



## **BÀI 3: JQUERY THUỘC TÍNH (PHẦN 1)**

- ⊙ Tổng quan về jquery
- ⊙ Thư viện jquery.
- ⊙ Các vấn đề liên quan
  - Các đối tượng cơ bản
  - Hàm (Function)
  - Callback
  - Proxy Pattern
  - Thuộc tính trong jQuery
  - sử dụng các Selector trong jQuery?
  - CSS Selector trong jQuery



- ❑ jQuery là một Framework được xây dựng dựa trên các tính năng của JavaScript. Vì thế trong khi phát triển các ứng dụng sử dụng jQuery, sử dụng tất cả các hàm và các tính năng khác được hỗ trợ trong JavaScript.



## ❑ Đối tượng String

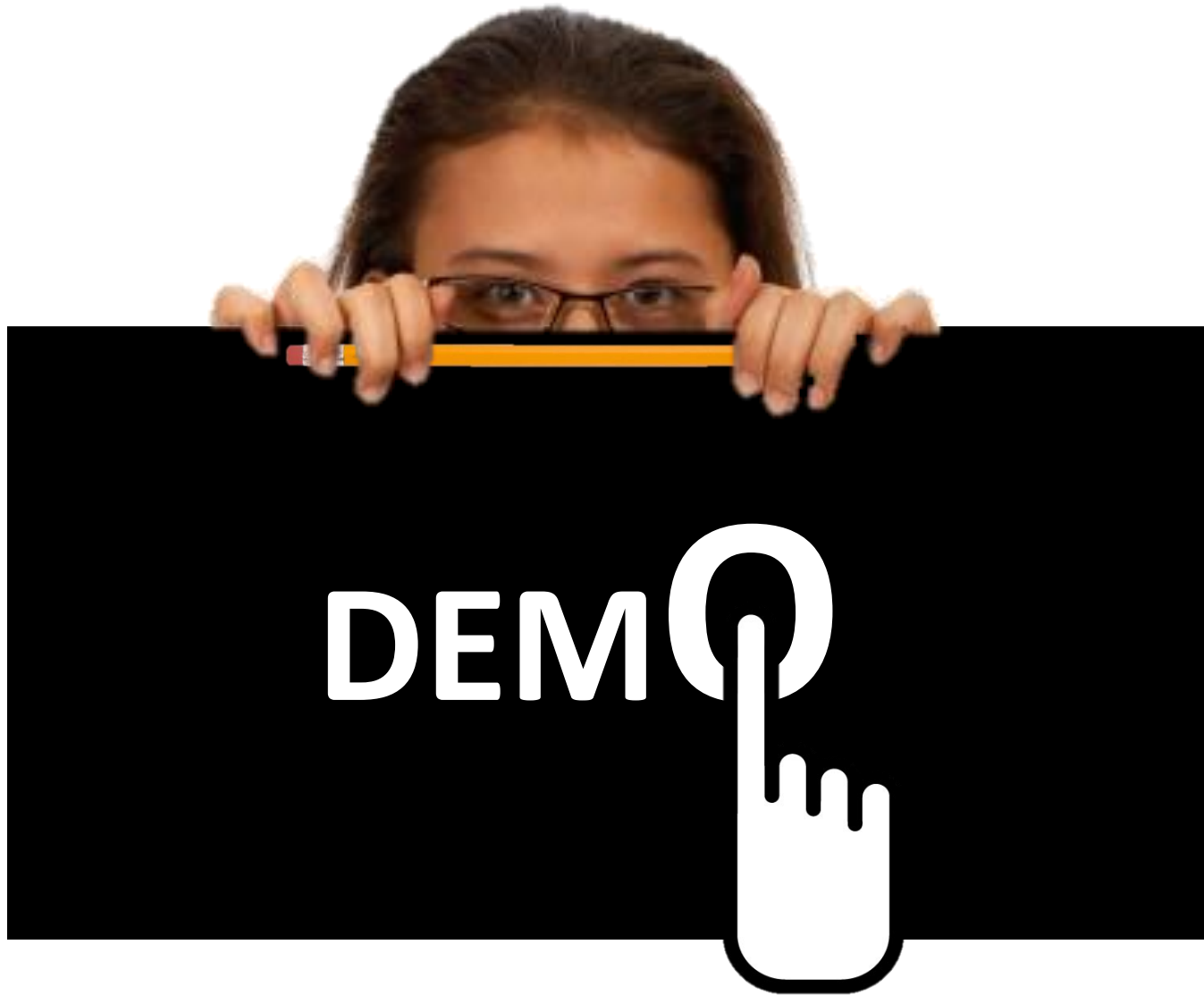
- ❖ Một chuỗi trong JavaScript là một đối tượng không đổi chứa 0, 1 hoặc nhiều ký tự.
- ❖ Sau đây là một ví dụ hợp lệ về một String trong JavaScript.

```
"This is JavaScript String"
```

```
'This is JavaScript String'
```

```
'This is "really" a JavaScript String'
```

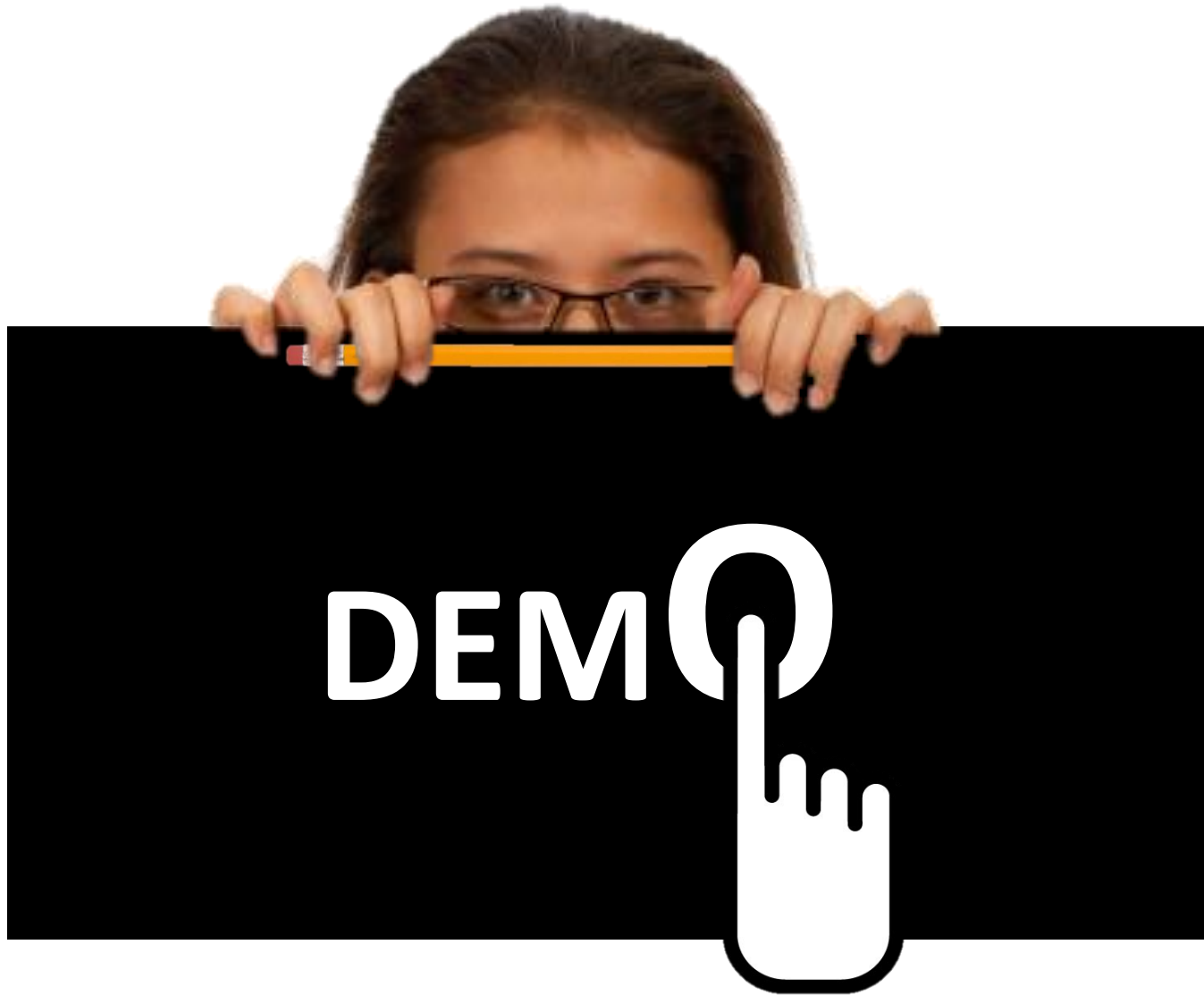
```
"This is 'really' a JavaScript String"
```



## ❑ Đối tượng Number trong jQuery

- ❖ Đối tượng Number trong JavaScript là định dạng độ chính xác kép (64 bit) theo chuẩn IEEE 754. Chúng là không đổi, như đối tượng String.
- ❖ Sau đây là ví dụ hợp lệ về một số trong JavaScript.

```
5350  
120.27  
0.26
```



## ❑ Đối tượng Object trong jQuery

- ❖ JavaScript hỗ trợ tốt khái niệm Object. Có thể tạo một Object bởi sử dụng Object Literal như sau:

```
var emp = {  
  name: "Zara",  
  age: 10  
};
```

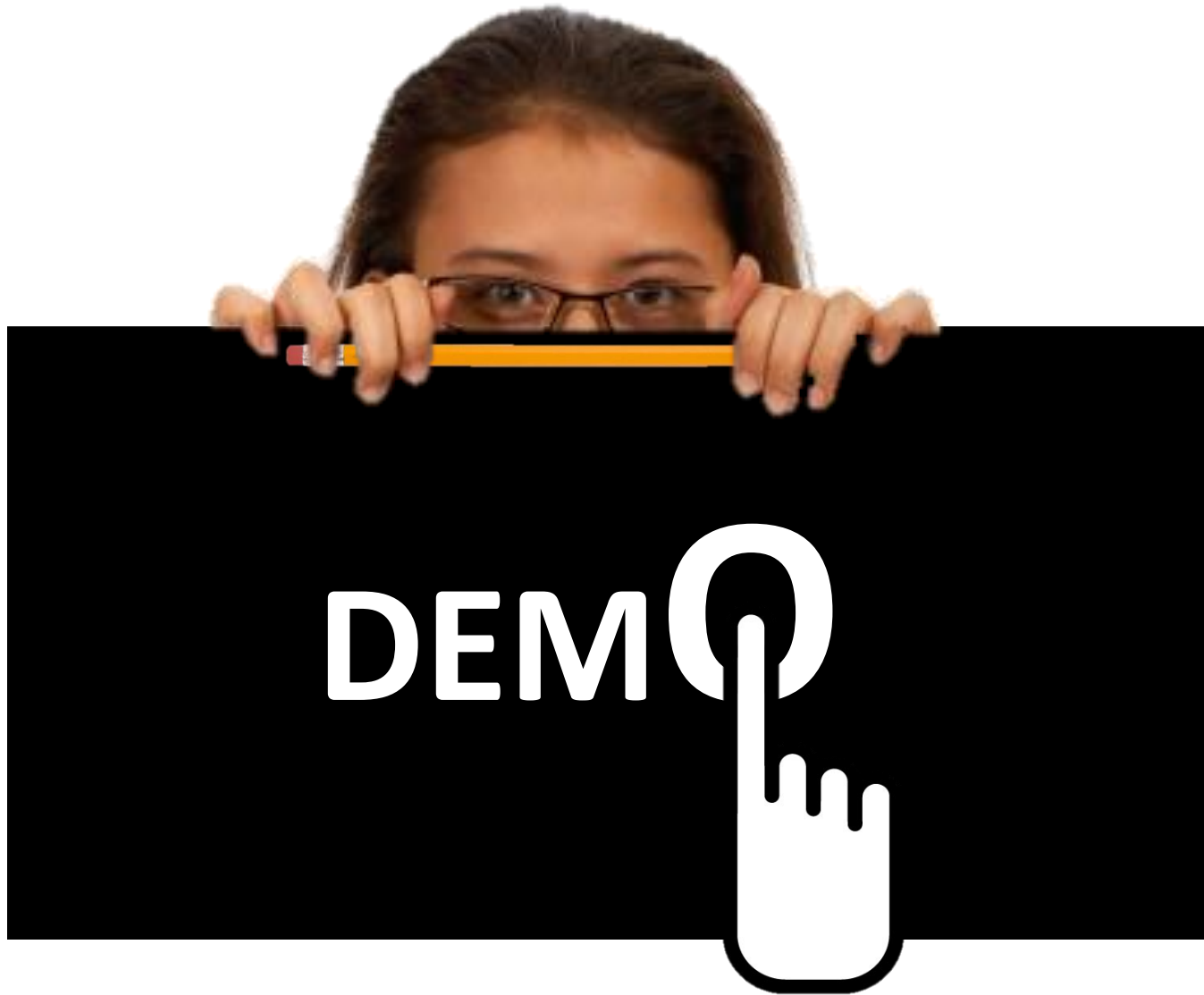
## ❑ Đối tượng Array trong jQuery

```
var x = [];  
var y = [1, 2, 3, 4, 5];
```

Một mảng có một thuộc tính **length** là hữu ích cho tính lặp:

```
var x = [1, 2, 3, 4, 5];  
for (var i = 0; i < x.length; i++) {  
  // Do something with x[i]  
}
```





## ❑ Hàm (Function) trong jQuery

- ❖ Một hàm trong JavaScript có thể được đặt tên hoặc ẩn danh. Một hàm được đặt tên có thể được định nghĩa bởi sử dụng từ khóa *function* như sau:

```
function named(){  
    // do some stuff here  
}
```

## ❖ Các tham số trong jQuery

```
function func(x){  
    console.log(typeof x, arguments.length);  
}  
func();           //==> "undefined", 0  
func(1);          //==> "number", 1  
func("1", "2", "3"); //==> "string", 3
```

## ❑ Phạm vi (Scope) trong jQuery

- ❖ **Các biến Global** – Một biến Global có phạm vi chung, nghĩa là nó được định nghĩa ở mọi nơi trong JavaScript code.
- ❖ **Các biến Local** – Một biến Local sẽ chỉ nhìn thấy bên trong một hàm nơi nó được định nghĩa. Các tham số hàm luôn luôn là Local cho hàm đó.

```
var myVar = "global";    // ==> Declare a global variable

function ( ) {
    var myVar = "local";  // ==> Declare a local variable
    document.write(myVar); // ==> local
}
```

## ❑ Callback trong jQuery

- ❖ Một số callback là các sự kiện, được gọi để cung cấp cho người sử dụng cơ hội để phản ứng lại khi một trạng thái nào đó được kích hoạt.

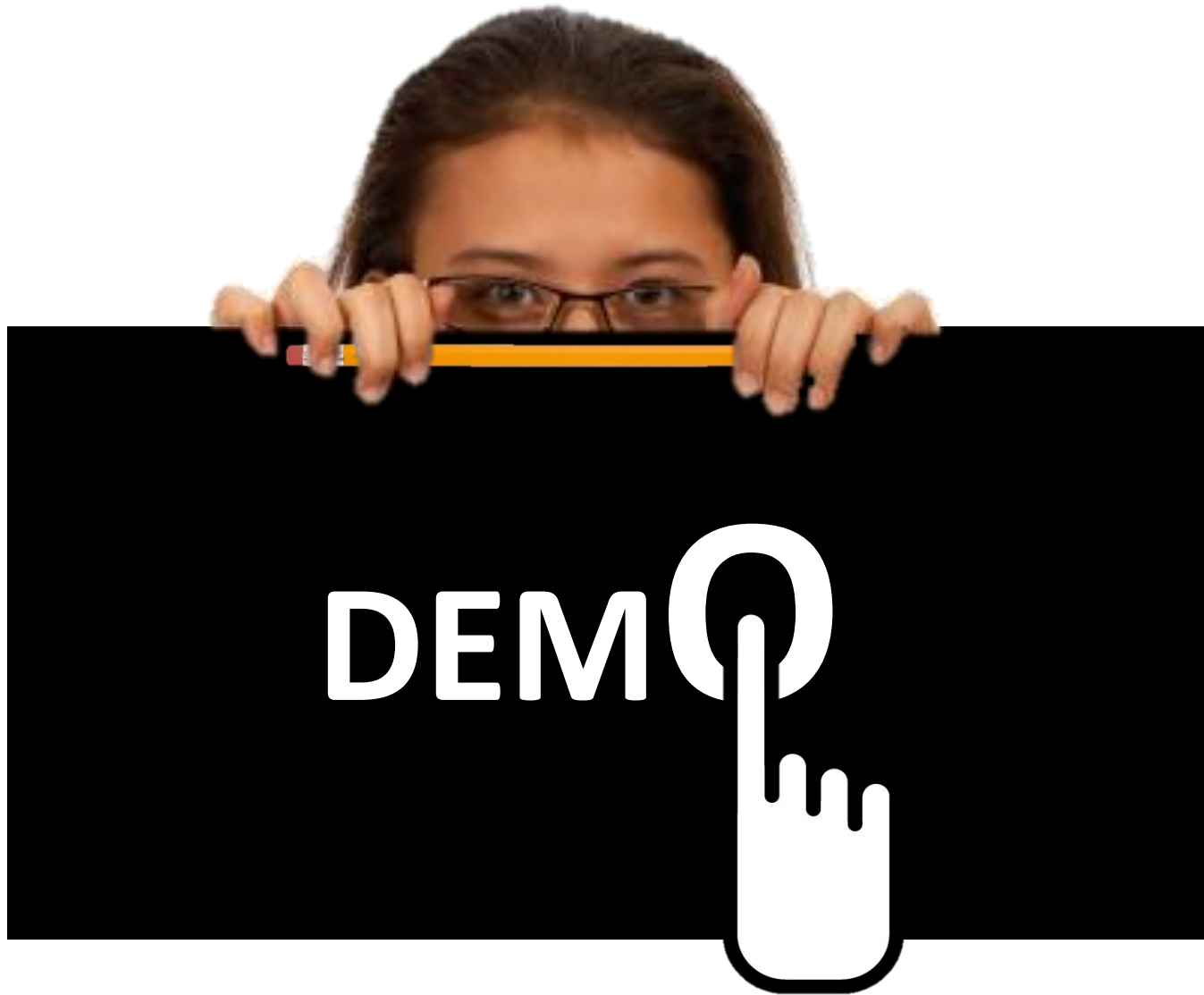
Hệ thống sự kiện trong jQuery sử dụng các callback này ở khắp mọi nơi, ví dụ:

```
$("#body").click(function(event) {  
    console.log("clicked: " + event.target);  
});
```

## ❑ Proxy Pattern trong jQuery

- ❖ Một Proxy có thể được khởi tạo tại vị trí của Real Subject này và cho phép nó để được truy cập ở chế độ từ xa. Chúng ta có thể lưu giữ phương thức **setArray** của jQuery trong một Closure và viết đè lên (overwrite) nó như sau:

```
(function() {  
    // log all calls to setArray  
    var proxied = jQuery.fn.setArray;  
  
    jQuery.fn.setArray = function() {  
        console.log(this, arguments);  
        return proxied.apply(this, arguments);  
    };  
})();
```



## ❑ Các hàm có sẵn trong jQuery

- ❖ JavaScript đi kèm một tập hợp các hàm hữu ích gắn liền với nó. Những phương thức này có thể được sử dụng để thao tác String, Number, và Date.
- ❖ Bảng dưới liệt kê các hàm JavaScript quan trọng:

STT	Phương thức & Miêu tả
1	<b>charAt()</b> Trả về ký tự tại chỉ mục (index) đã cho.
2	<b>concat()</b> Kết nối hai chuỗi văn bản và trả về một chuỗi mới.
3	<b>forEach()</b> Gọi một hàm cho mỗi phần tử của một mảng.
4	<b>indexOf()</b> Trả về chỉ mục về sự xuất hiện đầu tiên bên trong việc gọi đối tượng String với giá trị đã cho, hoặc -1 nếu không tìm thấy.

6	<b>pop()</b> Gỡ bỏ phần tử cuối của một mảng và trả về phần tử đó.
7	<b>push()</b> Thêm một hoặc nhiều phần tử tới phần cuối của một mảng và trả về độ dài mới của mảng đó.
8	<b>reverse()</b> Đảo ngược thứ tự các phần tử trong một mảng – phần tử đầu tiên thành cuối cùng và cuối cùng thành đầu tiên.
9	<b>sort()</b> Sắp xếp phân loại các phần tử của một mảng.
10	<b>substr()</b> Trả về các ký tự trong một mảng bắt đầu từ vị trí đã cho từ số các ký tự đã xác định.
11	<b>toLowerCase()</b> Trả về giá trị chuỗi đang gọi được biến đổi thành kiểu chữ thường.



## ❑ Nhận giá trị thuộc tính trong jQuery

- ❖ Phương thức **attr()** có thể được sử dụng để nhận giá trị của một thuộc tính từ phần tử đầu tiên trong tập hợp đã so khớp hoặc thiết lập các giá trị thuộc tính trên các phần tử được so khớp.

### Ví dụ:

Sau đây là một ví dụ đơn giản mà nhận thuộc tính title của thẻ <em> và thiết lập giá trị <div id="divid"> với cùng giá trị.

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

    <script type="text/javascript" language="javascript">
      $(document).ready(function() {
        var title = $("em").attr("title");
        $("#divid").text(title);
      });
    </script>
  </head>

  <body>
    <div>
      <em title="Bold and Brave">This is first paragraph.</em>
      <p id="myid">This is second paragraph.</p>
      <div id="divid"></div>
    </div>
  </body>

</html>
```

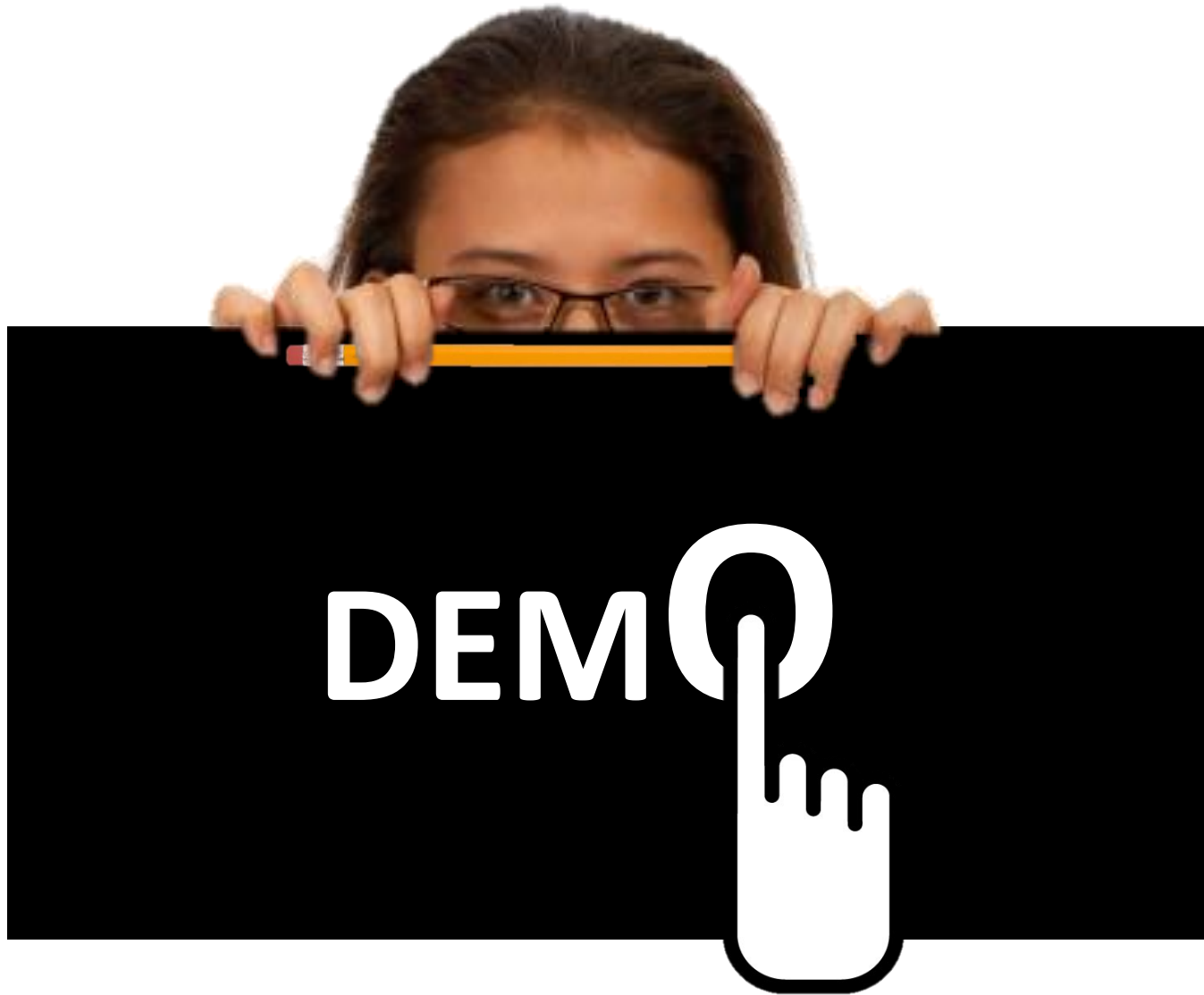
## ❑ Thiết lập giá trị thuộc tính trong jQuery

- ❖ Phương thức **attr(name, value)** có thể được sử dụng để thiết lập thuộc tính đã đặt tên trên tất cả phần tử trong tập hợp được bao bởi sử dụng value đã truyền.

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

    <script type="text/javascript" language="javascript">
      $(document).ready(function() {
        $("#myimg").attr("src", "../images/jquery.jpg");
      });
    </script>
  </head>

  <body>
    <div>
      
    </div>
  </body>
</html>
```





## **BÀI 3: JQUERY THUỘC TÍNH (PHẦN 2)**

- ❑ Thư viện jQuery khai thác sức mạnh của các CSS (*Cascading Style Sheets*). Selector để giúp chúng ta truy cập nhanh và dễ dàng hơn tới các phần tử hoặc nhóm các phần tử trong DOM.
- ❑ Selector được sử dụng để chọn một hoặc nhiều phần tử HTML bởi sử dụng jQuery. Khi một phần tử được chọn, thì chúng ta có thể thực hiện các hoạt động đa dạng trên phần tử đã chọn đó



## ❑ Hàm cơ sở \$() trong jQuery:

- ❖ jQuery Selector bắt đầu với ký hiệu đô la và cặp dấu ngoặc đơn **\$()**.

STT	Selector & Miêu tả
1	<b>Tag Name</b> Biểu diễn một tên thẻ có sẵn trong DOM. Ví dụ: <b>\$('p')</b> chọn tất cả đoạn văn trong phần tử.
2	<b>Tag ID</b> Biểu diễn một tên thẻ có sẵn với ID đã cho trong DOM. Ví dụ <b>\$('#some-id')</b> chọn tất cả phần tử đơn trong tài liệu mà có một ID là some-id
3	<b>Tag Class</b> Biểu diễn một thẻ có sẵn với lớp đã cho trong DOM. Ví dụ <b>\$('.some-class')</b> chọn tất cả các phần tử trong tài liệu mà có một lớp là some-class.

❑ **Ghi chú** – Hàm cơ sở **\$()** là đồng nghĩa với một hàm **jQuery()**. Vì thế trong trường hợp đang sử dụng bất kỳ thư viện JavaScript nào khác thì có thể xuất hiện xung đột ở đây, khi đó có thể đổi ký hiệu **\$** thành **jQuery** và có thể sử dụng hàm **jQuery()** thay cho hàm **\$()**.

❑ Ví dụ:

Sau đây là ví dụ đơn giản sử dụng Tag Selector. Nó sẽ chọn tất cả phần tử với tên thẻ **p** và sẽ thiết lập màu nền thành "yellow".



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

    <script type="text/javascript" language="javascript">
      $(document).ready(function() {
        $("p").css("background-color", "yellow");
      });
    </script>
  </head>

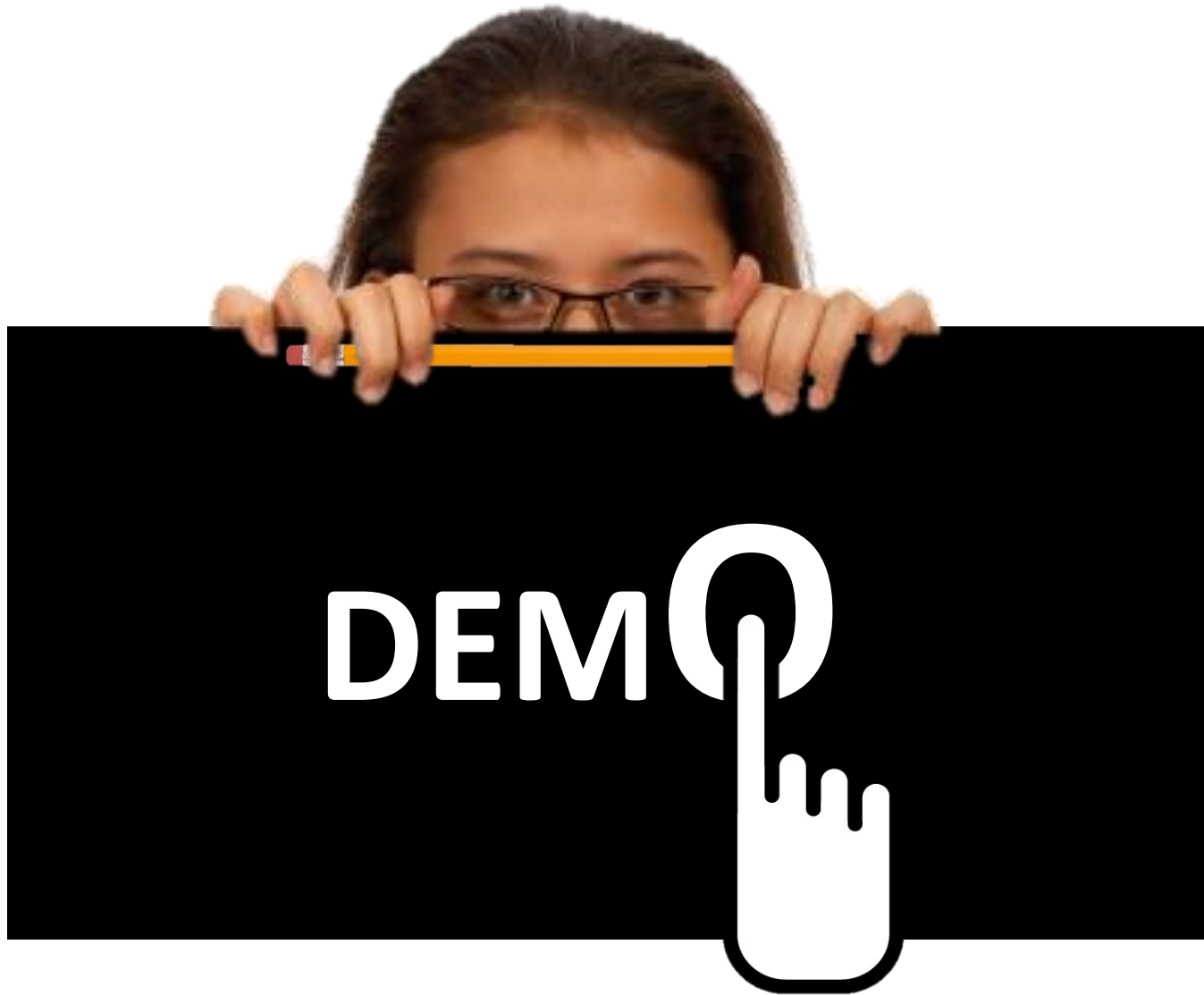
  <body>
    <div>
      <p class="myclass">This is a paragraph.</p>
      <p id="myid">This is second paragraph.</p>
      <p>This is third paragraph.</p>
    </div>
  </body>
```

❑ Kết quả:

This is a paragraph.

This is second paragraph.

This is third paragraph.



- ❑ Selector là rất hữu ích và sẽ cần yêu cầu ở mọi bước trong khi sử dụng jQuery. Chúng nhận phần tử chính xác khi bạn muốn từ tài liệu HTML.
- ❑ Bảng sau liệt kê các Selector cơ bản và giải thích về chúng bởi các ví dụ khi bạn click và link tương ứng:

STT	Selector & Miêu tả
1	<b>Name</b> Chọn tất cả phần tử mà so khớp với phần tử có <b>Name</b> đã cho.
2	<b>#ID</b> Chọn một phần tử đơn mà so khớp với <b>ID</b> đã cho.
3	<b>.Class</b> Chọn tất cả phần tử mà so khớp với <b>Class</b> đã cho.
4	<b>Universal (*)</b> Chọn tất cả phần tử có sẵn trong một DOM.
5	<b>Multiple Elements E, F, G</b> Chọn các kết quả được tổ hợp từ tất cả các bộ chọn <b>E, F</b> hoặc <b>G</b> đã cho..

## ❑ Cú pháp

`$("#id_của_element")`

## ❑ Mã HTML

`< a href="#" id="linkID">Link</a>`

## ❑ Truy cập đến element có **id** là **linkID**

`getElementById("linkID") ➔ $("#linkID")`

## ❑ Truy cập đến element **<a>** có **id** là **linkID**

`$("#a#linkID")`

## ❑ Cú pháp

`$(".tên_lớp")`

## ❑ Mã HTML

`< a class="link">Link</a>`

## ❑ Chọn tất cả các element có **tên lớp** là **link**

`$(".link")`

## ❑ Chọn tất cả các element **<a>** có **tên lớp** là **link**

`$("a.link")`

## ☐ Cú pháp

`$("tên_thẻ_HTML")`

## ☐ Chọn tất cả các element **div** trên trang

`$("div")`



## ❑ Mã HTML

```
<body>
<a href="link1.html"></a>
<div>
    <a href="link2.html"></a>
</div>
<div id="divContent">
    <a href="link3.html"> Google </a>
    <a href="link4.html"> w3schools </a>
</div>
</body>
```

❑ Chọn tất cả các element **<a>** nằm trong thẻ **div**

`$("div a")`

❑ Chọn tất cả các element **<a>** nằm trong thẻ **div**

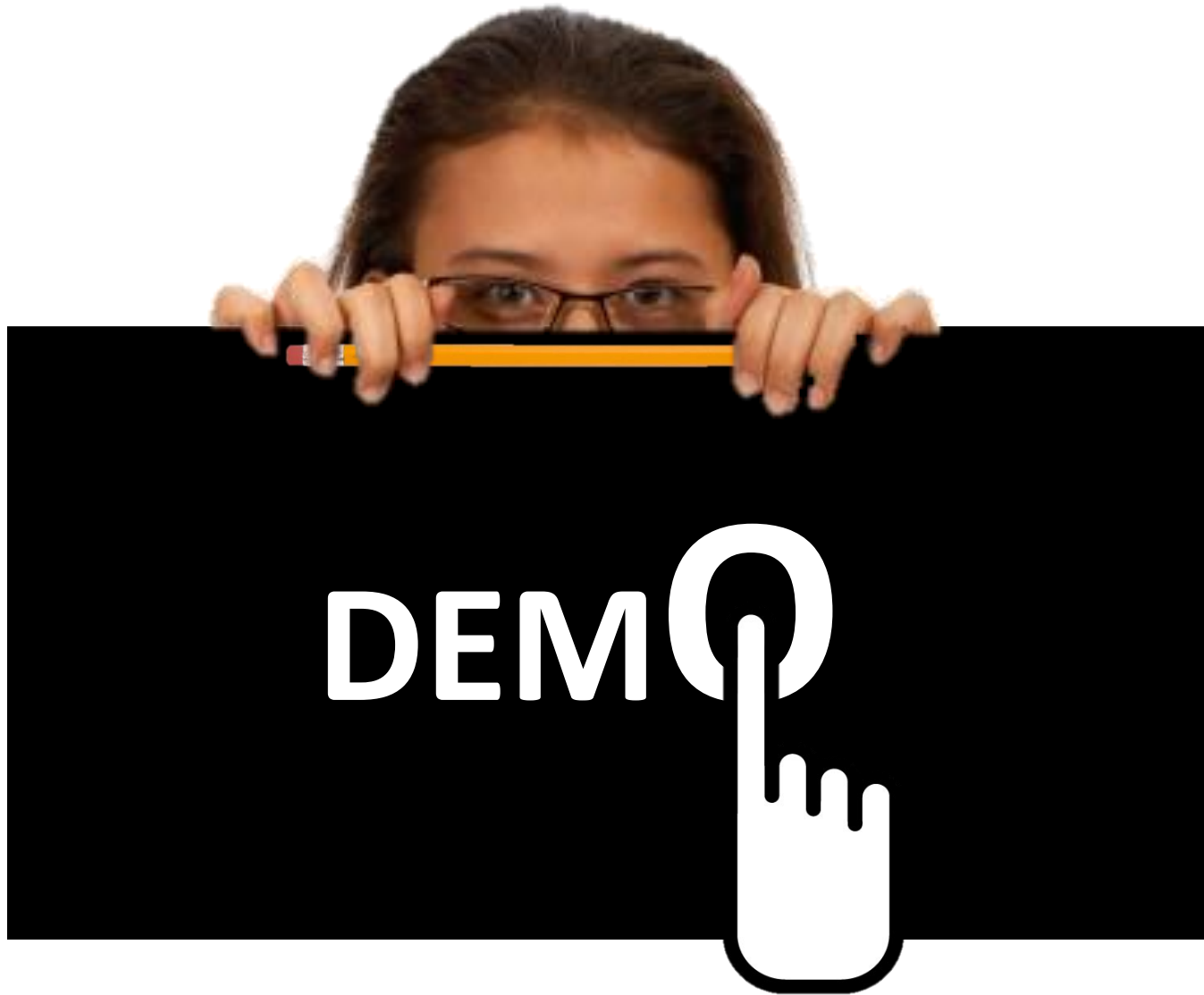
**divContent**

`$("#divContent a")`

- ❑ Mã HTML `<body>`  
`<p>Đầu tiên</p>`  
`<p>Thứ hai</p>`  
`<p>Thứ ba</p>`  
`<body>`
- ❑ Chọn element `<p>` **đầu tiên** trong tài liệu  
`$("#p:first")`
- ❑ Chọn element `<p>` **cuối cùng** trong tài liệu  
`$("#p:last")`
- ❑ Chọn element `<p>` **thứ hai** trong tài liệu  
`$("#p")[1]`
- ❑ Chọn các element `<p>` **lẻ** trong tài liệu
  - ❖ Phần tử đầu tiên là 0  
`$("#p:odd")`

- ❑ Chọn tất cả các element có **attribute href**  
`$("[href]")`
- ❑ Chọn tất cả các element **<a>** có **attribute href**  
`$("a[href]")`
- ❑ Chọn tất cả các element với **attribute href** có giá trị là **"#"**  
`$("a[href = '#']")`
- ❑ Một số ký tự đặc biệt

Cú pháp	Giải thích
attributeName*=value	chọn các element mà giá trị của attribute chứa value
attributeName~=value	chọn các element mà giá trị attribute bằng value
attributeName!=value	chọn các element mà giá trị attribute không bằng value hoặc không có attribute đó
attributeName\$=value	chọn các element mà giá trị attribute kết thúc bằng value
attributeName^=value	chọn các element mà giá trị attribute bắt đầu bằng value



- ❑ Thư viện jQuery hỗ trợ gần như tất cả các Selector trong Cascading Style Sheet (CSS), khi đã được đề cập trên World Wide Web Consortium.

```
selector.css( propertyName, PropertyValue );
```

- ❑ Ví dụ sau thêm màu Font tới *list item* thứ ba

```
<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() {
        $("li").eq(2).css("color", "red");
      });
    </script>
  </head>

  <body>
    <div>

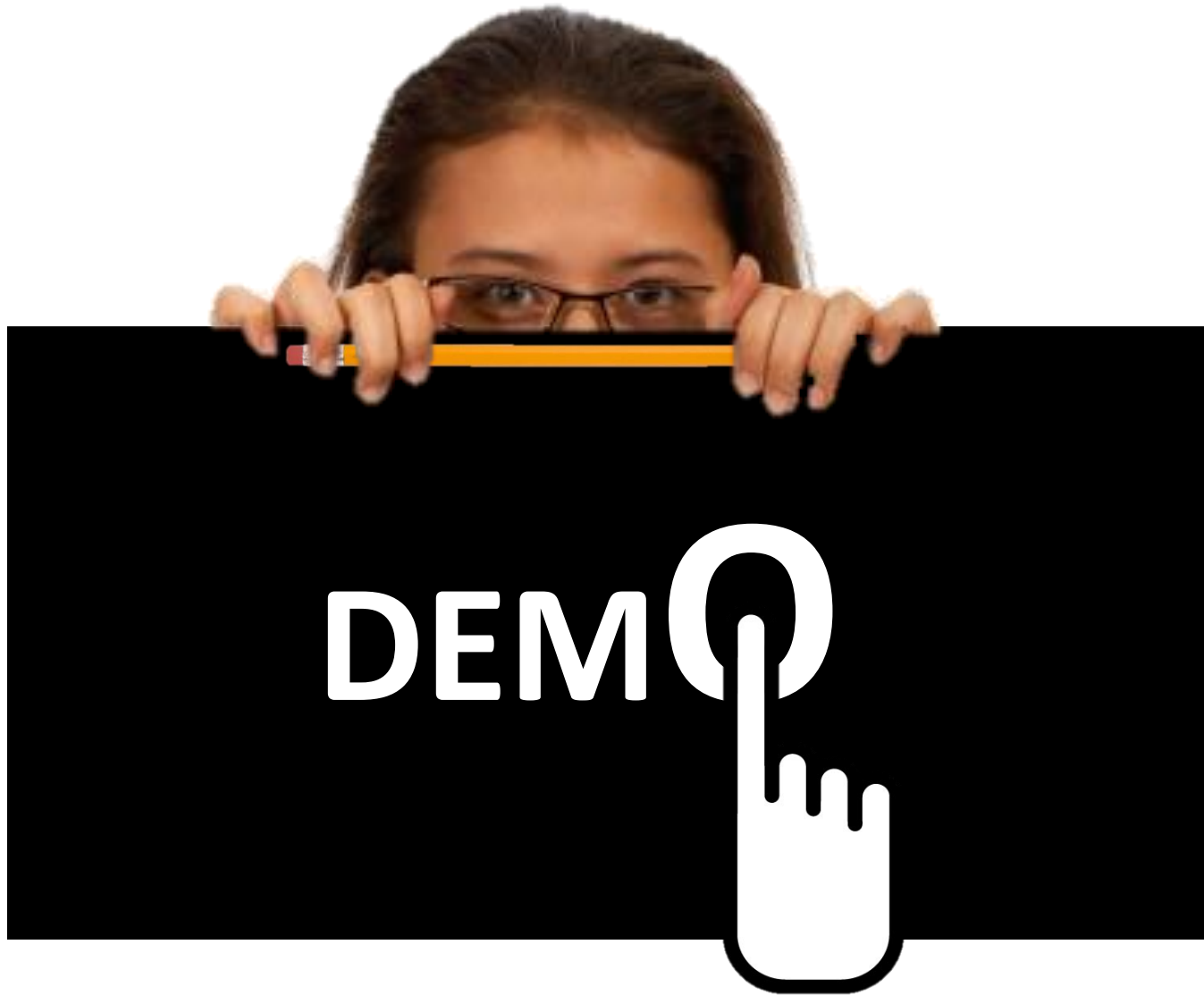
      <ul>
        <li>list item 1</li>
        <li>list item 2</li>
        <li>list item 3</li>
        <li>list item 4</li>
        <li>list item 5</li>
        <li>list item 6</li>
      </ul>

    </div>
  </body>
</html>
```



Nó sẽ cho kết quả sau:

- list item 1
- list item 2
- **list item 3**
- list item 4
- list item 5
- list item 6



- ❑ Áp dụng nhiều thuộc tính CSS trong jQuery
  - ❖ Áp dụng nhiều thuộc tính CSS bởi sử dụng phương thức **CSS( {key1:val1, key2:val2....} )** của jQuery.
  - ❖ Sau đây là cú pháp cho phương thức này:

```
selector.css( {key1:val1, key2:val2....keyN:valN})
```

- ❖ Ví dụ:



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

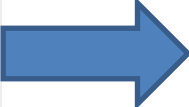
    <script type="text/javascript" language="javascript">
      $(document).ready(function() {
        $("li").eq(2).css({"color":"red", "background-color":"green"});
      });
    </script>
  </head>

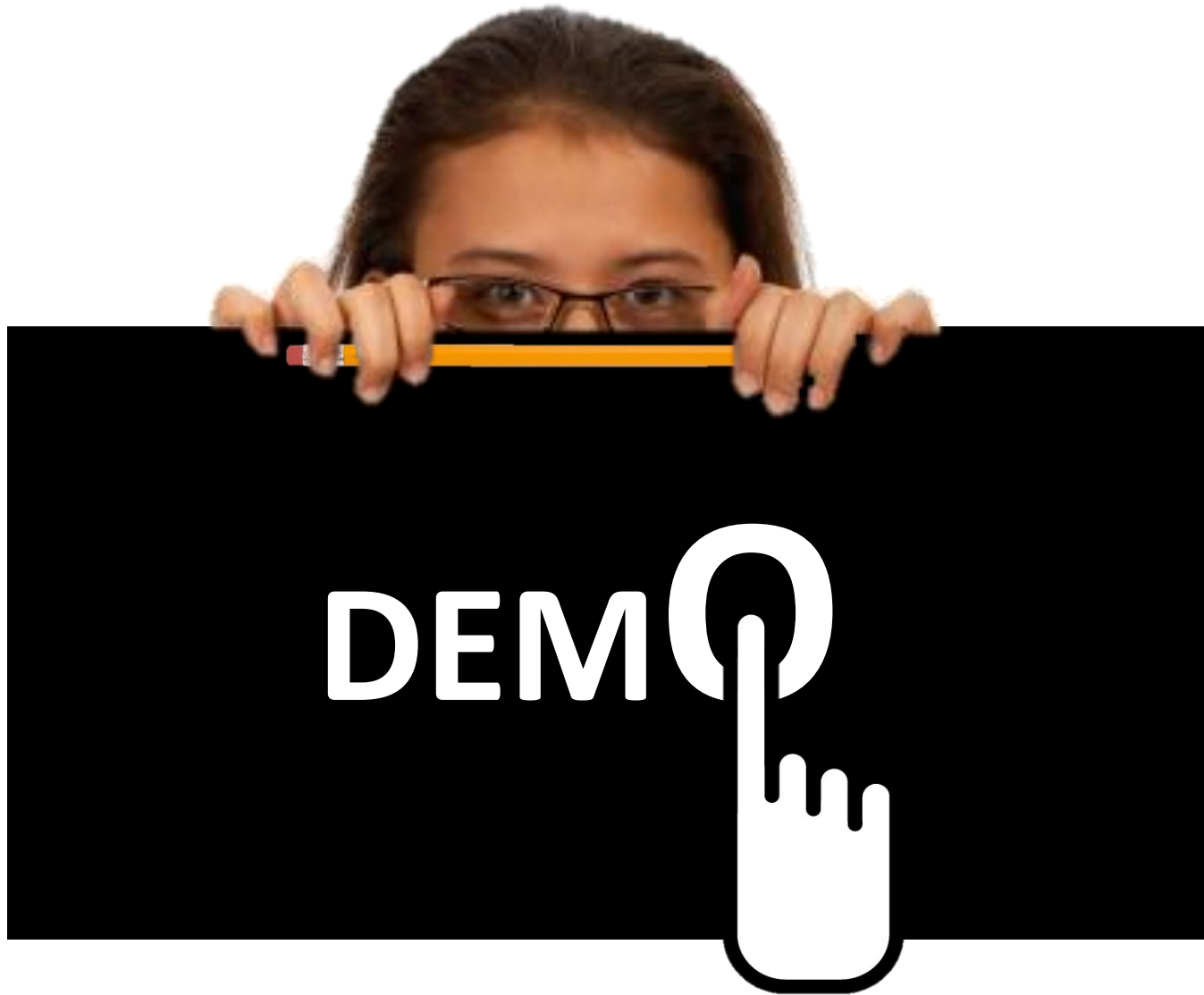
  <body>
    <div>

      <ul>
        <li>list item 1</li>
        <li>list item 2</li>
        <li>list item 3</li>
        <li>list item 4</li>
        <li>list item 5</li>
        <li>list item 6</li>
      </ul>

    </div>
  </body>
</html>
```

Nó sẽ cho kết quả sau:

- 
- list item 1
  - list item 2
  - **list item 3**
  - list item 4
  - list item 5
  - list item 6



## ❑ Các phương thức CSS trong jQuery

STT	Phương thức & Miêu tả
1	<b>css( name )</b> Trả về thuộc tính style trên phần tử đã so khớp đầu tiên
2	<b>css( name, value )</b> Thiết lập thuộc tính style đơn tới một giá trị trên tất cả phần tử đã so khớp
3	<b>css( properties )</b> Thiết lập một đối tượng key/value như là các thuộc tính style tới tất cả các phần tử đã so khớp
4	<b>height( val )</b> Thiết lập chiều cao CSS của mỗi phần tử đã so khớp
5	<b>height( )</b> Nhận chiều cao, giá trị pixel, được tính toán hiện tại của phần tử đã so khớp đầu tiên

6	<b>innerHeight( )</b> Nhận chiều cao bên trong (trừ Border và bao gồm padding) cho phần tử đã so khớp đầu tiên
7	<b>innerWidth( )</b> Nhận độ rộng bên trong (trừ Border và bao gồm padding) cho phần tử đã so khớp đầu tiên.
8	<b>offset( )</b> Nhận offset hiện tại của phần tử so khớp đầu tiên, bằng giá trị pixel, liên quan tới tài liệu
9	<b>offsetParent( )</b> Trả về một tập hợp jQuery với phần tử cha đã xác định vị trí của phần tử so khớp đầu tiên
10	<b>outerHeight( [margin] )</b> Nhận chiều cao bên ngoài (bao gồm Border và padding theo mặc định) cho phần tử đã so khớp đầu tiên

- Tổng quan về jquery
- Thư viện jquery.
- Các vấn đề liên quan
  - Các đối tượng cơ bản
  - Hàm (Function)
  - Callback
  - Proxy Pattern
  - Thuộc tính trong jQuery
  - sử dụng các Selector trong jQuery?
  - CSS Selector trong jQuery





**Cảm ơn**