JavaScript & Advanced Technologies

Style object

ITI – Assiut Branch

Eng. Hany Saad

How to Apply

❑ Dynamic Styles.

❑ Positioning Page Elements.

❑ Dynamic Content.

Dynamic Styles

❑ Font Styles

❑ Text Styles

❑ Text and Background Colors

❑ Borders, Padding, and Margins

❑ Background Images

❑ Opacity

Font Styles

CSS and DOM Reference	Values
font-family:*name* *object***.style.fontFamily="***name***"**	Font *name* can be any system font; multiple names can be specified in order of preference, separated by commas.
font-size:*size* *object***.style.fontSize="***size***"**	Font *size* is specified as in a unit of measurement, normally point size (12pt).

font-style:*style*
*object***.style.fontStyle="***style***"**
	Font *style* specified as
	    normal
	    italic

font-weight:*weight*
*object***.style.fontWeight="***weight***"**
	Font *weight* specified as
	    normal
	    bold

font-variant:*variant*
*object***.style.fontVariant="***variant***"**
	Font *variant* specified as
	    normal
	    small-caps

Text Styles

CSS and DOM Reference	Values

text-align:*alignment*
*object***.style.textAlign="***alignment***"**
	Sets the horizontal *alignment* of text within an element. The *alignment* can be: left
	                center
	                right
	                justify

line-height:*height*
*object***.style.lineHeight="***height***"**
	Sets the *height* of lines of text in an element; specify a
	    measurement (px, pt, n%, em, en)
	    normal

letter-spacing:*spacing* *object***.style.letterSpacing="***spacing***"**	Sets the *spacing* between letters in an element; specify a measurement (px, pt, n%, em, en) normal
text-indent:*size* *object***.style.textIndent="***size***"**	Sets the *size* of indentation of the first line of a block of text; specify units of measurement (px, pt, n%, em, en)

text-transform:*case*
*object***.style.textTransform="***case***"**
	Sets the *case* of words in a text block using
	    capitalize
	    lowercase
	    uppercase
	    none

text-decoration:*style*
*object***.style.textDecoration="***style***"**
	Sets a *style* using: underline
	                overline
	                line-through
	                none

Text and Background Colors

CSS and DOM Reference	Values

color:*color*
*object***.style.color="***color***"**
	Foreground color specified as a color name, hexadecimal value, or RGB value:
	  color:red
	  color:#FF0000
	  color:rgb(255,0,0)

background-color:*color*
*object***.style.backgroundColor="***color***"**
	Background color specified as a color name, hexadecimal value, or RGB value:
	  background-color:red
	  background-color:#FF0000
	  background-color:rgb(255,0,0)

Borders, Padding, and Margins (Cont.)

❑ **CSS Box Model**
- o All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.
- o The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.
- o The image below illustrates the box model:
    - **Content** - The content of the box, where text and images appear
    - **Padding** - Clears an area around the content. The padding is transparent
    - **Border** - A border that goes around the padding and content
    - **Margin** - Clears an area outside the border. The margin is transparent

Borders, Padding, and Margins (cont.)

CSS and DOM Reference.	Values

border-style:*style*
object.style.borderStyle
 	
border-top-style:*style*
object.style.borderTopStyle
 	
border-right-style:*style*
object.style.borderRightStyle
 	
border-bottom-style:*style*
object.style.borderBottomStyle
 	
border-left-style:*style*
object.style.borderLeftStyle
	Sets the *style* of a border surrounding a page element.
	
	The *style* can be applied to all borders (border-style, borderStyle) or to selected borders. Style types can be
	  dashed
	  dotted
	  double
	  groove
	  inset
	  none
	  outset
	  ridge
	  solid

Borders, Padding, and Margins (Cont.)

CSS and DOM Reference.	Values

object.style.borderWidth="*width*"
 	
border-top-width:*width*
object.style.borderTopWidth="*width*"
 	
border-right-width:*width*
object.style.borderRightWidth="*width*"
 	
border-bottom-width:*width*
object.style.borderBottomWidth="*width*"
 	
border-left-width:*width*
object.style.borderLeftWidth="*width*"
	Sets the *width* of a border surrounding a page element.
	
	The *width* can be applied to all borders (border-width, borderWidth) or to selected borders. Widths can be
	  thin
	  medium
	  thick
	  *n*px

Borders, Padding, and Margins (Cont.)

CSS and DOM Reference	Values
border-color:*color* **borderColor** **border-top-color:***color* **borderTopColor** **border-right-color:***color* **borderRightColor** **border-bottom-color:***color* **borderBottomColor** **border-left-color:***color* ***border*LeftColor**	Sets the *color* of a border surrounding a page element. The *color* can be applied to all borders (border-color, borderColor) or to selected borders. The *color* is specified as a color name, hexadecimal value, or RGB value.

Borders, Padding, and Margins (Cont.)

CSS and DOM Reference	Values
border:*style width color* ***object*.style.border="***style width color***"**	Border styles, widths, and colors can be set with the single border specification by coding these values separated by a blank space: border:solid 1px red border="solid 1px red"

Background Images

CSS and DOM Reference	Values
background-image:url(*url***)** *object***.backgroundImage="***url***"**	Sets the URL of a background image; *url* can be set to none to prevent an image from loading.
background-position:*location* *object***.backgroundPosition="***location***"**	Sets the *location* of the left and top edges of the background image with a pair of values separated by a space. Values are left|center|right paired with top|center|bottom OR **x% y%** Locations can also be specified as pairs of percentages or pixels for the left and top values. https://www.w3schools.com/cssref/pr_background-position.asp

background-repeat:*axes*
*object***.backgroundRepeat="***axes***"**
	Sets whether a background image should repeat along the horizontal and/or vertical axes. *Axes* values are:
	  no-repeat
	  repeat
	  repeat-x
	  repeat-y

background-attachment:*value*
*object***.backgroundAttachment="***value***"**
	Describes whether a background image remain fixed in place or scrolls with the document. *Values* are:
	  fixed
	  scroll

Background Images (Cont.)

CSS and DOM Reference	Values
background-size *object*.**backgroundSize=""**	The background-size property specifies the size of the background images. There are four different syntaxes you can use with this property: the keyword syntax ("auto", "cover" and "contain"), the one-value syntax (sets the width of the image (height becomes "auto"), the two-value syntax (first value: width of the image, second value: height), and the multiple background syntax (separated with comma). https://www.w3schools.com/cssref/css3_pr_background-size.asp

❑ Background image properties reference:
https://www.w3schools.com/cssref/css3_pr_background.asp

opacity

❑ **Example: Image Opacity**

❑ CSS Format:

> opacity: 0.4; /* from 0 to 1*/
> filter: alpha(opacity=40); /* For IE8 and earlier */

❑ DHTML Format:

> object.style.opacity="0.4";
> object.style.filter='alpha(opacity=40)'; /* For IE8 and earlier */

How to Apply

❑ Dynamic Styles

❑ Positioning Page Elements

❑ Dynamic Content

Positioning Page Elements

❑ Attributes used:

- o Positioning Styles
- o Dynamic position
- o Dynamic sizing
- o float
- o overflow
- o z-index
- o Visibility
- o display
- o clip

Positioning styles

❑ **The CSS positioning properties allow you to position an element.**

❑ **CSS syntax:**

position: static|absolute|fixed|relative|initial|inherit;

❑ **DHTML Syntax**

object.style.position="static|absolute|fixed|relative|initial|inherit"

Positioning styles (cont.)

❑ **Elements can be positioned as:**

- ○ **Static Positioning**
  - • HTML elements are positioned static by default.
  - • A static positioned element is always positioned according to the normal flow of the page.
  - • Static positioned elements are not affected by the top, bottom, left, and right properties.

- ○ **Fixed Positioning**
  - • An element with fixed position is positioned relative to the browser window.
  - • It will not move even if the window is scrolled.

- ○ **Sticky Positioning**
  - • A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).
  - • https://www.w3schools.com/howto/howto_css_sticky_element.asp

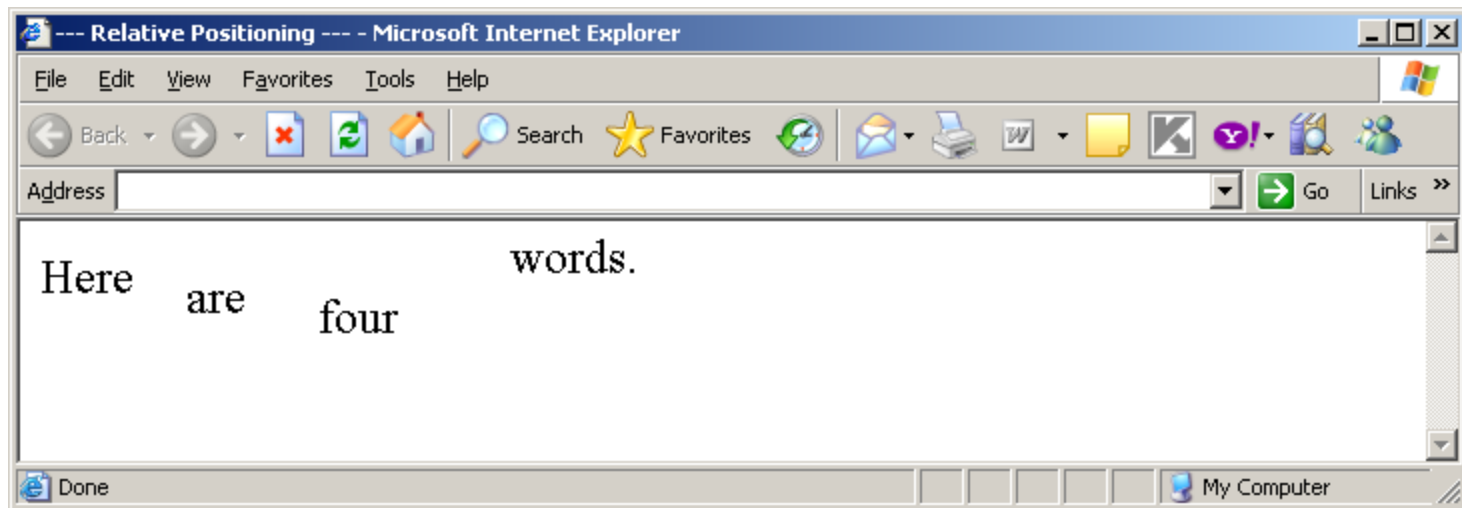# Positioning styles (cont.)

❑ **Elements can be positioned as:**

o **Relative Positioning**

- A relative positioned element is positioned relative to its normal position.
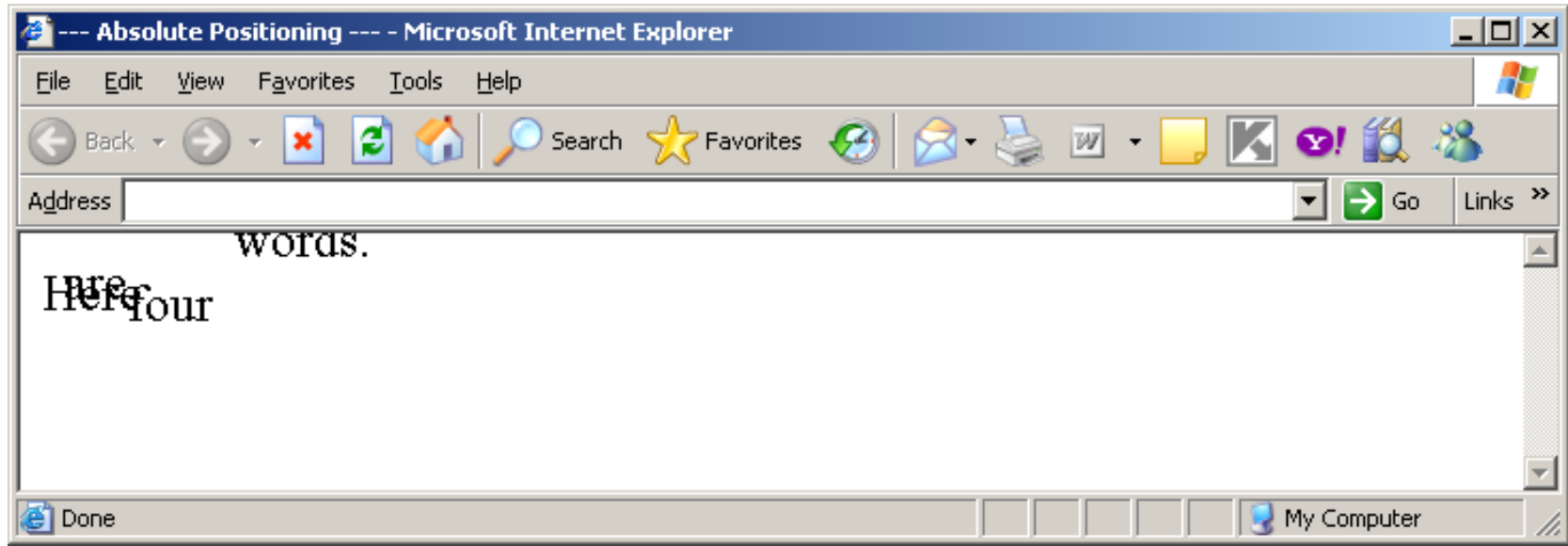
o **Absolute Positioning**

- An absolute position element is positioned relative to the first parent element that has a position other than static.

- Absolutely positioned elements are removed from the normal flow. The document and other elements behave like the absolutely positioned element does not exist.

- Absolutely positioned elements can overlap other elements.

position:relative


```
<p style="font-size:18pt">
        <span>Here</span>
        <span style="position:relative; left:20px; top:10px">are</span>
        <span style="position:relative; left:50px; top:20px">four</span>
        <span style="position:relative; left:100px; top:-10px">words.</span>
</p>
```

position:absolute


```
<p style="font-size:18pt">
    <span>Here</span>
    <span style="position: absolute; left:20px; top:10px">are</span>
    <span style="position: absolute; left:50px; top:20px">four</span>
    <span style="position: absolute; left:100px; top:-10px">words.</span>
</p>
```

Positioning Page Elements (cont.)

❑ Attributes used:

- o Positioning Styles
- o Dynamic position
- o Dynamic sizing
- o float
- o overflow
- o z-index
- o Visibility
- o display
- o clip

Dynamic Position

CSS and DOM Reference	Effects
left:*n px* *object***.style.left="***n px***"**	Sets the left edge of the element relative to its container element; *n* is a ***string*** measurement unit, e.g., left:100px.
top:*n px* *object***.style.top="***n px***"**	Sets the top edge of the element relative to its container element; *n* is a ***string*** measurement unit, e.g., top:100px
right:*n* *object***.style.right="***n px***"**	Sets the right edge of the element relative to its container element; *n* is a ***string*** measurement unit, e.g., right:100px.
bottom:*n* *object***.style.bottom="***n px***"**	Sets the bottom edge of the element relative to its container element; *n* is a ***string*** measurement unit, e.g., bottom:100px

Dynamic Position(Cont.)

CSS and DOM Reference	Effects
object.style.pixelLeft=*n* **IE Only, deprecated in others**	Sets the left edge of the element relative to its container element; *n* is ***numeric*** for use in calculations, e.g., 100.
object.style.pixelTop=*n* **IE Only, deprecated in others**	Sets the top edge of the element relative to its container element; *n* is ***numeric*** for use in calculations, e.g., 100.

❑ Instead of using pixelLeft and pixelTop, you can use left, top as following:

```
//To get its value:
var x=parseInt(object.style.left)+100;
//To set its value:
Object.style.left = x + "px";
```

Positioning Page Elements (cont.)

❑ Attributes used:

- o Positioning Styles
- o Dynamic position
- o Dynamic sizing
- o float
- o overflow
- o z-index
- o Visibility
- o display
- o clip

Dynamic sizing

CSS and DOM Reference	Effects
width:*value* *object*.**style.width="***n px***"**	Sets the width of the element; *n* is a **string** measurement, either in pixels or percentages.
height:*n* *object*.**style.height="***n px***"**	Sets the height of the element; *n* is a **string** measurement, either in pixels or percentages.
object.**style.pixelWidth=***n* **IE Only, deprecated in others**	Sets the width of the element; *n* is **numeric** for use in calculations.
object.**style.pixelHeight=***n* **IE Only, deprecated in others**	Sets the height of the element; *n* is **numeric** for use in calculations.

Positioning Page Elements (cont.)

❑ Attributes used:

- o Positioning Styles
- o Dynamic position
- o Dynamic sizing
- o float
- o overflow
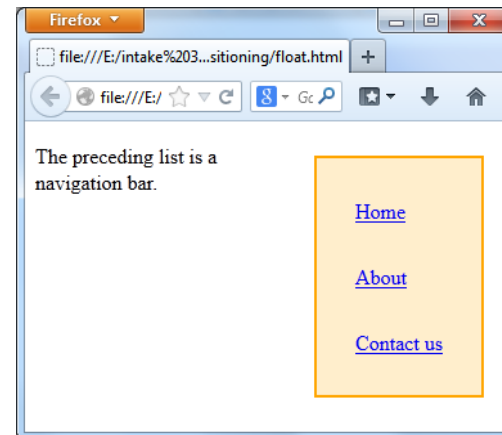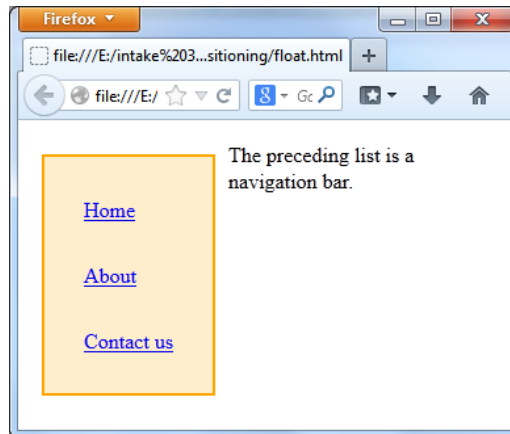- o z-index
- o Visibility
- o display
- o clip

float

❑ With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it.

❑ CSS

float:left|right|none|inherit|initial

❑ DHTML

object.style. float="left|right|none|inherit|initial"

float (cont.)

- ❑ **Turning off Float - Using Clear**
    - o Elements after the floating element will flow around it. To avoid this, use the clear property.
    - o The clear property specifies which sides of an element other floating elements are not allowed.

- ❑ CSS syntax
    Clear: left|right|both|none|inherit|initial

- ❑ DHTML
    object.style.clear="left|right|both|none|inherit|initial"

Positioning Page Elements (cont.)

❑ Attributes used:

- o Positioning Styles
- o Dynamic position
- o Dynamic sizing
- o float
- o overflow
- o z-index
- o Visibility
- o display
- o clip

overflow

❑ specifies what happens if content overflows an element's box

❑ CSS syntax

> overflow: visible|hidden|scroll|auto|initial|inherit;

❑ DHTML syntax

> object.style.overflow="visible|hidden|scroll|auto|initial|inherit"

Overflow (cont.)

❑Property values:

Value	Description
visible	The overflow is not clipped. It renders outside the element's box. This is default
hidden	The overflow is clipped, and the rest of the content will be invisible
scroll	The overflow is clipped, but a scroll-bar is added to see the rest of the content
auto	If overflow is clipped, a scroll-bar should be added to see the rest of the content
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

Positioning Page Elements (cont.)

❑ Attributes used:

- o Positioning Styles
- o Dynamic position
- o Dynamic sizing
- o float
- o overflow
- o z-index
- o Visibility
- o display
- o clip

z-index

- The z-index property is used to place an element "behind" another element.

- Default z-index is 0.

- The higher number the higher priority. z-index: -1 has lower priority.

- CSS syntax

    z-index:n

- DHTML syntax

    object.style.zIndex=n

Positioning Page Elements (cont.)

❑ Attributes used:

- o Positioning Styles
- o Dynamic position
- o Dynamic sizing
- o float
- o overflow
- o z-index
- o Visibility
- o display
- o clip

Visibility

❑ The visibility property determines if an element is visible or not.

❑ CSS syntax

> visibility:hidden|visible|collapse

❑ DHTML syntax

> object.style.visibility="hidden|visible|collapse"

❑ Collapse: Only for table elements. collapse removes a row or column, but it does not affect the table layout. The space taken up by the row or column will be available for other content.

❑ If collapse is used on other elements, it renders as "hidden"

http://www.w3schools.com/cssref/playit.asp?filename=playcss_visibility_collapse

Positioning Page Elements (cont.)

❑ Attributes used:

- o Positioning Styles
- o Dynamic position
- o Dynamic sizing
- o float
- o overflow
- o z-index
- o Visibility
- o display
- o clip

display

❑ The display property determines how the element id displayed.

❑ CSS syntax

> display:block|inline|none ...

❑ DHTML syntax

> object.style.display=":block|inline|none …"

Display (cont.)

❑ **Display values:**

Value	Description
inline	Default value. Displays an element as an inline element (like <span>)
block	Displays an element as a block element (like <p>)
none	The element will not be displayed at all (has no effect on layout)
list-item	Let the element behave like a element
table	Let the element behave like a <table> element
table-cell	Let the element behave like a <td> element
table-column	Let the element behave like a <col> element
table-row	Let the element behave like a <tr> element

❑ **More properties:**

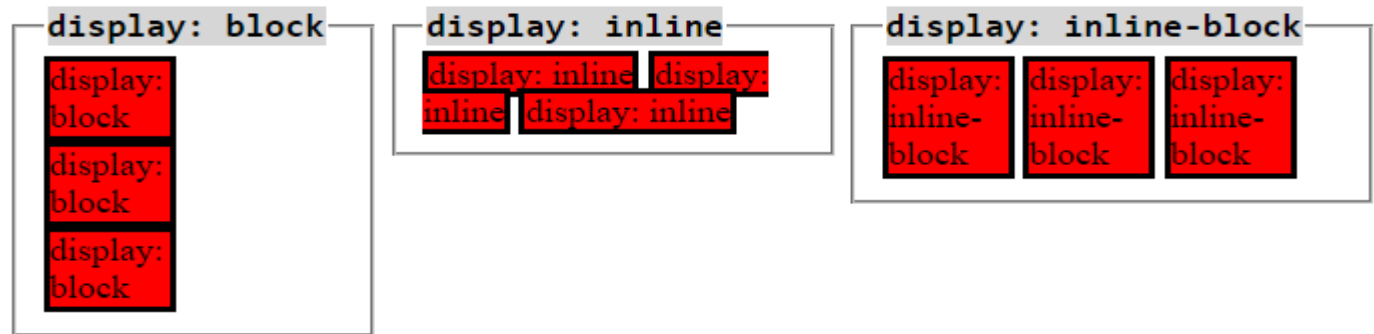http://www.w3schools.com/cssref/pr_class_display.asp

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

Display (cont.)

❑ The display: inline-block Value

- o Compared to display: inline, the major difference is that display: inline-block allows to set a width and height on the element.

- o Also, with display: inline-block, the top and bottom margins/paddings are respected, but with display: inline they are not.

- o Compared to display: block, the major difference is that display: inline-block does not add a line-break after the element, so the element can sit next to other elements.

❑ More:

https://www.w3schools.com/css/css_inline-block.asp

https://stackoverflow.com/questions/9189810/css-display-inline-vs-inline-block

http://dustwell.com/div-span-inline-block.html

https://alligator.io/css/display-inline-vs-inline-block

http://learnlayout.com/inline-block.html

Positioning Page Elements (cont.)

❑ Attributes used:

- o Positioning Styles
- o Dynamic position
- o Dynamic sizing
- o float
- o overflow
- o z-index
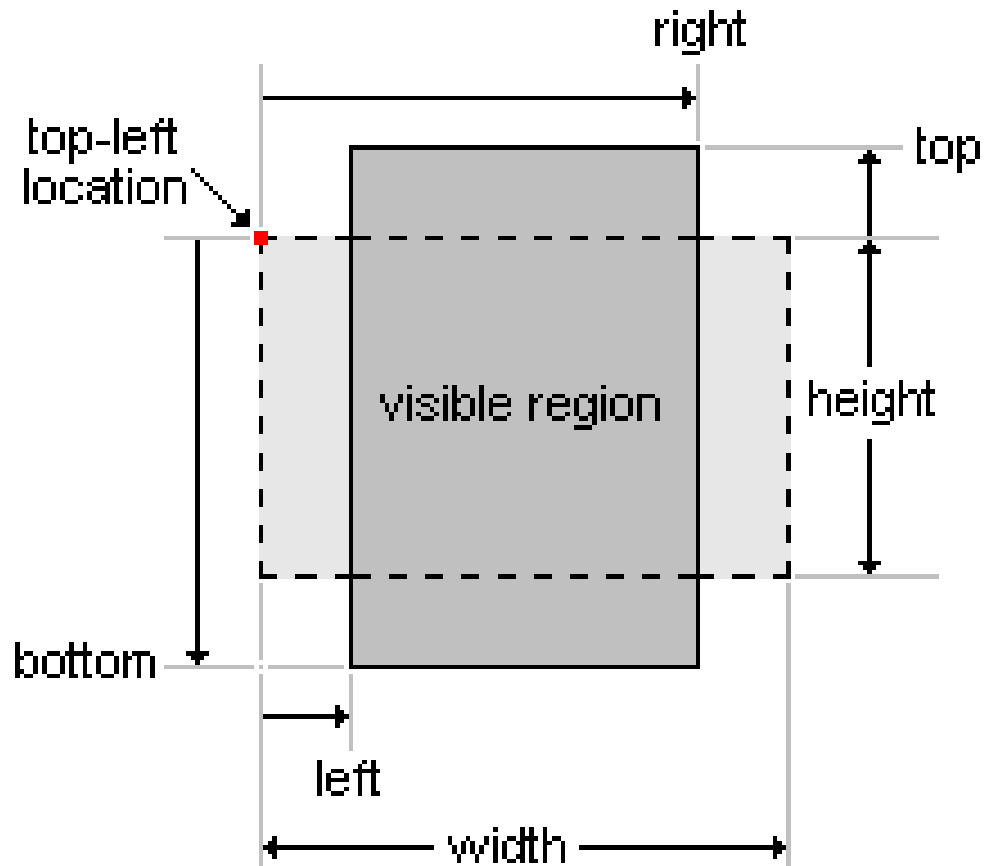- o Visibility
- o display
- o clip

clip

- ❑ Specifies how an element is clipped for display; i.e. which part is visible.

- ❑ The clipping region is defined as a rectangle by setting the clip value for each of the 4 edges (top, right, bottom, left).

- ❑ For each edge you can clip a portion of the viewing space away, or to add extra viewing space.

- ❑ CSS syntax

clip:rect(top,right,bottom,left)

- ❑ DHTML syntax

object.style.clip=(top,right,bottom,left)

Clip(Cont.)

clip:rect(top,right,bottom,left)

How to Apply

❑ Dynamic Styles

❑ Positioning Page Elements

❑ Dynamic Content

Dynamic Content

❏ All HTML containers--paragraphs, spans, divisions, and the like—have innerText and innerHTML properties representing the content that appears inside their tags.
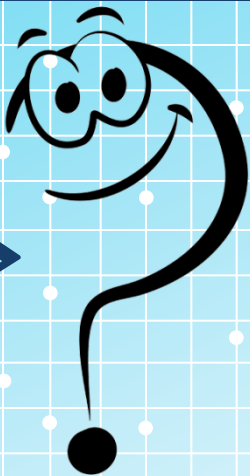
❏ These properties can be changed through scripting.

DOM Reference	Values
object.**innerText** = "*text*" **IE Only, use textContent() instead.**	Sets the text content of the container object, replacing any existing text.
object.**innerHTML** = "*text + HTML*"	Sets the text and XHTML content of a container object, replacing any existing text and XHTML tags. The XHTML is rendered when assigned to this property.

<SCRIPT > </SCRIPT>

<script>document.writeln("Thank You!")</script>