



BÀI 6: JQUERY HIỆU ỨNG (PHẦN 1)

- ◎ Tổng quan hiệu ứng trong jquery
- ◎ Các hiệu trong jquery
 - Hiển thị và ẩn các phần tử
 - Toggle các phần tử
 - Các phương thức jQuery Effect
 - Animation thao tác tay
 - Animation tự động hóa
 - Rollover với một Mouse Event



❑ jQuery cung cấp một giao diện đơn giản để thực hiện các loại hiệu ứng tuyệt vời đa dạng. Các phương thức jQuery cho phép chúng ta áp dụng nhanh chóng các hiệu ứng phổ biến với cấu hình tối thiểu.

❑ Cú pháp

```
[selector].show( speed, [callback] );
```

- ❖ **speed** – Một chuỗi biểu diễn một trong 3 tốc độ đã định trước ("slow", "normal", hoặc "fast") hoặc số mili giây để chạy hiệu ứng (ví dụ: 1000).
- ❖ **callback** – tham số tùy ý này biểu diễn một hàm để được thực thi bất cứ khi nào hiệu ứng hoàn thành; thực thi một lần cho mỗi hiệu ứng.

- ❑ Cú pháp đơn giản cho phương thức **hide()** trong jQuery.

```
[selector].hide( speed, [callback] );
```

- ❑ **speed** – Một chuỗi biểu diễn một trong 3 tốc độ đã định trước ("slow", "normal", hoặc "fast") hoặc số mili giây để chạy hiệu ứng (ví dụ: 1000).
- ❑ **callback** – tham số tùy ý này biểu diễn một hàm để được thực thi bất cứ khi nào hiệu ứng hoàn thành; thực thi một lần cho mỗi hiệu ứng.
- ❑ Ví dụ

```
<html>
  <head>
    <title>The jQuery Example</title>
    <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.min.js"></script>

    <script type="text/javascript" language="javascript">
      $(document).ready(function() {

        $("#show").click(function () {
          $(".mydiv").show( 1000 );
        });

        $("#hide").click(function () {
          $(".mydiv").hide( 1000 );
        });
      });
    </script>

    <style>
      .mydiv{ margin:10px;padding:12px; border:2px solid #666; width:100px; height:100px; }
    </style>
  </head>

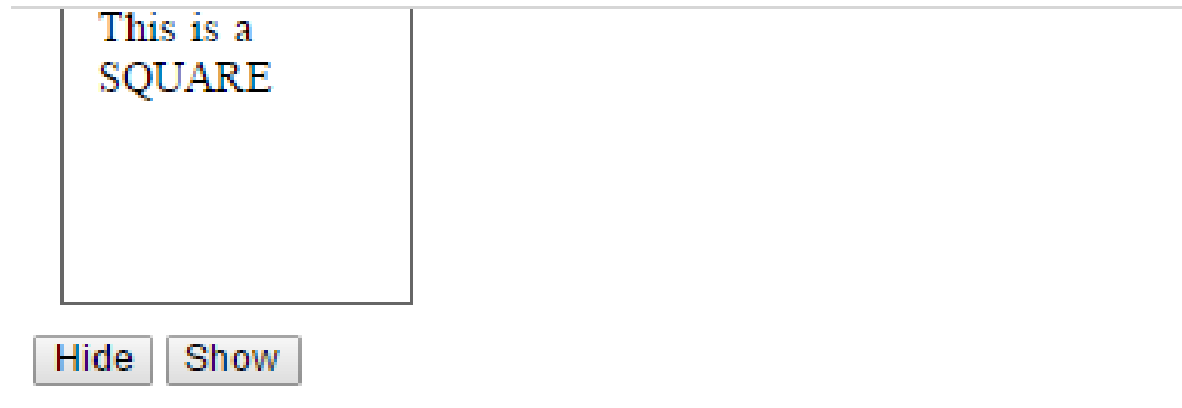
  <body>

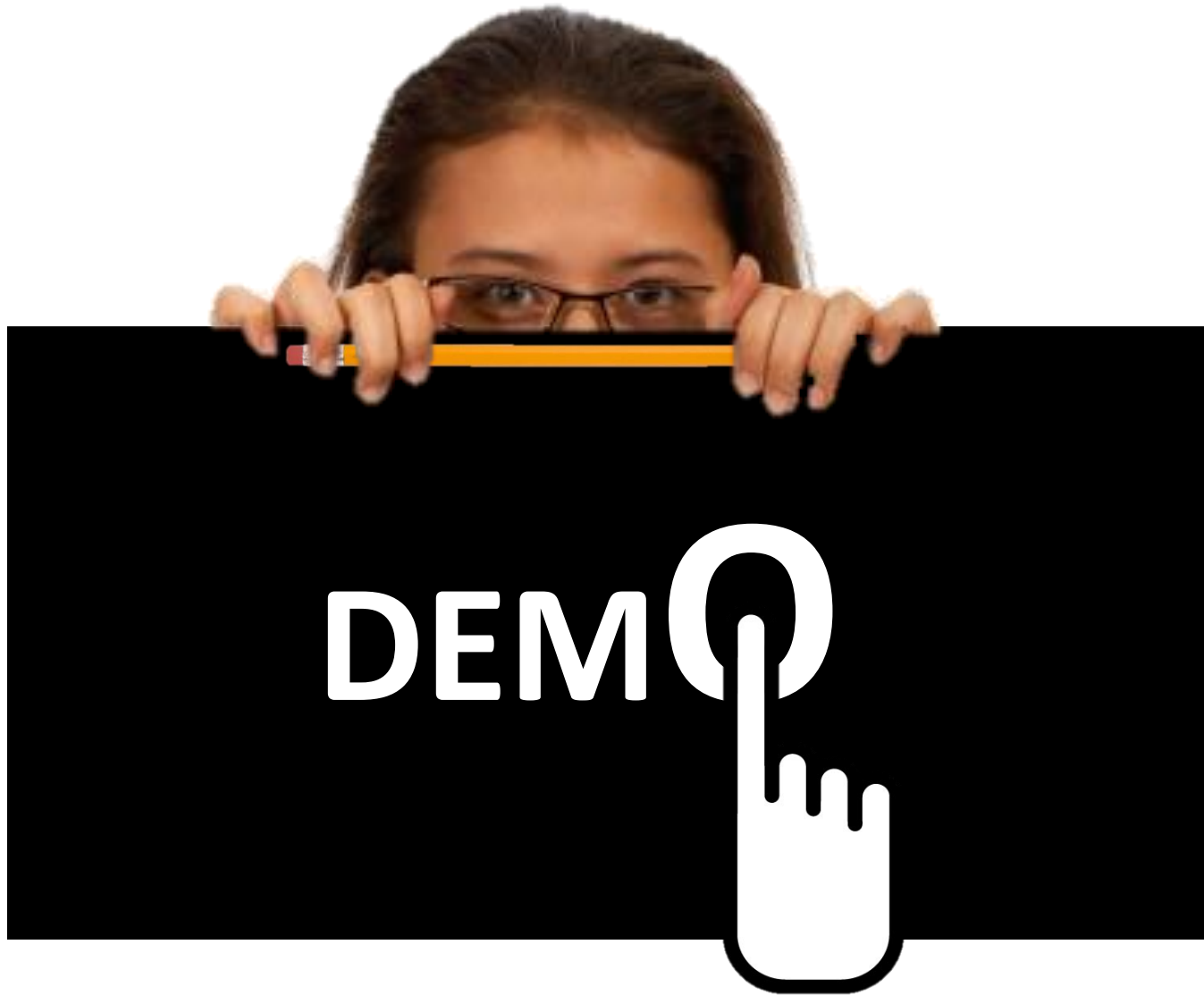
    <div class="mydiv">
      This is a SQUARE
    </div>

    <input id="hide" type="button" value="Hide" />
    <input id="show" type="button" value="Show" />

  </body>
</html>
```

❑ Kết quả:





- ❑ jQuery cung cấp các phương thức để toggle trạng thái hiển thị của các phần tử giữa hiển thị và ẩn. Nếu phần tử khi khởi tạo là hiển thị, nó sẽ được ẩn; nếu là ẩn, nó sẽ được hiển thị.
- ❑ Cú pháp đơn giản cho một phương thức **toggle()** trong jQuery:

`[selector].toggle([speed][callback]);`

- ❑ **speed** – Một chuỗi biểu diễn một trong 3 tốc độ đã định trước ("slow", "normal", hoặc "fast") hoặc số mili giây để chạy hiệu ứng (ví dụ: 1000).
- ❑ **callback** – tham số tùy ý này biểu diễn một hàm để được thực thi bất cứ khi nào hiệu ứng hoàn thành; thực thi một lần cho mỗi hiệu ứng.
- ❑ Ví dụ

```
<html>
  <head>
    <title>The jQuery Example</title>
    <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/

    <script type="text/javascript" language="javascript">

      $(document).ready(function() {
        $(".clickme").click(function(event){
          $(".target").toggle('slow', function(){
            $(".log").text('Transition Complete');
          });
        });
      });
    </script>

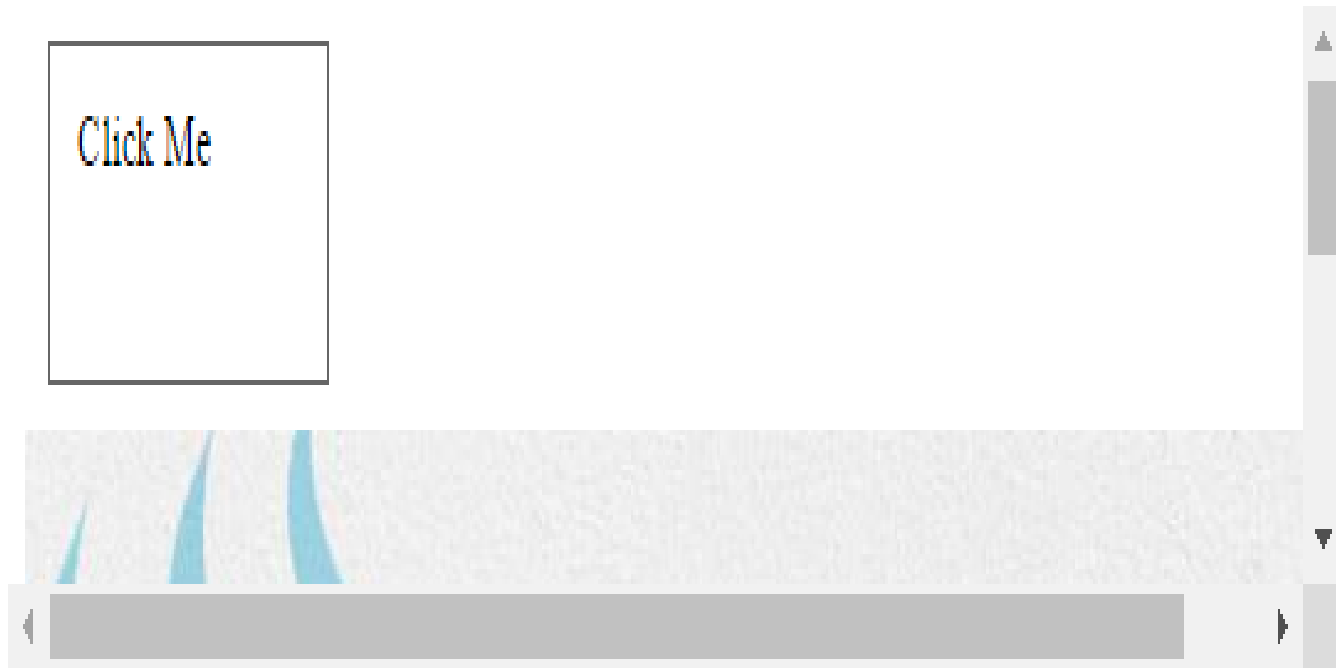
    <style>
      .clickme{ margin:10px;padding:12px; border:2px solid #666; width:100px; height
    </style>

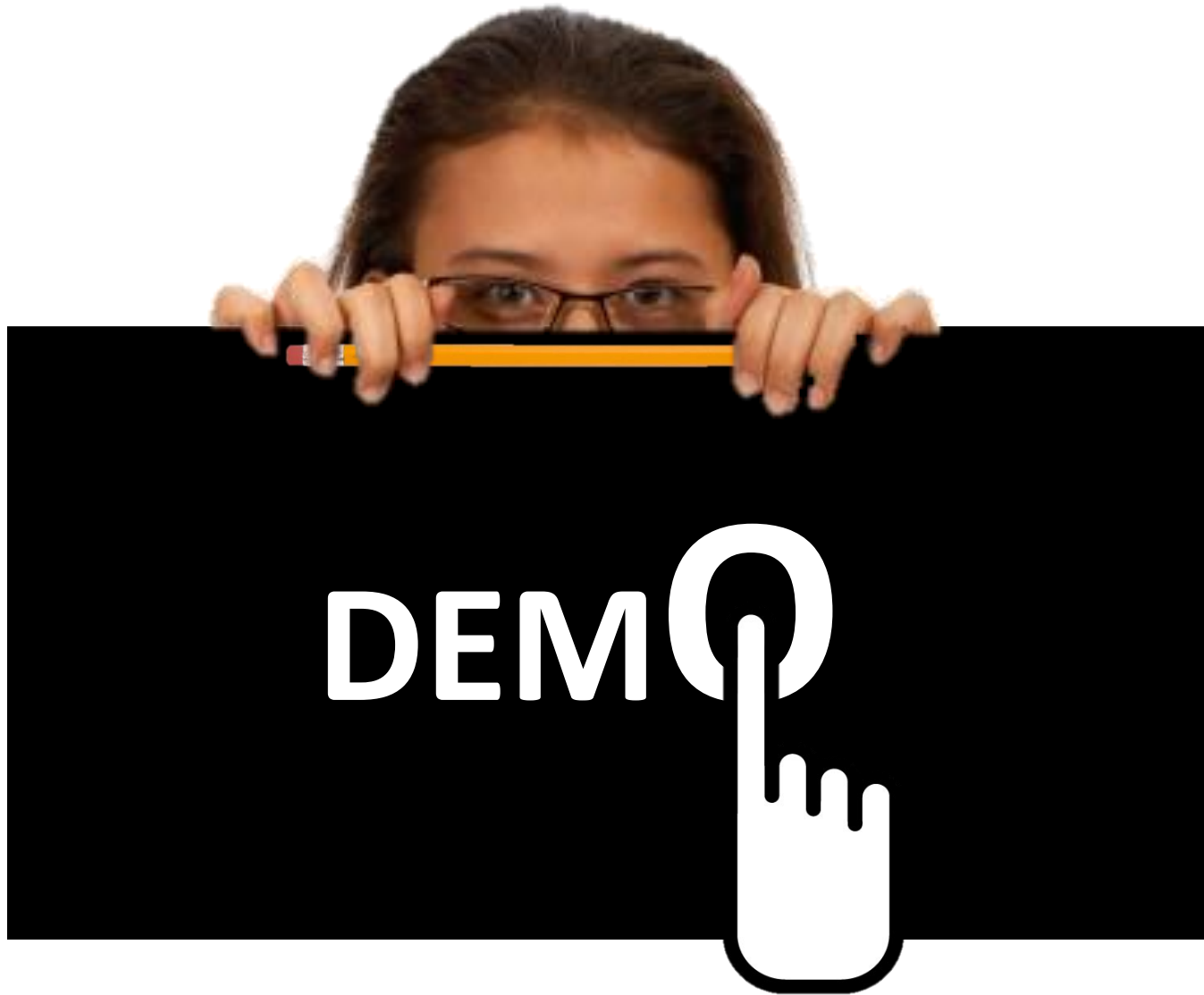
  </head>

  <body>

    <div class="content">
      <div class="clickme">Click Me</div>
      <div class="target">
        
      </div>
      <div class="log"></div>
    </div>
  </body>
```

❑ Kết quả:





- ❑ Bảng dưới liệt kê tất cả phương thức cơ bản để tạo các kiểu hiệu ứng khác nhau:

STT	Phương thức & Miêu tả
1	animate(params, [duration, easing, callback]) Một hàm để tạo các hiệu ứng custom
2	fadeIn(speed, [callback]) <i>Fade in</i> tất cả các phần tử đã so khớp bởi việc điều chỉnh <i>opacity</i> (độ chắn sáng) và kích hoạt một callback tùy ý sau khi hoàn thành.
3	fadeOut(speed, [callback]) <i>Fade out</i> tất cả phần tử đã so khớp bởi điều chỉnh <i>opacity</i> về 0, sau đó thiết lập hiển thị về "none" và kích hoạt một callback tùy ý sau khi hoàn thành
4	fadeTo(speed, opacity, callback) <i>Fade</i> độ chắn sáng của tất cả phần tử đã so khớp tới một <i>opacity</i> đã cho và kích hoạt một callback tùy ý sau khi hoàn thành.
5	hide() Ẩn mỗi phần tử trong tập hợp đã so khớp nếu chúng đã hiển thị

6	hide(speed, [callback]) Ẩn tất cả phần tử đã so khớp bởi sử dụng một hiệu ứng đẹp và kích hoạt một callback tùy ý sau khi hoàn thành
7	show() Hiển thị mỗi phần tử trong tập hợp đã so khớp nếu chúng bị ẩn
8	show(speed, [callback]) Hiển thị tất cả phần tử đã so khớp bởi sử dụng một hiệu ứng đẹp và kích hoạt một callback tùy ý sau khi hoàn thành
10	slideDown(speed, [callback]) Bộc lộ tất cả phần tử đã so khớp bởi điều chỉnh chiều cao của chúng và kích hoạt một callback tùy ý sau khi hoàn thành
11	slideToggle(speed, [callback]) Toggle sự nhìn thấy của tất cả phần tử đã so khớp bởi điều chỉnh chiều cao và kích hoạt một callback tùy ý sau khi hoàn thành

12	slideUp(speed, [callback]) Ẩn tất cả phần tử đã so khớp bởi điều chỉnh chiều cao của chúng và kích hoạt một callback tùy ý sau khi hoàn thành.
13	stop([clearQueue, gotoEnd]) Dừng tất cả các hiệu ứng đang chạy hiện tại trên tất cả phần tử đã xác định
14	toggle() Toggle sự hiển thị mỗi phần tử trong tập hợp các phần tử đã so khớp
15	toggle(speed, [callback]) Toggle sự hiển thị mỗi phần tử trong tập hợp các phần tử đã so khớp bởi sử dụng một hiệu ứng đẹp và kích hoạt một callback tùy ý sau khi hoàn thành
16	toggle(switch) Toggle sự hiển thị mỗi phần tử trong tập hợp đã so khớp dựa trên việc chuyển mạch giữa: true là hiển thị tất cả phần tử, false là ẩn tất cả phần tử
17	jQuery.fx.off Vô hiệu hóa toàn bộ tất cả hiệu ứng



BÀI 6: JQUERY HIỆU ỨNG (PHẦN 2)

- ❑ Thực hiện một animation đơn giản sử dụng các thuộc tính đối tượng DOM và các hàm JavaScript như sau.
- ❑ Sử dụng hàm **getElementById()** để nhận một đối tượng DOM và sau đó gán nó tới một biến Global là **imgObj**.
- ❑ Định nghĩa một hàm khởi tạo **init()** để khởi tạo **imgObj**, thiết lập các thuộc tính **position** và **left** của nó.
- ❑ Cuối cùng, đang gọi hàm **moveRight()** để tăng khoảng cách trái bởi 10 pixel. Có thể cũng thiết lập nó một giá trị âm để di chuyển nó sang cạnh trái.

❑ Ví dụ

```
<html>

  <head>
    <title>JavaScript Animation</title>

    <script type="text/javascript">
      <!--
        var imgObj = null;

        function init(){
          imgObj = document.getElementById('myImage');
          imgObj.style.position= 'relative';
          imgObj.style.left = '0px';
        }

        function moveRight(){
          imgObj.style.left = parseInt(imgObj.style.left) + 10 + 'px';
        }

        window.onload =init;
      //-->
    </script>

  </head>

  <body>

    <form>
      
      <p>Click button below to move the image to right</p>
      <input type="button" value="Click Me" onclick="moveRight();" />
    </form>

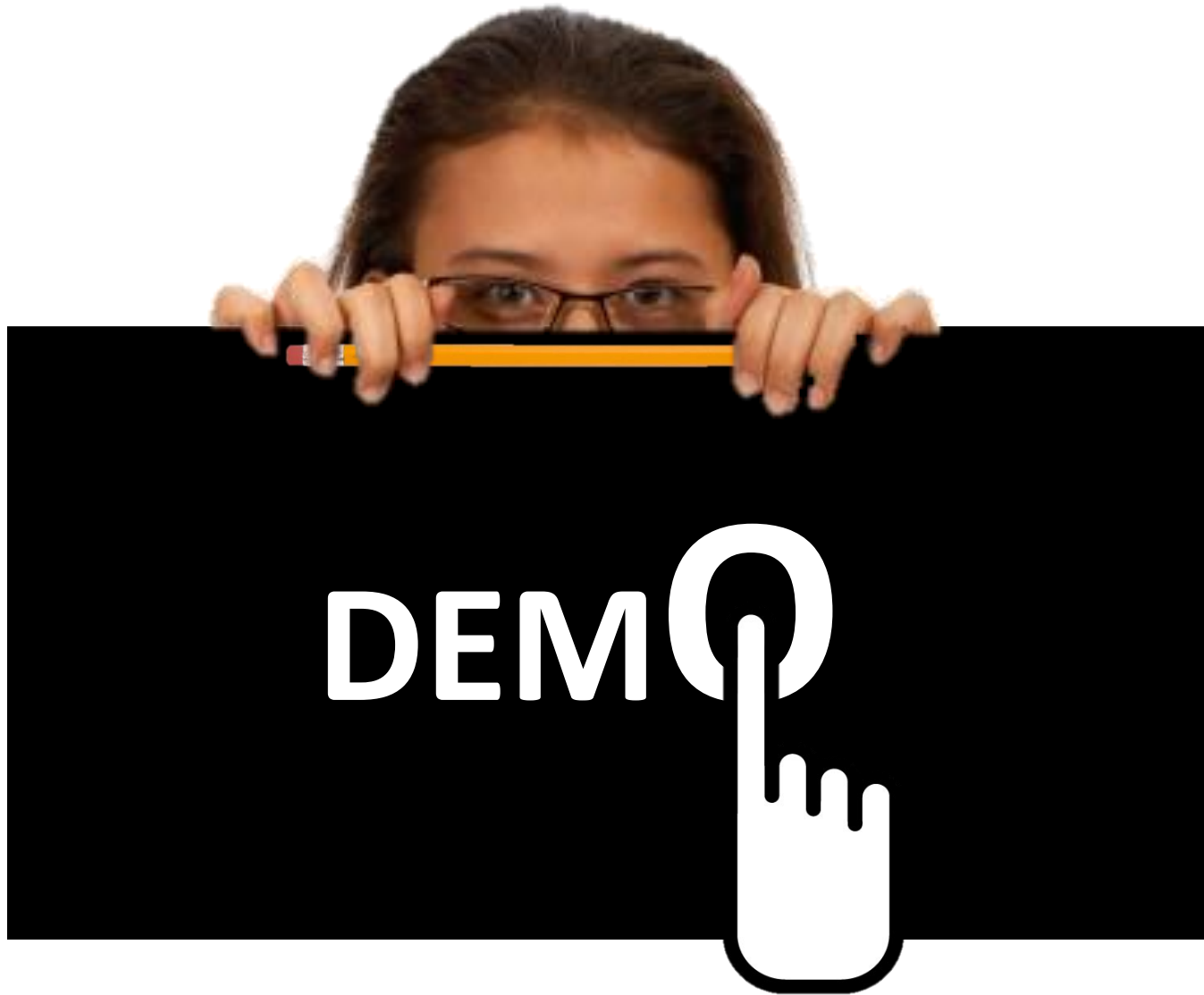
  </body>
</html>
```

❑ Kết quả



Click button below to move the image to right

Click Me



- ❑ Chúng ta có thể tự động hóa tiến trình này bởi sử dụng hàm **setTimeout()** như sau:
- ❑ Dưới đây, chúng tôi thêm nhiều phương thức hơn.
- ❑ Hàm **moveRight()** đang gọi hàm **setTimeout()** để thiết lập vị trí của *imgObj*.
- ❑ Thêm một hàm mới là **stop()** để xóa bộ bấm giờ được thiết lập bởi hàm **setTimeout()** và để thiết lập đối tượng tại vị trí khởi tạo.

Code

```
<html>

<head>
  <title>JavaScript Animation</title>

  <script type="text/javascript">
    <!--
      var imgObj = null;
      var animate ;

      function init(){
        imgObj = document.getElementById('myImage');
        imgObj.style.position= 'relative';
        imgObj.style.left = '0px';
      }

      function moveRight(){
        imgObj.style.left = parseInt(imgObj.style.left) + 10 + 'px';
        animate = setTimeout(moveRight,20); // call moveRight in 20msec
      }

      function stop(){
        clearTimeout(animate);
        imgObj.style.left = '0px';
      }
    </script>
  </head>

  <body>
    <div>
      <img alt="myImage" id="myImage" style="display: block; margin: 0 auto; width: 100px; height: 100px; background-color: #ccc; border: 1px solid #000;"/>
    </div>
  </body>
</html>
```

```
        window.onload =init;
    //-->
</script>

</head>

<body>

    <form>
        
        <p>Click the buttons below to handle animation</p>
        <input type="button" value="Start" onclick="moveRight();" />
        <input type="button" value="Stop" onclick="stop();" />
    </form>

</body>
</html>
```

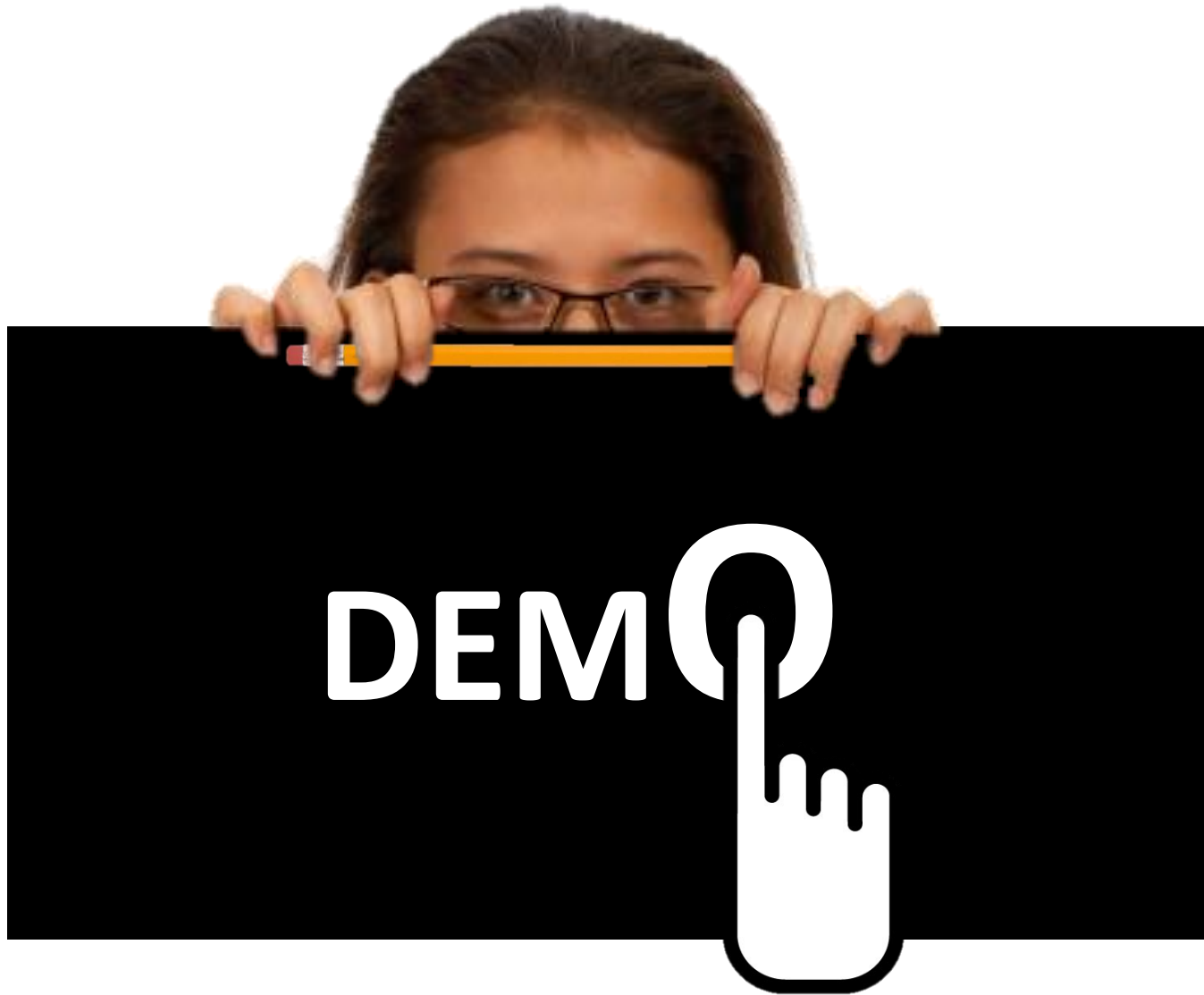
❑ Kết quả



Click the buttons below to handle animation

Start

Stop



- ❑ Quan sát những gì đang được sử dụng trong ví dụ này:
- ❑ Tại thời điểm tải trang này, lệnh if kiểm tra cho sự tồn tại của đối tượng Image. Nếu đối tượng Image này không có sẵn, khối này sẽ không được thực thi.
- ❑ **Image()** constructor tạo và tải lại một đối tượng Image mới gọi là **image1**.
- ❑ Thuộc tính src được gán tên của image file ngoại vi gọi là ./images/html.gif.
- ❑ Tương tự, chúng ta tạo đối tượng **image2** và gán ./images/http.gif trong đối tượng này.
- ❑ Dấu # vô hiệu hóa đường link để mà trình duyệt không cố gắng tới URL khi được click. Link này là một hình ảnh.

- ❑ Phương thức xử lý lỗi **onMouseOver** được kích hoạt khi người dùng di chuyển chuột qua link (hình ảnh đó), và **onMouseOut** event handler được kích hoạt khi người dùng di chuyển chuột ra khỏi link đó.
- ❑ Khi di chuyển chuột qua hình ảnh, HTTP image thay đổi từ hình ảnh đầu tiên tới hình ảnh thứ hai. Khi di chuyển chuột ra khỏi hình ảnh, thì hình ảnh ban đầu được hiển thị.
- ❑ Khi di chuyển chuột ra khỏi link, hình ảnh khởi tạo đầu tiên `html.gif` sẽ tái xuất hiện trên màn hình.

Code

```
<html>

<head>
  <title>Rollover with a Mouse Events</title>

  <script type="text/javascript">
    <!--
      if(document.images){
        var image1 = new Image(); // Preload an image
        image1.src = "../images/html.gif";
        var image2 = new Image(); // Preload second image
        image2.src = "../images/http.gif";
      }
    <!-->
  </script>

</head>

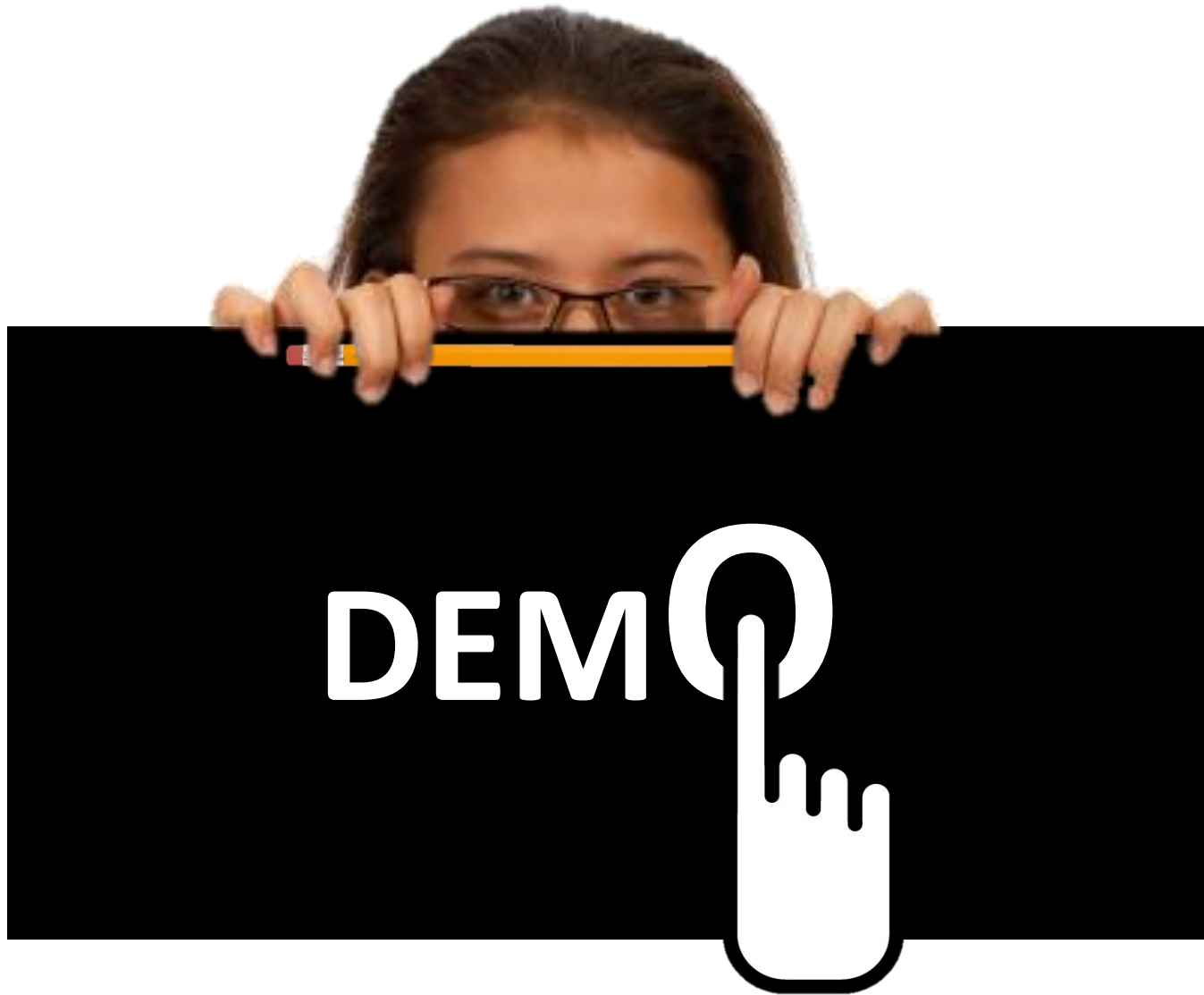
<body>
  <p>Move your mouse over the image to see the result</p>

  <a href="#" onMouseOver="document.myImage.src=image2.src;" onMouseOut="document.myImage.src=image1.src;">
    
  </a>
</body>
</html>
```

❑ Kết quả

Move your mouse over the image to see the result





- Tổng quan hiệu ứng trong jquery
- Các hiệu trong jquery
 - Hiển thị và ẩn các phần tử
 - Toggle các phần tử
 - Các phương thức jQuery Effect
 - Animation thao tác tay
 - Animation tự động hóa
 - Rollover với một Mouse Event





Cảm ơn