

Dynamic Documents

1. Positioning Elements

Absolute Positioning 1

<!-- absPos.html

Illustrates absolute positioning of elements

-->

<html >

<head>

<title> Absolute positioning </title>

<style type = "text/css">

/* A style for a paragraph of text */

.regtext {font-family: Times; font-size: 14pt; width: 600px}

/* A style for the text to be absolutely positioned */

.abstext {position: absolute; top: 25px; left: 50px;

font-family: Times; font-size: 24pt;

font-style: italic; letter-spacing: 1em;

color: red; width: 500px}

```
</style>

</head>

<body>

  <p class = "regtext">

    Apple is the common name for any tree of the genus Malus, of
    the family Rosaceae. Apple trees grow in any of the temperate
    areas of the world. Some apple blossoms are white, but most
    have stripes or tints of rose. Some apple blossoms are bright
    red. Apples have a firm and fleshy structure that grows from
    the blossom. The colors of apples range from green to very
    dark red. The wood of apple trees is fine grained and hard.
    It is, therefore, good for furniture construction. Apple trees
    have been grown for many centuries. They are propagated by
    grafting because they do not reproduce themselves.

  </p>

  <p class = "abstext">

    APPLES ARE GOOD FOR YOU

  </p>

</body>

</html>
```

Absolute Positioning 2

```
<?xml version = "1.0" encoding = "utf-8" ?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!-- absPos2.html
```

Illustrates nested absolute positioning of elements

```
-->
```

```
<html xmlns = "http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<title> Nested absolute positioning </title>
```

```
<style type = "text/css">
```

```
/* A style for a paragraph of text */
```

```
.regtext {font-family: Times; font-size: 14pt; width: 500px;
position: absolute; top: 100px; left: 100px;}
```

```
/* A style for the text to be absolutely positioned */
```

```
.abstext {position: absolute; top: 25px; left: 50px;
font-family: Times; font-size: 24pt;
font-style: italic; letter-spacing: 1em;
color: red; width: 400px;}
```

</style>

</head>

<body>

<div class = "regtext">

Apple is the common name for any tree of the genus *Malus*, of the family Rosaceae. Apple trees grow in any of the temperate areas of the world. Some apple blossoms are white, but most have stripes or tints of rose. Some apple blossoms are bright red. Apples have a firm and fleshy structure that grows from the blossom. The colors of apples range from green to very dark red. The wood of apple trees is fine grained and hard. It is, therefore, good for furniture construction. Apple trees have been grown for many centuries. They are propagated by grafting because they do not reproduce themselves.

APPLES ARE GOOD FOR YOU

</div>

</body>

</html>

Relative Positioning

```
<?xml version = "1.0" encoding = "utf-8" ?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<!-- relPos.html
```

Illustrates relative positioning of elements

```
-->
```

```
<html xmlns = "http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<title> Relative positioning </title>
```

```
</head>
```

```
<body style = "font-family: Times; font-size: 24pt;">
```

```
<p>
```

Apples are <span style =

"position: relative; top: 5px;

font-family: Times; font-size: 48pt;

font-style: italic; color: red;">

GOOD for you.

```
</p>
```

```
</body>
```

```
</html>
```

Moving Elements

<!-- mover.html

Uses mover.js to move an image within a document

-->

```
<html xmlns = "http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<title> Moving elements </title>
```

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

```
</script>
```

```
</head>
```

```
<body>
```

```
<form action = "">
```

```
<p>
```

```
<label>
```

x coordinate:

```
<input type = "text" id = "leftCoord" size = "3" />
```

```
</label>
```

```
<br />
```

```
<label>
```

y coordinate:

```
<input type = "text" id = "topCoord" size = "3" />
```

```

</label>

<br />

<input type = "button" value = "Move it"
      onclick =
        "moveIt('baby',
          document.getElementById('topCoord').value,
          document.getElementById('leftCoord').value)" />

</p>

</form>

<div id = "baby" style = "position: absolute;
      top: 115px; left:0px;">

  <img src = "baby.jpg"
      alt = "(Pictures of Saturn)" />

</div>

</body>

</html>

```

// mover.js

// Illustrates moving an element within a document

// The event handler function to move an element

```
function moveIt(movee, newTop, newLeft) {
```

```
dom = document.getElementById(movee).style;

// Change the top and left properties to perform the move
// Note the addition of units to the input values

dom.top = newTop + "px";
dom.left = newLeft + "px";
}
```

Element Visibility

```
<!-- showHide.html

    Uses showHide.js

    Illustrates visibility control of elements

-->

<html xmlns = "http://www.w3.org/1999/xhtml">

<head>

    <title> Visibility control </title>

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

    </script>

</head>

<body>

    <form action = "">
```



```
<div id = "baby" style = "position: relative;
    visibility: visible;">
    <img src = "baby.jpg"
        alt = "(my baby)" />
</div>

<p>
    <br />
    <input type = "button" value = "Toggle Baby"
        onclick = "flipImag()" />
</p>
</form>
</body>
</html>
```

```
// showHide.js
```

```
// Illustrates visibility control of elements
```

```
// The event handler function to toggle the visibility
```

```
// of the images of Saturn
```

```
function flipImag() {
```

```
    dom = document.getElementById("baby").style;
```

```
// Flip the visibility adjective to whatever it is not now
if (dom.visibility == "visible")
    dom.visibility = "hidden";
else
    dom.visibility = "visible";
}
```

Changing Colors

<!-- dynColors.html

Illustrates dynamic foreground and background colors

-->

<html xmlns = "http://www.w3.org/1999/xhtml">

<head>

<title> Dynamic colors </title>

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

</script>

</head>

<body>

<p style = "font-family: Times; font-style: italic;

font-size: 24pt" >

This small page illustrates dynamic setting of the

foreground and background colors for a document

</p>

<form action = "">

<p>

<label>

Background color:

<input type = "text" name = "background" size = "10"
onchange = "setColor('background', this.value)" />

</label>

<label>

Foreground color:

<input type = "text" name = "foreground" size = "10"
onchange = "setColor('foreground', this.value)" />

</label>

</p>

</form>

</body>

</html>

// **dynColors.js**

// Illustrates dynamic foreground and background colors

```
// The event handler function to dynamically set the  
// color of background or foreground
```

```
function setColor(where, newColor) {  
    if (where == "background")  
        document.body.style.backgroundColor = newColor;  
    else  
        document.body.style.color = newColor;  
}
```

Changing Fonts

<!-- dynFont.html

Illustrates dynamic font styles and colors

-->

<html xmlns = "http://www.w3.org/1999/xhtml">

<head>

<title> Dynamic fonts </title>

<style type = "text/css">

.regText {font: 16pt 'Times New Roman'}

.wordText {color: blue;}

</style>

</head>

```

<body>

<p class = "regText">

The state of

<span class = "wordText";

    onmouseover = "this.style.color = 'red';

        this.style.fontStyle = 'italic';

        this.style.fontSize = '24pt';"

    onmouseout = "this.style.color = 'blue';

        this.style.fontStyle = 'normal';

        this.style.fontSize = '16pt';">

    Washington

</span>

    produces many of our nation's apples.

</p>

</body>

</html>

```

Dynamic Content

```
<!-- dynValue.html
```

Illustrates dynamic values

```
-->
```

```
<html xmlns = "http://www.w3.org/1999/xhtml">
```

```
<head>

<title> Dynamic values </title>

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

</script>

</head>

<body>

<form action = "">

<p style = "font-weight: bold">

<span style = "font-style: italic">

    Customer information

</span>

<br /><br />

<label>

    Name:

    <input type = "text"  onmouseover = "messages(0)"

        onmouseout = "messages(4)" />

</label>

<br />

<label>

    Email:

    <input type = "text"  onmouseover = "messages(1)"

        onmouseout = "messages(4)" />
```

</label>

To create an account, provide the following:

<label>

User ID:

<input type = "text" onmouseover = "messages(2)"
onmouseout = "messages(4)" />

</label>

<label>

Password:

<input type = "password"
onmouseover = "messages(3)"
onmouseout = "messages(4)" />

</label>

<textarea id = "adviceBox" rows = "3" cols = "50"
style = "position: absolute; left: 250px;
top: 100px">

This box provides advice on filling out the form on this page. Put the mouse cursor over any input field to get advice.

</textarea>


```
<input type = "reset" value = "Reset" />
```

</p>

```
// dynValue.js
```

```
// Illustrates dynamic values
```

```
var helpers = ["Your name must be in the form: \n \n  
first name, middle initial., last name", "Your email address must have the form: \n  
user@domain",  
"Your user ID must have at least six characters",  
"Your password must have at least six \n  
characters and it must include one digit",  
"This box provides advice on filling out \n  
the form on this page. Put the mouse cursor over any \n  
field to see the advice."]
```



```
input field to get advice"]
```

```
// *****
```

```
// The event handler function to change the value of the
```

```
// textarea
```

```
function messages(adviceNumber) {  
    document.getElementById("adviceBox").value =  
        helpers[adviceNumber];  
}
```

Locating the Mouse Cursor

```
<!-- where.html
```

```
    Uses where.js
```

```
    Illustrates x and y coordinates of the mouse cursor
```

```
-->
```

```
<html xmlns = "http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<title> Where is the cursor? </title>
```

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

```
</script>
```

```
</head>
```

```
<body onclick = "findIt(event)">

  <form action = "">

    <p>

      Within the client area: <br />

      x:

      <input type = "text" id = "xcoor1" size = "4" />

      y:

      <input type = "text" id = "ycoor1" size = "4" />

      <br /><br />

      Relative to the origin of the screen coordinate system:

      <br />

      x:

      <input type = "text" id = "xcoor2" size = "4" />

      y:

      <input type = "text" id = "ycoor2" size = "4" />

    </p>

  </form>

  <p>

    <img src = "yellowplane3.jpeg" alt = "(Picture of C172)" />

  </p>

</body>

</html>
```

//where.js

// shown the coordinates of the mouse cursor position

// in an image and anywhere on the screen when the mouse is clicked JavaScript
Document

//event handler function to get and display the

// coordinates of the cursor, both in an element and on the screen

function findIt(evt)

{

document.getElementById("xcoor1").value=evt.clientX;

document.getElementById("ycoor1").value=evt.clientY;

document.getElementById("xcoor2").value=evt.screenX;

document.getElementById("ycoor2").value=evt.screenY;

}

Reacting to a Mouse Click

<!-- anywhere.html

Display a message when the mouse button is pressed,
no matter where it is on the screen

-->

<html xmlns = "http://www.w3.org/1999/xhtml">

<head>

```

<title> Sense events anywhere </title>

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

</script>

</head>

<body onmousedown = "displayIt(event);"

    onmouseup = "hideIt();">

<p>

    <span id= "message"

        style = "color: red; visibility: hidden;

            position: relative;

            font-size: 20pt; font-style: italic;

            font-weight: bold;">

        Please don't click here!

    </span>

    <br /><br /><br /><br /><br /><br /><br /><br />

    <br /><br /><br /><br /><br /><br /><br /><br />

</p>

</body>

</html>

// anywhere.js

//  Display a message when the mouse button is pressed,

```

```
// no matter where it is on the screen

// The event handler function to display the message

function displayIt(evt) {
    var dom = document.getElementById("message");
    dom.style.left = (evt.clientX - 130) + "px";
    dom.style.top = (evt.clientY - 25) + "px";
    dom.style.visibility = "visible";
}

// *****

// The event handler function to hide the message

function hideIt() {
    document.getElementById("message").style.visibility =
        "hidden";
}
```

Slow Movement of Elements

<!-- moveText.html

Illustrates a moving text element

Uses the JavaScript from file moveTextfun.js

-->

<html xmlns = "http://www.w3.org/1999/xhtml">

<head>

<title> Moving text </title>

<script type = "text/javascript"

src = "moveTextfun.js">

</script>

</head>

<!-- Call the initializing function on load, giving the
destination coordinates for the text to be moved

-->

<body onload = "initText()">

<!-- The text to be moved, including its initial position -->

<p>

<span id = 'theText' style =

"position: absolute; left: 500px; top: 500px;

font: bold 20pt 'Times Roman';

color: blue;"> Hello B1 section :)

</p>

```
</body>
```

```
</html>
```

```
// moveTextfuncs.js
```

```
- used with moveText.html
```

```
var dom, x=500, y=500, finalx = 0, finaly = 0;
```

```
// ***** //
```

```
// A function to initialize the x and y coordinates
```

```
// of the current position of the text to be moved,
```

```
// and then call the mover function
```

```
function initText() {
```

```
    dom = document.getElementById('theText').style;
```

```
/* Get the current position of the text */
```

```
    var x = dom.left;
```

```
    var y = dom.top;
```

```
/* Convert the string values of left and top to
```

```
numbers by stripping off the units */
```

```

x = x.match(/\d+/);

y = y.match(/\d+/);


/* Call the function that moves it */


moveText(x, y);
} /*** end of function initText */


// ***** //

// A function to move the text from its original
// position to (finalx, finaly)

function moveText(x, y) {

/* If the x coordinates are not equal, move
x toward finalx */

if (x != finalx)
    if (x > finalx) x--;
    else if (x < finalx) x++;

```



```
/* If the y coordinates are not equal, move
   y toward finaly */

if (y != finaly)
    if (y > finaly) y--;
    else if (y < finaly) y++;

/* As long as the text is not at the destination,
   call the mover with the current position */

if ((x != finalx) || (y != finaly)) {

/* Put the units back on the coordinates before
   assigning them to the properties to cause the
   move */

    dom.left = x + "px";
    dom.top = y + "px";

/* Recursive call, after a 1-millisecond delay */

    setTimeout("moveText(" + x + ", " + y + ")", 1);
```

```
}
```

```
} /*** end of function moveText */
```

Dragging and Dropping Elements

<!-- dragNDrop.html

An example to illustrate the DOM 2 Event model

Allows the user to drag and drop words to complete

a short poem

Does not work with IE8

```
-->
```

```
<html xmlns = "http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<title> Drag and drop </title>
```

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

```
</script>
```

```
</head>
```

```
<body style = "font-size: 20;">
```

```
<p>
```

```
Roses are red <br />
```

```
Violets are blue <br />
```

```
<span style = "position: absolute; top: 200px; left: 0px;
    background-color: green;"
    onmousedown = "grabber(event);"> candy </span>
<span style = "position: absolute; top: 200px; left: 75px;
    background-color: lightgrey;"
    onmousedown = "grabber(event);"> cats </span>
<span style = "position: absolute; top: 200px; left: 150px;
    background-color: lightgrey;"
    onmousedown = "grabber(event);"> cows </span>
<span style = "position: absolute; top: 200px; left: 225px;
    background-color: lightgrey;"
    onmousedown = "grabber(event);"> glue </span>
<span style = "position: absolute; top: 200px; left: 300px;
    background-color: lightgrey;"
    onmousedown = "grabber(event);"> is </span>
<span style = "position: absolute; top: 200px; left: 375px;
    background-color: lightgrey;"
    onmousedown = "grabber(event);"> is </span>
<span style = "position: absolute; top: 200px; left: 450px;
    background-color: lightgrey;"
    onmousedown = "grabber(event);"> meow </span>
<span style = "position: absolute; top: 250px; left: 0px;
```

```

        background-color: lightgrey;"

    onmousedown = "grabber(event);"> mine </span>
<span style = "position: absolute; top: 250px; left: 75px;
        background-color: lightgrey;"

    onmousedown = "grabber(event);"> moo </span>
<span style = "position: absolute; top: 250px; left: 150px;
        background-color: lightgrey;"

    onmousedown = "grabber(event);"> new </span>
<span style = "position: absolute; top: 250px; left: 225px;
        background-color: lightgrey;"

    onmousedown = "grabber(event);"> old </span>
<span style = "position: absolute; top: 250px; left: 300px;
        background-color: lightgrey;"

    onmousedown = "grabber(event);"> say </span>
<span style = "position: absolute; top: 250px; left: 375px;
        background-color: lightgrey;"

    onmousedown = "grabber(event);"> say </span>
<span style = "position: absolute; top: 250px; left: 450px;
        background-color: lightgrey;"

    onmousedown = "grabber(event);"> so </span>
<span style = "position: absolute; top: 300px; left: 0px;
        background-color: lightgrey;"

```

```

        onmousedown = "grabber(event);"> sticky </span>
<span style = "position: absolute; top: 300px; left: 75px;
        background-color: lightgrey;"
        onmousedown = "grabber(event);"> sweet </span>
<span style = "position: absolute; top: 300px; left: 150px;
        background-color: lightgrey;"
        onmousedown = "grabber(event);"> syrup </span>
<span style = "position: absolute; top: 300px; left: 225px;
        background-color: lightgrey;"
        onmousedown = "grabber(event);"> too </span>
<span style = "position: absolute; top: 300px; left: 300px;
        background-color: lightgrey;"
        onmousedown = "grabber(event);"> yours </span>
</p>
</body>
</html>

```

// **dragNDrop.js**

```

// An example to illustrate the DOM 2 Event model
// Allows the user to drag and drop words to complete
// a short poem
// Does not work with IE8

```

```
// Define variables for the values computed by
// the grabber event handler but needed by mover
// event handler

    var diffX, diffY, theElement;

// *****

// The event handler function for grabbing the word

function grabber(event) {

// Set the global variable for the element to be moved

    theElement = event.currentTarget;

// Determine the position of the word to be grabbed,
// first removing the units from left and top

    var posX = parseInt(theElement.style.left);
    var posY = parseInt(theElement.style.top);
```

```
// Compute the difference between where it is and
// where the mouse click occurred

diffX = event.clientX - posX;
diffY = event.clientY - posY;

// Now register the event handlers for moving and
// dropping the word

document.addEventListener("mousemove", mover, true);
document.addEventListener("mouseup", dropper, true);

// Stop propagation of the event and stop any default
// browser action

event.stopPropagation();
event.preventDefault();

} /** end of grabber

// *****
```

```
// The event handler function for moving the word

function mover(event) {

    // Compute the new position, add the units, and move the word

    theElement.style.left = (event.clientX - diffX) + "px";
    theElement.style.top = (event.clientY - diffY) + "px";

    // Prevent propagation of the event

    event.stopPropagation();
} /** end of mover

// *****

// The event handler function for dropping the word

function dropper(event) {

    // Unregister the event handlers for mouseup and mousemove

    document.removeEventListener("mouseup", dropper, true);
```



```
document.removeEventListener("mousemove", mover, true);
```

```
// Prevent propagation of the event
```

```
event.stopPropagation();
```

```
} /** end of droppers
```