7;
```

Working with strings:

```
var name1;
var name2;
name1 = “Scooby”;
name2 = “Doo”;
var dog;
dog=name1+name2;
dog=name1+" "+name2;
var firstName, lastName;
```

Variable declaration is not required, but it is a good practice

Local vs Global Variables

```
<script type="text/javascript">
<!—
var globalCount=0;
function example1() {
   var count1=0;
   count1=count1+1; // local variables used for temp calculation
   globalCount=globalCount+1;
}

//-->
</script>
```

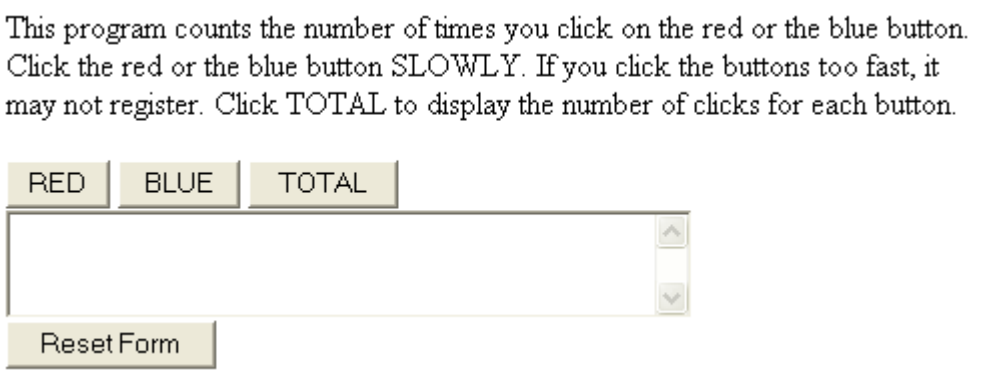
What if we declare the same name?

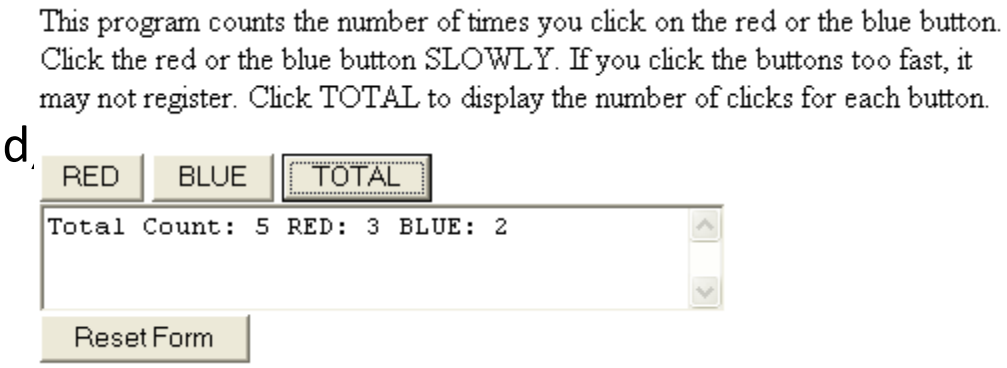
```
<script type="text/javascript">
<!—
var globalCount=0;
function example1() {
    var count1=0;
    count1=count1+1;
    globalCount=globalCount+1;
    alert("Global Count: "+globalCount);
}
function example2() {
    var count1=0;
    var globalCount=99;
    alert("Global Count: "+globalCount);
}

//-->
</script>
```

Note the second declaration of globalCount in example2()

Exercise 13: jex4.html

- Write a html page that has three buttons, a text area and reset button.
- The program counts the number of times you click the red or the blue button
- When you click the total, it gives you a running total of red, blue clicks.
- The reset button clears the display area and resets the count





Displaying strings – help!

If you have to display strings such as the following:

"I have 3 days left. "

You basically concatenate the strings as follows where count was a number you calculated:

"I have "+ count +" days left. "

Getting input: 2 ways of doing

Inside javascript

```
myName = prompt("What is your name?");
alert("My name is : "+ myName);
```

Inside form

```
<input type="text" name="nameField" value="Enter your name" size=20>
```

To access the string entered inside javascript

```
var myName=document.formName.nameField.value;
alert("My name is : "+ myName);
```

Note that the input is ALWAYS in STRING

Exercise 14: jex5.html

- Write a html page that sums or multiplies two numbers (see picture below).
- Caution: Because you are dealing with numbers, you have to make sure that you convert the input string to numbers.
- To change a value from a string to a number, you use the *parseFloat* command in javascript such as:

  num1=parseFloat(num1);

  ie. var num1=**document.formName.nameField.value**;
  Here num1 is actually a string ie. “4"; instead of the number 4 so you need to do num1=parseFloat(num1); to convert