

## 1. Mục tiêu

- Sử dụng jQuery
  - Nhúng jQuery
  - Sử dụng: \$
- Chọn phần tử trong jQuery
- Bắt sự kiện
- Get/set
- Filter
- Plugin: Autocomplete, Fine Uploader, Slider, Validation, Zoom image ...

## 2. Tóm tắt lý thuyết

### 2.1 JQuery là gì?

**JQuery** là một thư viện JavaScript dùng để tương tác với HTML, được thiết kế để giúp việc viết mã JavaScript trở nên đơn giản và thuận tiện hơn. Để sử dụng **jQuery**, ta cần nhúng thư viện **jQuery** vào trang web của mình.

a) Sử dụng CDN:

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">  
</script>
```

b) Tải file xuống và chèn trong thẻ <script>:

Tải thư viện jQuery từ trang : <http://jquery.com/download>

```
<script type="text/javascript" src="../jQuery/jquery-2.1.3.min.js">  
</script>
```

Với `src="..."` là đường dẫn tới file thư viện .js trên local hoặc trên *web hosting*.

### 2.2 Các thuật ngữ cần biết.

jQuery chứa đựng nhiều thuật ngữ khác nhau. Một trong số những thuật ngữ quan trọng nhất của jQuery mà ta cần biết như:

- jQuery **Selectors**
- jQuery **Tag Name**
- **Tag ID**
- **Tag Class**
- **Function** trong jQuery
- **Callback** trong jQuery
- Các **Closure** trong jQuery
- **Proxy Pattern** trong jQuery
- Phạm vi (**Scope**) trong jQuery
- Các tham số trong jQuery
- Context trong jQuery

## 2.3 jQuery Selector.

jQuery Selectors được sử dụng để chọn và thao tác các phần tử HTML. Đây là phần rất quan trọng trong thư viện jQuery. Với jQuery Selectors, ta có thể tìm hoặc chọn các phần tử HTML dựa trên ID, classes, attributes, types và nhiều thứ khác từ DOM. Mỗi jQuery Selector bắt đầu bằng ký tự \$ (). Ký tự này được gọi là **factory function**.

Trong Javascript thuần, muốn lấy một phần tử trong HTML có cú pháp:  
`document.getElementById{Id|Class|Tagname}("Tên_element")...`

Trong jQuery để lấy ra một phần tử trong HTML ta chỉ cần sử dụng:  
`$(selector)`. Ví dụ:

- `$(This)` : Lấy phần tử hiện tại.
- `$("h1")` : Lấy các phần tử thẻ H1.
- `$(".TopDev")` : Lấy các phần tử có class "TopDev".
- `$("#TopDev")` : Lấy phần tử có ID là "TopDev".
- `$("*")` : Lấy tất cả phần tử.
- `$(p.intro)` : Lấy các phần tử p có class là "intro".
- `$(p:first)` : Lấy phần tử p đầu tiên trong tài liệu HTML.
- `$(p:last)` : Lấy phần tử p cuối cùng trong tài liệu HTML.
- `$(ul li:first)` : Lấy phần tử li đầu tiên trong phần tử ul.
- `$([href])` : Lấy các phần tử có thuộc tính href.
- `$(a[target='_blank'])` : Lấy các phần tử a có thuộc tính target='\_blank'.
- `$(a[target!='_blank'])` : lấy các phần tử a có thuộc tính target nhưng không phải là '\_blank'.
- `$(button)` : Lấy các phần tử button hoặc các phần tử input là button.
- `$(tr:even)` : Lấy các phần tử tr chẵn trong bảng.
- `$(tr:odd)` : Lấy các phần tử tr lẻ trong bảng.

### 2.3.1 jQuery TagName.

TagName đại diện cho một tên thẻ có sẵn trong DOM.

Ví dụ: `$(p)` chọn tất cả các đoạn văn 'p' trong tài liệu.

### 2.3.2 jQuery TagID

TagID đại diện cho một thẻ có sẵn với một ID cụ thể trong DOM.

Ví dụ: `$(#real-id)` chọn một phần tử cụ thể trong tài liệu có ID là real-id.

### 2.3.3 jQuery TagClass

Tag Class đại diện cho một tag có sẵn với một class cụ thể trong DOM.

Ví dụ: `$(.real-class)` chọn tất cả các phần tử trong tài liệu có class là real-class.

## 2.4 Callback trong jQuery

Callback là một hàm thuần JavaScript được truyền một số phương thức như là một tham số hoặc tùy chọn. Một số callback là các sự kiện được gọi để người sử dụng thực hiện kiểm tra một trạng thái nào đó được kích hoạt. Ví dụ:

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

Hầu hết callback cung cấp các tham số và một context. Trong ví dụ event-handler, callback được gọi với một tham số, một Event.

Hoặc một số callback được yêu cầu để trả một kết quả gì đó, cái khác trả về giá trị tùy ý. Để ngăn cản sự đệ trình (gửi dữ liệu đi) của form, một xử lý sự kiện Submit có thể trả về false như sau:

```
$("#myform").submit(function() {  
    return false;  
});
```

Cú pháp sử dụng sự kiện: `$(selector).sự_kiện(Mã_thực_thi)`

**jQuery có những sự kiện sau:**

Các sự kiện đến từ hành động của “chuột”

- o **click**: khi click vào đối tượng.
- o **dblclick**: khi double click vào đối tượng.
- o **mouseenter**: khi rê chuột vào đối tượng.
- o **mouseleave**: khi đưa chuột rời khỏi đối tượng.

Các sự kiện đến từ hành động của bàn phím

- o **keypress**: khi phím được nhấn vào đối tượng.
- o **keydown**: khi phím được nhấn xuống và giữ nguyên.
- o **keyup**: khi phím được thả ra.

Các sự kiện đến từ form

- o **submit**: khi nhấn gửi dữ liệu đi (loại button với type submit).
- o **change**: khi thay đổi giá trị trong thẻ input.
- o **focus**: khi người dùng nhấn vào thẻ input hoặc dùng nút tab chuyển đến thẻ input khác.
- o **blur**: khi rời khỏi thẻ input.

Các sự kiện đến từ trình duyệt

- o **load**: khi trang đang tải.
- o **resize**: khi trang thay đổi kích thước tổng thể.
- o **scroll**: khi trang đang cuộn lên xuống.
- o **unload**: khi trang đang được load lại.

## 2.5 Get/Set dữ liệu

Một số phương thức jQuery có thể được sử dụng để gán hoặc đọc một số giá trị trên vùng chọn. Một số phương pháp này là `text()`, `html()`, `attr()` và `val()`.

Khi các phương thức này được gọi mà không có đối số, nó được gọi là getters vì nó nhận (hoặc đọc) giá trị của phần tử. Khi các phương thức này được gọi với một giá trị làm đối số, nó được gọi là setter vì nó đặt (hoặc gán) giá trị đó.

### 1. jQuery `text()`

Phương thức jQuery `text()` được sử dụng để lấy nội dung văn bản kết hợp của các phần tử đã chọn, bao gồm cả phần tử con của chúng hoặc đặt nội dung văn bản của các phần tử đã chọn.

Ví dụ nhận nội dung với `text()`

```
<script>
$(document).ready(function(){
    // Nhận nội dung văn bản kết hợp của tất cả các đoạn văn
    $(".btn-one").click(function(){
        var str = $("p").text();
        alert(str);
    });

    // Lấy nội dung văn bản của đoạn đầu tiên
    $(".btn-two").click(function(){
        var str = $("p:first").text();
        alert(str);
    });
});
</script>
```

**Lưu ý:** jQuery `text()` truy xuất các giá trị của tất cả các phần tử đã chọn (tức là văn bản kết hợp), trong khi các getters khác như `html()`, `attr()` và `val()` chỉ trả về giá trị từ phần tử đầu tiên trong vùng chọn.

Ví dụ đặt nội dung với `text()`

```
<script>
$(document).ready(function(){
    // Đặt nội dung văn bản của tất cả các đoạn văn
    $(".btn-one").click(function(){
        $("p").text("This is demo text.");
    });

    // Đặt nội dung văn bản của đoạn đầu tiên
    $(".btn-two").click(function(){
        $("p:first").text("This is another demo text.");
    });
});
</script>
```

**Lưu ý:** Khi các phương thức jQuery `text()`, `html()`, `attr()` và `val()` được gọi với một giá trị làm đối số, nó sẽ đặt giá trị đó cho mọi phần tử phù hợp.

## 2. jQuery html()

Phương thức jQuery `html()` được sử dụng để lấy hoặc đặt nội dung HTML của các phần tử.

Ví dụ lấy nội dung HTML với `html()`

```
<script>
$(document).ready(function(){
    // Nhận nội dung HTML của đoạn được chọn đầu tiên
    $(".btn-one").click(function(){
        var str = $("p").html();
        alert(str);
    });

    // Nhận nội dung HTML của một phần tử có vùng chứa ID
    $(".btn-two").click(function(){
        var str = $("#container").html();
        alert(str);
    });
});
</script>
```

Lưu ý: Nếu nhiều phần tử được chọn, `html()` chỉ trả về nội dung HTML của phần tử đầu tiên từ tập hợp các phần tử phù hợp.

Ví dụ đặt nội dung HTML với `html()`

```
<script>
$(document).ready(function(){
    // Đặt nội dung HTML cho nội dung tài liệu
    $("button").click(function(){
        $("body").html("<p>Hello World!</p>");
    });
});
</script>
```

## 3. jQuery attr()

Phương thức jQuery `attr()` dùng để lấy giá trị thuộc tính của phần tử hoặc đặt một hoặc nhiều thuộc tính cho phần tử được chọn.

Ví dụ lấy giá trị thuộc tính với `attr()`

```
<script>
$(document).ready(function(){
    // Nhận giá trị thuộc tính href của siêu liên kết được chọn đầu tiên
    $(".btn-one").click(function(){
        var str = $("a").attr("href");
        alert(str);
    });

    // Nhận giá trị thuộc tính alt của hình ảnh có ID sky
    $(".btn-two").click(function(){
        var str = $("img#sky").attr("alt");
        alert(str);
    });
});
</script>
```

Lưu ý: Nếu nhiều phần tử được chọn, `attr()` chỉ trả về giá trị thuộc tính của phần tử đầu tiên trong tập hợp các phần tử.

Ví dụ đặt giá trị thuộc tính với `attr()`

```
<script>
$(document).ready(function(){
    // Kiểm tra tất cả các hộp kiểm
    $("button").click(function(){
        $('input[type="checkbox"]').attr("checked", "checked");
    });
});
</script>
```

#### 4. jQuery val()

Phương thức jQuery `val()` chủ yếu được sử dụng để lấy hoặc đặt giá trị hiện tại của các thành phần biểu mẫu HTML như `<input>`, `<select>` và `<textarea>`.

Ví dụ nhận và đặt giá trị với `val()`

```
<script>
$(document).ready(function(){
    // Nhận giá trị của văn bản nhập có tên ID
    $("button.get-name").click(function(){
        var name = $('input[type="text"]#name').val();
        alert(name);
    });

    // Nhận giá trị của vùng văn bản có nhận xét ID
    $("button.get-comment").click(function(){
        var comment = $("textarea#comment").val();
        alert(comment);
    });

    // Nhận giá trị của hộp chọn có ID thành phố
    $("button.get-city").click(function(){
        var city = $("select#city").val();
        alert(city);
    });
});
</script>
```

```
<script>
$(document).ready(function(){
    // Đặt giá trị của tất cả các đầu vào văn bản
    $("button").click(function(){
        var text = $(this).text();
        $('input[type="text"]').val(text);
    });
});
</script>
```

Lưu ý: Nếu nhiều phần tử biểu mẫu được chọn, `val()` chỉ trả về giá trị của phần tử đầu tiên từ tập hợp các phần tử.

## 2.6 Filter

jQuery cung cấp một số phương thức như `filter()`, `first()`, `last()`, `eq()`, `slice()`, `has()`, `not()` ... được sử dụng để thu hẹp tìm kiếm các phần tử trong cây DOM.

### jQuery first()

Phương thức jQuery `first()` lọc các phần tử phù hợp và trả về phần tử đầu tiên từ tập hợp đó. Ví dụ sau sẽ chỉ làm nổi bật phần tử `<li>` đầu tiên trong `<ul>` bằng cách thêm lớp `.highlight` vào tài liệu.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery first() Demo</title>
  <style>
    .highlight {
      background: yellow;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script>
    $(document).ready(function() {
      $("ul li").first().addClass("highlight");
    });
  </script>
</head>
<body>
  <ul>
    <li>First list item</li>
    <li>Second list item</li>
    <li>Third list item</li>
    <li>Last list item</li>
  </ul>
</body>
</html>
```



### jQuery last()

Phương thức jQuery `last()` lọc các phần tử phù hợp và trả về phần tử cuối cùng từ tập hợp. Ví dụ sau sẽ chỉ làm nổi bật phần tử `<li>` cuối cùng trong `<ul>`.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery last() Demo</title>
  <style>
    .highlight {
      background: yellow;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script>
    $(document).ready(function() {
      $("ul li").last().addClass("highlight");
    });
  </script>
</head>
<body>
  <ul>
    <li>First list item</li>
    <li>Second list item</li>
    <li>Third list item</li>
    <li>Last list item</li>
  </ul>
</body>
</html>
```

- First list item
- Second list item
- Third list item
- Last list item

**jQuery eq()**

Phương thức jQuery `eq()` lọc các phần tử phù hợp và chỉ trả về một phần tử có số chỉ mục được chỉ định. Ví dụ làm nổi bật phần tử `<li>` thứ hai trong `<ul>`.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery eq() Demo</title>
  <style>
    .highlight {
      background: yellow;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script>
    $(document).ready(function() {
      $("ul li").eq(1).addClass("highlight");
    });
  </script>
</head>
<body>
  <ul>
    <li>First list item</li>
    <li>Second list item</li>
    <li>Third list item</li>
    <li>Last list item</li>
  </ul>
</body>
</html>
```

- First list item
- Second list item
- Third list item
- Last list item

**Lưu ý:** Chỉ mục được cung cấp cho phương thức `eq()` cho biết vị trí dựa trên phần tử 0, nghĩa là chỉ mục 0 là phần tử đầu tiên, chỉ mục 1 là phần tử thứ hai, v.v.  
⇒ Chỉ mục này đề cập đến vị trí của phần tử trong đối tượng jQuery, không phải trong cây DOM.

Ngoài ra số chỉ mục âm cho biết vị trí bắt đầu từ cuối tập hợp, thay vì bắt đầu. Ví dụ: `eq(-2)` xác định phần tử cuối cùng thứ hai trong tập hợp các phần tử phù hợp.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>jQuery eq() Demo</title>
  <style>
    .highlight {
      background: yellow;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script>
    $(document).ready(function() {
      $("ul li").eq(-2).addClass("highlight");
    });
  </script>
</head>

<body>
  <ul>
    <li>First list item</li>
    <li>Second list item</li>
    <li>Third list item</li>
    <li>Fourth list item</li>
  </ul>
</body>
</html>
```

- First list item
- Second list item
- Third list item
- Fourth list item

### jQuery filter()

Phương thức jQuery `filter()` có thể lấy bộ chọn hoặc hàm làm đối số để lọc tập hợp các phần tử phù hợp dựa trên một tiêu chí cụ thể.

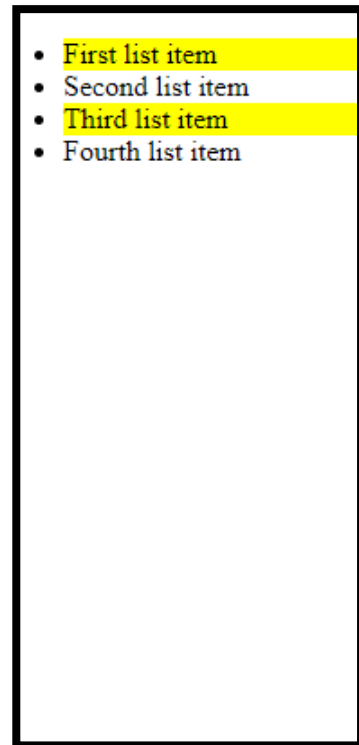
Bộ chọn hoặc hàm được cung cấp cho phương thức `filter()` được kiểm tra dựa trên từng phần tử trong tập hợp các phần tử phù hợp và tất cả các phần tử khớp với bộ chọn được cung cấp hoặc được kiểm tra bởi hàm sẽ được đưa vào kết quả.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>jQuery filter() Demo</title>
  <style>
    .highlight {
      background: yellow;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script>
    $(document).ready(function() {
      $("ul li").filter(":even").addClass("highlight");
    });
  </script>
</head>

<body>
  <ul>
    <li>First list item</li>
    <li>Second list item</li>
    <li>Third list item</li>
    <li>Fourth list item</li>
  </ul>
</body>

</html>
```



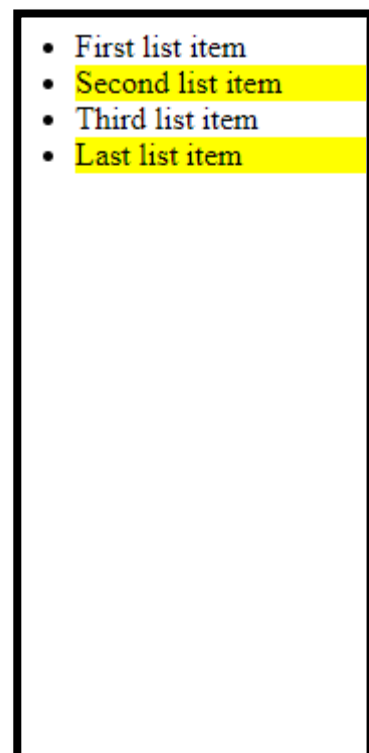
Ta cũng có thể truyền một hàm cho phương thức `filter()` để lọc tập hợp các phần tử phù hợp dựa trên các điều kiện nhất định. Ví dụ sau sẽ kiểm tra từng phần tử `<li>` trong `<ul>` và đánh dấu những phần tử `<li>` có chỉ mục là số lẻ, tức là chỉ làm nổi bật mục danh sách thứ hai và thứ tư vì chỉ mục dựa trên số 0.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>jQuery filter() Demo</title>
  <style>
    .highlight {
      background: yellow;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script>
    $(document).ready(function() {
      $("ul li").filter(function(index) {
        return index % 2 !== 0;
      }).addClass("highlight");
    });
  </script>
</head>

<body>
  <ul>
    <li>First list item</li>
    <li>Second list item</li>
    <li>Third list item</li>
    <li>Last list item</li>
  </ul>
</body>

</html>
```



### jQuery has()

Phương thức jQuery `has()` lọc tập hợp các phần tử phù hợp và chỉ trả về những phần tử có phần tử con được chỉ định. Ví dụ sau làm nổi bật tất cả các phần tử `<li>` của `<ul>` là phần tử con.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>jQuery filter() Demo</title>
  <style>
    .highlight {
      background: yellow;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script>
    $(document).ready(function() {
      $("ul li").has("ul").addClass("highlight");
    });
  </script>
</head>

<body>
  <ul>
    <li>Section 1</li>
    <li>Section 2</li>
    <li>
      <ul>
        <li>Section 2.1</li>
        <li>Section 2.2</li>
        <li>Section 2.3</li>
      </ul>
    </li>
    <li>Section 4</li>
  </ul>
</body>
</html>
```

- Section 1
- Section 2
- Section 2.1
- Section 2.2
- Section 2.3
- Section 4

### jQuery not()

Phương thức jQuery `not()` lọc tập hợp các phần tử phù hợp và trả về tất cả các phần tử không đáp ứng các điều kiện đã chỉ định. Nó có thể lấy bộ chọn hoặc một hàm làm đối số.

Bộ chọn hoặc hàm được cung cấp cho phương thức `not()` được kiểm tra dựa trên từng phần tử trong tập hợp các phần tử phù hợp và các phần tử không khớp với bộ chọn được cung cấp hoặc được kiểm tra bởi hàm sẽ được đưa vào kết quả.

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>jQuery not() Demo</title>
  <style>
    .highlight {
      background: yellow;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script>
    $(document).ready(function() {
      $("ul li").not(":even").addClass("highlight");
    });
  </script>
</head>

<body>
  <ul>
    <li>First list item</li>
    <li>Second list item</li>
    <li>Third list item</li>
    <li>Fourth list item</li>
  </ul>
</body>

</html>

```



Phương thức `not()` có thể lấy một hàm làm đối số tương tự `filter()`, nhưng nó hoạt động ngược lại với phương thức `filter()`.

Ví dụ sau kiểm tra từng phần tử `<li>` trong `<ul>` và đánh dấu các `<li>` có chỉ mục không phải là số lẻ.

```

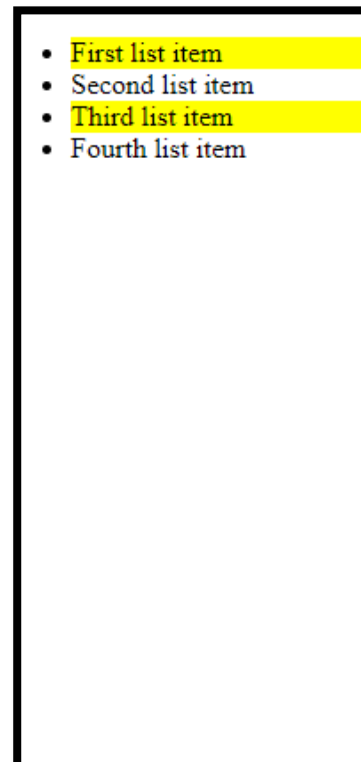
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>jQuery not() Demo</title>
  <style>
    .highlight {
      background: yellow;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script>
    $(document).ready(function() {
      $("ul li").not(function(index) {
        return index % 2 != 0;
      }).addClass("highlight");
    });
  </script>
</head>

<body>
  <ul>
    <li>First list item</li>
    <li>Second list item</li>
    <li>Third list item</li>
    <li>Fourth list item</li>
  </ul>
</body>

</html>

```



### jQuery slice()

Phương thức jQuery `slice()` lọc tập hợp các phần tử phù hợp được chỉ định bởi một *phạm vi* chỉ mục. Phương thức này chấp nhận số chỉ mục bắt đầu và kết thúc (tùy chọn) làm đối số, trong đó chỉ số bắt đầu chỉ định vị trí mà tại đó các phần tử bắt đầu được chọn và chỉ số kết thúc chỉ định vị trí mà tại đó các phần tử dừng được chọn.

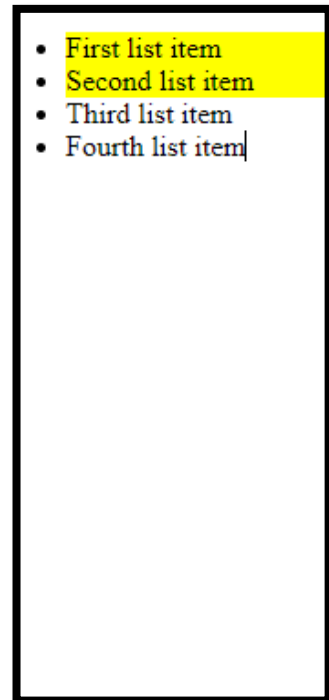
Ví dụ làm nổi bật phần tử `<li>` thứ nhất và thứ hai trong `<ul>` bằng cách thêm lớp `.highlight` vào.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>jQuery slice() Demo</title>
  <style>
    .highlight {
      background: yellow;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script>
    $(document).ready(function() {
      $("ul li").slice(0, 2).addClass("highlight");
    });
  </script>
</head>

<body>
  <ul>
    <li>First list item</li>
    <li>Second list item</li>
    <li>Third list item</li>
    <li>Fourth list item</li>
  </ul>
</body>

</html>
```



Ngoài ra cũng có thể sử dụng số chỉ mục âm để xác định vị trí bắt đầu từ cuối tập hợp. Ví dụ: phần `slice(-2, -1)` đánh dấu duy nhất mục danh sách thứ ba, vì đây là mục duy nhất trong phạm vi giữa hai mục từ cuối (-2) và một mục từ cuối (-1), vì vị trí cuối cùng không được đưa vào kết quả.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>jQuery slice() Demo</title>
  <style>
    .highlight {
      background: yellow;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script>
    $(document).ready(function() {
      $("ul li").slice(-2, -1).addClass("highlight");
    });
  </script>
</head>

<body>
  <ul>
    <li>First list item</li>
    <li>Second list item</li>
    <li>Third list item</li>
    <li>Fourth list item</li>
  </ul>
</body>
</html>
```

- First list item
- Second list item
- Third list item
- Fourth list item

## 2.7 Sử dụng Plug-in

Các plugin được tạo bởi người dùng jQuery dựa trên code trong thư viện jQuery. Các plugin trên kho lưu trữ jQuery UI chính thức có thể được coi là rất an toàn và mạnh mẽ, vì chúng được quản lý bởi cộng đồng chuyên nghiệp.

### *Các Plug-in jQuery phổ biến như:*

#### 1. *Effect*: Plug-in jQuery tạo hiệu ứng mạnh mẽ

Effect là một plugin jQuery đơn giản cho phép gán một số hiệu ứng animation cho các thành phần trên trang web.

#### 2. *TaggingJS*: Hỗ trợ tạo thẻ Tag phân loại nội dung

Plug-in TaggingJS của jQuery sẽ cho phép thêm hệ thống gắn thẻ vào trang web của mình một cách nhanh chóng. (Ví dụ Hashtag trên Facebook)

Chỉ với một thẻ tag, ta có thể nhóm các bài viết liên quan đến cùng chủ đề lại với nhau. Bằng cách này, Google có thể xếp hạng thẻ Tag của lên trên Google tìm kiếm.

#### 3. *Autocomplete*: Hỗ trợ dự đoán từ ngữ

Plugin Autocomplete của jQuery có tính năng tự động hoàn thành hoặc cung cấp thêm các đề xuất để hoàn thành nội dung đang gõ.

Plugin này cho phép thêm các thuật ngữ tự động hoàn thành của riêng mình vào danh sách để gia tăng trải nghiệm tìm kiếm tốt hơn.

#### 4. *Scrollmagic*: Plug in hỗ trợ hiệu ứng cuộn

Plugin ScrollMagic tác động các thành phần trang web dựa trên định vị của thanh cuộn. Ta có thể tạo animation khi cuộn trang hoặc đồng bộ hóa animation với vị trí cuộn.

#### 5. *Fine Uploader*: Plug in hỗ trợ chức năng upload

Plugin Fine Uploader cho phép kéo thả hình ảnh hoặc nhấp vào nút để chọn file hình ảnh từ máy tính.

#### 6. *Blueimp Gallery*: Plugin hỗ trợ tạo thư viện ảnh

Blueimp Gallery hỗ trợ tạo thư viện hình ảnh responsive (hiển thị tốt trên PC và Mobile). Nó có thể được điều khiển bằng bàn phím và chuột trên máy tính để bàn hoặc bằng cách vuốt trên điện thoại hoặc máy tính bảng.

Plugin này có thể được thiết lập để hiển thị hình ảnh hoặc video ở định dạng băng chuyền và cũng có thể hiển thị hình ảnh ở chế độ lightbox.

#### 7. *Slick*: Plug in hỗ trợ tạo carousel thú vị

Slick là một thư viện jQuery cho phép tạo Carousel với nhiều kiểu khác nhau thú vị hơn Blueimp Gallery ở trên.

Slick cho phép trình chiếu một hình ảnh hay nhiều hình ảnh, "lazy loading", và nhiều hiệu ứng khác.

#### 8. *Slider*: Plugin hỗ trợ tạo slide đẹp

Slide là một tính năng rất phổ biến trên các trang web. Plugin Slider này sử dụng thư viện jQuery để gán các giá trị số trên thanh ngang. Thanh trượt sau đó có thể được di chuyển lên và xuống thanh bằng phím mũi tên chuột hoặc bàn phím.

#### 9. *Infinite Ajax Scroll*: Cuộn không ngừng

Một trong những hàm liên quan đến JavaScript mà jQuery có thể đơn giản hóa dành cho các lập trình viên gọi là AJAX.

Kỹ thuật AJAX lấy nội dung từ máy chủ và tải nội dung lên chính trang người dùng đang xem mà không cần phải tải lại trang.

Plugin Infinite AJAX Scroll này sử dụng AJAX để tải thêm nội dung trên một trang khi người dùng tiếp tục cuộn xuống (Thay vì phải nhấn Next và tải trang mới).

⇒ Ví dụ hiệu ứng trên Facebook khi cuộn không ngừng trên newfeed và nội dung cứ tiếp tục đẩy về mà không cần load lại trang.

### 10. *AnimateScroll*: Plug in hỗ trợ cuộn đến vị trí định sẵn

Plugin AnimateScroll này giúp ta tạo ra các menu, khi người dùng click vào từng menu thì sẽ chuyển đến nội dung của phần đó.

Khi người dùng thao tác cuộn trên trang web, cuộn đến phần nội dung tương ứng thì hiệu ứng active cũng kích hoạt trên menu tương ứng.

### 11. *Jquery Validation*: Plug in hỗ trợ kiểm tra dữ liệu trên form

Plugin này cung cấp các phương thức kiểm tra định dạng của URL, email, số,... Ngoài ta cũng có thể sử dụng các validation method được cung cấp sẵn hoặc xây dựng các customized method.

Plugin này cung cấp 3 method chính:

- validate() – Thực hiện validate cho form được chọn.
- valid() – Kiểm tra xem các input trong form có hợp lệ hay không
- rules() – Đọc, thêm hoặc bớt các quy tắc xác định cho các input

## 3. Thời gian thực hành

120 phút

## 4. Đánh giá

Thang điểm tối đa: 10 điểm/Lab

## 5. Ví dụ

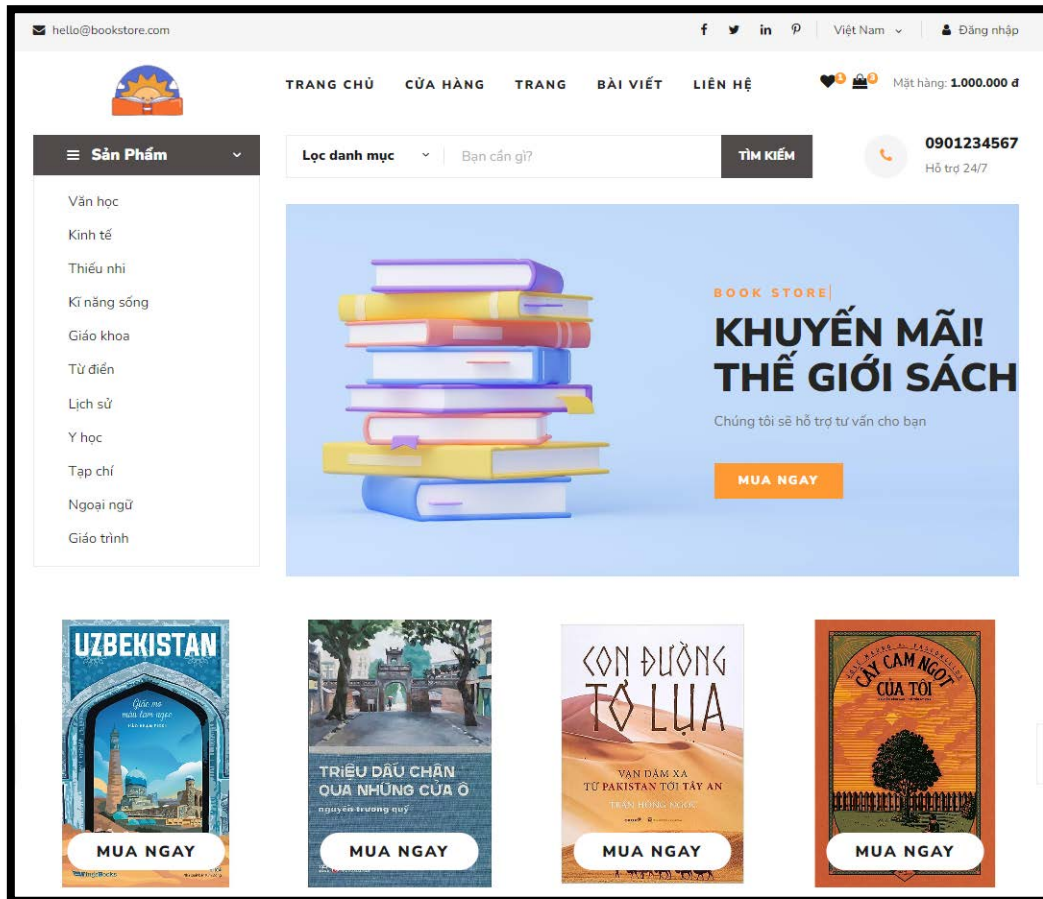
- Hãy chạy **Vi\_du\_1**, đây là ví dụ sử dụng jQuery để liệt kê danh sách:

The image shows two screenshots of a web form titled "Nhập danh sách". The left screenshot shows the initial state with an empty input field labeled "Thêm danh sách" and a "Thêm" button. The right screenshot shows the state after clicking "Thêm", with three rows of text: "Toán (2-4-6): 15h - 17h", "Lý (3-5-7) 13h - 15h", and "Hóa (3-5-7): 15h - 17h".

- ⇒ Ví dụ 1 sử dụng jQuery để chọn các phần tử HTML và gắn kết sự kiện. Khi người dùng nhập vào nút "Thêm", sự kiện click sẽ lấy giá trị từ ô nhập để thêm một mục mới vào danh sách phía dưới. Khi người dùng nhấp vào một mục trong danh sách, chúng ta đánh dấu mục đó là đã hoàn thành bằng cách

thêm hoặc xóa lớp completed. Sinh viên mở file **Lab09\Vi\_du\_1\js\script.js** để xem code xử lý.

- Trong **Vi\_du\_2**, sử dụng jQuery selectors và events vào trang BookStore.



⇒ Trong ví dụ này, sinh viên chú ý đến hiệu ứng khi nạp (load) trang. Khi load trang, click vào một liên kết hay khi responsive đều có hiệu ứng fadeOut. Các bạn mở file **Lab09\Vi\_du\_2\js\main.js** để xem mã code xử lý.

```

1 'use strict';
2
3 (function ($) {
4
5     /*-----
6     Preloader
7     -----*/
8     $window.on("load", function () {
9         $(".loader").fadeOut();
10        $(".preloader").delay(200).fadeOut("slow");
11
12        /* Mã này đang xử lý trình tải trước, là một thành phần giao diện người dùng thường
13        hiển thị hoạt ảnh đang tải trong khi nội dung của trang web đang được tải.
14        Trong trường hợp này, nó nhằm mục tiêu các phần tử có lớp .loader và #preloader
15        (có thể đại diện cho các phần tử trình tải và trình tải trước) và làm mờ chúng sau khi của số tải xong.
16        Ngoài ra còn có mã bổ sung cho bộ lọc thư viện bên trong trình xử lý sự kiện này. */
17
18        /*-----
19        Gallery Filter
20        -----*/
21        $(".featured_controls li").on("click", function () {
22            $(".featured_controls li").removeClass("active");
23            $(this).addClass("active");
24        });
25        if ($(".featured_filter").length > 0) {
26            var containerEl = document.querySelector(".featured_filter");
27            var mixer = mixitup(containerEl);
28        }
29
30        /* Mã này có liên quan đến bộ lọc thư viện. Khi một mục danh sách bên trong một phần tử có lớp
31        .featured_controls được nhấp vào, nó sẽ xóa lớp 'hoạt động' khỏi tất cả các mục danh sách và thêm nó vào mục được nhấp vào. Nó cũng kiểm tra xem một phần tử có lớp
32        .featured_filter có tồn tại hay không và nếu có, nó sẽ khởi tạo một plugin Mixitup (giả sử nó được bao gồm ở nơi khác trong mã). */
33    });
34
35    /*-----
36    Background Set
37    -----*/
38    $(".set-bg").each(function () {

```

## 6. Yêu cầu thực hành

**Bài 1:** (4đ) Hãy tạo form đăng nhập như bên dưới, sử dụng jQuery thực hiện các yêu cầu sau. Lưu thành *lab09\_bai1\_jqlogin.html*.

- Hãy tự tạo sẵn **3 tài khoản** đăng nhập ví dụ như:

```
var accounts = [  
  { username: 'user1', password: 'password1' },  
  { username: 'user2', password: 'password2' },  
  { username: 'user3', password: 'password3' }  
];
```

- Đăng nhập đúng thì hiện dòng chữ màu xanh “**Đăng nhập thành công**”;
- Đăng nhập sai thì hiện dòng chữ màu đỏ “**Đăng nhập thất bại, vui lòng kiểm tra lại thông tin**”;

### Login

Username:  Password:

### Login

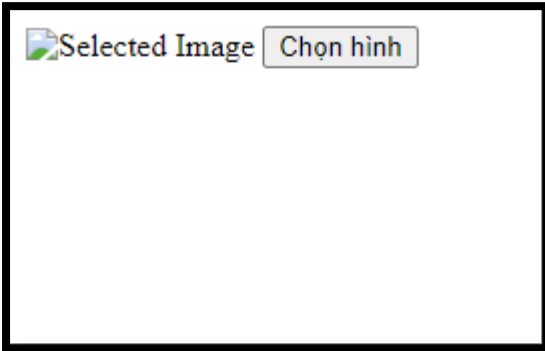
Username:  Password:    
**Đăng nhập thành công**

### Login

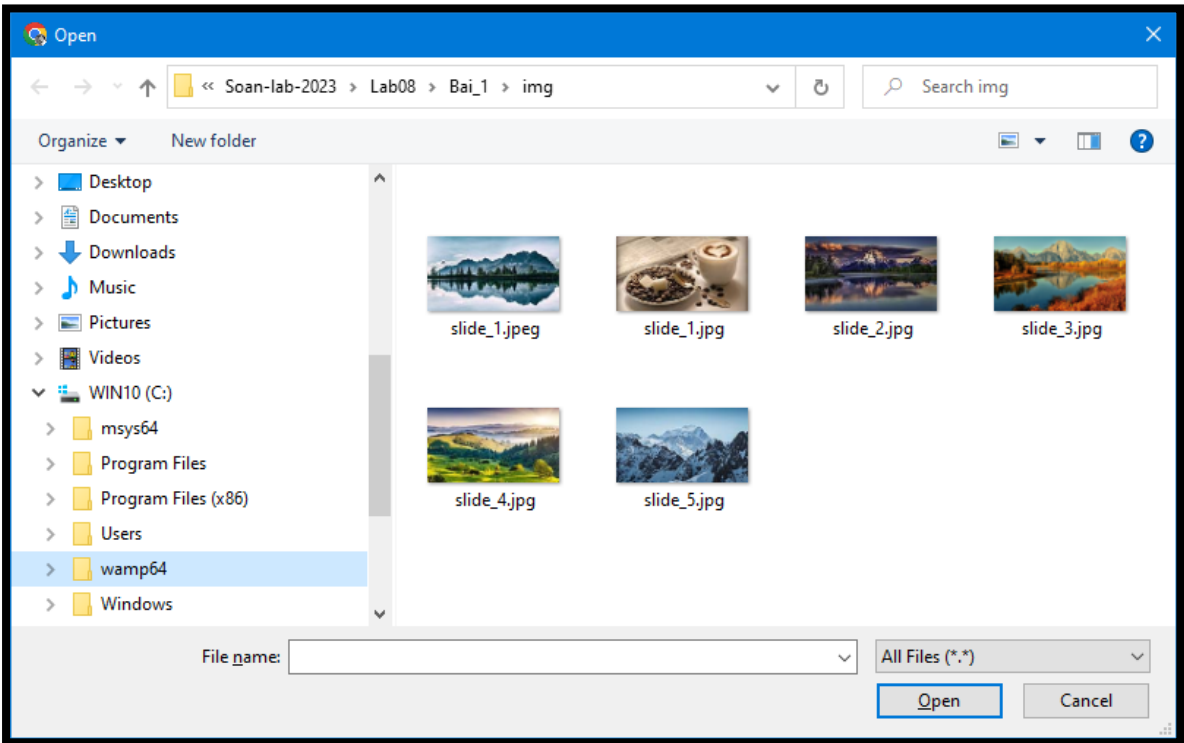
Username:  Password:    
**Đăng nhập thất bại, Vui lòng kiểm tra lại thông tin**

**Bài 2:** (4đ): Hãy tạo nút `<button>` **Chọn hình** có chức năng thêm file hình từ máy tính. Sử dụng jQuery để hiển thị hình đã chọn. Lưu thành *lab09\_bai2\_jqfile.html*.

**Gợi ý:** Sử dụng sự kiện click chuyển đến phần tử ẩn `fileInput` để mở cửa sổ chọn file. Sau khi người dùng chọn một file hình, chúng ta sử dụng `FileReader` để đọc nội dung của file và hiển thị nó trong phần tử hình ảnh (`selectedImage`).



```
$(document).ready(function() {  
    $('#changeImageButton').on('click',  
    function() {  
        $('#fileInput').click();  
    });  
    ...  
});
```



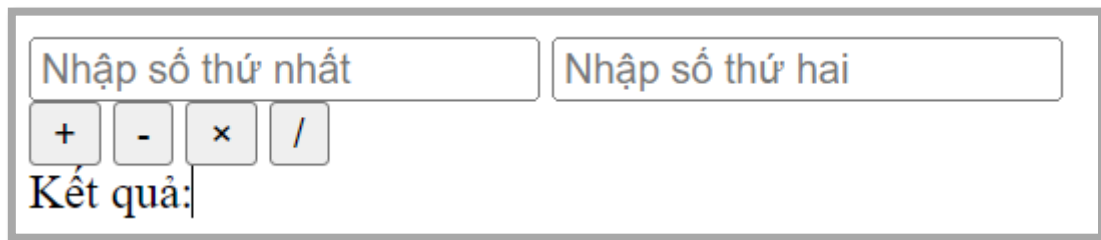
**Bài 3:** (2đ): Hãy tạo form tính các phép toán như bên dưới. Sử dụng jQuery với yêu cầu sau. Lưu thành *lab09\_bai3\_jqcaculate.html*.

- Có 2 input dùng để nhập 2 giá trị là số nguyên;
- Có 4 nút bấm: cộng , trừ, nhân, chia;
- Khi chọn nút bấm tương ứng sẽ xuất ra kết quả tương ứng.

Các ràng buộc điều kiện:

- Không chia cho 0;
- Nếu người dùng nhập thiếu giá trị thì sẽ không thực hiện phép tính;

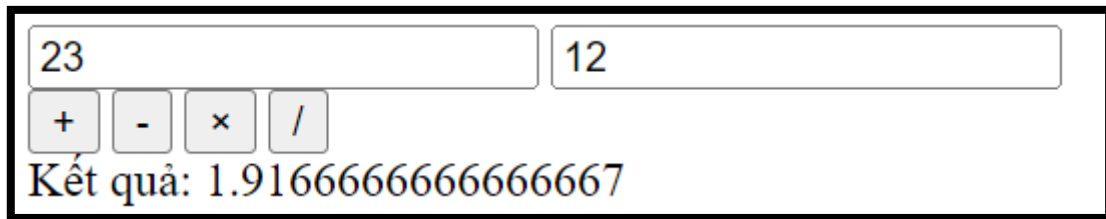
Ví dụ minh họa:



Nhập số thứ nhất      Nhập số thứ hai

+   -   ×   /

Kết quả:

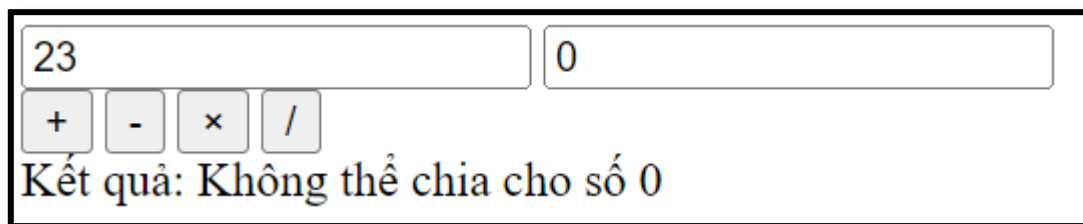


23      12

+   -   ×   /

Kết quả: 1.9166666666666667

Hình 1: Thực hiện phép chia

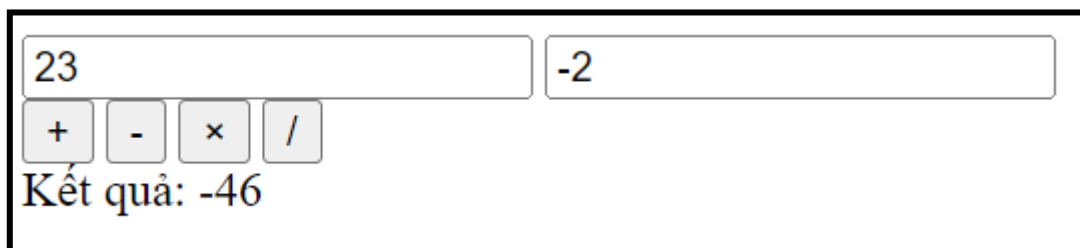


23      0

+   -   ×   /

Kết quả: Không thể chia cho số 0

Hình 2: Phép chia không hợp lệ



23      -2

+   -   ×   /

Kết quả: -46

Hình 3: Phép nhân với số âm