

**SOEN 287**

# **Chapter 6: Dynamic Documents with JavaScript (1)**

**Dr. Yuhong Yan**  
**CSE, Concordia University**  
**Winter, 2023**

**1**



# Topics

- Positioning elements
- Moving elements
- Element visibility
- Changing colors and fonts
- Dynamic content

**This set of slides is modified from the slides accompanied the textbook.**



# Changing the Document Dynamically

- What to change
  - Tag contents
  - Tag attribute values
  - Element style property values
- When
  - after the document has been displayed
  - react to event



# Positioning Elements

- CSS-P was released by W3C in 1997
- Completely supported by major browsers
- The position of any element is dictated by the three style properties: `position`, `left`, and `top`
  - The **five** possible values of `position` are `absolute`, `relative`, `static`, `fixed`, and `inherit`.



# Absolute Positioning

- Example 1:

```
<p style = "position: absolute; left: 50px;  
          top: 100px;">
```

→ **SHOW** `absPos.html`

- Example 2: absolute positioning for nested elements

→ **SHOW** `absPos2.html`

- You can move this element!



# Relative Positioning

- Example:

```
<span style = "position: relative; left: 50px;  
            top: 100px;"> GOOD </span>
```

→ **SHOW** relPos.html

- top and left properties are the offsets from its normal place
- If no top and left properties are specified, the element is placed in its normal place
- You can move this element!



# Fixed Positioning

- Example:

```
<span style = "position: fixed; left: 50px;  
            top: 100px;"> GOOD </span>
```

- `top` and `left` properties define the position relative to the browser window
- You can move this element!



# Static Positioning

- Example:

```
<span style = "position: static;"> GOOD </span>
```

→ **SHOW** relPos.html

- `static` is the default value
- no `top` and `left` properties
- You cannot move this element!





# Moving Elements

- If `position` is set to either `absolute` or `relative`, the element can be moved after it is displayed
- Just change the `top` and `left` property values with a script

→SHOW `mover.html` & `mover.js`

```
sty = document.getElementById("movee").style;  
  
sty.top = newTop + "px";  
sty.left = newLeft + "px";
```



## iClicker question

- In the previous example, what if I change "position:absolute" to "position:static"?
  - A. The example works as before.
  - B. The image does not move.
  - C. At least one error found in the error console.

**Answer: B**



## iClicker question

- Can I change the "position" from "static" to "absolute" using JavaScript?
  - A. No. Once the value of the position is defined, you cannot change it via JavaScript.
  - B. Yes. You can change it with JavaScript.
  - C. Not sure.

**Answer: B**



# Element Visibility

- The `visibility` property of an element controls whether it is displayed
- The values are `visible` and `hidden`

```
sty = document.getElementById("aID").style;  
  
if (sty.visibility == "visible")  
    sty.visibility = "hidden";  
else  
    sty.visibility = "visible";
```

→ **SHOW** `showHide.html` & `showHide.js`



# Property "visibility" vs. "display"

## ○ Visibility:

- hidden: hidden, but take place
- visible: normal

## ○ Display

- inline: element like `<span>`
- block: element like `<div>`, `<p>`, `<li>`
- none: hidden, take no place

**[http://www.w3schools.com/cssref/tryit.asp?filename=trycss\\_display\\_inline](http://www.w3schools.com/cssref/tryit.asp?filename=trycss_display_inline)**

**displayNone.html**



## iClicker question

- If an element is hidden, is its place occupied by other element?
  - A. NO.
  - B. YES.
  - C. Not sure.

**Answer: A**



# Changing Colors and Fonts

- Background color is controlled by the `backgroundColor` property
- Foreground color is controlled by the `color` property

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

→ **SHOW** `dynColors.html` & `dynColors.js`

# Challenge question

- How to change the background image using JavaScript?





# Changing Fonts

- A frequent scenario: change the font properties of a link by using the `mouseover` and `mouseout` events

```
onmouseover = "this.style.color = 'blue';  
              this.style.font = 'italic 16pt Times';"  
onmouseout = "this.style.color = 'black';  
              this.style.font = 'normal 16pt Times';"
```

→ **SHOW** `dynLink.html`

# Dynamic Content

- The content of an HTML element is addressed with the `value` property of its associated JavaScript object
- Example: a help box for a form

→ **SHOW** `dynValue.html` & `dynValue.js`

## iClicker question

- SHOW `dynValue.html` & `dynValue.js`
- Inside the above code, what if I change the message to HTML format? e.g.

```
var helpers = ["Your name must be in the form:  
<br/> first name, middle initial., last name", ...
```

- A. The message is shown in two lines.
- B. The `<br/>` is printed.
- C. Error message in error console.
- D. Nothing is shown in the text area.

**Answer: B**



## iClicker question

- SHOW `dynValue.html` & `dynValue.js`
- How the event handler is registered?
  - A. Assign the event handler to attribute
  - B. Assign the event handler to element property
  - C. None of the above

**Answer: A**



# Stacking Elements

- The `z-index` (in css) determines which element is in front and which are covered by the front element
- The JavaScript property associated with the `z-index` is `zIndex` (in JavaScript)
- `z-index` can be changed dynamically (by changing `zIndex`)

→ **SHOW** `stack.html` & `stack.js`

## iClicker question

- SHOW `stack.html` & `stack.js`
- Inside the above code, if image A's z-index is 10 and image B's z-index is 20, which one is displayed on the top?
- - A. Image A.
  - B. Image B.

**Answer: B**



# The End

