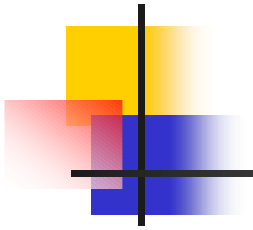
CSC309 Tutorial
– HTML DOM & JavaScript

TA: Lei Jiang

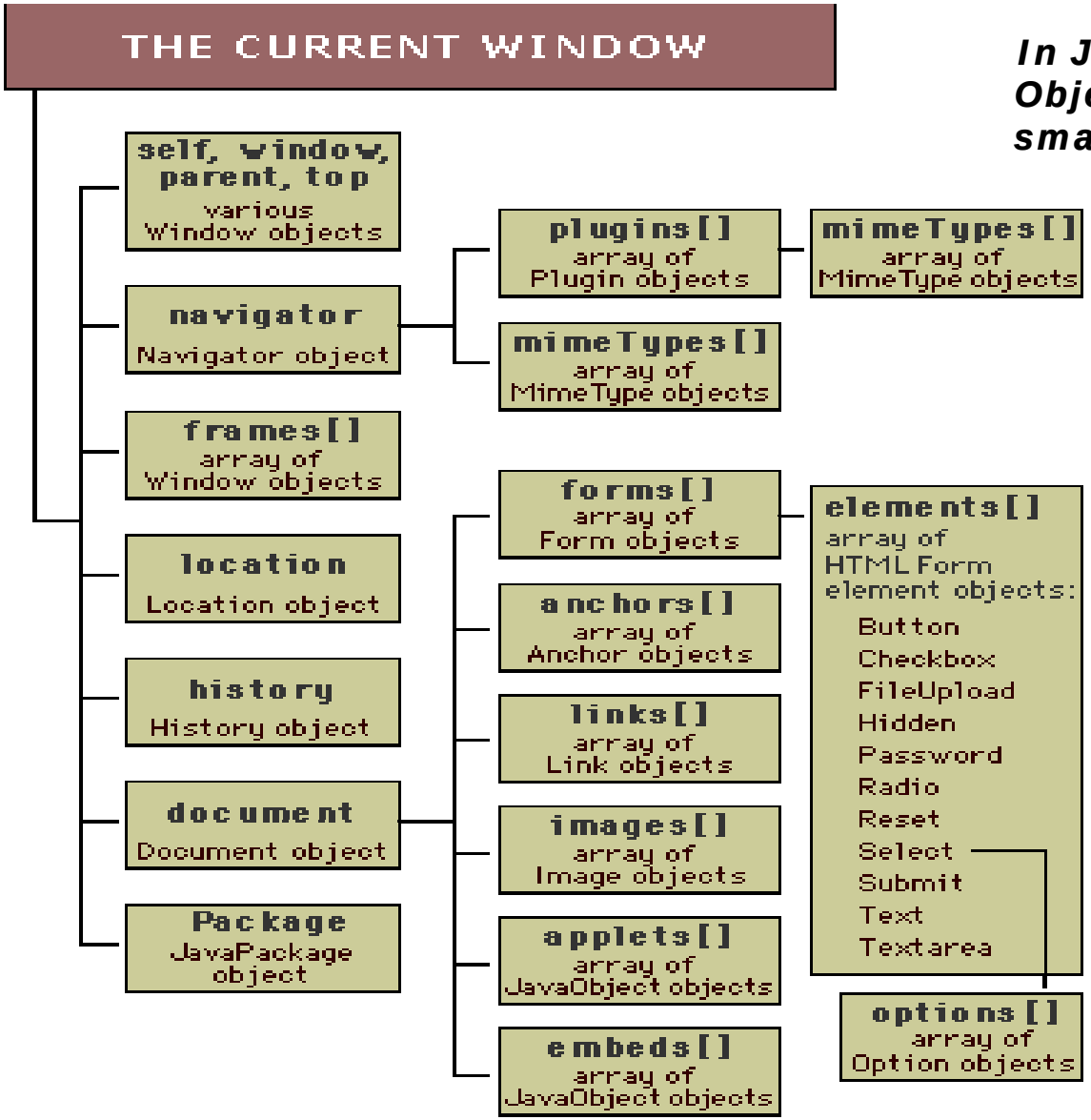
Feb. 24, 2003



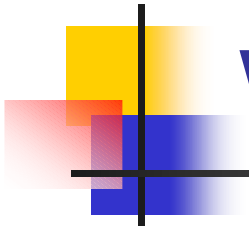
HTML DOM

- The Document Object Model is a way of accessing and modifying elements in a *HTML* or *XML* document.
- Core DOM v.s. HTML DOM
  - Core: mainly XML oriented
  - HTML: extends core, more specific to HTML documents

HTML DOM in JavaScript

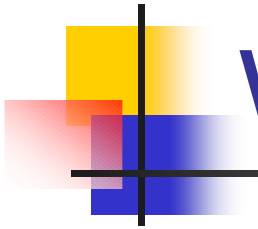


In JavaScript, Document Object Model describes a small hierarchy of objects.



Window & Frame Object

- The window object is the "parent" object for all other objects.
  - Other objects are accessible through window object's properties.
- A Frame object is defined by the FRAME tag in a FRAMESET document.
- Frame objects have the same properties and methods as window objects and differ only in the way they are displayed.



Window Object - Methods

- The window object has methods including:
  - **open** and **close**: Opens and closes a browser window
  - **alert**: Displays an Alert dialog box with a message.
  - **confirm**: Displays a Confirm dialog box with OK and Cancel buttons.
  - **prompt**: Displays a Prompt dialog box with a text field for entering a value.
  - **blur** and **focus**: Removes focus from, or gives focus to a window.
  - **setTimeout**: Evaluates an expression or calls a function once after the specified period elapses.

Opening Windows

- Using HTML

  ** UofT CS Dept **

- Using JavaScript
  - **window.open("URL","name","features");**
    - **name: the name of the window being open.**
    - **features: a list of the different components a window can have.**
  - **Example:**

    <a href="#" onClick="window.open('www.cs.toronto.edu','new_window');
    **return false;"> UofT CS Dept. **

    <a href="#"
    onClick="window.open('www.cs.toronto.edu','new_window','location,height=100
    **,width=100'); return false;"> UofT CS Dept. **

    <a href="#"
    onClick="window.open('www.cs.toronto.edu','new_window','location=no ,status
    **=no'); return false;"> UofT CS Dept. **

"features"
menubar
status
scrollbars
toolbar
width
height
location
resizable

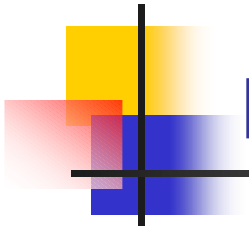
Windows Communication

```
<html>
<head>
    <title>Getting and using a window reference</title>
    <script language="JavaScript">
        // open a new window and get a reference to it
        var cs_window =
    window.open("http://www.cs.toronto.edu/","cshome","width=200,height=200");
        // blur the new window
        cs_window.blur();
    </script>
</head>
<body>
    <center>
    <h1>A new window has been opened and moved to the background.</h1>
    <a href="#" onMouseOver="cs_window.focus();">Bring it forward</a><br>
    <a href="#" onMouseOver="cs_window.blur();">Put it backward</a><br>
    </center>
</body>
</html>
```

Document Object - an Example

- The properties of the particular document object are largely content-dependent.
  - They are created based on the HTML in the document.
- Sample HTML

```
<HTML>
<HEAD>
    <TITLE>A Sample Document</TITLE>
    <SCRIPT>
        function update(form) { alert("Form being updated")}
    </SCRIPT>
</HEAD>
<BODY>
    <FORM NAME="myform" ACTION="foo.cgi" METHOD="get" >
        Enter a value: <INPUT TYPE="text" NAME="text1" VALUE="blank" SIZE=20 >
        Check if you want:  <INPUT TYPE="checkbox" NAME="Check1"
        CHECKED  onClick="update(this.form)"> Option #1
        <P> <INPUT TYPE="button" NAME="button1" VALUE="Press Me " onClick="update(this.form)">
    </FORM>
</BODY>
</HTML>
```

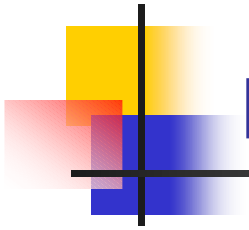


Document Object - an Example

- Given the previous HTML example, the **document** object has properties like:

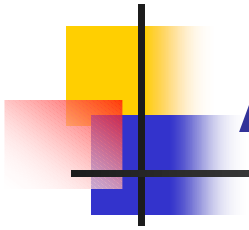
Properties	**Value**
document.title	"A Sample Document"
document.fgColor	#000000
document.bgColor	#ffffff
location.href	"http://www... .../sample.html"

The value of **document.title** reflects the value specified in the TITLE tag. The values for **document.fgColor** (the color of text) and **document.bgColor** (the background color) are based on the default values.



Document Object - an Example

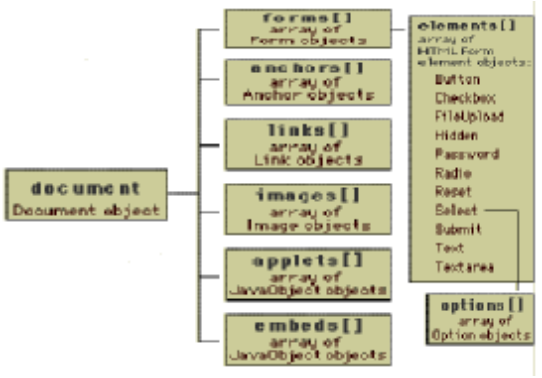
- Because there is a form in the document, there is also a Form object called document.myform
  - based on the form's NAME attribute
- The myform has child objects for the checkbox and the button.
  - document.myform.Check1, the checkbox
  - document.myform.button1, the button
- The myform object also has child objects named text1, corresponding to the text field in the form.
  - document.myform. text1.value is “blank"
  - document.myform. text1.name is "text1"



Arrarys

- **One way to access an element in the document is to use the name of the element.**
  - **Names are defined in the name attributes**
  - **Eg:<INPUT TYPE="text" NAME="text1" VALUE="blank" SIZE=20 >**
- **If you don't know the name of an element, you can use Arrarys.**

Arrarys



- For example, images[] array can be used to access **all the images in the current page.**
  - Eg:

    for (var loop = 0; loop < document.images.length; loop++) {

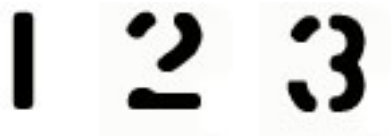
      document.images[loop].src =

      "http://www.utoronto.ca/images/uoftlogo1.gif";

    }
  - Changes all the images in the current page to UofT Logo

Swapping Images

```
<html>
<head>
    <title>Change-o-rama</title>
    <script language="JavaScript">
    function swap(img) {
        var change=prompt("Change to which one (1, 2 or 3)?","");
        if (change == "1" ) {
            window.document.images[img].src="one.jpg";
        }
        if (change == "2" ) {
            window.document.images[img].src="two.jpg";
        }
        if (change == "3" ) {
            window.document.images[img].src="three.jpg";
        }}
    function random_swap(img) {
        var change = Math.ceil(Math.random()*10) % 4;
        if (change == "1" ) {
            window.document.images[img].src="one.jpg";
        }
        if (change == "2" ) {
            window.document.images[img].src="two.jpg";
        }
        if (change == "3" ) {
            window.document.images[img].src="three.jpg";
        }}
    </script>
</head>
<body bgcolor="#ffffff">
Try clicking on the "Change" link to see what happens to the
    images below:
<p><img src="one.jpg" onMouseOver="random_swap(0)">
  <img src="two.jpg" onMouseOver="random_swap(1)">
  <img src="three.jpg" onMouseOver="random_swap(2)">
<p><a href="#" onClick="swap(0)">Change</a>
  <a href="#" onClick="swap(1)">Change</a>
  <a href="#" onClick="swap(2)">Change</a>
</body>
</html>
```



Change Change Change

Animating Images

```
<html>
<head>

<title>Automated Animations</title>

<script type="text/javascript">

function initial() {
  timedelay = 1000;
  counter = 0;
  numberOfImages = 3;
  animation = new Array();

  animation[0] = 'one.jpg';
  animation[1] = 'two.jpg';
  animation[2] = 'three.jpg';
  animateImages();
}
```

```
function animateImages(){
  document.animatedimage.src = animation[counter++];
  if(counter >= numberOfImages)
    counter= 0;
  timeId = setTimeout("animateImages()", timedelay);
}
</script>
</head>
<body onload="initial()" onunload="clearTimeout(timeId)">
<center><h2>Automated Animations</h2>
<img src="one.gif" name="animatedimage" width="50"
height="60" border="0"><br>
</center>
</body>
</html>
```

Form Manipulation

```
<html>
<head>
<title>Form value</title>
<script language="JavaScript">
    var des = "Overview This course provides an introduction to the technologies used for developing Web
    applications. We start with a brief overview of the networking technologies that make the Web possible,
    such as TCP and DNS. We then discuss technologies for static and architectures. We also discuss general
    Web design principles, with a special focus on security and scalability. ";
    var info = "Instructor:  Prof. Eyal de Lara, Location:  LM 159, Lectures:  Tuesdays and Thursdays 3-4
    Tutorials:  Mondays 4-5";
</script>
</head>
<body bgcolor="#ffffff">
<form name="form1">
    <textarea name="textarea1" rows=10 cols=60>
            Manipulating the Value of a Text Field
    </textarea>
</form>
<a href="#" onMouseOver="window.document.form1.textarea1.value= des;">Course
    Description</a>
<a href="#" onMouseOver="window.document.form1.textarea1.value=info;">Course Info</a>
</body>
</html>
```

Manipulating the Value of a Text Field

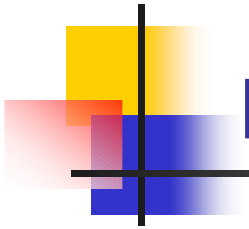
Course Description Course Info

Overview This course provides an introduction to the technologies used for developing Web applications. We start with a brief overview of the networking technologies that make the Web possible, such as TCP and DNS. We then discuss technologies for static and architectures. We also discuss general Web design principles, with a special focus on security and scalability.

Course Description Course Info

Instructor: Prof. Eyal de Lara, Location: LM 159, Lectures: Tuesdays and Thursdays 3-4 Tutorials: Mondays 4-5

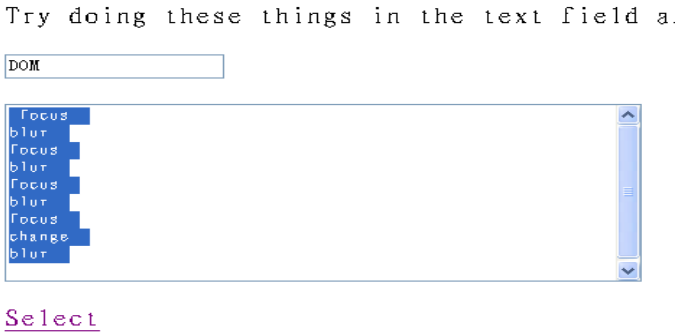
Course Description Course Info



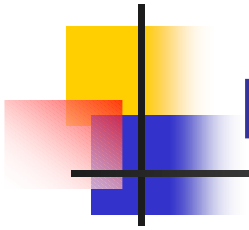
Form Manipulation

- **Text fields (areas) understand onBlur, onFocus, and onChange events.**
  - **onFocus: occurs when clicking inside a text field.**
  - **onBlur: happens when moving out of a text field by clicking outside of it, or hitting "tab."**
  - **onChange: happens when changing text field's content and then moves outside the text field.**

Form Manipulation

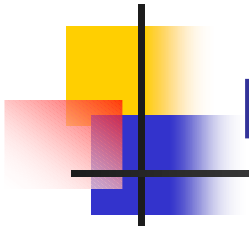



```
<html>
<head>
<title>Form2</title>
</head>
<script language="JavaScript">
function writeIt(the_word)
{
  var word_with_return = the_word + "\n";
  window.document.first_form.the_textarea.value += word_with_return;
}
</script>
<body bgcolor="#ffffff">
Try doing these things in the text field and see what happens in the textarea below it.
<form name="first_form">
      <input type="text" name="first_text" onFocus="writeIt('focus');" onBlur="writeIt('blur');"
      onChange="writeIt('change');">
      <p> <textarea name="the_textarea" rows=10 cols=60> </textarea>
      <p> <a href="#" onMouseOver="document.first_form.the_textarea.select()">Select</a>
</form>
</body>
</html>
```



Form Handler

- **Forms are objects as well,**
  - **they have their own methods, properties, and event handlers.**
  - **one important event handler is onSubmit.**
- **onSubmit gets called in two situations:**
  - **When a user clicks on a Submit button,**
  - **When a user hits Return in a text field.**
- **If this event is not handled, the JavaScript may behave differently than expected.**



Form Handler

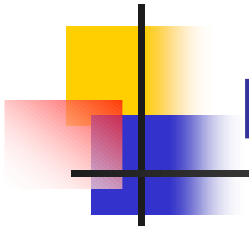
- **In Netscape, clicking on an unhandled Submit button generally leads to reloading the page.**
- **To avoid this use**

```
<form onSubmit="return false;">
        <input type="submit" value="Submit">
</form>
```

- **Generally, `return false;` is a way to change browsers' normal behavior.**

```
<a href="http://www.cs.toronto.edu/"
  onClick="return false;">UofT CS Dept.</a>
```

  - The link won't lead you anywhere because the return false; statement .

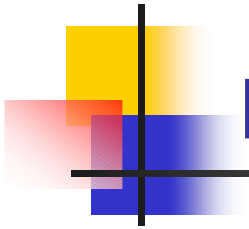


Form - Checkbox

- **Checkboxes have one main property checked**
  - **If a checkbox is checked, the checked property will be *true*; otherwise false.**
  - **Eg:**

```
var is_checked = window.document.the_form.the_checkbox.checked;
if (is_checked == true) {
        alert("Yup, it's checked!");
} else {
        alert("Nope, it's not checked.");
}
```

  - **Test if the_checkbox has been checked.**



Form - Checkbox

- **Checkboxes have one interesting event handler: onClick.**
  - **When someone clicks on a checkbox, the onClick event handler goes into action.**
  - **The following example switch the background color of the document.**

Form - Checkbox

```
<html>
<head>
<title>Form3</title>
</head>
<script language="JavaScript">
function switchLight()
{
  var the_box = window.document.form_3.check_1;
  var the_switch = "";
  if (the_box.checked == false) {
    alert("Hey! Turn that back on!");
    document.bgColor='black';
  } else { alert("Thanks!");  document.bgColor='white'; }
}
</script>
<body bgcolor="#ffffff">
<form name="form_3">
 <input type="checkbox" name ="check_1" onClick="switchLight();" checked=true>The Light Switch
</form>
</body>
</html>
```

Form - Select

- **There are two primary kinds of selects:**
  - **pulldown select**

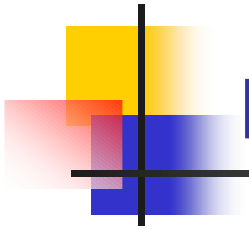
```
<select name="pulldown1">
   <option>spider
   <option>lemur
   <option>chimp
</select>
```

  - **list selects**

```
<select name="pulldown1" size=3>
   <option>spider
   <option>lemur
   <option>chimp
</select>
```

Form - Select


My Favorite Pets
Birds
robin
hummingbird
crow
Which one(s) you like?
Microsoft Internet Explorer
you selected: robin crow
确定



Form - Select

- **The form**

```
<form name="the_form" onSubmit="reportMultiple(); return false;">
    My Favorite Pets
    <select name="choose_category" size=1
    onChange="swapOptions(window.document.the_form.choose_category.options
    [selectedIndex].text);">
            <option selected>Dogs
            <option>Fish
            <option>Birds
    </select>
    <p> <select name="the_examples" size=3 >
            <option>poodle
            <option>puli
            <option>greyhound
    </select> </p>
    <p> <input type="submit" value="Which one(s) you like?"> </p>
</form>
```

Form - Select

- In the header, three arrays are created to match the pulldown select options.

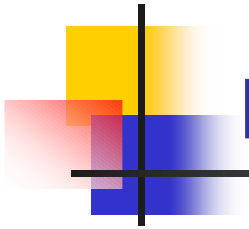
```
var Dogs = new Array("poodle","puli","greyhound");
var Fish = new Array("trout", "mackerel", "bass");
var Birds = new Array("robin", "hummingbird", "crow");
```

- swapOptions is called when the pulldown select is changed.
  - **If you select "Birds" in the pulldown select, the_array_name will equal the string "Birds"**

```
function swapOptions(the_array_name)
{
    var the_array = eval(the_array_name);
    setOptionText(window.document.the_form.the_examples, the_array);
}
```

- setOptionText is used to change the list select according to the pulldown select.

```
function setOptionText(the_select, the_array)
{
  for (loop=0; loop < the_select.options.length; loop++)
  { the_select.options[loop].text = the_array[loop]; }
}
```



Form - Select

- reportMultiple function popup an alert window to show your selection.

```
function reportMultiple() {
        var options_string = "";
        var the_select = window.document.the_form.the_examples;
        for (loop=0; loop < the_select.options.length; loop++) {
                if (the_select.options[loop].selected == true)
                { options_string += the_select.options[loop].text + " ";}
        }
        alert("you selected: " + options_string);
}
```