

JAVASCRIPT

What makes a web site dynamic?

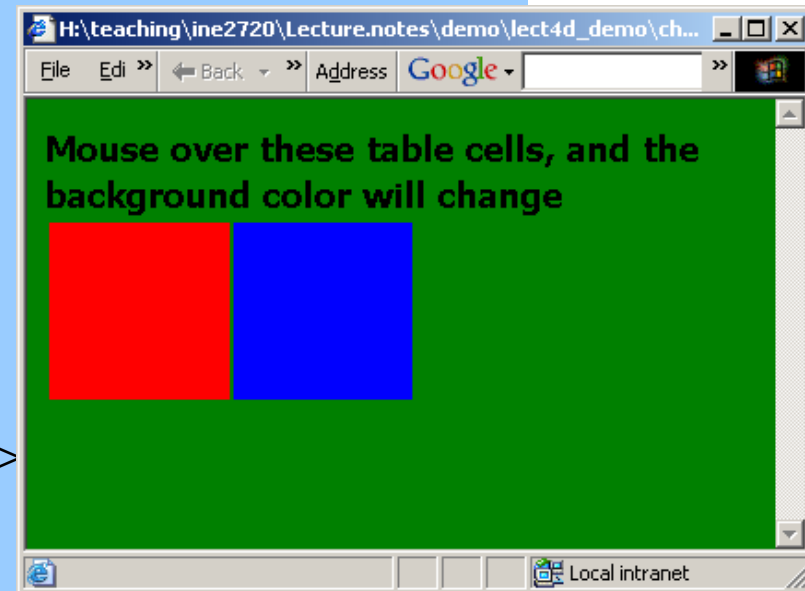
- **Interactivity** — adapt and react to the visitor's actions as quick as possible.
- **Synchronicity** — bring together relevant information from a variety of sources.
- **Flexibility** — give the visitor different ways to find information in the site.
- **Adaptability** — adjusts to cater to individual visitor's needs.
- **Activity** — uses motion and sound to draw user's attention to changes on the site.

The Role of each component in DHTML

- With CSS, we can change the style of any HTML elements.
- With DOM, we can have a map on every elements in the HTML page.
- With JavaScript, we can access and have operations on the elements in the DOM tree.
- With event handler, we can execute the predefined scripts at any time.

Change Background Color

```
<html><head>
<script language="JavaScript">
function bgChange(bg)
{   document.body.style.background=bg; }
</script></head>
<body><b>Mouse over these table cells, and the background color will change</b>
<table width="300" height="100">
<tr>
<td onmouseover="bgChange('red')"
    onmouseout="bgChange('transparent')" bgcolor="red">
</td>
<td onmouseover="bgChange('blue')"
    onmouseout="bgChange('transparent')" bgcolor="blue">
</td>
<td onmouseover="bgChange('green')"
    onmouseout="bgChange('transparent')" bgcolor="green">
</td>
</tr>
</table>
</body></html>
```

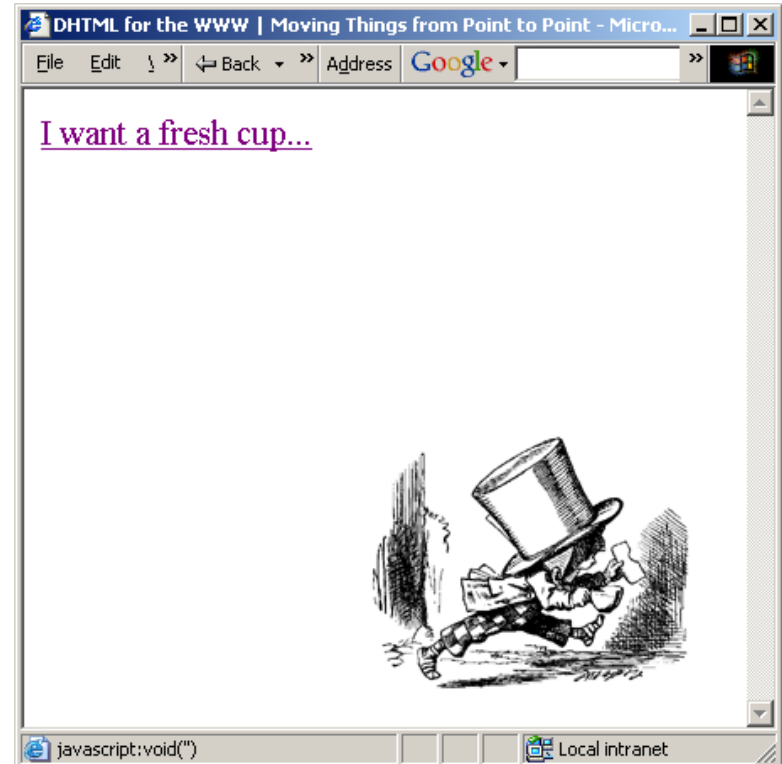
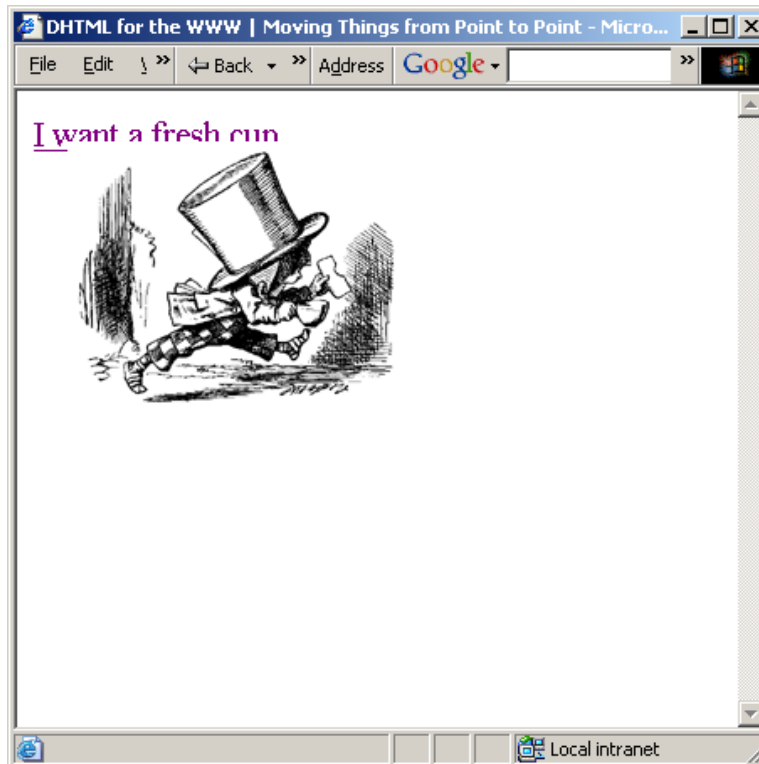


Dynamic Positioning

- ❑ It means the HTML elements can be positioned by using JavaScript.
- ❑ The element is moved by manipulating any of the “top”, “left”, “right” and “bottom” CSS properties.
- ❑ The more table you use, the slower your page displays.
- ❑ Positioning elements with CSS is more accurate than tables and the results are displayed much faster.

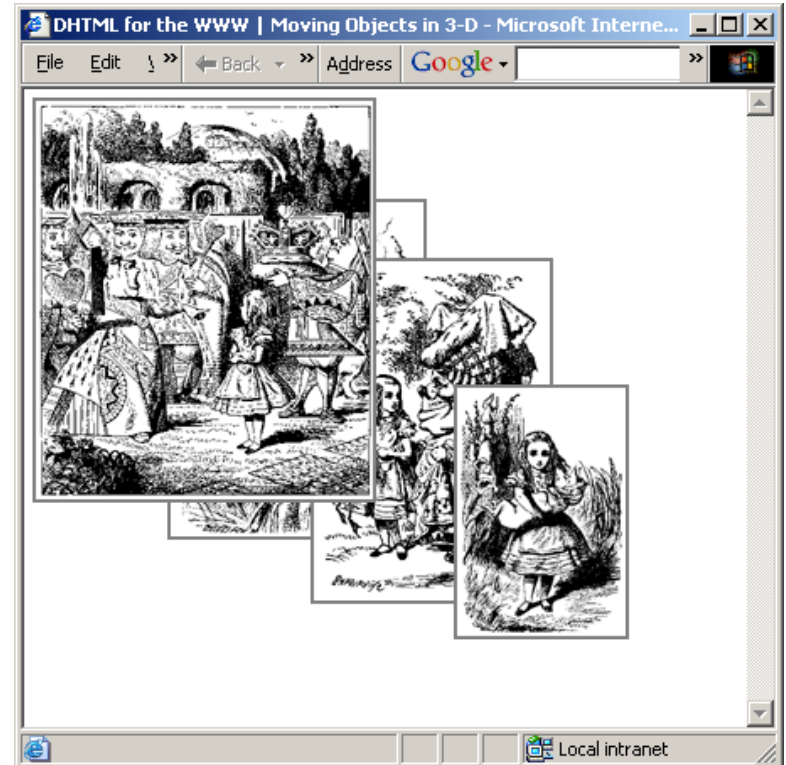
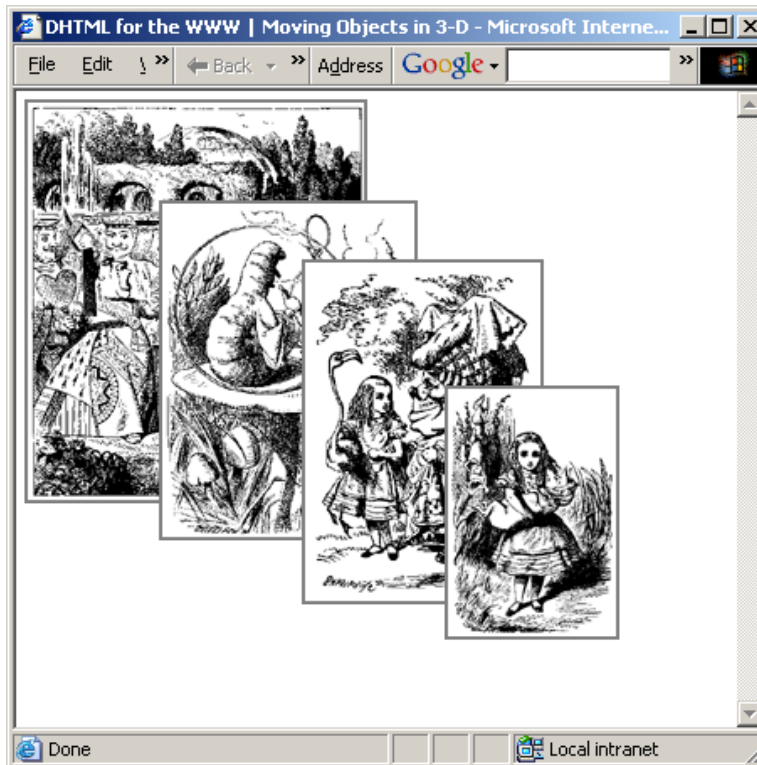
Moving Objects from Point to Point

- Using CSS, you can change the position of an object on the screen.



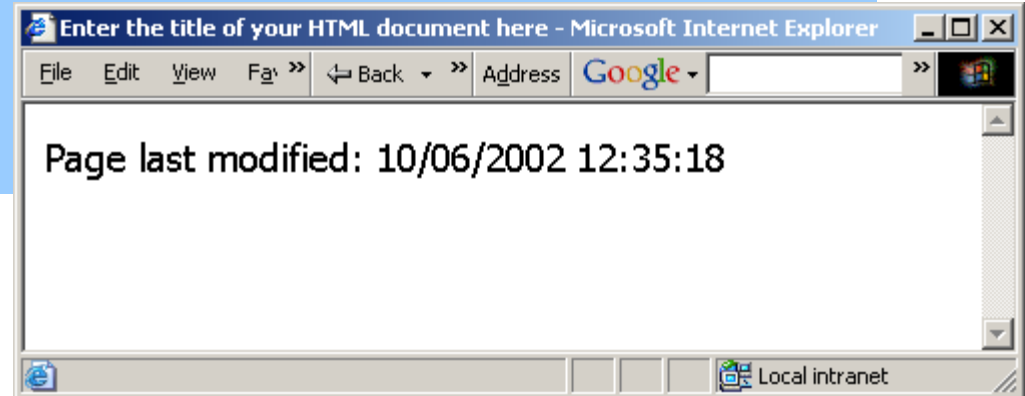
Moving Objects in 3-D

- By specifying the z-index, we can put the objects in 3-D manner.



Adding last modified Date

```
<html>
<head>
<title>Enter the title of your HTML document here</title>
</head>
<body>
<script language="JavaScript">
document.write("Page last modified: " + document.lastModified)
</script>
</body>
</html>
```



More Events and Validation

Page/window events

name	description
load	the browser loads the page
unload	the browser exits the page
resize	the browser window is resized
error	an error occurs when loading a document or an image

Form events

event name	description
submit	form is being submitted
reset	form is being reset
change	the text or state of a form control has changed

```
<script type="text/javascript" src="prototype.js"></script>
```

```
document.observe("dom:loaded", function() {  
    $("orderform").observe('submit', function(Event) {  
        if ($("zipcode").value.length < 5){  
            Event.stop();  
            alert('zipcode<5');  
        }  
    });  
});  
..  
<form name="orderform" id="orderform">  
    <input type="text" name="tZip" id="zipcode" value="">  
    <input type="submit" name="bSub" value="GO" id="submit"> JS
```

Prototype and forms

- `$("#id")["name"]`: gets parameter with given name from form with given id
- `$F("id")`: `$F` returns value of form with the given id

```
<script type="text/javascript"src="prototype.js"></script>
<script>function f1() {
    var value = $F('username');
    if (value.length < 4) {
        $("#username").clear();
        $("#username").disable();
    }
}
</script></head><body>
<input type="text" name="username" id="username"
value="ini">
<input type="button" name="btn1" value="GO"
onclick="f1();">
```

Client-side validation code

```
<form id="exampleform">  
City: <input type="text" name="ct" id="city"><BR>  
State: <input type="text" name="st" id="state"><BR>  
<input type="submit" name="submit" value="GO">
```

HTML

```
window.onload = function() {  
    $("exampleform").onsubmit = checkData;  
};  
function checkData(event) {  
    if($("city").value==" "||$("state").value.length!=2){  
        Event.stop(event);  
        alert("Error, invalid city/state.");  
    }  
}
```

JS

- forms expose onsubmit and onreset events
- to abort a form submission, call Prototype's Event.stop on the event

Regular expressions in JavaScript

- `string.match(regex)`
 - if string fits the pattern, returns the matching text; else returns null
 - can be used as a Boolean truthy/falsey test:

```
var name = $("name").value;  
if (name.match(/[a-z]+/)) { ... }
```

- an `i` can be placed after the regex for a case-insensitive match
 - `name.match(/Abdul/i)` will match “abdul”, “AbDuL”, ...

http://www.w3schools.com/jsref/jsref_obj_regexp.asp

Replacing text with regular expressions

- `string.replace(regex, "text")`
 - ▣ replaces the first occurrence of given pattern with the given text
 - ▣ `var str = "Abdul Rahman";`
`str.replace(/[a-z]/, "a")` returns " Aadul Rahman"
- a `g` can be placed after the regex for a global match (replace all occurrences)
 - ▣ `str.replace(/[a-z]/g, "x")` returns "Axxxxx Rxxxxxx"

Keyboard/text events

name	description
<u>keydown</u>	user presses a key while this element has keyboard focus
<u>keyup</u>	user releases a key while this element has keyboard focus
<u>keypress</u>	user presses and releases a key while this element has keyboard focus
<u>focus</u>	this element gains keyboard focus
<u>blur</u>	this element loses keyboard focus
<u>select</u>	this element's text is selected or deselected)

Key event objects

property name	description
keyCode <small>rototype's key code constants</small>	ASCII integer value of key that was pressed (convert to char with String.fromCharCode)
altKey, ctrlKey, shiftKey	true if Alt/Ctrl/Shift key is being held

Key event objects

Event.KEY_BACKSPACE	Event.KEY_DELETE	Event.KEY_DOWN	Event.KEY_END
Event.KEY_ESC	Event.KEY_HOME	Event.KEY_LEFT	Event.KEY_PAGEDOWN
Event.KEY_PAGEUP	Event.KEY_RETURN	Event.KEY_RIGHT	Event.KEY_TAB
Event.KEY_UP			

- issue: if the event you attach your listener to doesn't have the focus, you won't hear the event
 - ▣ possible solution: attach key listener to entire page body, outer element, etc.