DHTML: Oggetti e Collezioni

Sommario

- Introduzione
- Riferimenti a Oggetti
- Collezioni all e children
- Stili dinamici
- Posizionamento dinamico
- Collezione frames
- Oggetto navigator
- Riepilogo

Obiettivi

- Usare il modello dinamico degli oggetti HTML per creare pagine dinamiche
- Capire la gerarchia di oggetti dinamici
- Usare le collezioni all e children
- Usare stili e posizionamenti dinamici
- Usare la collezione frames collection
- Usare l'oggetto navigator

Introduzione

- Il modello degli Oggetti Dinamici HTML
  - Permette all'autore di controllare la presentazione delle pagine
  - Permette l'accesso a tutti gli elementi delle pagine
- Una pagina Web
  - contiene elementi, moduli, frame, tabelle
  - rappresentati secondo una gerarchia di oggetti
- Scripting
  - È l'attività per recuperare e modificare le proprietà e gli attributi

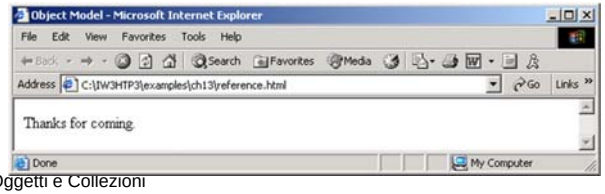
Riferimento a Oggetti

- Il modo più semplice per riferirsi ad un elemento è usare l'attributo `id` dell'elemento
  - Gli attributi XHTML sono considerati proprietà che possono essere manipolate dalle azioni dello scripting


```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 13.1: reference.html  -->
6  <!-- Object Model Introduction -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Object Model</title>
11
12       <script type = "text/javascript">
13          <!--
14          function start()
15          {
16             alert( pText.innerText );
17             pText.innerText = "Thanks for coming.";
18          }
19          // -->
20       </script>
21
22    </head>
```



```
23
24    <body onload = "start()">
25       <p id = "pText">Welcome to our Web page!</p>
26    </body>
27 </html>
```



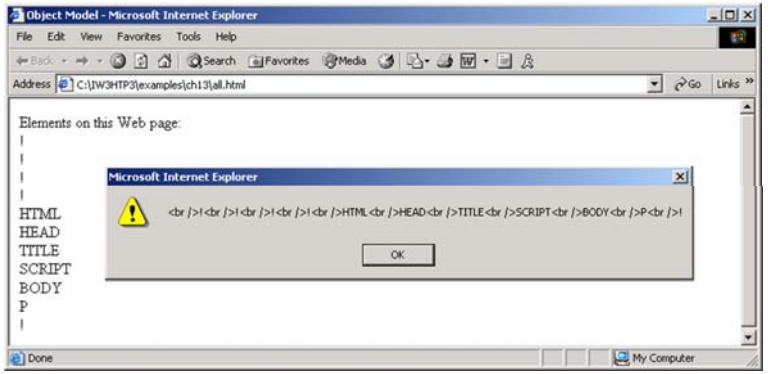
Collezioni all e children

- Collezione
  - Array di oggetti correlati su una pagina
  - `all`
    - Tutti gli elementi XHTML in un documento
  - `children`
    - Lo specifico elemento contiene gli elementi figli di quell'elemento

```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig 13.2: all.html       -->
6  <!-- Using the all collection -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Object Model</title>
11
12       <script type = "text/javascript">
13          <!--
14          var elements = "";
15
16          function start()
17          {
18             for ( var loop = 0; loop < document.all.length; ++loop )
19                elements += "<br />" + document.all[ loop ].tagName;
20
21             pText.innerHTML += elements;
22             alert( elements );
23          }
24          // -->
```



```
25       </script>
26    </head>
27
28    <body onload = "start()">
29       <p id = "pText">Elements on this Web page:</p>
30    </body>
31 </html>
```



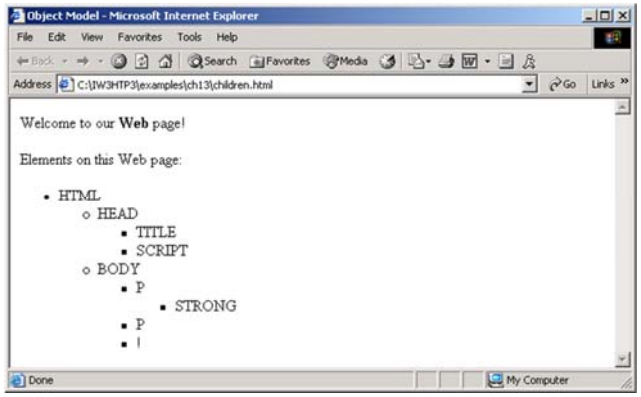

```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig 13.3: children.html -->
6  <!-- The children collection -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Object Model</title>
11
12       <script type = "text/javascript">
13          <!--
14          var elements = "<ul>";
15
16          function child( object )
17          {
18             var loop = 0;
19
20             elements += "<li>" + object.tagName + "<ul>";
21
```



```
22             for ( loop = 0; loop < object.children.length; loop++ )
23             {
24                if ( object.children[ loop ].children.length )
25                   child( object.children[ loop ] );
26                else
27                   elements += "<li>" +
28                      object.children[ loop ].tagName +
29                      "</li>";
30             }
31
32             elements += "</ul>" + "</li>";
33          }
34          // -->
35       </script>
36    </head>
37
```

```
38    <body onload = "child( document.all[ 4 ] );
39       myDisplay.outerHTML += elements;
40       myDisplay.outerHTML += '</ul>';">
41
42       <p>Welcome to our <strong>Web</strong> page!</p>
43
44       <p id = "myDisplay">
45          Elements on this Web page:
46       </p>
47
48    </body>
49 </html>
```


Esecuzione

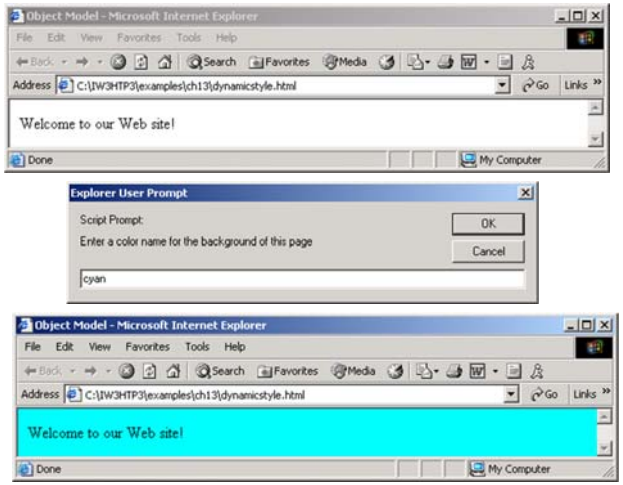


Stili Dinamici

- Lo stile degli elementi può essere gestito dinamicamente
- È possibile modificare i valori dell'attributo `class`


```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 13.4: dynamicstyle.html -->
6  <!-- Dynamic Styles              -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Object Model</title>
11
12       <script type = "text/javascript">
13          <!--
14          function start()
15          {
16             var inputColor = prompt(
17                "Enter a color name for the " +
18                "background of this page", "" );
19             document.body.style.backgroundColor = inputColor;
20          }
21          // -->
22       </script>
23    </head>
```

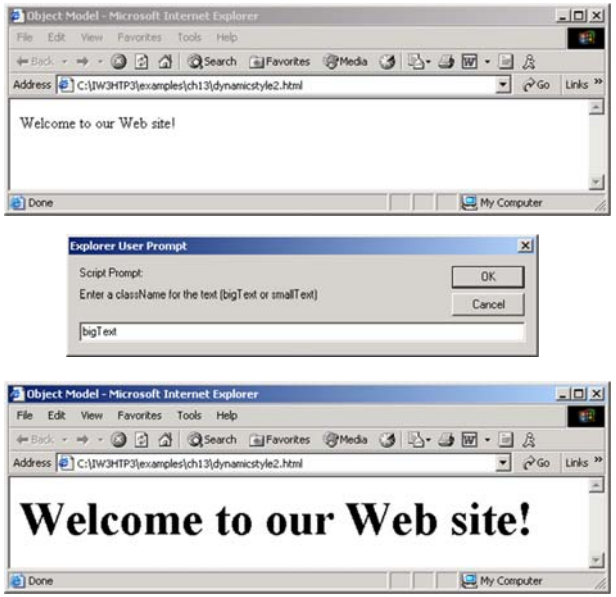
```
24
25    <body onload = "start()">
26       <p>Welcome to our Web site!</p>
27    </body>
28 </html>
```




```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 13.5: dynamicstyle2.html -->
6  <!-- More Dynamic Styles           -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Object Model</title>
11
12       <style type = "text/css">
13
14          .bigText   { font-size: 3em;
15                       font-weight: bold }
16
17          .smallText { font-size: .75em }
18
19       </style>
20
```



```
21       <script type = "text/javascript">
22          <!--
23          function start()
24          {
25             var inputClass = prompt(
26                "Enter a className for the text " +
27                "(bigText or smallText)", "" );
28             pText.className = inputClass;
29          }
30          // -->
31       </script>
32    </head>
33
34    <body onload = "start()">
35       <p id = "pText">Welcome to our Web site!</p>
36    </body>
37 </html>
```

Posizionamento Dinamico

- Gli elementi XHTML possono essere posizionati nella pagina mediante azioni di scripting
  - Il posizionamento può essere sia `absolute` che `relative`
  - L'elemento può essere spostato usando le proprietà CSS `top`, `left`, `right` e `bottom`

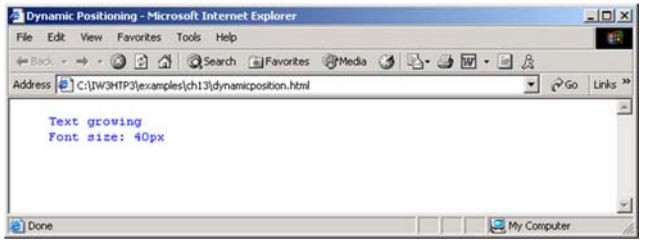

```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 13.6: dynamicposition.html -->
6  <!-- Dynamic Positioning             -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Dynamic Positioning</title>
11
12       <script type = "text/javascript">
13          <!--
14          var speed = 5;
15          var count = 10;
16          var direction = 1;
17          var firstLine = "Text growing";
18          var fontStyle = [ "serif", "sans-serif", "monospace" ];
19          var fontStylecount = 0;
20
21          function start()
22          {
23             window.setInterval( "run()", 100 );
24          }
25
```

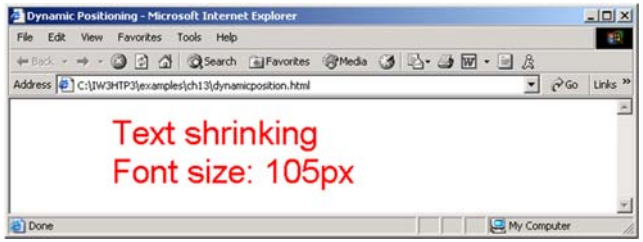


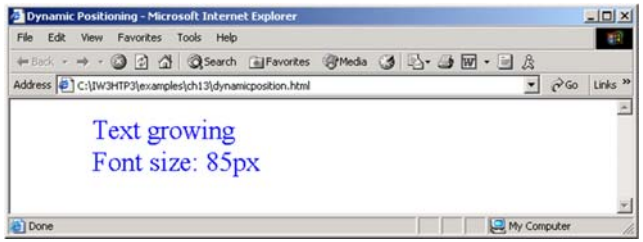
```
26          function run()
27          {
28             count += speed;
29
30             if ( ( count % 200 ) == 0 ) {
31                speed *= -1;
32                direction = !direction;
33
34                pText.style.color =
35                   ( speed < 0 ) ? "red" : "blue" ;
36                firstLine =
37                   ( speed < 0 ) ? "Text shrinking" : "Text growing";
38                pText.style.fontFamily =
39                   fontStyle[ ++fontStylecount % 3 ];
40             }
41
42             pText.style.fontSize = count / 3;
43             pText.style.left = count;
44             pText.innerHTML = firstLine + "<br /> Font size: " +
45                               count + "px";
46          }
47          // -->
48       </script>
49    </head>
50
```



```
51    <body onload = "start()">
52          <p id = "pText" style = "position: absolute; left: 0;
53                                   font-family: serif; color: blue">
54          Welcome!</p>
55    </body>
56 </html>
```







Collezioni di frame

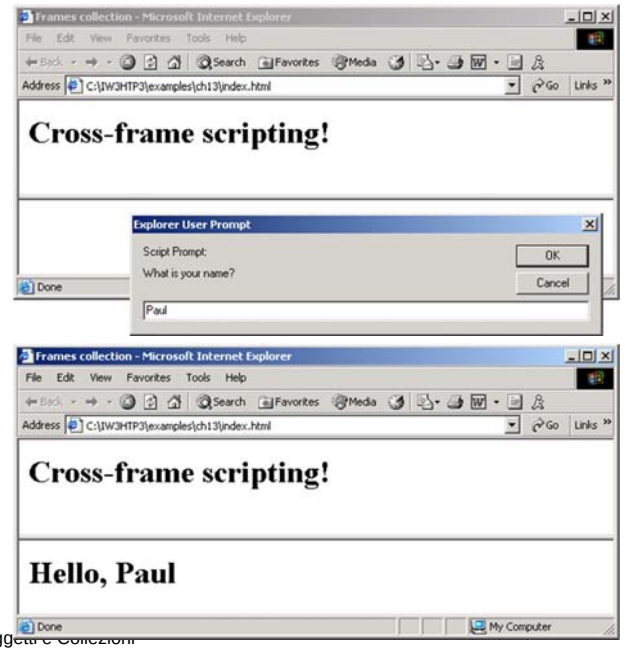
- È possibile gestire elementi e oggetti nei frame usando la collezione `frames`


```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
4
5  <!-- Fig. 13.7: index.html       -->
6  <!-- Using the frames collection -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>Frames collection</title>
11    </head>
12
13    <frameset rows = "100, *">
14       <frame src = "top.html" name = "upper" />
15       <frame src = "" name = "lower" />
16    </frameset>
17
18 </html>
```



```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 13.8: top.html   -->
6  <!-- Cross-frame scripting -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>The frames collection</title>
11
12       <script type = "text/javascript">
13          <!--
14          function start()
15          {
16             var text = prompt( "What is your name?", "" );
17             parent.frames( "lower" ).document.write(
18                "<h1>Hello, " +  text + "</h1>" );
19          }
20          // -->
21       </script>
22    </head>
23
```

```
24    <body onload = "start()">
25       <h1>Cross-frame scripting!</h1>
26    </body>
27 </html>
```

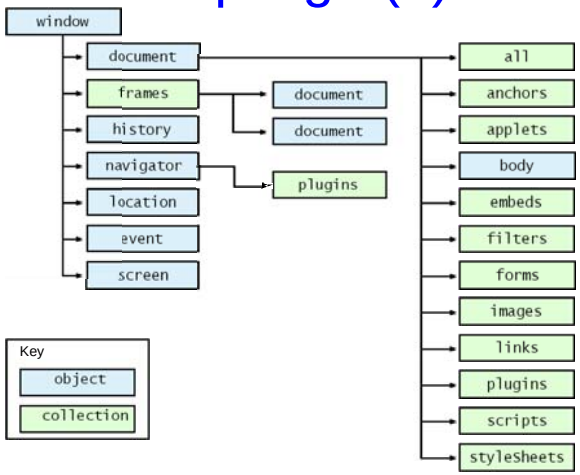
Oggetto navigator

- Permette di riconoscere il browser con cui si accede alla pagina
  - Netscape, Mozilla, Microsoft's Internet Explorer, Safari, …


```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig 13.9: navigator.html   -->
6  <!-- Using the navigator object -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10       <title>The navigator Object</title>
11
12       <script type = "text/javascript">
13          <!--
14          function start()
15          {
16             if (navigator.appName=="Microsoft Internet Explorer")
17             {
18                if ( navigator.appVersion.substring( 1, 0 ) >= "4" )
19                   document.location = "newIEversion.html";
20                else
21                   document.location = "oldIEversion.html";
22             }
23             else
24                document.location = "NSversion.html";
25          }
```

```
26          // -->
27       </script>
28    </head>
29
30    <body onload = "start()">
31       <p>Redirecting your browser to the appropriate page,
32       please wait...</p>
33    </body>
34 </html>
```

Modello Oggetti Dinamici: Riepilogo (1)



Modello Oggetti Dinamici: Riepilogo (2)

Object or collection	Description
Objects	
`window`	Represents the browser window and provides access to the `document` object contained in the `window`. If the `window` contains frames a separate `window` object is created automatically for each frame, to provide access to the `document` rendered in the frame. Frames are considered to be subwindows in the browser.
`document`	Represents the XHTML document rendered in a `window`. The `document` object provides access to every element in the XHTML document and allows dynamic modification of the XHTML document.
`body`	Provides access to the `body` element of an XHTML document.
`history`	Keeps track of the sites visited by the browser user. The object provides a script programmer with the ability to move forward and backward through the visited sites, but for security reasons does not allow the actual site URLs to be manipulated.
`navigator`	Contains information about the Web browser, such as the name of the browser, the version of the browser, the operating system on which the browser is running and other information that can help a script writer customize the user's browsing experience.
`location`	Contains the URL of the rendered document. When this object is set to a new URL, the browser immediately switches (navigates) to the new location.
`event`	Can be used in an event handler to obtain information about the event that occurred (e.g., the mouse *x-y* coordinates during a mouse event).
`screen`	Contains information about the computer screen for the computer on which the browser is running. Information such as the width and height of the screen in pixels can be used to determine the size at which elements should be rendered in a Web page.

Fig. 13.11 Objects in the Internet Explorer 6 Object Model.

Modello Oggetti Dinamici: Riepilogo (3)

Object or collection	Description
Collections	
`all`	Many objects have an `all` collection that provides access to every element contained in the object. For example, the `body` object's `all` collection provides access to every element in the `body` element of an XHTML document.
`anchors`	Collection contains all the anchor elements (`a`) that have a `name` or `id` attribute. The elements appear in the collection in the order they were defined in the XHTML document.
`applets`	Contains all the `applet` elements in the XHTML document. Currently, the most common `applet` elements are Java applets.
`embeds`	Contains all the `embed` elements in the XHTML document.
`forms`	Contains all the `form` elements in the XHTML document. The elements appear in the collection in the order they were defined in the XHTML document.
`frames`	Contains `window` objects that represent each frame in the browser window. Each frame is treated as its own subwindow.
`images`	Contains all the `img` elements in the XHTML document. The elements appear in the collection in the order they were defined in the XHTML document.
`links`	Contains all the anchor elements (`a`) with an `href` property. This collection also contains all the `area` elements that represent links in an image map.

Fig. 13.11 Objects in the Internet Explorer 6 Object Model.

Modello Oggetti Dinamici: Riepilogo (4)

Object or collection	Description
`plugins`	Like the `embeds` collection, this collection contains all the `embed` elements in the XHTML document.
`scripts`	Contains all the `script` elements in the XHTML document.
`styleSheets`	Contains `styleSheet` objects that represent each `style` element in the XHTML document and each style sheet included in the XHTML document via `link`.

Fig. 13.11 Objects in the Internet Explorer 6 Object Model.