JavaScript and jQuery Course

Session 10 Links, Images, Timers

Default Actions

Common HTML elements that have a default action

- <a href=url>
  - accesses that URL
- <input type=submit> <button type=submit>
  - calls the action of the form
- <input type=reset> <button type=reset>
  - resets elements to initial values

All click events

2

Cancel Default Actions

Cancel the default action

- Event object
  - Created when an event occurs
- preventDefault()
  - Method of the event object
  - Used to prevent the default action from occurring

3

Cancel Default Actions

```
<a href="http://www.google.com">Click to go to Google</a>
```

By default goes to Google

```
<a href="image.jpg" id="link">Click to do something</a>
```

We want to execute JavaScript when we click the link

```
function showImage(event) {
        // code here;
        event.preventDefault();
    }

document.getElementById('link').onclick = showImage;
```

4

Image Object

Allows us to access the properties of an image

- Create instance of image object
```
var image = new Image();
```
- Load an image in that object
```
image.src = "image_name.jpg";
```
- Set other properties
```
image.alt    = “description";
image.width  = “200";
image.height = “200";
```

5

Preload images

Caches the images for use in slide show etc

```
function preloadImages() {
        // counter
        var i = 0;

        // create object
        var imageObj = new Image();

        // set image list
        images = new Array();
        images[0] = "image1.jpg"
        images[1] = "image2.jpg"
        images[2] = "image3.jpg"
        images[3] = "image4.jpg"

        // start preloading
        for (i = 0; i < images.length; i++) {
            imageObj.src = images[i];
            alert(images[i]);
        }
```

```
window.onload = function() {
        preloadImages();
}
```

6

Preload images

Caches the images for use in book application

```
var links = document.getElementsByTagName("a");
var i, link, image;
for (i = 0; i < links.length; i++) {
    link = links[i];
    image = new Image();
    image.src = link.href;
}
```

```
<a href="images/release.jpg">
<img src="thumbnails/release.jpg" alt=""></a>
```

7

Timers

The window object allows execution of code at specified time intervals set in milliseconds.

There are 1000 milliseconds in one second.

```
setTimeout(function, milliseconds)
```
Executes a function, after waiting a specified number of milliseconds.

If the function has not already been executed, you can stop the execution by calling the clearTimeout() method –

```
clearTimeout(timeoutVariable)
```

8

Timers

`setInterval(function, milliseconds)`

Same as setTimeout(), but repeats the execution of the function continuously.

The clearInterval() method stops the executions of the function specified in the setInterval() method.

`clearInterval(intervalVariable)`

9

Clock Example

Creates a continuous clock

```
<script>
    function showTime() {
        var today = new Date();
        document.getElementById('time').innerHTML =
          today.toLocaleTimeString();
        setInterval(showTime, 1000);
    } // end showDate

    window.onload = function() {
        showTime();
    }

</script>
```

7:46:56 PM

10