



BÀI 5: JQUERY EVENT (PHẦN 1)

- ⊙ Sự kiện trong jquery
- ⊙ Một số sự kiện cơ bản
- ⊙ Closure và một số vấn đề liên quan



- ❑ Chúng tôi có khả năng để tạo ra các trang web động bằng cách sử dụng các sự kiện. Sự kiện là những hành động có thể được phát hiện bởi ứng dụng Web.
- ❑ Sau đây là những sự kiện ví dụ:
 - ❖ Một cú nhấn chuột
 - ❖ Một tải trang web
 - ❖ Lấy chuột trên một phần tử
 - ❖ Nộp dạng HTML
 - ❖ Một tổ hợp phím trên bàn phím

- ❑ Sử dụng jQuery Event Model, chúng tôi có thể thiết lập xử lý sự kiện trên các yếu tố DOM với **bind ()** phương pháp như sau:

❑ Code



```
<html>
  <head>
    <title>The jQuery Example</title>
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>

    <script type="text/javascript" language="javascript">
      $(document).ready(function() {
        $('div').bind('click', function( event ){
          alert('Hi there!');
        });
      });
    </script>

    <style>
      .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}
    </style>

  </head>

  <body>

    <p>Click on any square below to see the result:</p>

    <div class="div" style="background-color:blue;">ONE</div>
    <div class="div" style="background-color:green;">TWO</div>
    <div class="div" style="background-color:red;">THREE</div>

  </body>
</html>
```

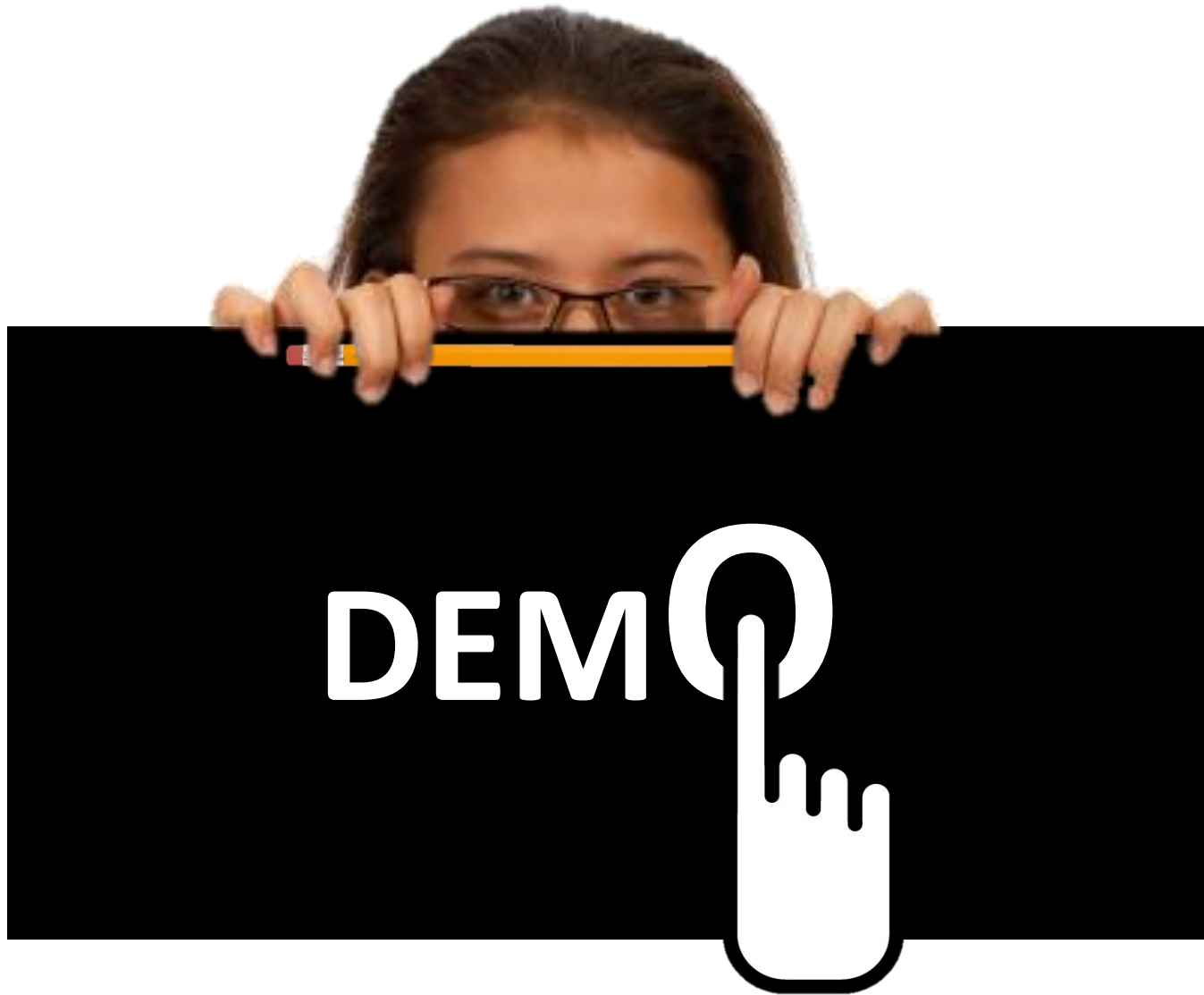
❑ Kết quả:

Click on any square below to see the result:

ONE

TWO

THREE



- ❑ Mã này sẽ gây ra các yếu tố phân chia để đáp ứng với các sự kiện nhấp chuột; khi người dùng nhấp bên trong bộ phận này sau đó, cảnh báo sẽ được hiển thị.
- ❑ Điều này sẽ tạo ra kết quả như sau:

Click on any square below to see the result:

ONE

TWO

- ❑ Cú pháp đầy đủ của bind () lệnh này như sau:

```
selector.bind( eventType[, eventData], handler)
```

- ❑ Sau đây là mô tả của các thông số -

- ❖ **eventType** - Một chuỗi có chứa một loại sự kiện JavaScript, chẳng hạn như nhấp chuột hoặc trình. Tham khảo các phần tiếp theo cho một danh sách đầy đủ các loại sự kiện.
- ❖ **eventData** - Đây là thông số tùy chọn là một bản đồ dữ liệu đó sẽ được thông qua để xử lý sự kiện.
- ❖ **xử lý** - Một chức năng để thực hiện mỗi lần sự kiện này được kích hoạt.

❑ Loại bỏ các xử lý sự kiện

- ❖ Thông thường, khi một xử lý sự kiện được thiết lập, nó vẫn có hiệu lực cho đến hết cuộc đời của trang. Có thể có một nhu cầu khi bạn muốn loại bỏ xử lý sự kiện.
- ❖ jQuery cung cấp **unbind()** lệnh để loại bỏ một xử lý sự kiện xuất cảnh. Cú pháp của unbind () như sau:

```
selector.unbind(eventType, handler)  
  
or  
  
selector.unbind(eventType)
```

❑ Sau đây là mô tả của các thông số:

❖ **eventType** - Một chuỗi có chứa một loại sự kiện JavaScript, chẳng hạn như nhấp chuột hoặc trình. Tham khảo các phần tiếp theo cho một danh sách đầy đủ các loại sự kiện.

❖ **xử lý** - Nếu được cung cấp, xác định người nghe cụ thể đó là để được gỡ bỏ.

Số TT	Loại sự kiện & Mô tả
1	mở hồ Xảy ra khi các yếu tố mất tập trung.
2	thay đổi Xảy ra khi các yếu tố thay đổi.
3	nhấp chuột Xảy ra khi một cú click chuột.
4	dblclick Xảy ra khi một con chuột nhấp đúp chuột.
5	lỗi Xảy ra khi có một lỗi trong việc xếp dỡ, vv
6	chú trọng Xảy ra khi các yếu tố được tập trung.

7	KeyDown Xảy ra khi phím được nhấn.
số 8	bấm phím Xảy ra khi phím được nhấn và thả.
9	KeyUp Xảy ra khi phím đàn được thả.
10	tải Xảy ra khi tài liệu được tải.
11	mousedown Xảy ra khi nút chuột được nhấn.
12	mouseenter Xảy ra khi chuột vào trong một khu vực phần tử.
13	MouseLeave Xảy ra khi chuột rời khỏi một khu vực phần tử.

DANH SÁCH CÁC SỰ KIỆN

14	MouseMove Xảy ra khi con trỏ chuột di chuyển.
15	mouseout Xảy ra khi con trỏ chuột di chuyển khỏi các phần tử.
16	Di chuột lên trên Xảy ra khi con trỏ chuột di chuyển trên một phần tử.
17	mouseup Xảy ra khi nút chuột được thả ra.
18	thay đổi kích thước Xảy ra khi cửa sổ được thay đổi kích cỡ.
19	cuộn giấy Xảy ra khi cửa sổ là cuộn.
20	lựa chọn Xảy ra khi một văn bản được chọn.
21	đệ trình Xảy ra khi mẫu được gửi.
22	dỡ bỏ Xảy ra khi các tài liệu được dỡ xuống.

❑ Các thuộc tính sự kiện

Số TT	Bất động sản & Mô tả
1	<code>altKey</code> Thiết lập là <code>true</code> nếu phím Alt được nhấn khi sự kiện này được kích hoạt, <code>false</code> nếu không. Các phím Alt được dán nhãn Option trên hầu hết các bàn phím máy Mac.
2	<code>ctrlKey</code> Thiết lập là <code>true</code> nếu phím Ctrl được nhấn khi sự kiện này được kích hoạt, <code>false</code> nếu không.
3	<code>data</code> Các giá trị, nếu có, thông qua như là tham số thứ hai đến bind () lệnh khi xử lý được thành lập.
4	<code>keyCode</code> Đối với KeyUp và các sự kiện KeyDown, này trả phím được nhấn.
5	<code>metaKey</code> Thiết lập là <code>true</code> nếu phím Meta được ép khi sự kiện này được kích hoạt, <code>false</code> nếu không. Chìa khóa Meta là phím Ctrl trên máy và phím Command trên máy Mac.

6	Trang X Đối với các sự kiện chuột, quy định cụ thể ngang phối hợp của các sự kiện tương đối từ nguồn gốc trang.
7	Trang Y Đối với các sự kiện chuột, quy định cụ thể dọc phối hợp của các sự kiện tương đối từ nguồn gốc trang.
số 8	relatedTarget Đối với một số các sự kiện chuột, xác định các yếu tố đó con trỏ sang trái hoặc nhập khi sự kiện này được kích hoạt.
9	screenX Đối với các sự kiện chuột, quy định cụ thể ngang phối hợp của các sự kiện tương đối từ nguồn gốc màn hình.
10	ScreenY Đối với các sự kiện chuột, quy định cụ thể dọc phối hợp của các sự kiện tương đối từ nguồn gốc màn hình.
11	phím Shift Thiết lập là true nếu phím Shift được nhấn khi sự kiện này được kích hoạt, false nếu không.

13	<p>Dấu thời gian</p> <p>Các dấu thời gian (tính bằng mili giây) khi sự kiện này được tạo ra.</p>
14	<p>kiểu</p> <p>Đối với tất cả các sự kiện, quy định các loại sự kiện được kích hoạt (ví dụ, nhấp chuột).</p>
15	<p>cái nào</p> <p>Đối với các sự kiện bàn phím, chỉ định mã số cho chìa khóa mà gây ra sự kiện, và cho các sự kiện chuột, quy định cụ thể mà nút được nhấn (1 trái, 2 trung, 3 cho đúng).</p>

Code

```
<html>
  <head>
    <title>The jQuery Example</title>
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>

    <script type="text/javascript" language="javascript">
      $(document).ready(function() {
        $('div').bind('click', function( event ){
          alert('Event type is ' + event.type);
          alert('pageX : ' + event.pageX);
          alert('pageY : ' + event.pageY);
          alert('Target : ' + event.target.innerHTML);
        });
      });
    </script>

    <style>
      .div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}
    </style>

  </head>

  <body>

    <p>Click on any square below to see the result:</p>

    <div class="div" style="background-color:blue;">ONE</div>
    <div class="div" style="background-color:green;">TWO</div>
    <div class="div" style="background-color:red;">THREE</div>

  </body>

</html>
```

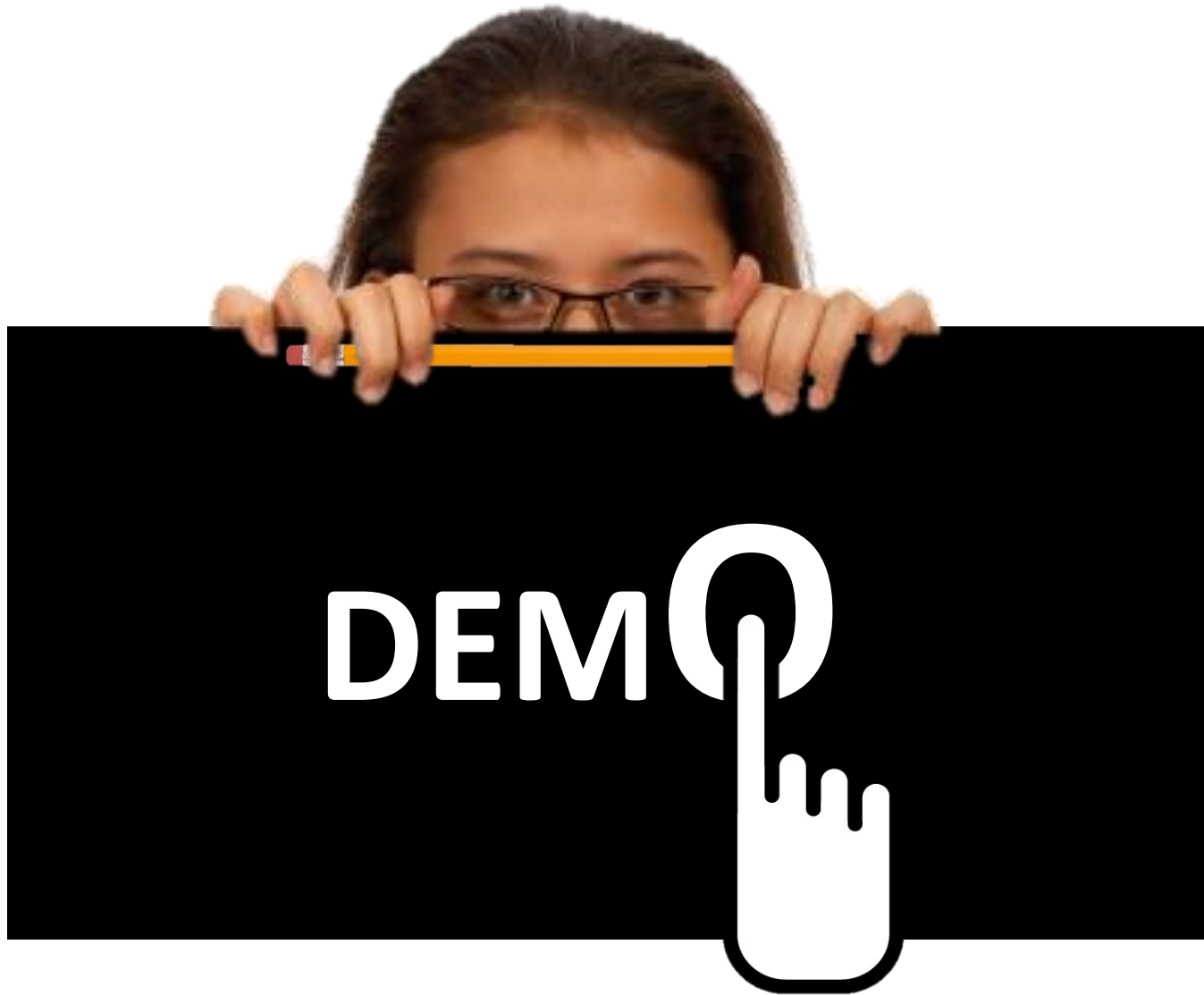
 Kết quả:

Click on any square below to see the result:

ONE

TWO

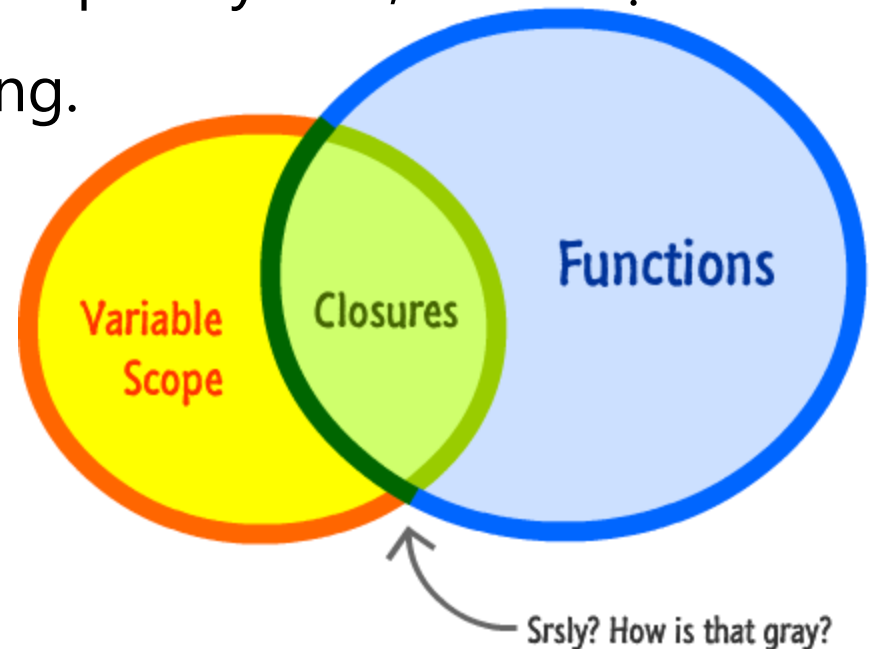
THREE





BÀI 5: JQUERY EVENT(PHẦN 2)

- ❑ Closure là một **hàm được tạo ra từ bên trong một hàm khác** (*hàm cha*), nó có thể sử dụng các biến toàn cục, biến cục bộ của hàm cha và biến cục bộ của chính nó. Việc viết hàm theo kiểu closure trong một số trường hợp sẽ giúp code nhìn sáng và dễ quản lý hơn, linh hoạt hơn trong việc xử lý các chức năng.



- ❑ **Ví dụ:** Xây dựng hàm closure hiển thị thông báo với câu thông báo được nhận từ một tham số, sau đó nối thêm chữ `freetuts.net` đằng sau chuỗi đó.

```
1 function showMessage(message){  
2     alert(message + ' - Freetuts.net');  
3 }  
4  
5 showMessage('Xin chào các bạn');
```

- ❑ Cách viết này rất đơn giản và dễ hiểu. Nhưng bây giờ muốn viết một dạng khác nhìn rồi hơn chút xíu đó là cách sử dụng closure function.

```
1 // Bước 1: Tạo hàm closure
2 function showMessage(message)
3 {
4     return function(){
5         alert(message + ' - Freetuts.net');
6     };
7 }
8
9 // Bước 2: khởi tạo hàm closure
10 var messageFunc = showMessage('Xin chào các bạn');
11
12 // Bước 3: Chạy hàm closure
13 messageFunc();
```

- ❑ Giải thích code
- ❑ Để ý trong hàm showMessage đã return về một function, vì vậy lúc khởi tạo và gán hàm nó vào biến messageFunc thì biến messageFunc sẽ là một #function chưa được khởi động, vì vậy câu thông báo sẽ chưa xuất hiện. Đoạn code ở bước 3 sẽ khởi động hàm trả về đó và câu thông báo sẽ xuất hiện.

❑ Các ví dụ hàm Closure trong Javascript

❖ Closure có tham số

- ❖ Trong ví dụ ở phần một tạo một closure không có tham số, vậy nếu trường hợp có tham số thì cách viết như thế nào?

```
<!DOCTYPE html>
<html>
  <head>
    <title>Closure trong Javascript</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script language="javascript">
      // Bước 1: Tạo hàm closure
      function showMessage(message)
      {
        return function(time){
          for (var i = 1; i <= time; i++){
            alert(message + ' - Freetuts.net');
          }
        };
      }

      // Bước 2: khởi tạo hàm closure
      var messageFunc = showMessage('Xin chào các bạn');

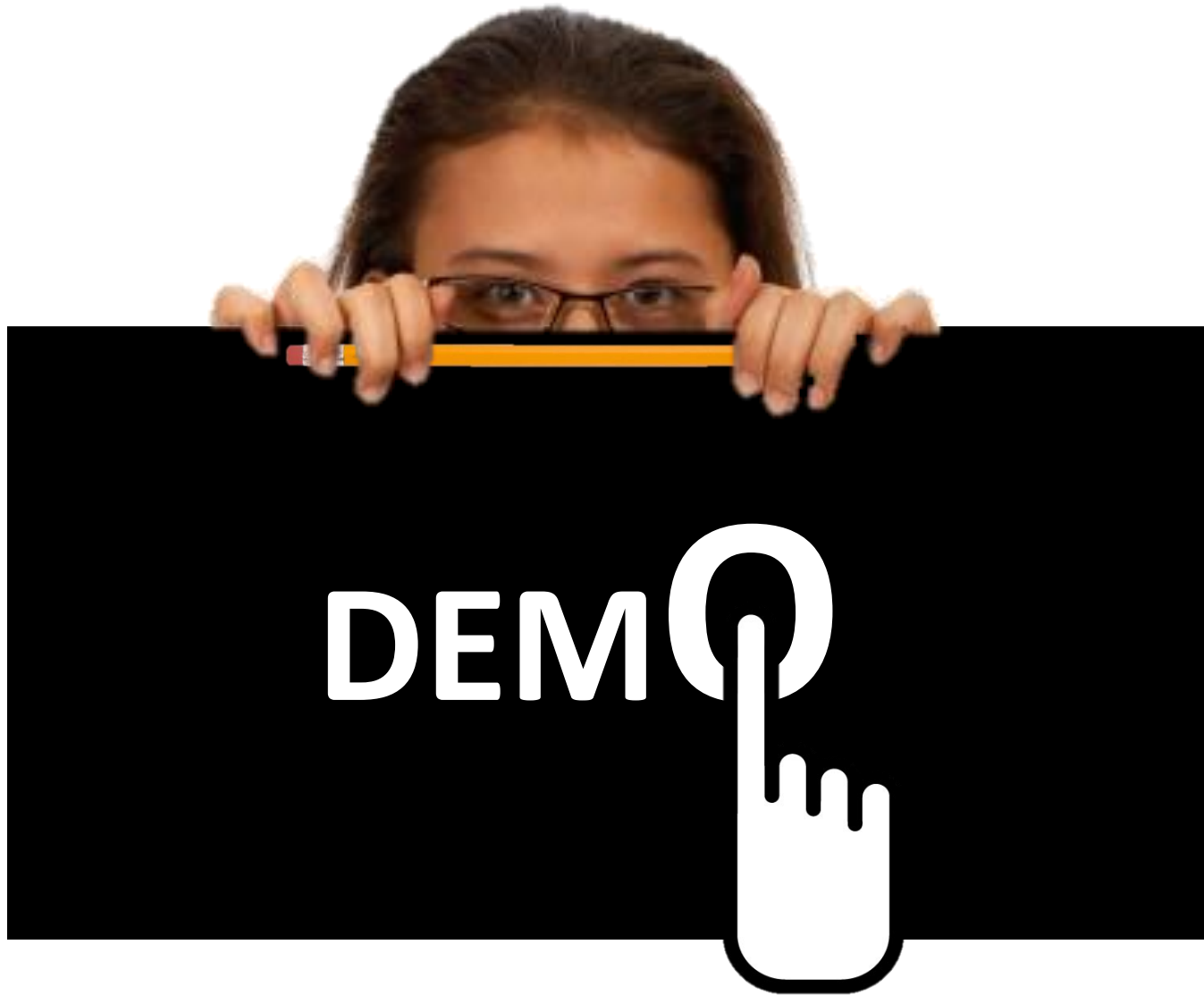
      // Bước 3: Chạy hàm closure
      messageFunc(2);
    </script>
  </head>
  <body>
  </body>
</html>
```

- ❑ Trong ví dụ này hàm closure trả về có một tham số time, lúc này tại bước 3 ta phải truyền nó vào thì chương trình mới hoạt động bình thường.
- ❑ **Closure thay đổi giá trị biến toàn cục lẫn cục bộ**
 - ❖ Các closure function có thể sử dụng biến ở 3 phạm vi, thứ nhất là biến toàn cục, thứ hai là biến cục bộ của hàm cha và thứ ba là biến cục bộ của chính nó. Không chỉ sử dụng được mà còn có khả năng thay đổi giá trị của các biến đó.

```

1  // Bước 1: Tạo hàm closure
2  function Student()
3  {
4      var name = '';
5      var age = '';
6
7      return {
8          set : function(in_name, in_age){
9              name = in_name;
10             age = in_age;
11         },
12         getName : function(){
13             return name;
14         },
15         getAge : function(){
16             return age;
17         }
18     };
19 }
20
21 // Bước 2: khởi tạo hàm closure
22 var studentObj = Student();
23
24 // Bước 3: Chạy hàm closure
25 studentObj.set('Nguyễn Văn Cường', '27');
26 alert(studentObj.getName());
27 alert(studentObj.getAge());

```



❑ Return nhiều Closure Function

- ❖ Muốn return nhiều hàm closure thì bạn phải sử dụng một object, trong đó mỗi phần tử sẽ là một closure function.

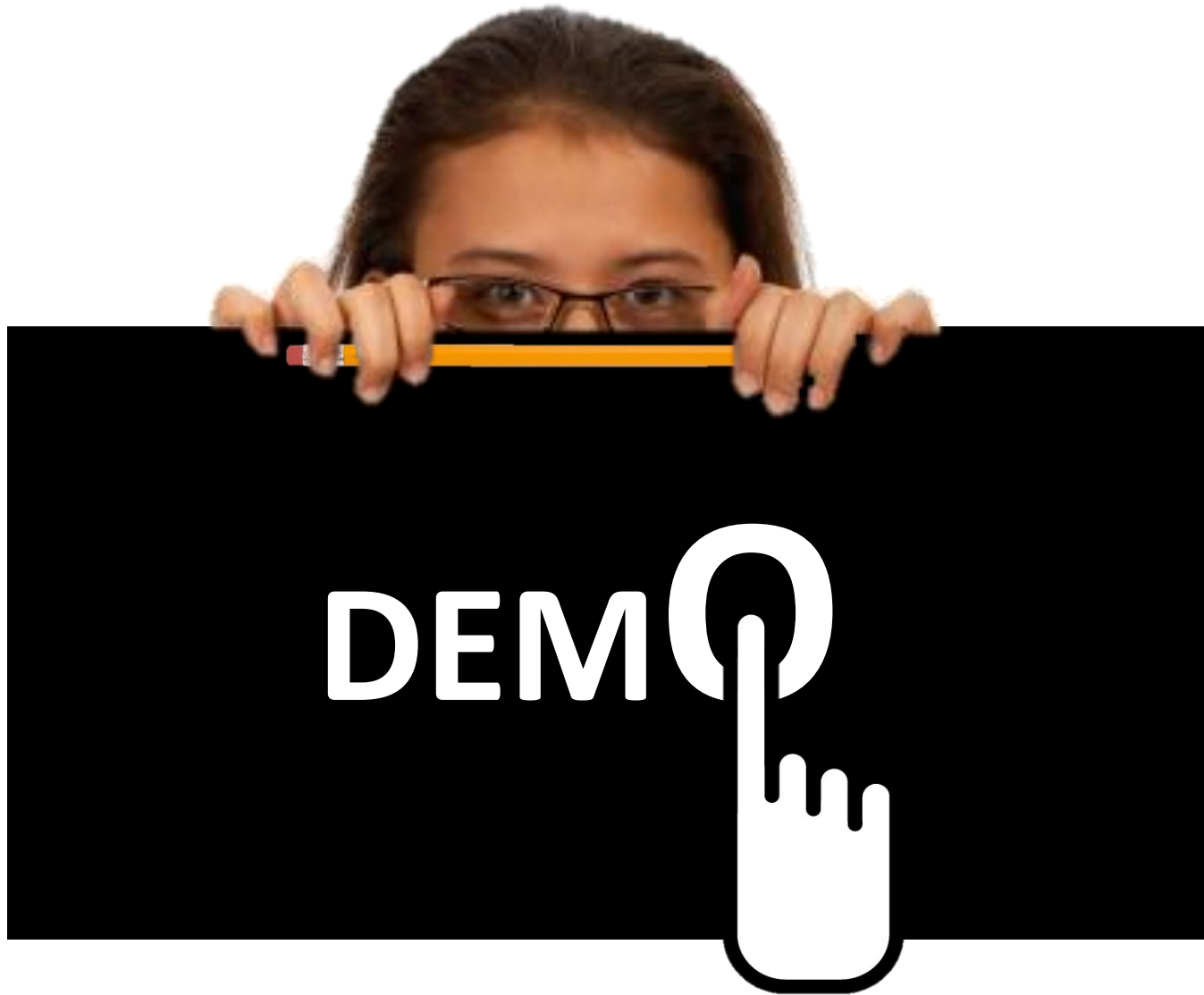
```
1  function multiClosure()  
2  {  
3      return {  
4          func1 : function(){  
5              console.log('Closure1');  
6          },  
7          func2 : function(){  
8              console.log('Closure2');  
9          }  
10     };  
11 }  
12  
13 // Cách sử dụng  
14 var object = multiClosure();  
15 object.func1();  
16 object.func2();
```

❑ Độ ưu tiên các biến

- ❖ Như ta biết closure function có thể sử dụng biến tại ba vị trí đó là biến toàn cục, biến cục bộ hàm cha và biến cục bộ của chính nó. Bây giờ có một trường hợp là tên các biến ở ba vị trí đó bị trùng nhau thì sẽ có chuyện gì xảy ra? Trường hợp này nó sẽ ưu tiên từ trong ra ngoài, nghĩa là nó sẽ thực hiện các bước như sau:

- ❑ **Bước 1:** Xem biến cục bộ trong hàm closure có trùng không? Nếu không trùng thì nó sẽ qua bước 2, còn nếu có trùng thì nó sẽ coi biến đó là biến cục bộ của chính nó.
- ❑ **Bước 2:** Xem biến cục bộ của hàm cha có trùng không? Nếu không trùng thì qua bước 3, nếu trùng thì nó sẽ coi biến đó là biến cục bộ của hàm cha.
- ❑ **Bước 3:** Xem biến toàn cục có trùng không? Nếu không trùng thì nó sẽ khởi tạo mới và đó sẽ là biến cục bộ của hàm closure, nếu trùng thì nó sẽ coi biến đó là biến toàn cục.


```
1 // Bước 1: Tạo hàm closure
2 var message = 'Biên toàn cục';
3 function showMessage()
4 {
5     var message = 'Biến cục bộ của hàm cha';
6     return function(){
7         alert(message);
8     };
9 }
10
11 // Bước 2: khởi tạo hàm closure
12 var messageFunc = showMessage();
13
14 // Bước 3: Chạy hàm closure
15 messageFunc();
```



- Sự kiện trong jquery
- Một số sự kiện cơ bản
- Closure và một số vấn đề liên quan





Cảm ơn