

## Designing Rollovers and Slide Shows

# Objectives

- Access the images object collection
- Create an image object
- Preload images
- Create a rollover effect with a custom property
- Use anonymous functions in event handlers
- Design an interactive image gallery

2

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*

- 1

- Load high-resolution graphic images
- Create slide captions
- Create a page overlay
- Set an object's opacity
- Create and apply transition effects
- Apply anonymous functions to timedelayed commands

The screenshot shows a presentation slide titled "The Internet". The main content is a high-resolution image of the Earth from space, with a satellite in orbit. The slide is part of a presentation, as evidenced by the navigation buttons at the bottom. Callouts from the previous slide point to the image, the slide title, and the navigation buttons.

slide show images must have a class name and be marked as hypertext

```

<ul id="slides">
  <li>
    <a href="#slide01.jpg" src="#slide01_thumb.jpg" class="slide">
      alt="Assault on the International Space Station [1999]"
    </a>
    
  <li>
    alt="the atlantis docks with the iss [2001]"
    <a href="#slide02.jpg" src="#slide02_thumb.jpg" class="slide">
      alt="The Atlantis approaches the ISS [2000]"
    </a>
    

```

```
function addEvent(object, eventName, fn) {
  if (object.addEventListener) {
    object.addEventListener(eventName, fn, false);
  } else if (object.attachEvent) {
    object.attachEvent("on" + eventName, fn);
  }
}

addEvent(window, "load", setupTimeline, false);

function setupTimeline() {
}
```

- ```
Function setupSlideshow() {  
    var slides = new Array();  
  
    // populate array of slide images  
    for (var i = 0; i < document.images.length; i++) {  
        var thumb = document.images[i];  
  
        if (thumb.className == "slide" && thumb.parentNode.className == "S")  
            slides.push(thumb);  
    }  
}
```

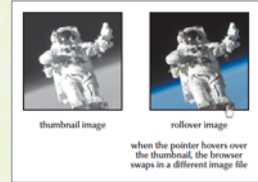
## Setting Up a Slide Show

- Referencing inline images
  - To reference the collection of inline images within a document, use the `document.images` object collection
  - To reference a specific inline image, use `document.images[idref]` where `idref` is either the index number of the image object or the text of its `id` or `name` attribute

## Creating an Image Rollover

- Rollover:** Visual effect in which an object changes appearance in response to the pointer hovering over it
  - Commonly created through CSS using the `hover` pseudo-class
- To change source of inline image from original thumbnail to rollover image, change the value of the `src` property

```
imgObject.src = rollover;
```



## Creating an Image Rollover

- Image object:** JavaScript object that stores an image and makes it available for quick retrieval
- To create an image object, run the command  

```
var imgObject = new Image();
```
- To set the dimensions of a new image object, use  

```
var imgObject = new Image(width, height);
```
- To define the source of an image object, use  

```
imgObject.src = url;
```

## Creating an Image Rollover

- Define image objects as **custom properties**
  - Object properties defined by the programmer
  - Can contain text strings, numeric values, Boolean values, or other objects
  - Advantage: They become attached to the object and can be referenced throughout the program code
  - Code to define these properties will be stored in the `createRollover()` function

## Creating an Image Rollover

- The `createRollover()` function

```
function setupSlideshow() {
  var slides = new Array();
  // populate array of slide images
  for (var i = 0; i < document.images.length; i++) {
    var thumb = document.images[i];
    if (thumb.className == "slide" && thumb.parameters.tagName == "A") {
      slides.push(thumb);
    }
  }
  for (var i = 0; i < slides.length; i++) {
    // create a rollover for each slide
    createRollover(slides[i]);
  }
}

function createRollover(thumb) {
  thumb.over = new Image();
  thumb.over.src = thumb.src;
}
```

creates a rollover for each slide

two image objects created for each thumbnail

## Creating an Image Rollover

- Use anonymous functions to swap images in response to `onmouseout` and `onmouseover` events
- To apply an anonymous function to an object event, enter

```
object.anevent = function() {
  commands
}
```

```
function createRollover(thumb) {
  thumb.over = new Image();
  thumb.over.src = thumb.src;
  thumb.onmouseover = function() {
    thumb.src = thumb.over.src;
  };
  thumb.onmouseout = function() {
    thumb.src = thumb.src;
  };
}
```

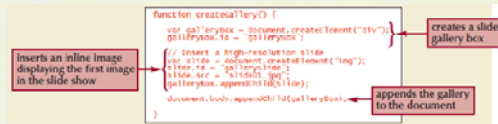
anonymous functions called by the mouseout and mouseover events

displays the over image when the pointer moves out from the slide

displays the over image when the pointer moves over the slide

## Creating an Image Gallery

- To create an application that requires minimal rewriting of the source Web page, write a function to create the gallery and then append it to the body of the current document



13

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Creating an Image Gallery

- Creating image objects for the high-resolution images:
  - Create a **custom property** to associate each thumbnail with a high-resolution version of the image
  - Preload image objects so that the slide show runs smoothly as it switches from one image to another
- Adding a slide caption:
  - Code retrieves caption text from image's alt attribute

14

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Hiding and Viewing the Slide Gallery

- To hide the gallery when clicked, add an image button to the gallery box
- To display the slide gallery by clicking any of the thumbnail images, create the showGallery() function, which will be run with the onclick event handler

```
function showGallery() {
    // Reveal the slide show
    document.getElementById("galleryBox").style.display = "block";
    // halt propagation of the click event
    return false;
}
```

- The changeSlide() function replaces the current slide image with a new slide and changes the caption

15

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Navigating Through the Slide Gallery

- Add navigation buttons to the slide gallery

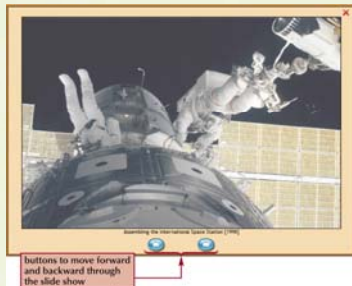
```
<p id="galleryFooter">
    <input type="image" src="back.png" />
    <input type="image" src="forward.png" />
</p>
```



16

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Navigating Through the Slide Gallery



17

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Navigating Through the Slide Gallery

- Program actions of navigation buttons to keep track of the slide number by attaching an index number to each slide and to each image as a custom property

```
for (var i = 0; i < slides.length; i++) {
    // create a thumbnail for each slide
    createThumbnail(slides[i]);
}

// attach a high-resolution image object to each slide
createHighRes(slides[i]);

if (slides.length > 0) {
    createCaption(slides[0]);
}

function createHighRes(thumb, index) {
    thumb.big = new Image();
    thumb.big.src = thumb.src.replace(".thumb", "");
    // display high-resolution image in slide gallery
    thumb.onlick = showGallery;
    // set the index of the slide
    thumb.big.index = index;
}
```

Annotations:

- Index values are based on the position of the slide in the slides array
- adds the index value of each slide as a custom property

18

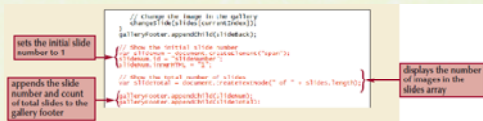
New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Navigating Through the Slide Gallery

- To display the slide number

`<span id="slideNumber">number</span>` of total

- Append the span element and the text node to the gallery footer by adding commands to the `createGallery()` function



19

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Creating a Page Overlay

- Design the page overlay effect

`<div id = "pageOverlay"></div>`

- The `createOverlay()` function



- Define the appearance of the overlay by adding new styles to the `gallery.css` style sheet

20

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Working with Opacity Values

- Makes the page overlay semitransparent so that the rest of the Web page can be dimly seen through it
- Opacity** defines the percentage to which an object is nontransparent or opaque



21

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Setting Opacity with CSS

- To set opacity value for objects under Internet Explorer, use the style

`filter: alpha(opacity = value)`

where *value* ranges from 0 (completely transparent) to 100 (completely opaque)

- To set opacity under non-IE browsers, use the style

`opacity: value`

where *value* ranges from 0 (completely transparent) to 1 (completely opaque)

22

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Setting Opacity with JavaScript

- To set opacity value for objects under Internet Explorer, use the object property

`object.style.filter = "alpha(opacity = value)";`  
where *object* is the object and *value* is the opacity level ranging from 0 to 100

- If filters have been defined for the object under IE, use

`object.filters["alpha"].opacity = value;`  
or  
`object.filters.alpha.opacity = value;`

- To set the opacity under non-IE browsers, use

`object.style.opacity = value;`

where *value* ranges from 0 to 1

23

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Creating Fade-Ins and Fade-Outs

- Transition:** Application of a visual effect
- Fade-in effect**
  - Opacity changes over time from completely transparent to completely opaque
  - Use JavaScript's `setTimeout()` method to set different opacity values at different delay times
  - Parameters of `fadeIn()` function: `objID`, `maxOpacity`, `fadeTime`, `delay`
- Fade-out effect**
  - An object changes from being opaque to completely transparent

24

New Perspectives on  
JavaScript and AJAX, 2nd Edition



## Creating Fade-Ins and Fade-Outs

```
function fadeIn(objID, maxopacity, fadeTime, delay) {
    // Calculate the interval between opacity changes
    var fadeInt = Math.round(fadeTime*1000)/maxopacity;
    // Loop up the range of opacity values
    for (var i = 0; i <= maxopacity; i++) {
        setTimeout("setOpacity('" + objID + "', " + i + "%)", delay);
        delay += fadeInt;
    }
}

function fadeOut(objID, maxopacity, fadeTime, delay) {
    // Calculate the interval between opacity changes
    var fadeInt = Math.round(fadeTime*1000)/maxopacity;
    // Loop down the range of opacity values
    for (var i = maxopacity; i >= 0; i--) {
        setTimeout("setOpacity('" + objID + "', " + i + "%)", delay);
        delay += fadeInt;
    }
}
```

gradually fades in the object

gradually fades out the object

25

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Applying a Fade-In

```
function changeSlide(slide) {
    // Set object references
    var galleryBox = document.getElementById("galleryBox");
    var oldSlide = document.getElementById("gallerySlide");
    var slideCaption = document.getElementById("slideCaption");
    var slideNum = document.getElementById("slideNumber");

    // Replace current slide with new slide
    setOpacity("gallerySlide", 0);
    var newSlide = oldSlide.cloneNode(true);
    newSlide.src = slide.big.src;
    newSlide.index = slide.big.index;
    galleryBox.replaceChild(newSlide, oldSlide);
    fadeIn("gallerySlide", 100, 0.5, 0);

    // Replace current caption with new caption
    slideCaption.innerHTML = slide.alt;

    // Update the slide number
    slideNum.innerHTML = newSlide.index + 1;
}
```

makes the current slide completely transparent

after replacing the old slide with the new slide, fades in the new slide

26

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Applying a Fade-Out

```
function createGallery(slides) {
    var galleryBox = document.createElement("div");
    galleryBox.id = "galleryBox";

    // Insert a button to close the gallery
    var galleryTitle = document.createElement("p");
    galleryTitle.id = "galleryTitle";

    var closeButton = document.createElement("input");
    closeButton.type = "image";
    closeButton.src = "galleryClose.png";
    closeButton.onclick = function() {
        fadeOut("galleryBox", 100, 0.5, 0);
        fadeOut("pageOverlay", 80, 0.5, 0);
        galleryBox.style.display = "none";
        document.getElementById("pageOverlay").style.display = "none";
    }
}
```

fades out the gallery box and page overlay

27

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Creating Fade-Ins and Fade-Outs

- Using anonymous functions with timed commands
  - To delay the implementation of a command block, place the block in an anonymous function nested within `setTimeout()` method:
 

```
setTimeout(function() {
                        commands;
                    }, delay);
```
  - To repeatedly run a command block at specified intervals, use an anonymous function and the `setInterval()` method:
 

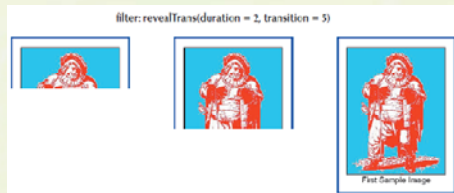
```
setInterval(function() {
                        commands;
                    }, interval);
```

28

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Creating Fade-Ins and Fade-Outs

- Internet Explorer supports a library of transition filters designed to dynamically change the appearance of an object over a specified time interval



29

New Perspectives on  
JavaScript and AJAX, 2nd Edition