



## **BÀI 2: JAVASCRIPT MẢNG – SỰ KIỆN (PHẦN 1)**

- ◎ Sử dụng mảng trong javascript
- ◎ Sử dụng các sự kiện trong javascript
  - Sự kiện onclick
  - Sự kiện onsubmit
  - Onmouseover
  - Onmouseout
  - Html 5 sự kiện tiêu chuẩn



- ❑ Các **mảng** đối tượng cho lưu trữ nhiều giá trị trong một biến duy nhất. Nó lưu trữ một bộ sưu tập tuần tự cố định kích thước của các yếu tố cùng loại.
- ❑ Một mảng được sử dụng để lưu trữ một tập hợp các dữ liệu, nhưng nó thường là hữu ích hơn để nghĩ về một mảng như là một bộ sưu tập của các biến cùng loại.
- ❑ **cú pháp**

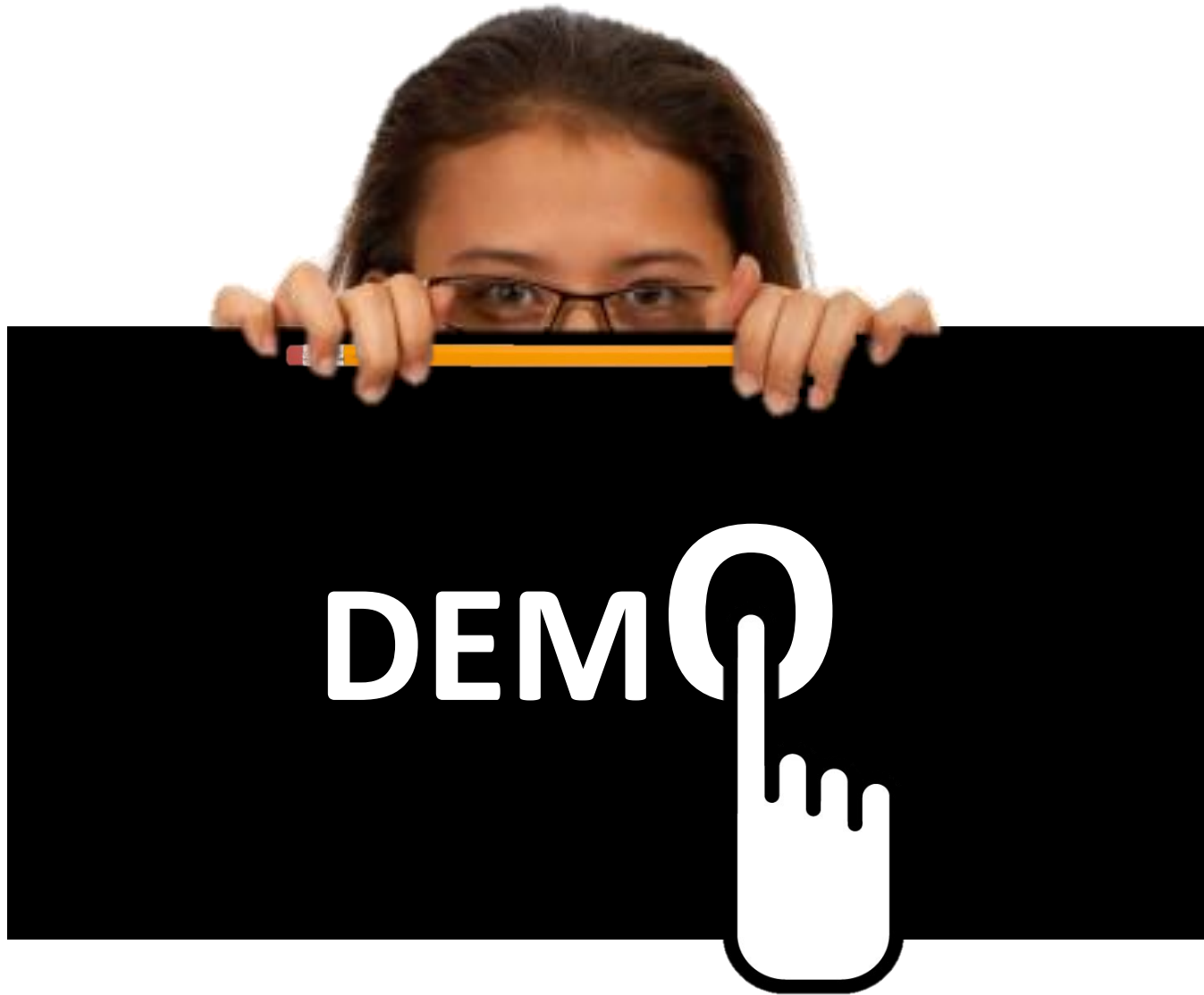
```
var fruits = new Array( "apple", "orange", "mango" );
```

- ❑ Có thể tạo mảng bằng cách gán các giá trị như sau:

```
var fruits = [ "apple", "orange", "mango" ];
```

- ❑ Sử dụng số thứ tự để truy cập và thiết lập giá trị bên trong một mảng như sau:

```
fruits[0] is the first element  
fruits[1] is the second element  
fruits[2] is the third element
```



- ❑ Dưới đây là danh sách các phương pháp của đối tượng Array cùng với mô tả của họ.

Phương thức	Miêu tả
<code>concat()</code>	Trả về một mảng mới bao gồm mảng này kết hợp với các mảng khác và/hoặc giá trị khác.
<code>every()</code>	Trả về true nếu mỗi phần tử trong mảng này thỏa mãn hàm kiểm tra đã cho.
<code>filter()</code>	Tạo một mảng mới với tất cả các phần tử của mảng này, mà hàm lọc đã cho trả về true.
<code>forEach()</code>	Gọi một hàm cho mỗi phần tử trong mảng.
<code>indexOf()</code>	Trả về chỉ mục đầu tiên (thấp nhất) của một phần tử trong mảng tương đương với giá trị đã cho, hoặc -1 nếu không được tìm thấy.
<code>join()</code>	Kết hợp tất cả phần tử trong một mảng vào trong một chuỗi.

<code>lastIndexOf()</code>	Trả về chỉ mục cuối cùng (lớn nhất) của một phần tử trong mảng tương đương với giá trị đã cho, hoặc -1 nếu không được tìm thấy.
<code>map()</code>	Tạo một mảng mới với các kết quả của việc gọi một hàm đã cho trên mỗi phần tử của mảng này.
<code>pop()</code>	Gỡ bỏ phần tử cuối cùng từ một mảng và trả về phần tử đó.
<code>push()</code>	Thêm một hoặc nhiều phần tử tới phần cuối của một mảng và trả về độ dài mới của mảng.
<code>reduce()</code>	Áp dụng một hàm đồng thời với hai giá trị của mảng (từ trái qua phải) khi để giảm nó tới một giá trị đơn.
<code>reduceRight()</code>	Áp dụng một hàm đồng thời với hai giá trị của mảng (từ phải qua trái) khi để giảm nó tới một giá trị đơn.
<code>reverse()</code>	Đảo ngược thứ tự của các phần tử của một mảng – Đầu tiên trở thành cuối cùng và cuối cùng trở thành đầu tiên.

<code>shift()</code>	Gỡ bỏ phần tử đầu tiên từ một mảng và trả về phần tử đó.
<code>slice()</code>	Extract – trích một khu vực của một mảng và trả về một mảng mới.
<code>some()</code>	Trả về true nếu có ít nhất một phần tử trong mảng này thỏa mãn hàm kiểm tra đã cho.
<code>toSource()</code>	Biểu diễn code nguồn của một đối tượng.
<code>sort()</code>	Sắp xếp các phần tử của một mảng.
<code>splice()</code>	Thêm và/hoặc gỡ bỏ các phần tử từ một mảng.
<code>toString()</code>	Trả về một chuỗi biểu diễn mảng đó và các phần tử của nó.
<code>unshift()</code>	Thêm một hoặc nhiều phần tử tới phần đầu của một mảng và trả về độ dài mới của mảng.





## **BÀI 2: JAVASCRIPT MẢNG – SỰ KIỆN (PHẦN 2)**

- ❑ Tương tác của JavaScript với HTML được xử lý thông qua các sự kiện xảy ra khi người sử dụng hoặc các trình duyệt khiến một trang.
- ❑ Khi tải trang web, nó được gọi là một sự kiện. Khi người dùng nhấp vào một nút, nhấp chuột đó cũng là một sự kiện. Các ví dụ khác bao gồm các sự kiện như nhấn phím bất kỳ, đóng cửa sổ, thay đổi kích thước một cửa sổ.

❑ Đây là loại sự kiện thường xuyên nhất được sử dụng trong đó xảy ra khi người dùng nhấp vào nút trái của chuột của mình. Có thể đặt xác nhận của bạn, cảnh báo, vv, đối với loại hình sự kiện này.

❑ Ví dụ

```
<html>
  <head>

    <script type="text/javascript">
      <!--
        function sayHello() {
          alert("Hello World")
        }
      //-->
    </script>

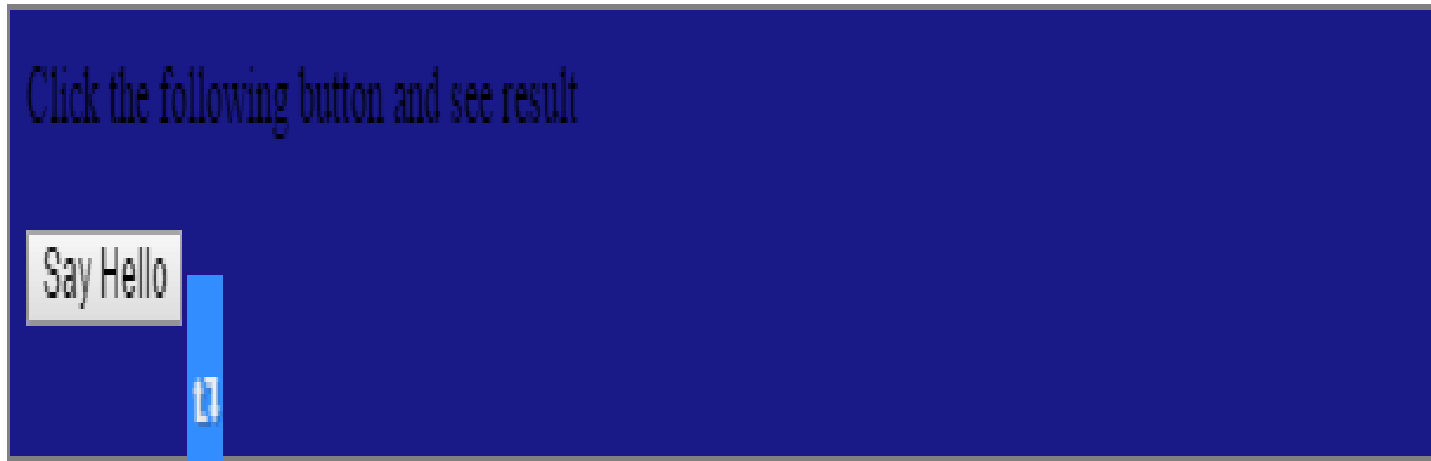
  </head>

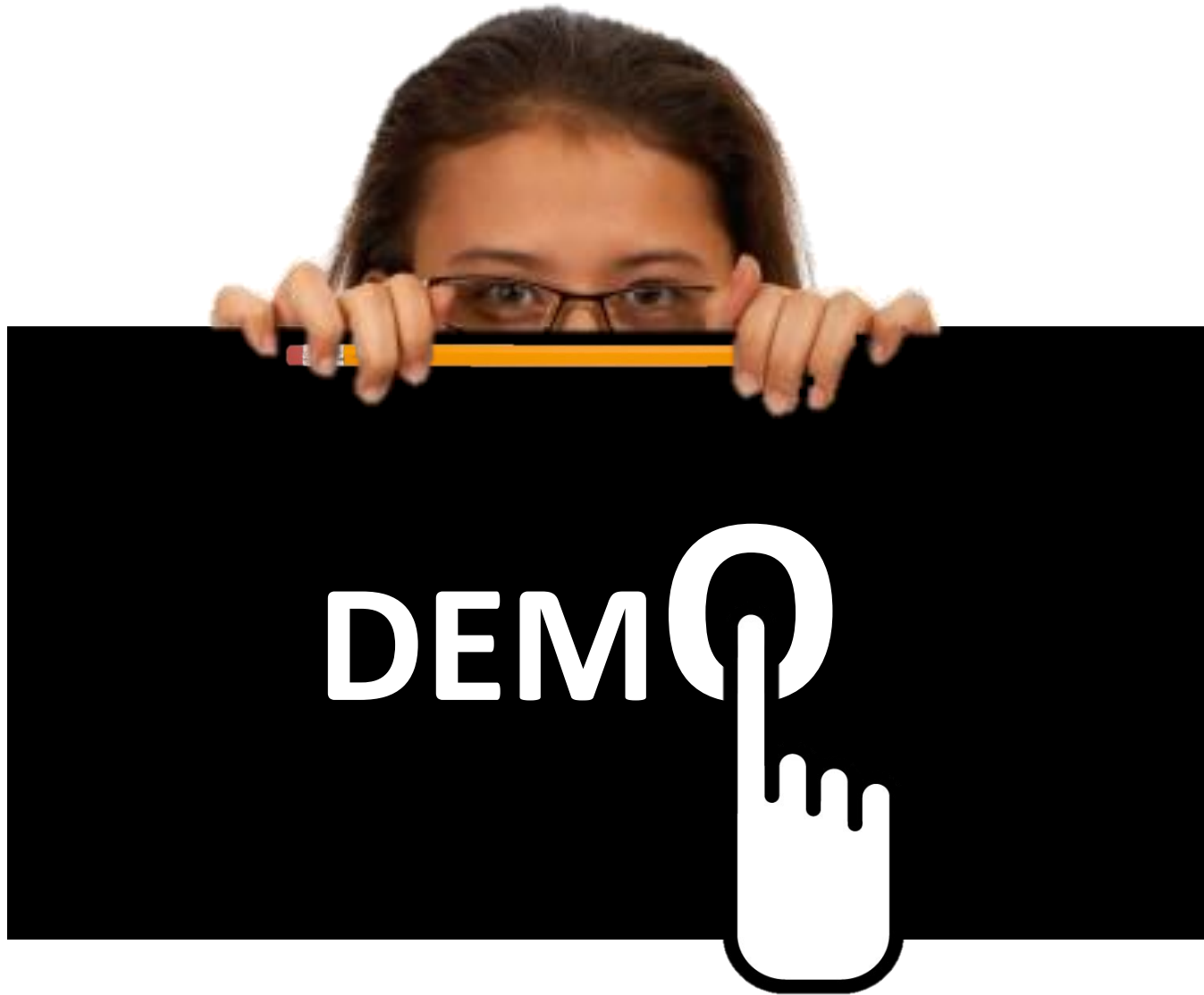
  <body>
    <p>Click the following button and see result</p>

    <form>
      <input type="button" onclick="sayHello()" value="Say Hello" />
    </form>

  </body>
</html>
```

❑ Kết quả:





- ❑ **onsubmit** là một sự kiện xảy ra khi bạn cố gắng để gửi một hình thức. Bạn có thể đặt hình thức xác nhận của bạn chống lại loại sự kiện này.
- ❑ **Ví dụ:** sau đây cho thấy làm thế nào để sử dụng onsubmit. Ở đây chúng ta đang kêu gọi một **validate ()** chức năng trước khi gửi một dữ liệu mẫu để các máy chủ web. Nếu **validate ()** chức năng trả về true, hình thức sẽ được gửi lên, nếu không nó sẽ không gửi dữ liệu.

```

<html>
  <head>

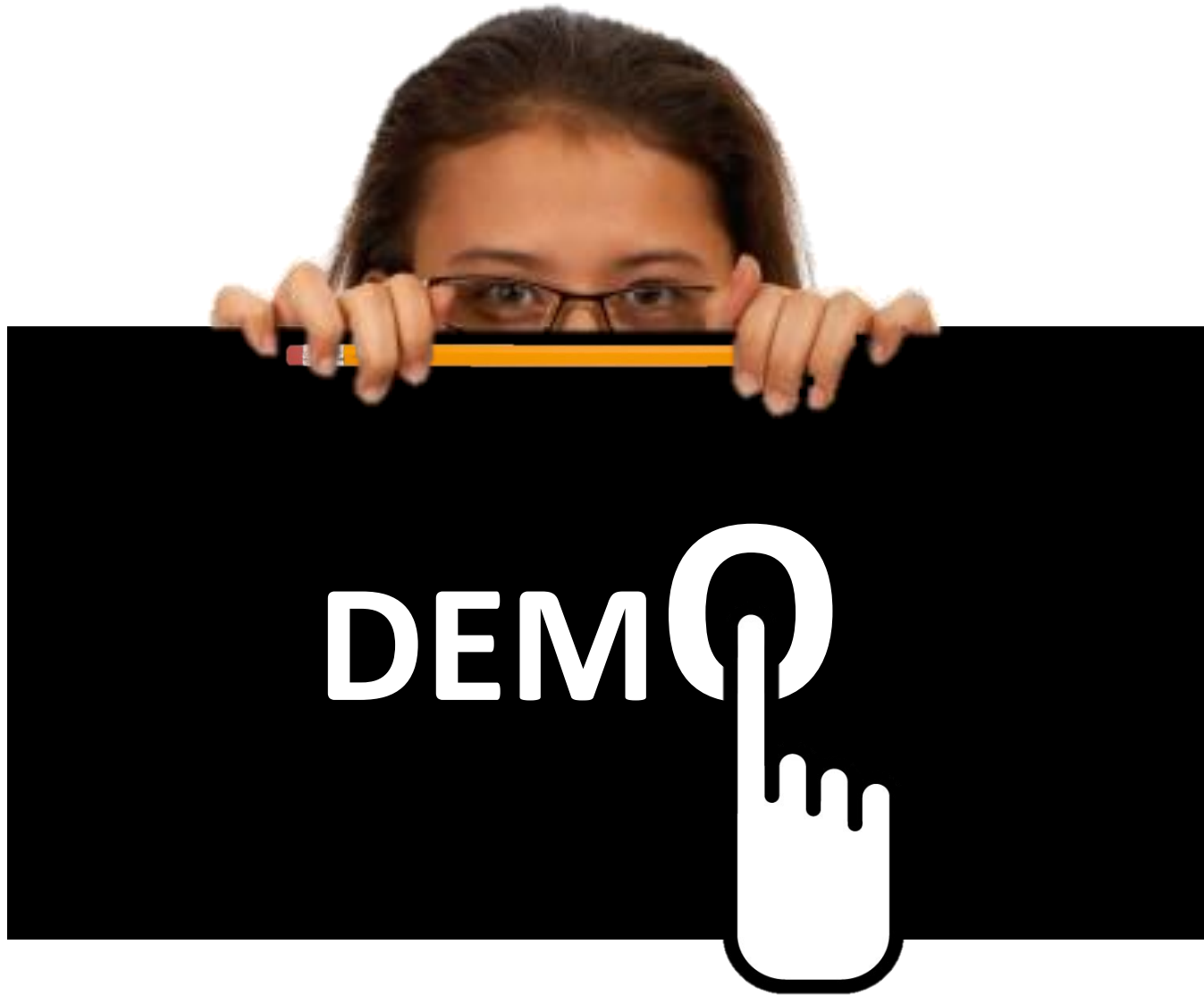
    <script type="text/javascript">
      <!--
        function validation() {
          all validation goes here
          .....
          return either true or false
        }
      //-->
    </script>

  </head>
  <body>

    <form method="POST" action="t.cgi" onsubmit="return validate()">
      .....
      <input type="submit" value="Submit" />
    </form>

  </body>
</html>

```





- ❑ Hai loại sự kiện sẽ giúp bạn tạo ra các hiệu ứng đẹp mắt với những hình ảnh hoặc thậm chí với văn bản là tốt. Các **onmouseover** sự kiện kích hoạt khi bạn mang chuột qua bất kỳ phần tử và các **onmouseout** kích hoạt khi bạn di chuyển chuột ra khỏi phần tử đó.

## ❑ Ví dụ:

```
<html>
<head>

  <script type="text/javascript">
    <!--
      function over() {
        document.write ("Mouse Over");
      }

      function out() {
        document.write ("Mouse Out");
      }

    //-->
  </script>

</head>
<body>
  <p>Bring your mouse inside the division to see the result:</p>

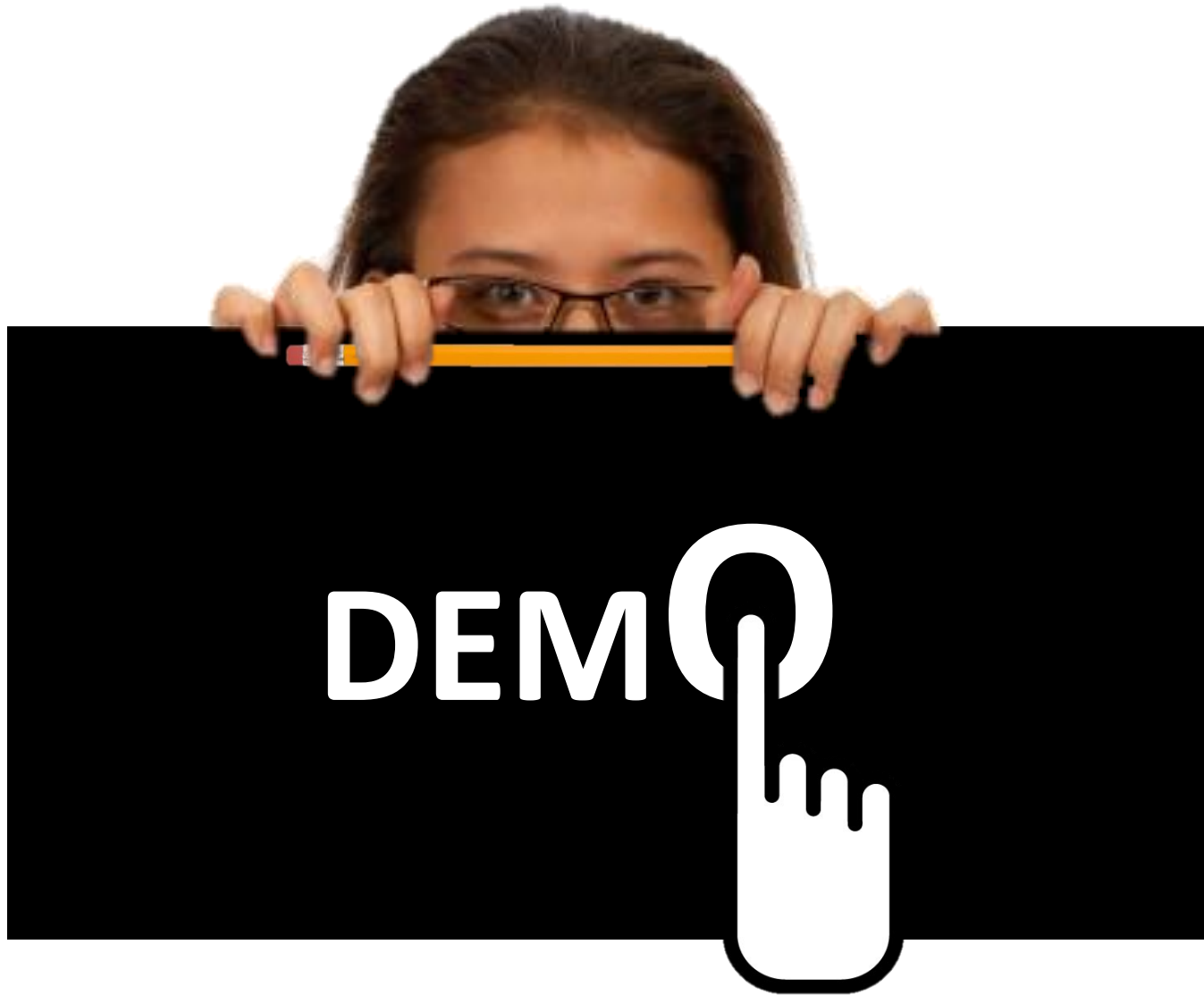
  <div onmouseover="over()" onmouseout="out()">
    <h2> This is inside the division </h2>
  </div>

</body>
</html>
```

❑ Kết quả:

Bring your mouse inside the division to see the result:

**This is inside the division**



❑ Dưới đây kịch bản chỉ là một chức năng Javascript để được thực thi dựa trên sự kiện đó.

đặc tính	Giá trị	Sự miêu tả
offline	kịch bản	Kích hoạt khi tài liệu đi offline
Onabort	kịch bản	Kích hoạt trên một sự kiện hủy bỏ
onafterprint	kịch bản	Kích hoạt sau khi tài liệu được in
onbeforeunload	kịch bản	Kích hoạt trước khi tải tài liệu
onbeforeprint	kịch bản	Kích hoạt trước khi tài liệu được in
onblur	kịch bản	Kích hoạt khi cửa sổ mất tập trung
oncanplay	kịch bản	Kích hoạt khi phương tiện truyền thông có thể bắt đầu chơi, nhưng sức có dừng lại để đệm
oncanplaythrough	kịch bản	Kích hoạt khi phương tiện truyền thông có thể được chơi đến cùng, mà không dừng lại để đệm

oncanplaythrough	kích bản	Kích hoạt khi phương tiện truyền thông có thể được chơi đến cùng, mà không dừng lại để đệm
onchange	kích bản	Kích hoạt khi một yếu tố thay đổi
trong một cái nhấp chuột	kích bản	Kích hoạt trên một click chuột
oncontextmenu	kích bản	Kích hoạt khi một trình đơn ngữ cảnh được kích hoạt
ondblclick	kích bản	Kích hoạt trên một con chuột nhấp đúp
ondrag	kích bản	Kích hoạt khi một phần tử được kéo
ondragend	kích bản	Kích hoạt vào cuối của một hoạt động kéo
ondragenter	kích bản	Kích hoạt khi một phần tử đã được kéo đến một mục tiêu thả hợp lệ

onselect	kích bản	Kích hoạt khi một phần tử được chọn
onstalled	kích bản	Kích hoạt khi có một lỗi trong khi tìm dữ liệu truyền thông
onstorage	kích bản	Kích hoạt khi một tài liệu
onsubmit	kích bản	Kích hoạt khi một mẫu được gửi
onsuspend	kích bản	Kích hoạt khi trình duyệt đã được lấy dữ liệu phương tiện truyền thông, nhưng dừng lại trước khi toàn bộ tập tin phương tiện truyền thông đã lấy
ontimeupdate	kích bản	Kích hoạt khi phương tiện truyền thông thay đổi vị trí chơi của nó
onundo	kích bản	Kích hoạt khi một tài liệu thực hiện một undo
unload	kích bản	Kích hoạt khi người dùng rời khỏi tài liệu
onvolumechange	kích bản	Kích hoạt khi phương tiện truyền thông thay đổi âm lượng, cũng có khi âm lượng được thiết lập để "tắt tiếng"
onwaiting	kích bản	Kích hoạt khi phương tiện truyền thông đã ngừng chơi, nhưng dự kiến sẽ tiếp tục

- Sử dụng mảng trong javascript
- Sử dụng các sự kiện trong javascript
  - Sự kiện onclick
  - Sự kiện onsubmit
  - Onmouseover
  - Onmouseout
  - HTML 5 Sự kiện tiêu chuẩn





**Cảm ơn**