

Chương 1: BOOTSTRAP

1.1. Tổng quan về Bootstrap

1.1.1. Bootstrap là gì:

- Bootstrap là một nền tảng (framework) miễn phí, mã nguồn mở, dựa trên HTML, CSS & Javascript. Nó được tạo ra để xây dựng các giao diện Website tương thích với tất cả các thiết bị có kích thước màn hình khác nhau.
- Bootstrap bao gồm những đối tượng cơ bản có sẵn như: typography, forms, buttons, tables, navigation, modals, image carousels và nhiều thứ khác. Nó cũng có nhiều Component, Javascript hỗ trợ cho việc thiết kế Responsive dễ dàng, thuận tiện và nhanh chóng hơn.

1.1.2. Ưu điểm của Bootstrap

- Bootstrap giúp người thiết kế giao diện website tiết kiệm rất nhiều thời gian. Các thư viện Bootstrap có những đoạn mã sẵn sàng cho lập trình viên áp dụng vào website của mình mà không phải tốn quá nhiều thời gian để tự viết code cho giao diện.
- Bootstrap dễ dàng sử dụng, chỉ cần có kiến thức cơ bản về html, css
- Ưu điểm nổi bật nhất của bootstrap là khả năng tự động điều chỉnh kích thước trang web phù hợp với các loại màn hình: điện thoại, tablets, laptop hay máy tính để bàn.
- Bootstrap tương thích với tất cả các trình duyệt hiện đại như: chrome, firefox, internet explorer, cốc cốc, safari và opera.
- Bootstrap Versions:

Phiên bản	Ngày phát hành	Nội dung chính
1.0	19/08/2011	Phiên bản đầu tiên của Bootstrap , chưa hỗ trợ Mobile .
2.0	31/01/2012	Bổ sung hệ thống Grid-Layout 12 cột. Thêm một số thành phần (component) mới. Và thay đổi một vài thành phần sẵn có. Vẫn chưa hỗ trợ Mobile .
3.0	19/08/2013	Các thành phần được thiết kế lại theo phong cách thiết kế phẳng (flat design). Và lần đầu tiên hỗ trợ các thiết bị Mobile .
4.0	19/01/2018	Bootstrap 4 gần như viết lại hoàn toàn từ Bootstrap 3 , và được đánh giá là dễ sử dụng hơn rất nhiều so với phiên bản trước.

1.1.3. Cách cài đặt Bootstrap vào Website

Cách 1: Sử dụng Bootstrap từ thư viện CDN.

+ Thêm thẻ link này vào phần thẻ head của file html

```
<link rel="stylesheet"
```

```
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css">
```

+ Thêm các thẻ script này vào cuối thẻ body

```
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
```

```
<script
```

```
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
```

```
<script
```

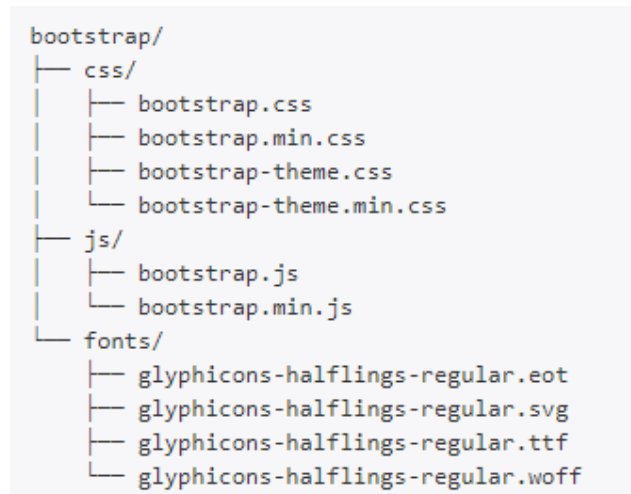
```
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>
```

Cách 2: Tải về và dùng trực tiếp local

- Bước 1: Download phiên bản Bootstrap tại <http://getbootstrap.com>



Cấu trúc các file của Bootstrap sau khi download về như sau:



- + Trong thư mục CSS, chúng ta chỉ cần quan tâm 2 file: **bootstrap.css** và **bootstrap.min.css**. Nội dung 2 file thì như nhau nhưng **bootstrap.css** là bản đầy đủ, và **bootstrap.min.css** là bản nén để khi sử dụng gọi vào sẽ giúp website load nhanh hơn.
- + Trong thư mục JS, chúng ta cũng chỉ quan tâm tới 2 file, đó là: file đầy đủ **bootstrap.js** và file nén là **file bootstrap.min.js**.

- Bước 2: Nhúng CSS, Javascript của Bootstrap vào HTML:
- + Thêm thẻ link, chỉ đường dẫn đến file bootstrap.css hoặc bootstrap.min.css:
`<link href="css/bootstrap.min.css" rel="stylesheet">`
- + Thêm thẻ script vào cuối thẻ body:
`<script src="js/jquery.min.js"></script>`
`<script src="js/bootstrap.min.js"></script>`

Chú ý: File *jquery.min.js* download tại *jquery.com* và chép vào thư mục JS

1.1.4. Cách sử dụng các thư viện của Bootstrap:

Để sử dụng thư viện Bootstrap, chỉ cần gọi tên class CSS phù hợp với thẻ HTML mà Bootstrap đã quy định sẵn tại trang chủ <http://getbootstrap.com>

Ví dụ:

```
<button class="btn btn-default" type="submit">Button</button>
```

Trong đó: btn và btn-default là tên class mà Bootstrap quy định

1.2. Các thư viện CSS trong Bootstrap

1.2.1. Container:

Trong thiết kế giao diện HTML – CSS thì việc sử dụng *container* rất quan trọng. Nó giúp tạo ra 1 khung có chiều rộng cố định nằm giữa trang web.

a. Class container-fluid

Lớp **container-fluid** khi được áp dụng cho một phần tử sẽ làm cho phần tử này có chiều rộng 100%.

Cú pháp:

```
<div class="container-fluid">  
    Nội dung  
</div>
```

b. Class .container

Dựa trên kích thước chiều rộng màn hình của các thiết bị, **Bootstrap4** chia chúng thành 5 loại:

- ☐ Các thiết bị có chiều rộng nhỏ hơn **567px** được gọi là **Extra Small** (Rất nhỏ).
- ☐ Các thiết bị có chiều rộng lớn hơn hoặc bằng **567px** được gọi là **Small** (Nhỏ), hoặc được gọi là thiết bị **sm**.

- ❑ Các thiết bị có chiều rộng lớn hơn hoặc bằng **768px** được gọi là **Medium** (Trung bình), hoặc được gọi là thiết bị **md**.
- ❑ Các thiết bị có chiều rộng lớn hơn hoặc bằng **992px** được gọi là **Large** (Lớn), hoặc được gọi là thiết bị **lg**.
- ❑ Các thiết bị có chiều rộng lớn hơn hoặc bằng **1200px** được gọi là **Extra Large** (Rất lớn), hoặc được gọi là thiết bị **xl**.

Cú pháp:

```
<div class="container">
  Nội dung
</div>
```

Chú ý:

- Nếu chiều rộng thiết bị nhỏ hơn **567px**, phần tử sẽ có chiều rộng 100%.
- Nếu chiều rộng thiết bị lớn hơn hoặc bằng **567px** và nhỏ hơn **768px** thì phần tử sẽ hiển thị ở giữa (center) và có chiều rộng **567px**.
- Nếu chiều rộng của thiết bị lớn hơn hoặc bằng **768px** và nhỏ hơn **992px**, phần tử sẽ hiển thị ở giữa (center) và có chiều rộng **768px**.
- Nếu chiều rộng của thiết bị lớn hơn hoặc bằng **992px** và nhỏ hơn **1200px**, phần tử sẽ hiển thị ở giữa (center) và có chiều rộng **992px**.
- Nếu chiều rộng của thiết bị lớn hơn hoặc bằng **1200px**, phần tử sẽ hiển thị ở giữa (center) và có chiều rộng **1200px**.

1.2.2 Grid System

Hệ thống lưới trong Bootstrap được chia thành 12 cột. Hệ thống lưới hoạt động hiệu quả khi đặt trong class `.container` (chiều rộng cố định) hoặc `.container-fluid` (chiều rộng full màn hình)

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
# of columns	12				
Gutter width	30px (15px on each side of a column)				

Cú pháp:

```
<!-- Thiết lập chiều rộng cho cột (column) theo từng thiết bị -->
<div class="row">
  <div class="col-*-*"></div>
```

```

        <div class="col-*-*"></div>
        <div class="col-*-*"></div>
    </div>

    <!-- Tự động bố trí theo Bootstrap -->
    <div class="row">
        <div class="col"></div>
        <div class="col"></div>
        <div class="col"></div>
    </div>

```

Chú ý: Một ưu điểm lớn trong Bootstrap4 là các cột(columns) nếu không được chỉ định chiều rộng sẽ tự động bố trí chiều rộng bằng nhau và chiều cao bằng nhau.

Ví dụ 1: Đối với các thiết bị có độ rộng màn hình < 576px, chúng ta sử dụng class **.col-** cho các column cần thiết lập. Để thiết lập 2 column với độ rộng màn hình lần lượt là 25% và 75% ta làm như sau:

```

<div class="container-fluid">
    <div class="row">
        <div class="col-3 bg-success">
            <p> Column 1 </p>
        </div>

        <div class="col-9 bg-warning">
            <p> Column 2</p>
        </div>
    </div>
</div>

```

Ví dụ 2: Đối với các thiết bị có độ rộng màn hình 992px – 1199px, chúng ta sử dụng class **.col-lg-** cho các column cần thiết lập. Để tạo ra 1 grid có 2 dòng: dòng 1 có 2 cột với tỷ lệ độ rộng lần lượt là 3; 9 và dòng 2 có 3 cột với độ rộng màn hình bằng nhau như hình sau:



Ta làm như sau:

```

<div class="container-fluid">

```

```
<div class="row">
  <div class="col-md-3 bg-primary">
    <p> Column 1 </p>
  </div>
  <div class="col-md-9 bg-secondary">
    <p> Column 2 </p>
  </div>
</div>
<div class="row">
  <div class="col-md bg-success">
    <p> Column 3 </p>
  </div>
  <div class="col-md bg-warning">
    <p> Column 4 </p>
  </div>
  <div class="col-md bg-primary">
    <p> Column 5 </p>
  </div>
</div>
</div>
```

1.1.4.1. Typography

- a. Mặc định văn bản thẻ <p> trong Bootstrap 4: Bootstrap 4 quy định sử dụng font chữ mặc định trên website có kích thước font-size là 16px, với chiều cao dòng là 1.5. Kiểu chữ mặc định là **“Helvetica Neue”, Helvetica, Arial, Sans-Serif**. Tất cả các phần tử <p> đều có margin-top: 0 và margin-bottom: 1rem (1rem = 16px).
- b. Có 6 thẻ tiêu đề (từ H1 đến H6) như sau:

```
<h1>h1. Bootstrap heading</h1>
<h2>h2. Bootstrap heading</h2>
<h3>h3. Bootstrap heading</h3>
<h4>h4. Bootstrap heading</h4>
<h5>h5. Bootstrap heading</h5>
<h6>h6. Bootstrap heading</h6>
```

Heading	Example
<code><h1></h1></code>	h1. Bootstrap heading
<code><h2></h2></code>	h2. Bootstrap heading
<code><h3></h3></code>	h3. Bootstrap heading
<code><h4></h4></code>	h4. Bootstrap heading
<code><h5></h5></code>	h5. Bootstrap heading
<code><h6></h6></code>	h6. Bootstrap heading

- c. Các tiêu đề display mở rộng tính phong phú cho các tiêu đề thông thường với font chữ lớn hơn và có độ đậm font chữ hơn.

Có 4 class display để lập trình viên có thể lựa chọn, như sau:

```
<h1 class="display-1">Display 1</h1>
<h1 class="display-2">Display 2</h1>
<h1 class="display-3">Display 3</h1>
<h1 class="display-4">Display 4</h1>
```

- d. Các class định nghĩa văn bản

- .font-weight-bold: in đậm
- .font-italic: in nghiêng
- .font-weight-light: in nhạt
- .font-weight-normal: in bình thường
- .lead: làm văn bản nổi bật
- .small: chữ nhỏ (85% bình thường)
- .text-left, .text-center, .text-right: canh trái, canh giữa, canh phải
- .text-*-left, .text-*-center, .text-*-right: canh trái, canh giữa, canh phải tùy theo màn hình small, medium, large hay xlarge.
- .text-monospace: văn bản khoảng trống đơn
- .text-nowrap: văn bản không bao
- .text-lowercase, .text-uppercase, .text-capitalize: văn bản thường, văn bản hoa chữ đầu, văn bản in hoa
- .text-justify: canh đều

1.1.4.2. Table

a. Table cơ bản:

```
<table class="table">
...
</table>
```

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

```
<table class="table table-dark">
```

```
</table>
```

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

b. Table head options

Thiết lập màu nền cho **Header** của bảng bằng cách sử dụng lớp **.thead-dark** hoặc **.thead-light** cho thẻ **<thead>**:

```
<table class="table">
  <thead class="thead-dark">
    ...
  </thead>
</table>
```

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

```
<table class="table">
  <thead class="thead-light">
    ...
  </thead>
</table>
```


#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

c.Striped Table

Sử dụng lớp **.table-striped** bạn có thể tạo ra một bảng có các kẻ sọc (Striped table), điều này có nghĩa là các dòng lẻ và các dòng chẵn trong **<tbody>** sẽ có màu nền khác nhau, nó giúp người dùng dễ dàng quan sát.

`<table class="table table-striped">`

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

`<table class="table table-striped table-dark">`

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

d. Bordered table:

`<table class="table table-bordered">`

Lớp **.table-bordered** sẽ tạo ra viền (border) cho 4 cạnh của bảng, và tất cả các cạnh của tất cả các ô của bảng.

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

<table class="table table-bordered table-dark">

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

<table class="table table-borderless">

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

<table class="table table-borderless table-dark">

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

e. Hoverable rows

Lớp **.table-hover** giúp tạo ra hiệu ứng (effect) đổi màu nền khi con trỏ (pointer) di chuyển phía trên các dòng (row) của bảng (Chỉ các dòng trong <tbody>).

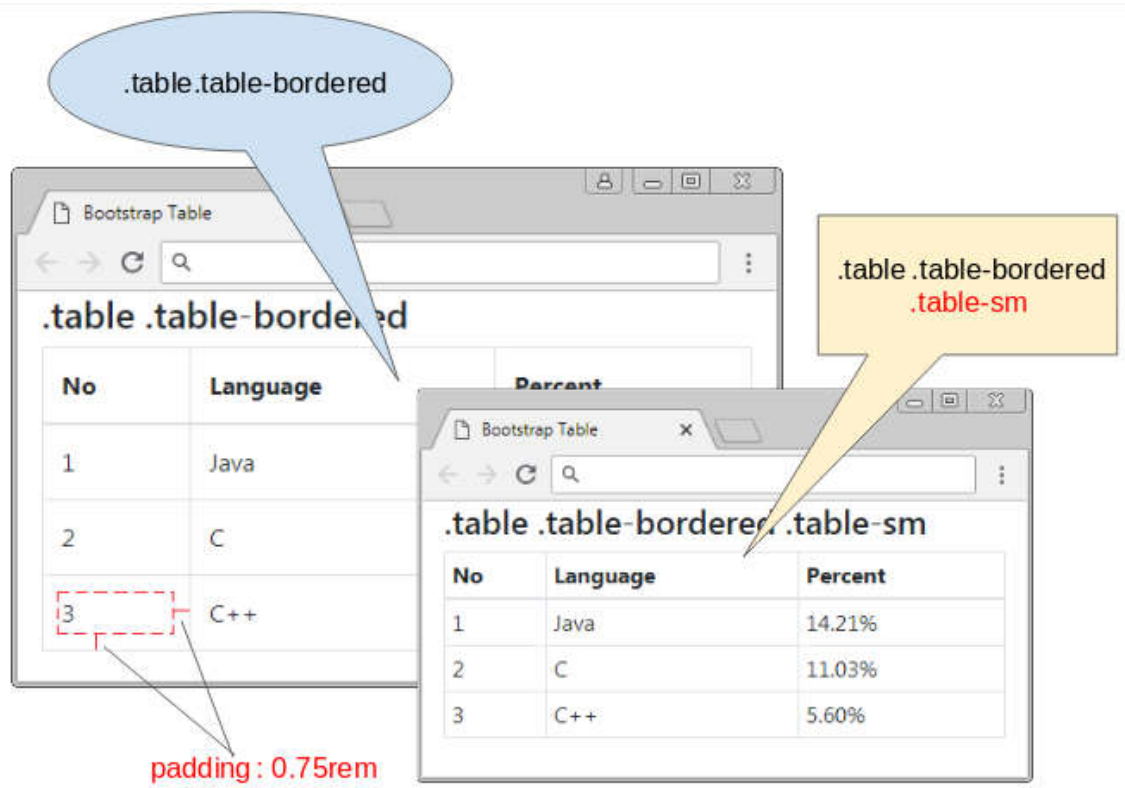
<table class="table table-hover">

<table class="table table-hover table-dark">

f. Small table

Bootstrap thiết lập **padding** cho các ô của bảng, giá trị này mặc định là **0.75rem**, lớp **.table-sm** tạo ra một bảng nhỏ gọn hơn bằng cách giảm giá trị của **padding** đi một nửa.

```
<table class="table table-sm">
```



g. Contextual classes

Có thể dùng các **lớp bối cảnh**(Contextual Classes) để định nghĩa màu sắc cho bảng. Các lớp này có thể dùng cho toàn bộ bảng (table), các dòng (tr) hoặc các ô trong bảng (td). các lớp bối cảnh sử dụng là:

Class	Heading	Heading
Active	Cell	Cell
Default	Cell	Cell
Primary	Cell	Cell
Secondary	Cell	Cell
Success	Cell	Cell
Danger	Cell	Cell
Warning	Cell	Cell
Info	Cell	Cell
Light	Cell	Cell
Dark	Cell	Cell

Ví dụ:

```
<!-- On rows -->
<tr class="table-active">...</tr>
<!-- On cells (`td` or `th`) -->
<tr>
  <td class="table-active">...</td>
</tr>
```

h. Responsive tables

Với nhu cầu thiết kế website đáp ứng trên các kích thước màn hình ở các thiết bị khác nhau (laptop, desktop, tablet, mobile) thì lớp **.table-responsive** sẽ giúp bảng biểu hiển thị trên tất cả màn hình mà không bị bể giao diện.

```
1 <div class="table-responsive">
2   <table class="table">
3     nội dung bảng biểu cần định nghĩa...
4   </table>
  </div>
```

Lớp **.table-responsive** sẽ tạo ra một bảng đáp ứng: một thanh cuộn ngang được thêm vào bảng nếu kích thước màn hình trình duyệt bé hơn 992px. Còn khi kích thước màn hình lớn hơn 992px thì không có gì thay đổi.

Tùy theo kích thước màn hình, có thể định nghĩa khi nào cần dùng thanh cuộn:

Css Class	Mô tả
.table-responsive	Nếu bảng không thể tự thu nhỏ chiều rộng của nó hơn nữa, thanh cuộn nằm ngang sẽ xuất hiện.
.table-responsive-sm	Nếu chiều rộng của thiết bị (hoặc trình duyệt) nhỏ hơn 567px và bảng không thể tự thu nhỏ chiều rộng của nó hơn nữa, thanh cuộn nằm ngang sẽ xuất hiện.
.table-responsive-md	Nếu chiều rộng của thiết bị (hoặc trình duyệt) nhỏ hơn 768px và bảng không thể tự thu nhỏ chiều rộng của nó hơn nữa, thanh cuộn nằm ngang sẽ xuất hiện.
.table-responsive-lg	Nếu chiều rộng của thiết bị (hoặc trình duyệt) nhỏ hơn 992px và bảng không thể tự thu nhỏ chiều rộng của nó hơn nữa, thanh cuộn nằm ngang sẽ xuất hiện.
.table-responsive-xl	Nếu chiều rộng của thiết bị (hoặc trình duyệt) nhỏ hơn 1200px và bảng không thể tự thu nhỏ chiều rộng của nó hơn nữa, thanh cuộn nằm ngang sẽ xuất hiện.

Ví dụ định nghĩa bảng đáp ứng nếu kích thước bé hơn 576px.

```

1 <div class="table-responsive-sm">
2   <table class="table">
3     nội dung bảng --- dammio.com
4   </table>
5 </div>
```

1.1.4.3. Form

Bootstrap có 3 cách để bố trí các phần tử form trên giao diện:

- Sử dụng lớp .form-group để bố trí các phần tử theo hướng thẳng đứng
- Sử dụng lớp .form-inline để bố trí các phần tử theo hướng nằm ngang.
- Đặt các phần tử Form trên 1 hệ thống lưới và khi đó các phần tử sẽ hiển thị theo quy tắc hệ thống lưới Grid System của Bootstrap.

a. Form-group:

Để bố trí các phần tử nằm theo hướng thẳng đứng thì cần bao bọc chúng bởi khối thẻ <div> với class .form-group. Các phần tử này sẽ có chiều rộng lấp đầy phần tử cha.

Email address

We'll never share your email with anyone else.

Password

☐ Check me out

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-
describedby="emailHelp" placeholder="Enter email">
    <small id="emailHelp" class="form-text text-muted">We'll never share your email
with anyone else.</small>
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1"
placeholder="Password">
  </div>
  <div class="form-group form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

b. Form-inline

Để bố trí các phần tử hiển thị trên 1 hàng bạn có thể bao bọc chúng bởi thẻ **<div class="form-inline">**. Tuy nhiên chúng có thể nằm trên các dòng khác nhau nếu phần tử cha có chiều rộng quá bé.

☐ Remember me

```
<form class="form-inline">
  <label class="sr-only" for="inlineFormInputName2">Name</label>
  <input type="text" class="form-control mb-2 mr-sm-2" id="inlineFormInputName2"
placeholder="Jane Doe">

  <label class="sr-only" for="inlineFormInputGroupUsername2">Username</label>
```

```

<div class="input-group mb-2 mr-sm-2">
  <div class="input-group-prepend">
    <div class="input-group-text">@</div>
  </div>
  <input type="text" class="form-control" id="inlineFormInputGroupUsername2"
placeholder="Username">
</div>

<div class="form-check mb-2 mr-sm-2">
  <input class="form-check-input" type="checkbox" id="inlineFormCheck">
  <label class="form-check-label" for="inlineFormCheck">
    Remember me
  </label>
</div>

<button type="submit" class="btn btn-primary mb-2">Submit</button>
</form>

```

c. Form Grid

Hệ thống lưới (Grid System) là một hệ thống mạnh mẽ để bố trí các phần tử trên giao diện, có thể sử dụng Grid System để áp dụng cho **Form**.

Email	Password	
<input type="text" value="Email"/>	<input type="text" value="Password"/>	
Address		
<input type="text" value="1234 Main St"/>		
Address 2		
<input type="text" value="Apartment, studio, or floor"/>		
City	State	Zip
<input type="text"/>	<input type="text" value="Choose..."/>	<input type="text"/>
<input type="checkbox"/> Check me out		
<input type="button" value="Sign in"/>		

```

<form>
  <div class="form-row">
    <div class="form-group col-md-6">
      <label for="inputEmail4">Email</label>
      <input type="email" class="form-control" id="inputEmail4"
placeholder="Email">
    </div>

```

```

<div class="form-group col-md-6">
  <label for="inputPassword4">Password</label>
  <input type="password" class="form-control" id="inputPassword4"
placeholder="Password">
</div>
</div>
<div class="form-group">
  <label for="inputAddress">Address</label>
  <input type="text" class="form-control" id="inputAddress" placeholder="1234
Main St">
</div>
<div class="form-group">
  <label for="inputAddress2">Address 2</label>
  <input type="text" class="form-control" id="inputAddress2"
placeholder="Apartment, studio, or floor">
</div>
<div class="form-row">
  <div class="form-group col-md-6">
    <label for="inputCity">City</label>
    <input type="text" class="form-control" id="inputCity">
  </div>
  <div class="form-group col-md-4">
    <label for="inputState">State</label>
    <select id="inputState" class="form-control">
      <option selected>Choose...</option>
      <option>...</option>
    </select>
  </div>
  <div class="form-group col-md-2">
    <label for="inputZip">Zip</label>
    <input type="text" class="form-control" id="inputZip">
  </div>
</div>
<div class="form-group">
  <div class="form-check">
    <input class="form-check-input" type="checkbox" id="gridCheck">
    <label class="form-check-label" for="gridCheck">
      Check me out
    </label>
  </div>
</div>
<button type="submit" class="btn btn-primary">Sign in</button>
</form>

```

Tóm lại: Đây là 3 kiểu form chính. Các kiểu form khác có thể xem chi tiết trên link:

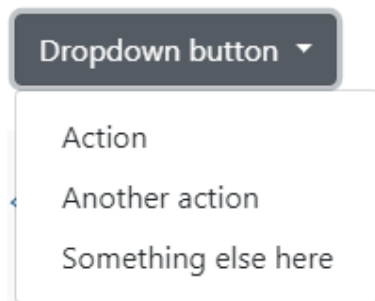
<https://getbootstrap.com/docs/4.3/components/forms/>

1.1.4.4. List group

Cras justo odio
Dapibus ac facilisis in
Morbi leo risus
Porta ac consectetur ac
Vestibulum at eros

```
<ul class="list-group">
  <li class="list-group-item">Cras justo odio</li>
  <li class="list-group-item">Dapibus ac facilisis in</li>
  <li class="list-group-item">Morbi leo risus</li>
  <li class="list-group-item">Porta ac consectetur ac</li>
  <li class="list-group-item">Vestibulum at eros</li>
</ul>
```

1.1.4.5. Dropdown



```
<div class="dropdown">
  <button class="btn btn-secondary dropdown-toggle" type="button"
  id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true" aria-
  expanded="false">
    Dropdown button
  </button>
  <div class="dropdown-menu" aria-labelledby="dropdownMenuButton">
    <a class="dropdown-item" href="#">Action</a>
    <a class="dropdown-item" href="#">Another action</a>
    <a class="dropdown-item" href="#">Something else here</a>
  </div>
</div>
```

1.1.4.6. Button

Bootstrap đã xây dựng sẵn một vài lớp **Css** để tạo một **Button**, với một vài kiểu dáng (style) dành cho các mục đích ngữ cảnh khác nhau. Trong đó lớp chính là **.btn**, và một vài lớp bổ sung để thiết lập màu sắc, kích thước, trạng thái...

a. Simple Button



```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

b. Button tags



```
<a class="btn btn-primary" href="#" role="button">Link</a>
<button class="btn btn-primary" type="submit">Button</button>
<input class="btn btn-primary" type="button" value="Input">
<input class="btn btn-primary" type="submit" value="Submit">
<input class="btn btn-primary" type="reset" value="Reset">
```

c. Outline Button



```
<button type="button" class="btn btn-outline-primary">Primary</button>
<button type="button" class="btn btn-outline-secondary">Secondary</button>
<button type="button" class="btn btn-outline-success">Success</button>
<button type="button" class="btn btn-outline-danger">Danger</button>
<button type="button" class="btn btn-outline-warning">Warning</button>
<button type="button" class="btn btn-outline-info">Info</button>
<button type="button" class="btn btn-outline-light">Light</button>
<button type="button" class="btn btn-outline-dark">Dark</button>
```

d. Kích thước Button:

Mặc định các **Button** trong **Bootstrap** có một kích thước vừa (Medium). Sử dụng lớp **.btn-sm** nếu bạn muốn có **Button** kích thước nhỏ hơn, sử dụng lớp **.btn-lg** nếu bạn muốn có **Button** với kích thước lớn.



```
<button type="button" class="btn btn-primary btn-lg">Large button</button>  
<button type="button" class="btn btn-secondary btn-lg">Large button</button>
```



```
<button type="button" class="btn btn-primary btn-sm">Small button</button>  
<button type="button" class="btn btn-secondary btn-sm">Small button</button>
```

e. Trạng thái Button

Sử dụng lớp **.active** để làm cho một **Button** có trạng thái giống như nó đang được kích hoạt (active). Với **Bootstrap** một **Button** ở trạng thái **"active"** sẽ có màu đậm hơn so với chính nó khi ở trạng thái thông thường. Sử dụng lớp **.disabled** để vô hiệu hóa một **Button**, người dùng sẽ không thể tương tác với **Button** này.

1.1.4.7. Image

- a. Responsive images: Để thực hiện hiển thị hình ảnh reponsive trên các thiết bị, chỉ cần thêm lớp **.img-fluid** vào hình cần thêm. Khi đó lớp này sẽ áp dụng phong cách **max-width: 100%;** và **height: auto;** vào hình ảnh.

```

```

- b. Image thumbnails: Nếu muốn tạo hình thu nhỏ có khung viền xung quanh sử dụng lớp **.img-thumbnail**

```

```

- c. Aligning images

Hai class **.float-right** và **.float-left** dùng để canh chỉnh hình ảnh xuất hiện ở bên trái hoặc bên phải nội dung

```
  

```

Để căn giữa hình ảnh, bạn có thể dùng class **.mx-auto** (margin:auto) và **.d-block** (display:block) vào hình ảnh cần canh giữa.

1.1.4.8. Menu trong Bootstrap

a. Menu nằm ngang

Active Link Link Disabled

```
<ul class="nav">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#" tabindex="-1" aria-
disabled="true">Disabled</a>
  </li>
</ul>
```

b. Canh lề cho menu:

Bootstrap cung cấp cho một vài lớp để căn lề cho **Nav**:

Lớp	Mô tả
.justify-content-start	Căn lề trái
.justify-content-center	Căn lề giữa
.justify-content-end	Căn lề phải

c. Responsive Nav:

Mặc định **Bootstrap Nav** là một Menu nằm ngang. Nhưng bạn có thể làm cho nó hiển thị thẳng đứng bằng cách sử dụng lớp **.flex-column**

Active

Link

Link

Disabled

```
<ul class="nav flex-column">
  <li class="nav-item">
```

```

    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#" tabindex="-1" aria-
disabled="true">Disabled</a>
  </li>
</ul>

```

d. Tab Nav

Active Link Link Disabled

```

<ul class="nav nav-tabs">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#" tabindex="-1" aria-
disabled="true">Disabled</a>
  </li>
</ul>

```

e. Tabs with Dropdowns Nav

Active Dropdown ▼ Link Disabled

```

<ul class="nav nav-tabs">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#"
role="button" aria-haspopup="true" aria-expanded="false">Dropdown</a>
    <div class="dropdown-menu">
      <a class="dropdown-item" href="#">Action</a>
      <a class="dropdown-item" href="#">Another action</a>
      <a class="dropdown-item" href="#">Something else here</a>
    </div>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>

```

```

    <div class="dropdown-divider"></div>
    <a class="dropdown-item" href="#">Separated link</a>
  </div>
</li>
<li class="nav-item">
  <a class="nav-link" href="#">Link</a>
</li>
<li class="nav-item">
  <a class="nav-link disabled" href="#" tabindex="-1" aria-
disabled="true">Disabled</a>
</li>
</ul>

```

1.1.4.9. Breadcrumb trong Bootstrap

Breadcrumb chính là thanh điều hướng trên website. Từ đó khách truy cập sẽ biết mình đang ở mục nào.

Home

Home / Library

Home / Library / Data

```

<nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item active" aria-current="page">Home</li>
  </ol>
</nav>

```

```

<nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item"><a href="#">Home</a></li>
    <li class="breadcrumb-item active" aria-current="page">Library</li>
  </ol>
</nav>

```

```

<nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item"><a href="#">Home</a></li>
    <li class="breadcrumb-item"><a href="#">Library</a></li>
    <li class="breadcrumb-item active" aria-current="page">Data</li>
  </ol>
</nav>

```

1.1.4.10. Các công cụ khác tham khảo trên trang Web trực tiếp sau:

<https://getbootstrap.com/docs/4.3/components/>

1.1.2 JQUERY

1.1.2.1 Định nghĩa

jQuery là một framework JavaScript có khả năng hỗ trợ đa trình duyệt (cross- browser). Bên cạnh đó, jQuery đơn giản hóa lập trình với JavaScript, tiết kiệm thời gian và công sức lập trình. Sử dụng jQuery cung cấp cho người lập trình một số tiện ích sau:

- Truy cập các phần tử của tài liệu web.
- Thay đổi hình thức/giao diện trang web.
- Thay đổi nội dung trang web.
- Tương tác với người dùng với các hiệu ứng động.
- Tương tác Ajax

1.1.2.2 Thêm thư viện vào trang Web

Bước 1: Truy cập địa chỉ <http://www.jquery.com> và download phiên bản cần dùng

[Download the compressed, production jQuery 3.4.1](#)

Bước 2: Tham chiếu đến thư viện theo cấu trúc như sau:

```
<script src="/đường_dẫn_đến_tệp_tin_jquery"></script>
```

Chú ý: Nếu không muốn tải tệp tin *jquery-3.4.1.min.js* theo cách trên, có thể sử dụng trực tiếp tệp tin *jquery-3.4.1.min.js* từ Google CDN, Microsoft CDN, Max CDN, ...như cách dùng Bootstrap.

1.1.2.3 Viết câu lệnh JQuery:

Các câu lệnh trong JQuery nằm trong cặp thẻ `<script></script>`

Cú pháp của JQuery: `$(selector).action()`

- + \$: xác định JQuery
- + (selector): truy vấn tới thành phần HTML
- + Action: thể hiện hành động trên thành phần được chọn

1.1.2.4 Bộ chọn (JQuery Selector)

Như vậy vấn đề quan trọng bậc nhất khi học JQuery là bộ chọn (selector). Để xử lý các thẻ bạn cần phải biết cách xác định nó. Bộ chọn gồm 2 phần là bộ chọn cơ bản và bộ lọc.

- Truy xuất nội dung (element) trong document dựa trên biểu thức selector cung cấp. Selector sử dụng cú pháp tương tự CSS.
- Tập kết quả do Selector và Filter trả về: JQuery objects (không phải DOM objects).

- Cú pháp và cách chọn tương tự CSS

□ a. Các bộ chọn cơ sở

SELECTOR	VÍ DỤ	Ý NGHĨA
<tên thẻ>	\$("p")	Chọn tất cả các thẻ có tên là <tên thẻ>
#<định danh>	\$("#basic")	Chọn tất cả các thẻ có ID là <định danh>
.<tên lớp>	\$(".java")	Chọn tất cả các thẻ với thuộc tính class có giá trị là <tên lớp>
<tên thẻ>.<tên lớp>	\$("li.app")	Chọn tất cả các thẻ có tên thẻ là <tên thẻ> với thuộc tính class có giá trị là <tên lớp>
*	\$("*")	Chọn tất cả các element trên document.

b. Các bộ chọn quan hệ phân cấp

SELECTOR	VÍ DỤ	Ý NGHĨA
Selector1,selector2 ...	\$("#basic,.java")	Chọn tất cả các thẻ được xác định bởi tất cả các bộ chọn
Parent > Child	\$("body>.app")	Chọn tất cả các thẻ con trực tiếp của các thẻ thuộc bộ chọn <Parent>
Ancestor Descendant	\$("body .java")	Chọn tất cả các thẻ con, cháu của các thẻ thuộc bộ chọn <Ancestor>
Prev + Next	\$(".java+p")	Chọn một thẻ nằm kế tiếp của cả các thẻ thuộc bộ chọn <Prev>
Prev ~ Siblings	\$("p.java~p")	Chọn tất cả các thẻ em của các thẻ thuộc bộ chọn <Prev>

Ví dụ: \$("body > *") – xác định tất cả các thẻ nằm trong thẻ <body>

`$("body>.java, #basic .app")` – xác định tất cả các thẻ `<p class='java'>` trên toàn bộ trang và các thẻ bất kỳ có `class='app'` nằm trong các thẻ có `id='#basic'`

c. Form Selector

SELECTOR	Ý NGHĨA
<code>:input</code>	Chọn tất cả thẻ input, textarea trên Form
<code>:text</code>	Chọn tất cả text field trên Form
<code>:password</code>	Chọn tất cả password field
<code>:radio</code>	Chọn tất cả radio button
<code>:checkbox</code>	Chọn tất cả checkbox
<code>:submit</code>	Chọn tất cả button submit
<code>:reset</code>	Chọn tất cả button reset
<code>:image</code>	Chọn tất cả image
<code>:button</code>	Chọn tất cả generalized button

1.1.2.5 Bộ lọc (jQuery Filter)

jQuery Selector thường trả về 1 tập đối tượng. jQuery Filter được dùng để lọc trên kết quả chọn của jQuery Selector. jQuery cung cấp các bộ lọc sau đây hỗ trợ xây dựng điều kiện tìm kiếm các thẻ chính xác và nhanh hơn. Có 6 loại bộ lọc:

- Basic: lọc phần tử ở vị trí đầu tiên, cuối cùng, chẵn, lẻ,
- Content: lọc dựa trên nội dung
- Visibility: lọc dựa trên trạng thái hiển thị của element
- Attribute: lọc dựa trên thuộc tính của element
- Child: lọc dựa trên mối quan hệ với element cha
- Form: lọc trên các thành phần khai báo trên Form

a. Bộ lọc cơ bản

FILTER	Ý NGHĨA
<code>:first</code>	Chọn phần tử đầu tiên trong tập kết quả do Selector trả về
<code>:last</code>	Chọn phần tử cuối cùng trong tập kết quả do Selector trả về
<code>:even</code>	Chọn phần tử chẵn
<code>:odd</code>	Chọn phần tử lẻ
<code>:eq (index)</code>	Chọn phần tử tại vị trí index
<code>:gt (index)</code>	Chọn phần tử có vị trí > index

:lt (index)	Chọn phần tử có vị trí < index
:header	Chọn tất cả header element (H1, H2, .. H6)
:not (selector)	Chọn phần tử không thỏa selector

b. Bộ lọc thuộc tính

FILTER	Ý NGHĨA
[attribute]	Lọc các phần tử có khai báo attribute
[attribute=value]	Lọc các phần tử có attribute với giá trị = value
[attribute!=value]	Lọc các phần tử có attribute với giá trị != value
[attribute^=value]	Lọc các phần tử có attribute với giá trị bắt đầu là value
[attribute\$=value]	Lọc các phần tử có attribute với giá trị kết thúc là value
[attribute*=value]	Lọc các phần tử có attribute chứa giá trị value
[attributeFilter1+ *attributeFilter2+...	Lọc các phần tử thỏa tất cả các attribute filter.

c. Bộ lọc nội dung

FILTER	Ý NGHĨA
:contains(text)	Lọc các phần tử có chứa chuỗi text
:empty	Lọc các phần tử rỗng
:has(selector)	Lọc các phần tử có chứa ít nhất 1 element thỏa selector
:parent	Lọc các phần tử có ít nhất 1 con hoặc text (Lấy các thẻ không rỗng)

d. Bộ lọc con cháu

FILTER	Ý NGHĨA
--------	---------

:nth-child(index) :nth-child(even) :nth-child(odd)	Lọc các phần tử theo vị trí so với cha của nó
:nth-child(equation)	Lọc phần tử theo vị trí (vị trí thỏa phương trình tham số) so với cha của nó
:first-child	Lấy phần tử đầu tiên so với cha của nó
:last-child	Lấy phần tử cuối cùng so với cha của nó
:only-child	Lấy phần tử nếu phần tử này là con duy nhất so với cha của nó

d. Xử lý tập kết quả

METHOD	Ý NGHĨA
size()	Lấy số phần tử trong tập kết quả của Selector
get()	Lấy tập DOM elements trong tập kết quả của Selector
get(index)	Lấy DOM element ở vị trí index
find(expression)	Lấy các element con cháu thỏa expression
each()	Gọi thực thi phương thức với từng element trong tập kết quả của Selector

1.1.2.6 Xử lý sự kiện:

a. Các sự kiện thường gặp:

EVENT	Ý NGHĨA	VÍ DỤ
click(func)	Nhấp chuột	<code>\$(“a:first”).click(function(),alert(“Hello”)-)</code>
dblclick(func)	Nhấp đúp chuột	<code>\$(“:button”).click(function(),alert(“Hello”)-)</code>
mouseover(func)	Chuột đi vào	<code>\$(“img”).mouseover(function(){\$(“div”).toggleClass("highlight");})</code>
mouseout(func)	Chuột đi ra	<code>\$(“img”).mouseout(function(){\$(“div”).toggleClass("highlight");})</code>

mousedown(func)	Đè chuột	<code>\$(“:text”).mousedown(function(),\$(this).css(“color”, “red”)-)</code>
mouseup(func)	Nhả chuột	<code>\$(“:text”).mouseup(function(),\$(this).css(“color”, “black”)-)</code>
mousemove(func)	Di chuột	<code>\$(“#b”). mousemove (function(),alert(“Hello”)-)</code>
keydown(func)	Đè phím	<code>\$(“:text”). keydown (function(),\$(this).css(“color”, “red”)-)</code>
keypress(func)	Gõ phím	<code>\$(“#b”).click(function(),alert(“Hello”)-)</code>
keyup(func)	Nhả phím	<code>\$(“:text”).mouseup(function(),\$(this).css(“color”, “black”)-)</code>
submit(func)	Submit form	<code>\$(“#b”). submit (function(),alert(“Good luck”)-)</code>
blur(func)	Mất tích cực	<code>\$(“#b”). blur (function(),alert(“Hello”)-)</code>
focus(func)	Tích cực	<code>\$(“#b”). focus (function(),alert(“Hello”)-)</code>
hover(func1, func2)	Vào/ra	<code>\$(“#b”). hover(function(),alert(“Hi”)-, function(),alert(“Bye”)-)</code>

b. Đối tượng Event:

Type	Loại event xảy ra, ví dụ: “click”
target	Element mà event xảy ra
data	Dữ liệu được truyền vào handler bởi phương thức bind
pageX, pageY	Tọa độ chuột khi event xảy ra
preventDefault ()	Ngăn trình duyệt không thực thi xử lý mặc định, ví dụ khi click vào liên kết

1.1.2.7. Các hiệu ứng:

Hiệu ứng thông thường có 2 loại là xuất hiện và biến mất với một hình thức nào đó. JQuery cung cấp rất nhiều hiệu ứng, giúp chúng ta tạo ra trang web thật sự sinh động và hấp dẫn. Trong phần này xin được giới thiệu các hiệu ứng thường gặp sau:

- Ấn, hiện element
- Fade-in, fade-out
- Sliding
- Di chuyển element
- Hiệu ứng chuyển động tùy chỉnh

Hầu hết các phương thức hiệu ứng đều có 2 tham số tùy chọn là speed và callback. Speed qui định thời gian kéo dài của hiệu ứng giá trị của nó là **“slow”**, **“normal”**, **“fast”** hoặc **milli giây**. Callback là hàm gọi ngược tức hàm được thực hiện sau khi hiệu ứng hoàn tất.

a. Ấn/ hiện Element

METHOD	Ý NGHĨA	VÍ DỤ
show ()	Hiện thị các element thỏa selector nếu trước đó bị ẩn	
show(speed, callback)	Hiện thị các element thỏa selector nếu trước đó bị ẩn, speed xác định tốc độ hiển thị. Sau khi hiển thị xong, phương thức callback sẽ được thực thi.	<code>\$("#div1").show("normal", function(), alert("Hello"));</code>
hide ()	Ẩn element nếu trước đó đang hiển thị.	
hide(speed, callback)	Ẩn element nếu trước đó đang hiển thị, tham số có ý nghĩa tương tự phương thức show.	<code>\$("#div1").hide("slow");</code> <code>\$("#div1").hide(4000);</code>
toggle ()	Chuyển qua lại trạng thái ẩn/hiện các element.	
toggle(speed, callback)	Chuyển qua lại trạng thái ẩn/hiện các element, tham số có ý nghĩa tương tự phương thức show.	<code>\$("#div1").toggle("fast");</code>

b. Fade in/fade out

METHOD	Ý NGHĨA	VÍ DỤ
fadeIn(speed,callback)	Hiện thị element bằng cách tăng dần độ trong suốt.	<code>\$("#div1").fadeIn("normal");</code>
fadeOut(speed,callback)	Ẩn element bằng cách giảm dần độ trong suốt về 0, sau đó thiết lập style display là none.	<code>\$("#div1").fadeOut("slow");</code>
fadeTo(speed,opacity,callback)	Thay đổi độ trong suốt của element.	<code>\$("#div1").fadeTo("slow",0.3, function() {alert("finished");}); \$("#div1").fadeTo("slow",1.0);</code>

c. Sliding

METHOD	Ý NGHĨA	VÍ DỤ
slideDown(speed,callback)	Hiện thị element bằng cách tăng chiều cao.	<code>\$("#div1").slideDown("slow");</code>
slideUp(speed,callback)	Ẩn element bằng cách giảm chiều cao.	<code>\$("#div1").slideUp("normal");</code>
slideToggle(speed,callback)	Chuyển đổi trạng thái ẩn/hiện element.	<code>\$("#div1").slideToggle(3000);</code>

d. Custom Animation

Sử dụng phương thức `animate()` để tạo hiệu ứng tùy chỉnh. Bạn có cơ hội tạo ra các hiệu ứng cho riêng mình bằng cách điều chỉnh các tham số truyền vào phương thức. Trong trường hợp bạn muốn dừng hiệu ứng thì bạn chỉ việc gọi phương thức `stop`.

```
$(“selector”).animate(properties,*duration+  
*easing+,*callback+); $(“selector”).stop();
```

THAM SỐ	Ý NGHĨA	VÍ DỤ
properties	Các thuộc tính xác trạng thái hiển thị sau khi animate.	<code>,width:”100”, color:”red”-</code>
duration	Animate kéo dài trong bao lâu (“slow”, “normal”, “fast”,	1000

	milisecond)	
easing	Hiệu ứng xóa : swing, linear	swing
Callback	Hàm được gọi sau khi animate xong	function(),alert("Hell o")-

1.1.2.8 Form Validation

- Không cho để trống ô nhập...
- Dữ liệu nhập vào phải theo một khuôn dạng nhất định nào đó: email, creditcard, url...
- Dữ liệu phải nhập vào phải đúng kiểu: số nguyên, số thực, ngày giờ...
- Dữ liệu nhập vào phải có giá trị tối thiểu, tối đa, trong phạm vi...
- Dữ liệu nhập phải đúng theo một kết quả tính toán riêng của bạn...

Ví dụ sau đây kiểm tra ô nhập "Name" phải nhập ít nhất 3 ký tự và ô nhập "Age" phải nhập số từ 25 đến 65.

```
$(document).ready(function () {
    $("#form1").validate(
    {
        rules:
        {
            txtName: { required: true, minlength: 3 },
            txtAge: { required: true, digits: true, range: [25,65] }
        },
        messages:
        {
            txtAge: { digits: "Nhập số !" },
            txtName: { required: "Không để trống !", minlength: "Ít nhất
3 ký tự
!" } },

        errorLabelContainer: "#myError",
        wrapper: "li",
        submitHandler: function (form) {
            if (confirm("Dữ liệu form đã hợp lệ. Bạn có muốn submit không
?")) {
                form.submit();
            }
        }
    });
});
```