Changing CSS Styles via JavaScript

No readings?

Review?

- You can change the CSS styling of an element with JavaScript
- Syntax is similar to accessing an attribute:
  - document.getElementById(“*id*”).*attribute;*

    vs.
  - document.getElementById(“*id*”).style.*property*;

Review

- Property names in JavaScript replace hyphens with capital letters
- CSS properties:

```
font-size, background-color, text-decoration
```

- Properties in JavaScript:

```
fontSize, backgroundColor, textDecoration
```

Getting the Property Value

- HTML snippet:

```
<p id=“text”>
        There is some text here.
        <input type=“button”
                value=“Click” onclick=“showFontSize()”; />
</p>
```

- JavaScript file:

```
function showFontSize() {
        var fontSize =
                document.getElementById(“text”).style.fontSize;
        alert(fontSize);
}
```

- This will show the size of the font in the paragraph

Setting the Property Value

- HTML snippet:

```
<p id=“text”>
        There is some text here.
        <input type=“button”
                value=“Click” onclick=“increaseFont()”; />
</p>
```

- JavaScript file:

```
function increaseFont() {
        document.getElementById(“text”).style.fontSize = “20pt”;
}
```

- This will change the font in the paragraph to size 20 font.

Setting the Property Value

- NOTE! Whenever you set the value of a style property, you should use a string:

```
function increaseFont() {
        document.getElementById(“text”).style.fontSize = 20pt;
}
```

```
function increaseFont() {
        document.getElementById(“text”).style.fontSize = “20pt”;
}
```

Example

- Changing the color and size of text

Timers

Using Time in your Programs

- Repeat tasks every set time period.
  - Making a clock tick every second
- Do something after a delay
  - Click a button that shows an image, and then hide it after 5 seconds

Timer Functions in JavaScript

- Built in functions, like alert:
  - setTimeout – calls a function after a time delay
  - setInterval – calls a function every given time period
  - clearTimeout – stops the timer
- We will only look at the setTimeout function

setTimeout function

- Same as calling any other function with parameters:
  - setTimeout(*function*, *delay*);
  - *function* is the function to call after *delay* milliseconds has passed

```
function startTimer() {
       setTimeout(alertBox, 1000);
}

function alertBox() {
       alert(“Hello!”);
}
```

Important Note!

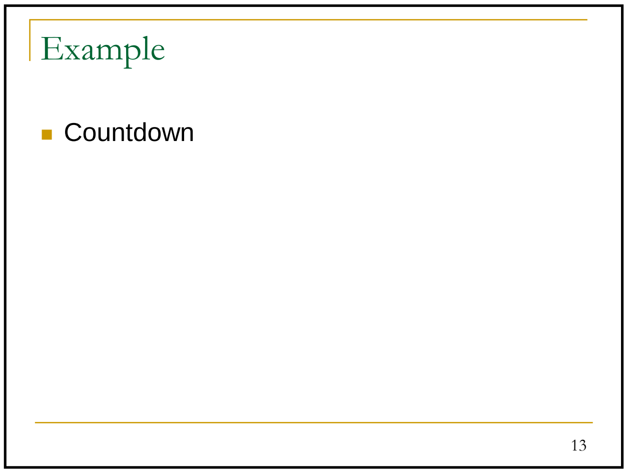
- The *function* parameter should be the name of the function you want to call with NO parentheses!

```
function startTimerGood() {
       setTimeout(alertBox, 1000);
}

function startTimerBad() {
       setTimeout(alertBox(), 1000);
}

function alertBox() {
       alert(“Hello!”);
}
```

- Why?



Example

- Countdown