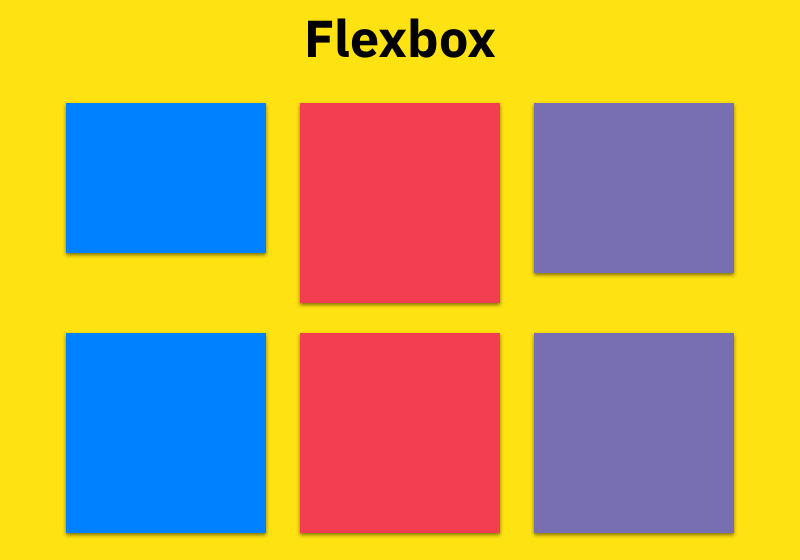
**Tìm hiểu về Flexbox và Grid:**

- Flexbox và Grid là một kỹ thuật dàn trang vô cùng tiện lợi có khả năng cân đối được các phần tử bên trong tự động trên mọi thiết bị điện tử như máy tính, desktop, điện thoại hay máy tính bảng.

Hay hiểu đơn giản hơn, bạn sẽ không phải thiết lập các kích thước của các phần tử, không cần cho nó float mà chỉ cần hiển thị chiều ngang hay chiều dọc của các phần tử đó theo mong muốn của bạn.

[](https://bizflyportal.mediacdn.vn/bizflyportal/images/fle16189054147826.jpeg)

Bạn nên dùng Flexbox để thiết lập bố cục trong phạm vi nhỏ như những khung trong website. Nếu là bố cục có phạm vi lớn hơn như chia cột website thì vẫn nên sử dụng dàn trang theo dạng Grid

**Các thuộc tính cơ bản của Flexbox**

Để hiểu rõ hơn Flexbox và khả năng dàn trang với Flexbox có hiệu quả thì bạn nên tìm hiểu và nắm vững thêm một số những thuộc tính cơ bản của flexbox dưới đây.

***flex-direction***

Khi bạn kích hoạt flex cho một phần tử thì các phần tử con trong đó sẽ được bố trí liên tục theo một hướng gọi là hướng chính. Và thuộc tính flex-direction chính là thuộc tính thiết lập lên hướng chính đó. Nếu bạn đã có hướng chính rồi thì hướng thứ hai sử dụng đến sẽ là hướng vuông góc với hướng chính.

***flex-flow***

Flex-flow có khả năng cho phép bạn căn chỉnh item theo một kích thước hợp lý với trang và mong muốn của mình.

[](https://bizflyportal.mediacdn.vn/bizflyportal/images/fle16189057984473.jpg)

Đưa ra một ví dụ giúp bạn dễ hiểu hơn như sau: Nếu bạn đặc tả các item là một thì container sẽ giúp sắp xếp và phân bố các item này trở lên cân bằng với nhau. Còn nếu bạn có 3 item và bạn đặc tả một trong số đó là hai thì item đó sẽ gấp đôi hai 2 item còn lại.

***flex-wrap***

Theo mặc định, các item sẽ được hiển thị trên cùng một hàng. Tuy nhiên, nếu bạn có nhu cầu hay mong muốn tách nó thành nhiều hàng hay nhóm các item lại với nhau khi dàn trang với Flexbox thì có thể sử dụng các chức năng của flex-wrap.

***justify-content***

Thuộc tính justify-content giúp đổi các phần tử trôi về một phía bất kỳ của hướng chính bằng cách nhận các giá trị như:

* flex-start: Các phần tử con được đẩy trôi về khu vực bắt đầu hướng chính.
* flex-end: Các phần tử được đẩy về khu vực cuối hướng chính.
* center: Các phần tử được đẩy vào giữa hướng chính.
* pace-around: Chia đều khoảng cách các phần tử nếu có không gian thừa.
* space-between: Tương tự như pace-around. Tuy nhiên, phần tử đầu và cuối sát méo khung.

**Một số thuật ngữ liên quan đến Flexbox**

Đến đây có lẽ bạn đã có thể hiểu rõ hơn Flexbox và những thuộc tính của nó rồi. Tuy nhiên, để dàn trang với Flexbox hiệu quả, mang lại lợi ích triệt để khi sử dụng thì bạn cũng nên hiểu rõ một số thuật ngữ cơ bản sau đây.

[](https://bizflyportal.mediacdn.vn/bizflyportal/images/fle16189060216726.jpg)

* Container: Container là thành phần bao quanh các phần tử bên trong giúp bạn có thể thiết lập các item này theo các kiểu hiển thị như sắp xếp theo chiều ngang hoặc sắp xếp theo chiều dọc.
* Item: Item là các phần tử con được bao trọn trong container. Khi thiết lập item, bạn có thể quy định nó sẽ được sử dụng bao nhiêu cột trong một container hoặc thiết lập thứ tự hiển thị của nó.
* main start, main end: Đây được xem là điểm bắt đầu (main start) và điểm kết thúc (main end) của container khi thiết lập flexbox. Điều này có thể hiểu đơn giản là các item bên trong container sẽ được hiển thị từ điểm bắt đầu cho đến điểm kết thúc. Cross start và cross end có ý nghĩa tương tự như main start, main end nhưng lại là trục vuông góc với nó.
* main axis: Đây chính là trục chính giúp điều hướng các item sẽ hiển thị. Trên màn hình chính main axis, item hiển thị theo chiều dọc. Bạn có thể sử dụng flex-direction để thay đổi trục main axis và item sẽ hiển thị theo đó. Trục vuông góc của main axis là cross axis.
* main size: Dựa theo trục main axis để xác định kích thước ( chiều rộng hoặc dọc) của item.
* cross size: Dựa theo trục cross axis để xác định kích thước (chiều rộng hoặc dọc) của item.
* Dislay: flexbox hiển thị container theo chiều dọc hoặc chiều dài dựa vào những giá trị sẵn có.

Flexbox chính là một kỹ thuật dàn trang hiệu quả và hữu ích với nhiều tính năng nổi bật có khả năng thay thế cho phương thức dàn trang truyền thống. Hy vọng, với những thông tin mà Bizfly đã cung cấp trong bài viết, bạn đã có thể hiểu được Flexbox là gì, những thuộc tính và một số thuật ngữ có liên quan đến Flexbox

**Tìm hiểu về Bootstrap**

## Bootstrap là gì?

Bạn có thể đã quen thuộc với chức năng của frameworks là gì. Nó tập hợp các cú pháp dành riêng cho nhiệm nào đó trong một bộ sưu tập để giúp các web developer xây dựng [**website**](https://www.hostinger.vn/huong-dan/lam-the-nao-de-tao-trang-web) nhanh hơn nhiều, vì họ không cần phải lo về các lệnh và chức năng cơ bản.

Mặc dù vậy, vẫn thiếu sự thống nhất do việc các thư viện được sử dụng quá nhiều, cần phải thay đổi liên tục. Bootstrap là câu trả lời cho vấn đề này.

Front-end framework là mã nguồn mở, ban đầu được xây dựng bởi Mark Otto và Jacob Thornton để lập trình web front-end nhanh hơn và dễ dàng hơn.

Nó chứa tất cả các mẫu thiết kế dựa trên [**HTML**](https://www.hostinger.vn/huong-dan/html-cheat-sheet-co-html5-tags-moi) và CSS với nhiều chức năng và thành phần khác nhau, như điều hướng, hệ thống lưới, băng chuyền hình ảnh và các nút.

Mặc dù Bootstrap giúp tiết kiệm thời gian của nhà phát triển khỏi việc phải quản lý các template nhiều lần, nhưng mục tiêu chính của nó là tạo ra các responsive website. Nó cho phép giao diện người dùng của trang web có thể hoạt động tối ưu trên mọi kích thước màn hình, trên điện thoại màn hình nhỏ hoặc máy tính để bàn màn hình lớn.

Do đó, các nhà phát triển không cần xây dựng các trang web dành riêng cho từng thiết bị và bị giới hạn phạm vi đối tượng nữa.

Do sự phổ biến của nó, ngày càng nhiều cộng đồng Bootstrap xuất hiện. Đó là một nơi tuyệt vời để các nhà phát triển và nhà thiết kế trao đổi kiến ​​thức và thảo luận về các bản vá framework mới nhất.

## 3 file chính của Bootstrap

Vì [**Bootstrap**](https://getbootstrap.com/) bao gồm một tập hợp các cú pháp thực hiện các chức năng cụ thể, điều đó chỉ có lý khi framework có các loại file khác nhau trong đó. Dưới đây là ba file chính quản lý giao diện người dùng và chức năng của trang web.

### Bootstrap.css

Bootstrap.css là một framework CSS sắp xếp và quản lý bố cục của trang web. Trong khi HTML quản lý nội dung và cấu trúc của trang web, CSS xử lý bố cục của trang web. Vì lý do đó, cả hai cấu trúc cần cùng tồn tại để thực hiện một hành động cụ thể.

Do các chức năng của nó, CSS cho phép bạn tạo giao diện thống nhất trên bao nhiêu trang web tùy thích. Giờ thì bạn có thể nói lời tạm biệt với việc phải ngồi hàng giờ để chỉnh sửa thủ công chỉ để thay đổi độ rộng của đường viền.

Với CSS, tất cả những gì bạn cần làm là giới thiệu các trang web đến file CSS. Bất kỳ thay đổi cần thiết có thể được thực hiện trong file đó một mình.

Các hàm CSS không chỉ giới hạn ở các kiểu văn bản vì chúng có thể được sử dụng để định dạng các khía cạnh khác của trang web như bảng và bố cục hình ảnh.

Vì CSS có rất nhiều khai báo và bộ chọn, việc ghi nhớ tất cả những cái đó có thể mất chút thời gian.

### Bootstrap.js

File này là phần cốt lõi của Bootstrap. Nó bao gồm các file [**JavaScript**](https://www.hostinger.vn/huong-dan/javascript-la-gi) chịu trách nhiệm cho việc tương tác của trang web.

Để tiết kiệm thời gian khi viết cú pháp JavaScript nhiều lần, các nhà phát triển có xu hướng sử dụng jQuery. Nó có một thư viện JavaScript đa nền tảng, mã nguồn mở phổ biến cho phép bạn thêm các chức năng khác nhau vào một trang web.

Dưới đây là một vài ví dụ về những gì jQuery có thể làm:

* Thực hiện các yêu cầu Ajax như loại trự dữ liệu từ một vị trí khác một cách linh hoạt
* Tạo tiện ích bằng bộ sưu tập plugin JavaScript
* Tạo hình động tùy chỉnh bằng các thuộc tính CSS
* Thêm tính năng động cho nội dung trang web

Mặc dù Bootstrap với các thuộc tính CSS và element HTML có thể hoạt động tốt, nhưng nó cần jQuery để tạo ra thiết kế responsive. Nếu không, bạn chỉ có thể sử dụng các phần tĩnh của CSS.

Bạn cần thêm thông tin? Hãy xem[**bài viết**](https://www.hostinger.vn/huong-dan/jquery-la-gi) này để biết thêm về jQuery.

### Glyphicons

Icons là một phần không thể thiếu của giao diện trang web. Chúng thường được liên kết với các hành động và dữ liệu nhất định trong giao diện người dùng. Bootstrap sử dụng Glyphicons để đáp ứng nhu cầu đó.

Bootstrap bao gồm một bộ Halflings Glyphicons đã được mở khóa để sử dụng miễn phí. Phiên bản miễn phí có giao diện chuẩn nhưng phù hợp với các chức năng thiết yếu.

Nếu bạn muốn tìm icon có phong cách hơn, [**Glyphicons**](https://www.glyphicons.com/) cũng bán các bộ icon premium khác nhau, chắc chắn sẽ trông đẹp hơn trên từng trang web cụ thể.

Bạn cũng có thể tải các biểu tượng riêng biệt và theo chủ đề xuống miễn phí trên các trang web khác nhau như [**Flaticon**](https://www.flaticon.com/packs/ultimate-glyphicons), [**GlyphSearch**](https://glyphsearch.com/) và [**Icons8**](https://icons8.com/icons/set/glyph).

Một số icon có thể bị ảnh hưởng bởi CSS để thay đổi giao diện trong khi các icon khác thì mặc định. Hãy sử dụng icons phù hợp nhất với trang web bạn cần.

## Cách sử dụng Bootstrap

Để hiểu được cách sử dụng bootstrap, hãy xem ví dụ dưới đây.

html lang="en">

<head>

<meta charset="utf-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>Bootstrap 101 Template</title>

<link href="css/bootstrap.min.css" rel="stylesheet" />

</head>

<body>

<h1>Hello, world!</h1>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

</body>

</html>

Mã hóa ký tự cho tài liệu HTML. Trong trường hợp này, UTF-8 đề cập đến Unicode.

meta charset="utf-8"

Cho biết bộ ký tự được sử dụng để viết trang web.

meta http-equiv="X-UA-Compatible"

Xác định phiên bản Internet Explorer sẽ hiển thị trang. Sử dụng chế độ Edge, nó được cài đặt để sử dụng chế độ cao nhất hiện có.

meta name="viewport"

Đảm bảo rằng trang có tỷ lệ 1: 1 với kích thước khung nhìn.

link href="css/bootstrap.min.css" rel="stylesheet"

Đây là phần chúng ta thêm Bootstrap core CSS.

src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"

Tải jQuery thông qua Google CDN. Sẽ tốt hơn để bạn tải nó từ CDN qua HTTP vì file có thể được lưu trong bộ nhớ cache trong một năm.

src="js/bootstrap.min.js

Thêm JavaScript lõi Bootstrap. Cú pháp này phải luôn nằm dưới cú pháp jQuery để hoạt động đúng. Quá trình bổ sung có thể được thực hiện thông qua URL của [**Google**](https://www.google.com/) hoặc tải xuống thủ công.

## Kết luận

Bootstrap là front-end framework miễn phí ngày càng phổ biến hơn trong số các nhà phát triển front-end. Các nhà phát triển có thể dễ dàng sử dụng và tiết kiệm rất nhiều thời gian khi  khỏi phải viết code thủ công.

Framework này cũng rất linh hoạt và có thể đáp ứng hầu hết mọi nhu cầu phát triển front end web. Các khả năng tốt nhất của nó bao gồm, nhưng không giới hạn tính năng responsive web tối ưu trên tất cả các kích thước màn hình.

Nếu bạn là nhà phát triển front end, bạn cần biết Bootstrap là gì và đây là thời điểm hoàn hảo để bạn bắt đầu sử dụng Bootstrap

**Tìm hiểu về Tailwind CSS**

# Tailwind CSS là gì ?

Thì nó là một utility-first CSS framework, nó cũng giống như Bootstrap, nó có những class built-in mà chúng ta có thể dùng. Tailwind CSS có nhiều các class bao gồm các thuộc tính CSS khác nhau và quan trọng, chúng ta có thể dễ dàng mở rộng tạo mới ra những class bằng chính những class của nó.

# Tại sao chúng ta nên dùng Tailwind ?

Nói chung là nó cũng na ná Boostrap thôi nhưng một điều tiện lợi khi dùng framework này là chúng ta có nhiều class mới hơn tiện lợi hơn Boostrap. Các bạn có thể tham khảo [tại đây](https://nerdcave.com/tailwind-cheat-sheet) trong Tailwind. Và hơn nữa, việc có nhiều thêm những class nhưng với quy tắc đặt tên cực kỳ thân thiện với người dùng, người dùng cũng có thể nhìn vào class đó và có thể biết được class này nó đang style cái gì. Chúng ta cũng phải nói đến khả năng tùy biến và mở rộng cao, đem đến cho ta sự linh hoạt không giới hạn

# Điểm mới khi dùng Tailwind CSS

Như các bạn đã biết đấy, khi chúng ta dùng framwork này ngoài việc nhiều class hơn tiện cho việc chúng ta style thì chúng ta còn có thể sử dụng lại các class build-in trong framework này để viết một class mới. Đó chính là khi chúng ta sử dụng từ khóa [**@apply**](https://viblo.asia/u/apply).

Bây giờ chúng ta lại mở file assets/styles/less/app.less ra chúng ta sẽ viết thêm css cho class như sau

@import "./libs/tailwind.less";

.demo {

@apply flex justify-between items-center py-4 bg-blue-900;

}

Và sau đó các bạn thay đoạn code thành như sau thì các bạn cũng nhận được kết quả giống y hệt như ban đầu

import '~/assets/styles/less/app.less';

const Demo = () => (

<div className="demo">

<div className="flex-shrink-0 ml-10 cursor-pointer">

<i className="fas fa-drafting-compass fa-2x text-orange-500"></i>

<span className="ml-1 text-3xl text-blue-200 font-semibold">Viblo</span>

</div>

<i className="fas fa-bars fa-2x visible md:invisible mr-10 md:mr-0 text-blue-200 cursor-pointer"></i>

<ul className="hidden md:flex overflow-x-hidden mr-10 font-semibold">

<li className="mr-6 p-1 border-b-2 border-orange-500">

<a className="text-blue-200 cursor-default" href="#">Home</a>

</li>

<li className="mr-6 p-1">

<a className="text-white hover:text-blue-300" href="#">Services</a>

</li>

<li className="mr-6 p-1">

<a className="text-white hover:text-blue-300" href="#">Projects</a>

</li>

<li className="mr-6 p-1">

<a className="text-white hover:text-blue-300" href="#">Team</a>

</li>

<li className="mr-6 p-1">

<a className="text-white hover:text-blue-300" href="#">About</a>

</li>

<li className="mr-6 p-1">

<a className="text-white hover:text-blue-300" href="#">Contacts</a>

</li>

</ul>

</div>

);

export default Demo;

Đây có thể nói là một đặc điểm riêng biệt mà mình thích nhất ở framework này, tại vì nó giúp mình sử dụng được những class built-in của framework trong một class mới, trông code chúng ta gọn gàng hơn rất là nhiều. Nếu dùng Bootstrap thì mình sẽ phải viết hết tất cả các class built-in của framework mà mình cần vào thẻ đó, như thế nhìn code front-end mình trông nó không gọn tí nào cả.

Tiếp theo đó chính là bạn có thể generate responsive, hover, focus, active, group-hovercác class này bằng cách chúng ta sẽ viết trong file css , wrap chúng bằng @variants

Ví dụ

/\* Input \*/

@variants hover, focus {

.banana {

color: yellow;

}

}

/\* Output \*/

.banana {

color: yellow;

}

.hover\:banana:hover {

color: yellow;

}

.focus\:banana:focus {

color: yellow;

}

Viết 1 mà sinh ra được nhiều thứ như này, thật ngắn gọn mà lại mang lại hiệu quả đúng ko các bạn.

Tiếp theo đến responsive nó cũng tương tự như vậy

.bg-gradient-brand {

background-image: linear-gradient(blue, green);

}

/\* ... \*/

@media (min-width: 640px) {

.sm\:bg-gradient-brand {

background-image: linear-gradient(blue, green);

}

/\* ... \*/

}

@media (min-width: 768px) {

.md\:bg-gradient-brand {

background-image: linear-gradient(blue, green);

}

/\* ... \*/

}

@media (min-width: 1024px) {

.lg\:bg-gradient-brand {

background-image: linear-gradient(blue, green);

}

/\* ... \*/

}

@media (min-width: 1280px) {

.xl\:bg-gradient-brand {

background-image: linear-gradient(blue, green);

}

/\* ... \*/

}

Sử dụng phương thức theme() để các bạn có thể truy cập là lấy ra giá trị trong Tailwind config. Ví dụ bạn có file tailwind.config.js như sau :

module.exports = {

important: true,

theme: {

screens: {

vs: '320px',

sm: '576px',

md: '768px',

lg: '992px',

xl: '1200px',

xxl: '1600px',

},

extend: {},

fontSize: {

xs: '.75rem',

sm: '.875rem',

tiny: '.875rem',

base: '1rem',

lg: '1.125rem',

xl: '1.25rem',

'2xl': '1.5rem',

'3xl': '1.875rem',

'4xl': '2.25rem',

'5xl': '3rem',

'6xl': '4rem',

'7xl': '5rem',

'8xl': '6rem',

'9xl': '7rem',

'10xl': '8rem',

'11xl': '9rem',

},

},

variants: {},

plugins: [],

}

Bây giờ các bạn có file css như sau

.btn-blue {

font-size: theme('fontSize.11xl');

}

# Pseudo-Class Variants

Đây là một điểm mình rất thích ở framework CSS này bởi vì chúng ta có thể viết class cho các sự kiện hover, focus, active, disabled, visited, first-child, last-child, odd-child, even-child, focus-within, cho các thẻ.

Ví như

<form>

<input class="bg-gray-200 hover:bg-white hover:border-gray-300 focus:outline-none focus:bg-white focus:shadow-outline focus:border-gray-300 ...">

<button class="bg-teal-500 hover:bg-teal-600 focus:outline-none focus:shadow-outline ...">

Sign Up

</button>

</form>

Đoạn code trên là chúng ta tạo ra 1 ô input với nút Sign up, thay vì phải viết css dài dướm dà chúng ta thêm tiền tố hover, focus vào trước class thôi là chúng ta đã có thể thực hiện thay đổi style khi hover hoặc focus vào thẻ html