

COMP335 Web Application Development

JavaScript

1

- Group Project
 - group name and short (one paragraph) description
 - <http://goo.gl/forms/eo5wnbMtXf>
- Today
 - JavaScript
- Next class
 - JQuery

- Assignment#2-CSS
 - by Friday 2/26
- In-class programming exam (tentative)
 - 3/14 (M)

2

Introduction

- History
 - Originally developed by Netscape
 - Joint Development with Sun Microsystems in 1995
 - JavaScript was influenced by many languages and was designed to look like Java
 - JavaScript was a trademark of Sun Microsystems, now is a trademark of Oracle.
- **HTML** to define the **content** of web pages
- **CSS** to specify the **styles** of web pages
- **JavaScript** to program the **behavior** of web pages

3

Where does JavaScript go?

```
<body>

<h1> Hello fellow Web programmers! </h1>
<h1> Don't forget to submit group name and project description</h1>
<button type="button" onclick="foo()">Click me</button>

<script >                                Embedded
function foo(){
    document.getElementsByTagName("h1")[0].innerHTML="Hello World!";
    window.alert ("Hello World!");
}
</script>

<script src="basic.js"></script>          External

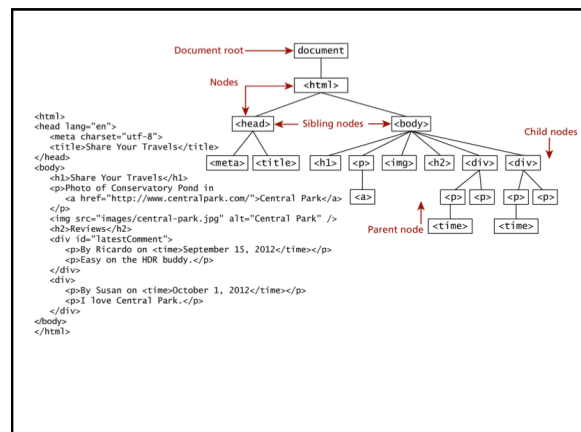
</body>

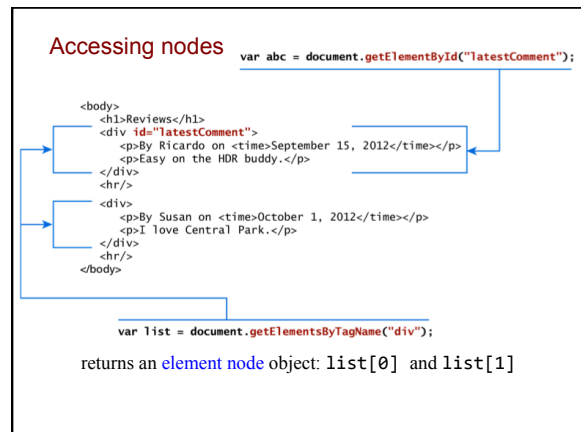
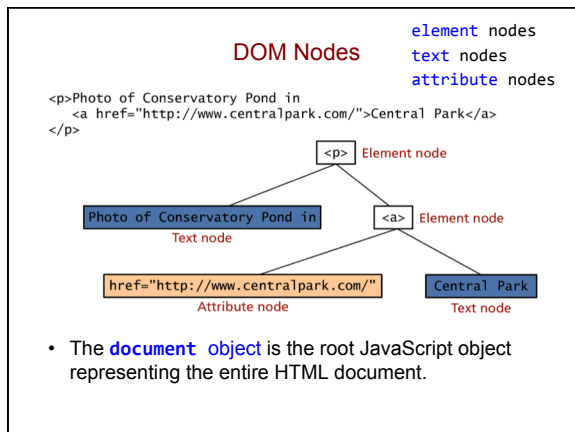
At the end of body tag
```

Example: basic.html

Document Object Model (DOM)

- JavaScript is almost always used to interact with the **HTML document** through **Document Object Model**.
- According to the W3C, the DOM is a:
Platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents.





Modify a DOM element

- Using the DOM document and HTML DOM element objects, we can do exactly that using the **innerHTML** property.
- Although the **innerHTML** technique works well (and is very fast), there is a more **verbose technique** available to us that builds output using the DOM.
 - DOM functions **createTextNode()**, **removeChild()**, and **appendChild()** allow us to modify an element in a more rigorous way

Example: node.html

Mini exercise

- Changing an element's style
 - Open your browser and go to <http://www.bridgew.edu>
 - get all <div> nodes and change its font color to "blue"

```

var divs = document.getElementsByTagName("div");

for(each in divs){
  if (divs[each].style) {
    divs[each].style.color = "blue";
  }
}
  
```

JavaScript Events

- A JavaScript **event** is an action that can be detected by JavaScript.
- We say then that an event is **triggered** and then it can be **caught** by JavaScript functions, which then do something in response.

Listener Approach

```

var greetingBox = document.getElementById('example1');
greetingBox.onclick = alert('Good Morning');
  
```

The "original" style of registering a listener

In the original JavaScript world, events could be specified right in the HTML markup with *hooks* to the JavaScript code (and still can).

```

var greetingBox = document.getElementById('example1');
greetingBox.addEventListener('click', alert('Good Morning'));
greetingBox.addEventListener('mouseout', alert('Goodbye'));
  
```

The "new" DOM2 approach to registering listeners

example: listener.html

Validating Forms

- Writing code to **pre-validate** forms on the client side will reduce the number of incorrect submissions, thereby reducing server load.
- There are a number of common validation activities including **email validation**, **number validation**, and **data validation**.

What does this code do?

```
<form>
  <input type="text" id="custName" placeholder="last, first name ">

  <input type="text" id="phone" placeholder="Phone No. (ddd-ddd-
  dddd)">

  <input type = "reset" id = "reset" >
  <input type = "submit" id = "submit" >

</form>

<script>
document.getElementById("custName").addEventListener('change',chkName);
document.getElementById("phone").addEventListener('change',chkPhone);
</script>
```

The **change** is used when the content of a field changes

14

What does this code do?

```
function chkPhone() {
  var myPhone = document.getElementById("phone");
  // -1 if no match was found
  var pos = myPhone.value.search(/^d{3}-d{3}-d{4}$/);

  if (pos == -1) {
    alert("The phone number you entered (" + myPhone.value +
    ") is not in the correct form. \n" +
    "The correct form is: ddd-ddd-dddd \n" +
    "Please go back and fix your phone number");

    return false;
  }
  else
    return true;
}
```

To keep the form active after the event handler is finished, the handler must return false

15

example: form.html

What does this code do?

```
function chkName() {
  var myName = document.getElementById("custName");
  // -1 if no match was found
  var pos = myName.value.search(
    /^[A-Z][a-z]+, ?[A-Z][a-z]+$/);

  if (pos == -1) {
    alert("The name you entered (" + myName.value +
    ") is not in the correct form. \n" +
    "The correct form is: " +
    "Last-name, First-name, Middle-Initial \n" +
    "Please go back and fix your name");

    return false;
  }
  else
    return true;
}
```

To keep the form active after the event handler is finished, the handler must return false

16

In-class exercise

- Modify "JS-ex1.html" to validate inputs



This page says:

The two passwords you entered are not the same
Please re-enter both now



This page says:

You did not enter a password
Please enter one now

☐ Prevent this page from creating additional dialogs.

OK

