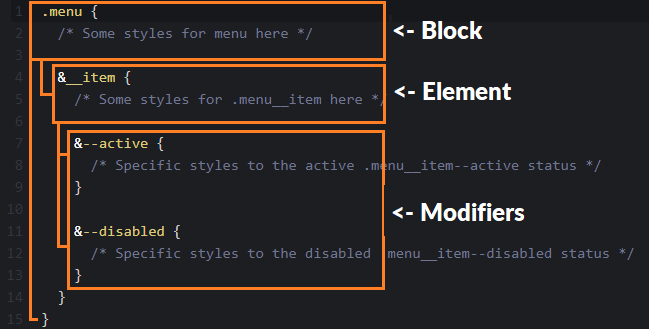
# Quy tắc đặt tên BEM

## BEM

* B: Block là khối
* E: Element là các thành phần trong khối
* M: Modifier là các ý nghĩa bổ sung cho Block và Element



# Reset CSS

## Cấu trúc thư mục CSS

Trong đó:

+ Thư mục **web** chứ các hình ảnh font chữ và code css, js làm việc trực tiếp với website

- Trong thư mục css chứa file base.css là nơi cứa các code chung cho toàn bộ project

- Thư mục chứ code của tùng phần là thư mục source

- Responsive là nơi chứa các code responsive

+ Thư mục **lib** chứa các thư viện import từ bên ngoài vào project như owl, carousel, fontawesome,…

## Code reset

Download [git@github.com:quangnd512/html-css\_basic.git](mailto:git@github.com:quangnd512/html-css_basic.git) với Mk: quang123@

Truy cập thư mục: D:\Desktop\github\html-css\_basic\Cut-Website

Tìm file: reset.css để lấy code reset sẵn

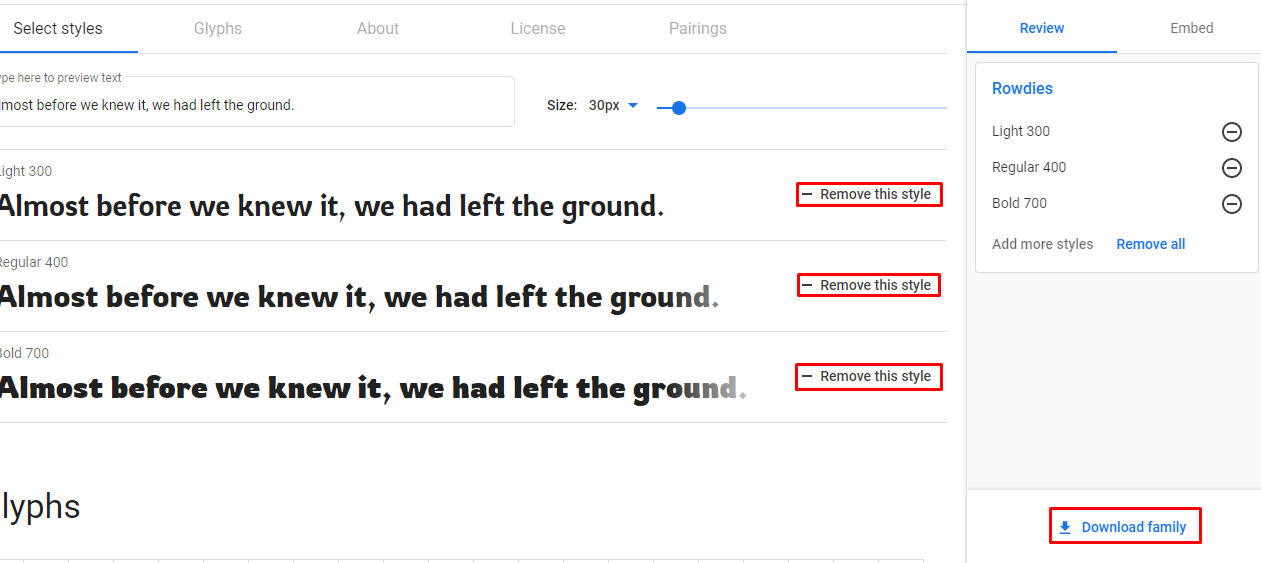
## Các bộ chọn trong CSS

|  |  |  |  |
| --- | --- | --- | --- |
| **Bộ chọn** | **Ví dụ** | **Mô tả các ví dụ** | **CSS** |
| .*class* | .intro | Chọn tất cả các phần tử có class=”intro” | 1 |
| #*id* | #firstname | Chọn tất cả các phần tử có id=”firstname” | 1 |
| \* | \* | Chọn tất cả các phần tử | 2 |
| *element* | p | Chọn tất cả các phần tử  <p> | 1 |
| *element,element* | div, p | Chọn tất cả các phần tử <div>và phần tử <p> | 1 |
| *element* *element* | div p | Chọn tất cả các phần tử <p> và bên trong phần tử  <div> | 1 |
| *element*>*element* | div > p | Chọn tất cả các phần tử <p> có phần tử cha là <div> | 2 |
| *element*+*element* | div + p | Chọn tất cả các phần tử <p> được đặt phía sau phần tử  <div> | 2 |
| *element1*~*element2* | p ~ ul | Chọn tất cả các phần tử  <ul> được đặt trước bởi một phần tử  <p> | 3 |
| [*attribute*] | [target] | Chọn tất cả các phần tử có cùng thuộc tính | 2 |
| [*attribute*=*value*] | [target=\_blank] | Chọn tất cả các phần tử có thuộc tính bằng giá trị( target=”\_blank”) | 2 |
| [*attribute*~=*value*] | [title~=flower] | Chọn tất cả các phần tử có tiêu đề của thuộc tính có chứa từ “flower” | 2 |
| [*attribute*|=*value*] | [lang|=en] | Chọn tất cả các phần tử có giá trị thuộc tính “lang” bắt đầu bằng “en” | 2 |
| [*attribute*^=*value*] | a[href^=”https”] | Chọn tất cả các phần tử  <a> có giá trị thuộc tính “href” bắt đầu bằng “https” | 3 |
| [*attribute*$=*value*] | a[href$=”.pdf”] | Chọn tất cả các phần tử <a> có giá trị thuộc tính “href” kết thúc bằng”.pdf” | 3 |
| [*attribute*\*=*value*] | a[href\*=”timoday”] | Chọn tất cả các phần tử <a> có giá trị thuộc tính “href” chứa chuỗi”timoday” | 3 |
| :active | a:active | Chọn tất cả các liên kết được kích hoạt | 1 |
| ::after | p::after | Chèn thêm nội dung ngay phía sau của các phần tử <p> | 2 |
| ::before | p::before | Chèn thêm nội dung ngay phía trước của các phần tử <p> | 2 |
| :checked | input:checked | Chọn tất cả các phần tử <input> đang được chọn (selected) | 3 |
| :disabled | input:disabled | Chọn tất cả các phần tử <input> đang được vô hiệu hoá (disabled) | 3 |
| :empty | p:empty | Chọn tất cả các phần tử  <p> không chứa phần tử con (bao gồm cả các nút văn bản) | 3 |
| :enabled | input:enabled | Chọn tất cả các phần tử <input> đang được kích hoạt | 3 |
| :first-child | p:first-child | Chọn các phần tử  <p> có phần tử đầu tiên của phần tử cha chứa nó | 2 |
| ::first-letter | p::first-letter | Chọn kí tự đầu tiên của phần tử  <p> | 1 |
| ::first-line | p::first-line | Chọn dòng đầu tiên của các phần tử <p> | 1 |
| :first-of-type | p:first-of-type | Chọn tất cả các phần tử <p> có phần tử đầu tiên <p> là phần tử cha | 3 |
| :focus | input:focus | Chọn các phần tử <input> nhận focus | 2 |
| :hover | a:hover | Chọn các liên kết khi chuột di chuyển qua | 1 |
| :in-range | input:in-range | Chọn phần tử <input> có giá trị  trong phạm vi nhất định | 3 |
| :invalid | input:invalid | Chọn tất cả các phần tử <input> có giá trị không hợp lệ | 3 |
| :lang(*language*) | p:lang(it) | Chọn tất cả các phần tử  <p> có giá trị thuộc tính “lang” bằng “it” | 2 |
| :last-child | p:last-child | Chọn tất cả các phần tử <p> là phần tử con cuối cùng của phần tử cha | 3 |
| :last-of-type | p:last-of-type | Chọn tất cả các phần tử <p> là thuộc tính cuối cùng của phần tử cha | 3 |
| :link | a:link | Chọn tất cả các liên kết khi chưa được click | 1 |
| :not(*selector*) | :not(p) | Chọn tất cả các phần tử không phải là một phần tử <p> | 3 |
| :nth-child(*n*) | p:nth-child(2) | Chọn tất cả các phần tử <p> là phần tử thứ hai của phần tử cha | 3 |
| :nth-last-child(*n*) | p:nth-last-child(2) | Chọn tất cả các phần tử <p> là phần tử con thứ hai của phần tử cha, tính từ phần tử con cuối cùng | 3 |
| :nth-last-of-type(*n*) | p:nth-last-of-type(2) | Chọn tất cả các phần tử <p>là phần tử thuộc tính thứ hai của phần tử cha, tính từ phần tử thuộc tính con cuối cùng | 3 |
| :nth-of-type(*n*) | p:nth-of-type(2) | Chọn tất cả các phần tử <p> là phần tử thuộc tính con thứ hai của phần tử cha | 3 |
| :only-of-type | p:only-of-type | Chọn tất cả các phần tử <p> là thuộc tính duy nhất của phần tử cha | 3 |
| :only-child | p:only-child | Chọn tất cả các phần tử <p> là con duy nhất của phần tử cha | 3 |
| :optional | input:optional | Chọn tất cả các phần tử đầu vào không có thuộc tính “required” | 3 |
| :out-of-range | input:out-of-range | Chọn tất cả các phần tử đầu vào có giá trị ngoài một phạm vi nhất định | 3 |
| :read-only | input:read-only | Chọn tất cả các phần tử đầu vào có thuộc tính xác định “readonly” | 3 |
| :read-write | input:read-write | Chọn tất cả các phần tử đầu vào có thuộc tính không xác định “readonly” | 3 |
| :required | input:required | Chọn tất cả các phần tử đầu vào có thuộc tính  “required” xác định | 3 |
| :root | :root | Chọn các phần tử gốc của văn bản | 3 |
| ::selection | ::selection | Chọn các phần tử được người dùng lựa chọn |  |
| :target | #news:target | Chọn các phần tử đang hoạt động hiện tại (click trong các liên kết  anchor name) | 3 |
| :valid | input:valid | Chọn tất cả các phần tử đầu vào có một giá trị hợp lệ | 3 |
| :visited | a:visited | Chọn tất cả các liên kết được truy cập | 1 |

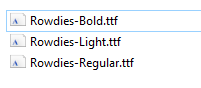
## Nhúng font

**Bước 1**: Download fonts

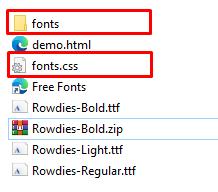
- Đầu tiên ta truy cập vào <https://fonts.google.com/> để tìm font cần tìm kiếm và download 3 loại font liên quan là Bold, Light, Regular



- Sau khi download xong giải nén file ta được 3 font.



- Truy cập vào <http://www.font2web.com/> để convert sang các định dạng font khác nhau.

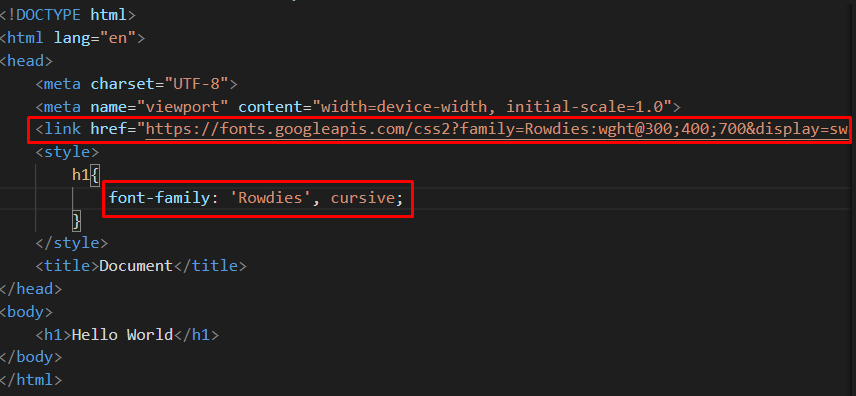


**Bước 2**: Nhúng font vào project

Cách 1: Nhúng bằng link google



- Copy các đoạn mã sau và dán vào css



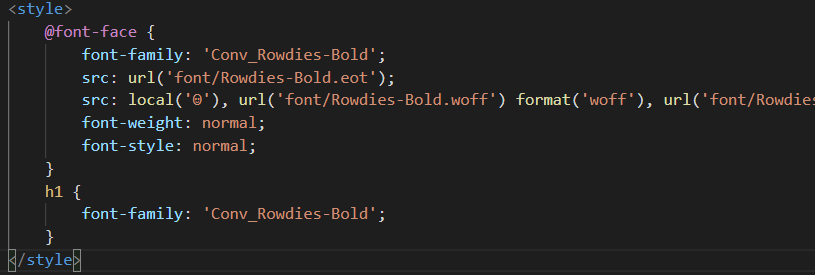
* Kết quả:



Cách 2: Nhúng bằng cách download fonts về

- Sau khi download thành công fonts ở bước 1 ta copy các font trong file fonts vừa download được vào thư mục fonts trong project của chúng ta.

- Và mở file fonts.css lên copy đoạn code font-face vào trong file css trong project



* Kết quả



## Nhúng icon

Bước 1: Download và nhúng vào project

- Truy cập vào trang website <https://fontawesome.com/> để download thư viện về

- Tìn đến mục other ways to use và nhấn vào download để download file

- Chuyển thư mục vừa download vào project.

- Vào file code link đến thư mục /css/all.min.css



Bước 2: Thêm icon

**Cách 1**: Thêm icon bằng cách sử dụng ::after hoặc ::before

Để thêm icon theo cách này ta cần phải có 3 thuộc tính sau:

content: "\f5d0";

    font-family: "Font Awesome 5 Free";

    font-weight:600;

Với: font-family phải là Font Awesome 5 Free

Content là content tùy ý

Font-weight phải từ 600 trở lên

**Cách 2**: Thêm icon bằng cách nhúng class vào thẻ i



**Cách 3**: Dùng dạng svg

- Download icon 

- Mở file bằng notepass++ và copy thẻ svg

- Sau đó dán trực tiếp vào code

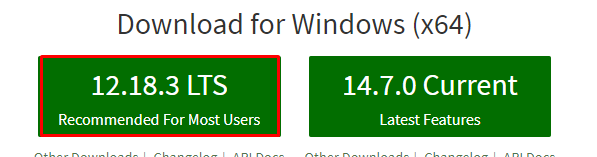
- Cách này thì không cần phải nhúng thư mục vừa download về

Chú ý: Ta cũng có thể lấy ảnh dạng svg tại trang web: <https://iconscout.com/unicons/explore/line> hoặc <https://www.flaticon.com/> và <https://freeicons.io/>

## Cách viết code với less

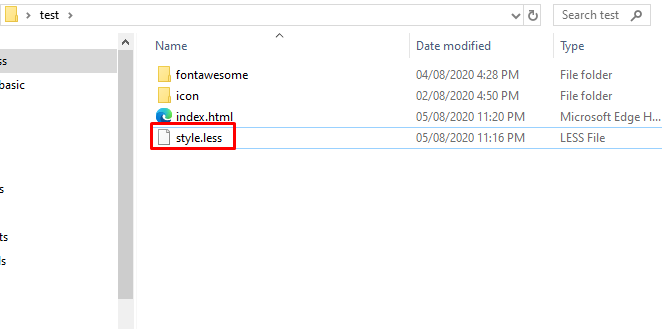
### Cài đặt less

Bước 1: Cài đặt Nodejs



Bước 2: Cài đặt npm npm install -g less

Bước 3: Gọi file css 

Bước 4: Tạo file less trong project cùng cấp với file css. Và viết code less vào trong css 

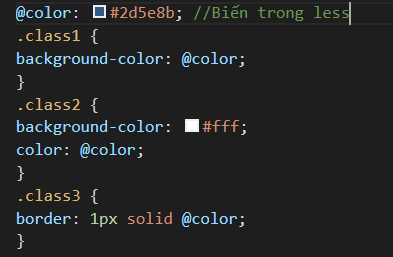
Bước 5: Chuyển code từ file less sang file css bằng câu lệnh lessc style.less style.css

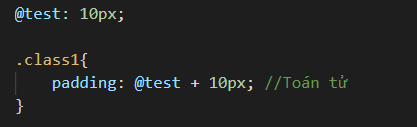


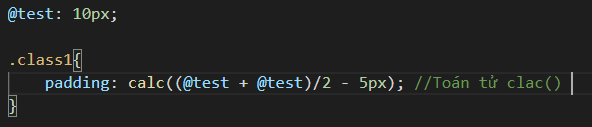
Reset lại trình duyệt và xem kết quả.!

### Tổng quan less

1. Biến và toán tử

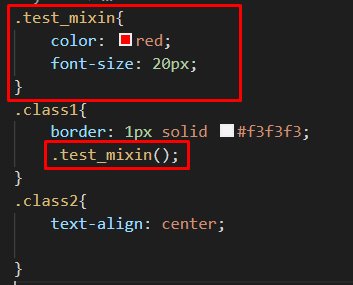




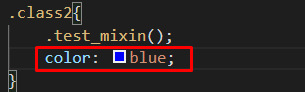


1. Mixin

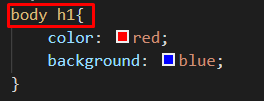
Tạo ra một class mixin gồm các thuộc tính chung rồi sau đó khi cần thì gọi class mixin đó vào các class riêng giống như cách gọi hàm trong OOP



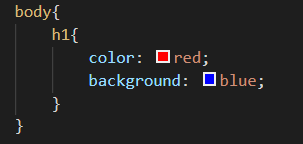
Nếu muốn thay đổi một thuộc tính đã khai báo trong class mixin ở class riêng thì ta khai báo thuộc tính cần thay đổi phía dưới hàm mixin vừa gọi vào



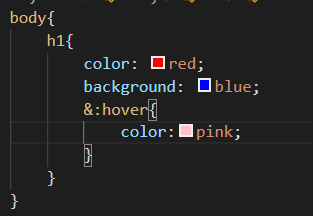
1. Nested Rules và Scope



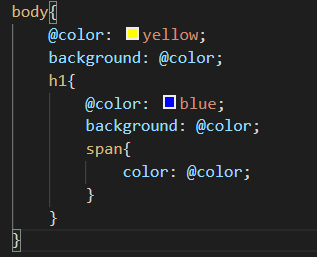
Ta có thể viết lại đoạn code trên cho ngắn gọn hơn như sau:



Ngoài ra ta cũng có thể thêm các thuộc tính hover cho các element trên bằng cách thêm dấu & VD: &:hover



Scope là khi element sẽ được thừa hưởng các biến gần nó nhất

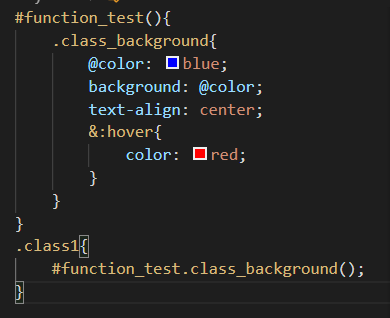


Ta thấy thuộc tính color của thẻ span nhận màu là blue

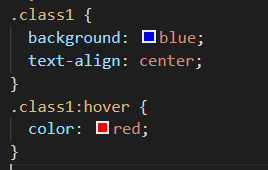


1. Function

Đôi khi ta cần kết hợp các hàm tính toán, các đoạn code, các class mixin vào để thực hiện một chức năng riêng biệt nào đó thì ta cần đến hàm. Cách khai báo hàm như sau:



Tương ứng css sẽ hiển thị như sau:



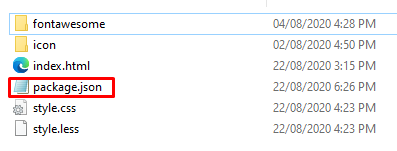
### Sử dụng công cụ grunt

Để cài grunt bạn cần phải cài nodejs

Sau đó cài grunt bằng câu lệnh: npm install grunt-cli -.-g

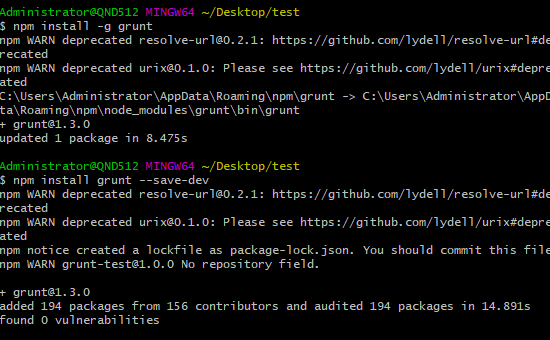
Bước 1: Tạo file package.json

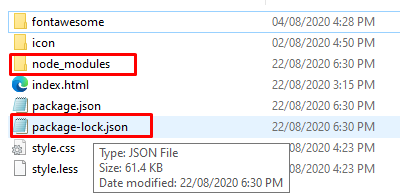
- Vào project mở cmd và gõ lệnh: npm init



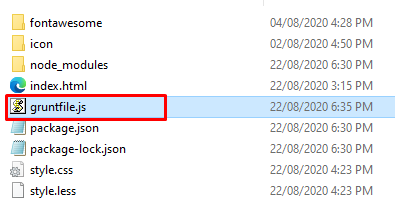
Bước 2: Cài đặt grunt toàn cục và cục bộ

Chạy lần lượt các lệnh: npm install -.-g grunt và npm install grunt -.-save-dev





Bước 3: Tạo file gruntfile.js



- Code mặc định bên trong.

module.exports = function(grunt) {

   // Project configuration.

   grunt.initConfig({

     pkg: grunt.file.readJSON('package.json'),

   });

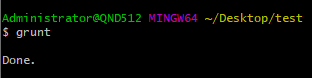
   // Default task(s).

   grunt.registerTask('default', []);

};

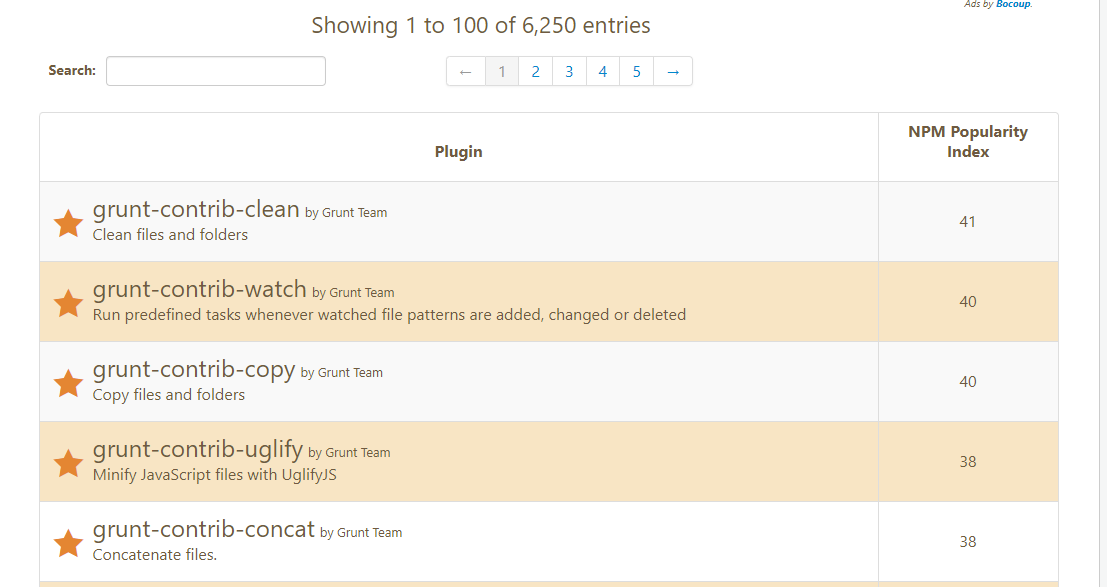
Đây là một file gruntfile tối giản để chỉ đọc file package.json và tạo một tác vụ default không chạy gì cả

- Chạy lệnh grunt để kiểm tra quá trình cài đặt.



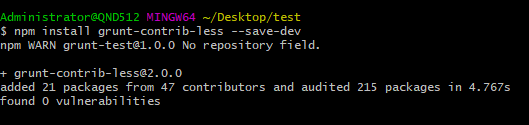
Bước 4: Thêm task vào gruntfile.js

- Xem các plugin tại địa chỉ: <https://gruntjs.com/plugins>

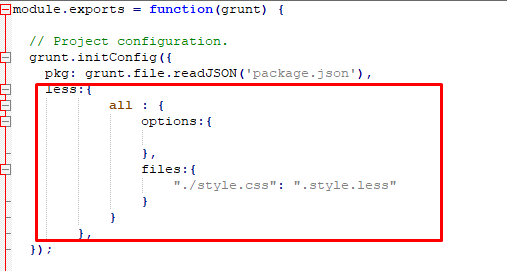


- Ở đây ta thêm plugin less để grunt biên dịch less sang css

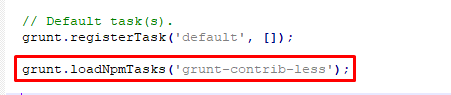
Dùng lệnh: npm install grunt-contrib-less -.-save-dev để cài đặt plugin less



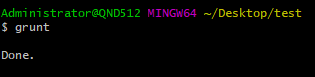
- Bổ sung task less vào file gruntfile.js



- Sau đó nạp task less



- Chạy grunt để chạy task less



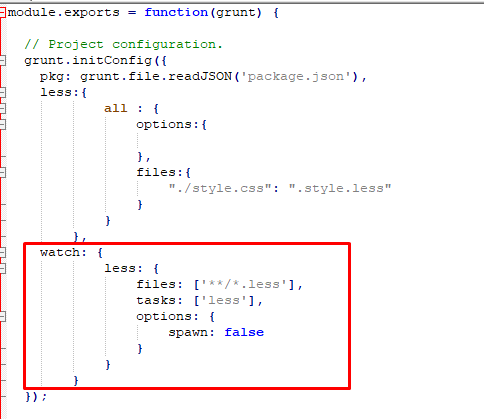
- Bây giờ mỗi khi những đoạn code less được thay đổi nó sẽ tự động biên dịch sang css khi ta sử dụng câu lệnh grunt

Bước 5: Thêm plugin chạy grunt tự động

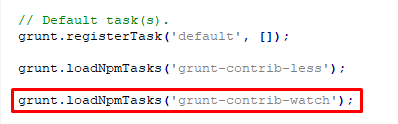
- Ta thậm chí không cần phải chạy grunt một cách thủ công nữa mà sẽ có plugin giúp ta thực hiện việc đó.

- Ta cài đặt plugin watch bằng câu lệnh sau: npm install grunt-contrib-watch –save-dev

- Bổ sung plugin watch vào file gruntfile.js



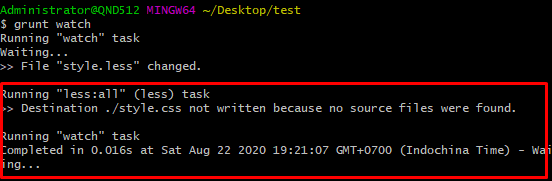
- Nạp task watch:



- Khởi chạy task watch: grunt watch

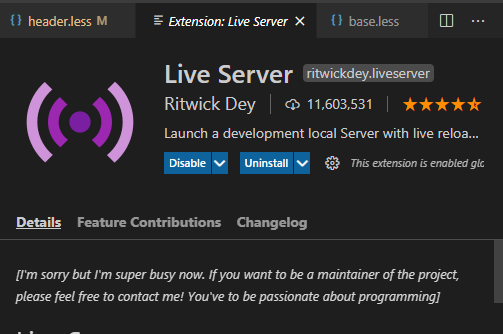
Kết quả:

- Khi ta thay đổi file less thì nó sẽ thông báo cho ta biết và tự động cập nhật vào file css đã khai báo trong file gruntfile.js

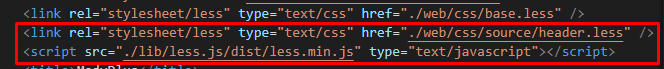


### Sử dụng less trực tiếp trong project mà không cần biên dịch ra css

* Đầu tiên ta cài live server trên công cụ Visual Studio Code.



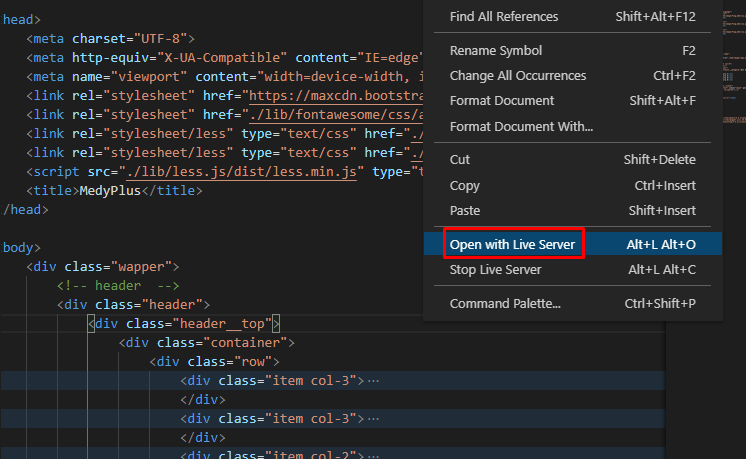
* Sau đó nhúng các file less vào project của chúng ta bằng cách.



* Với file header.less phải được đặt ở trên thẻ script nhúng thư viện less.min.js
* Thư viện less.min.js được download tại: <https://lesscss.org/>



* Sau khi các bước cài đặt đã hoàn tất ta bật live server lên.



# Slick và Owl

## Nhúng slick và owl

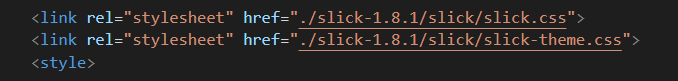
### Slick

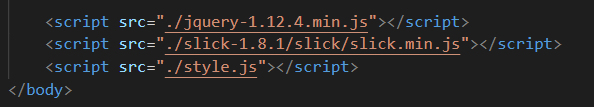
- Cách 1: Download slick

Bước 1: Download jQuery >1.7 tại <https://blog.jquery.com/2016/05/20/jquery-1-12-4-and-2-2-4-released/>

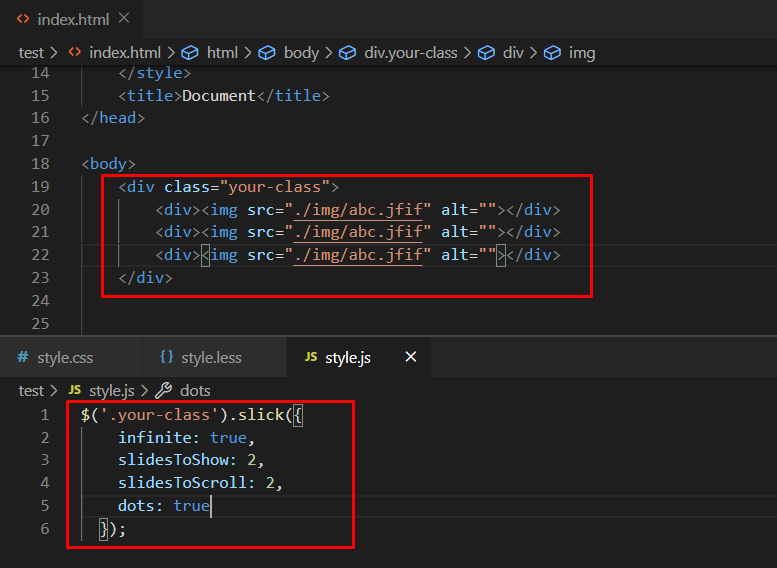
Bước 2: Download slick tại <https://kenwheeler.github.io/slick/>

Bước 3: Giải nén và gọi các file slick và jquery vào proje ct





Bước 4: Kiểm tra xem đã nhận slick chưa

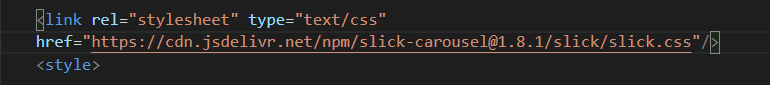


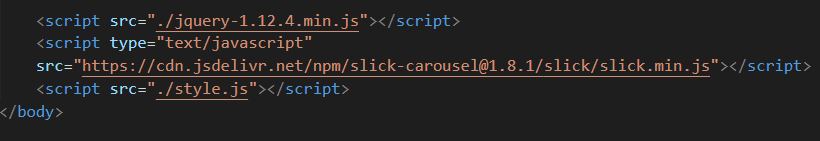


- Cách 2: Nhúng CDN

Bước 1: Truy cập vào trang website <https://www.jsdelivr.com/package/npm/slick-carousel> để lấy CDN

Bước 2: Dán các link vào thẻ link và script





### Owl

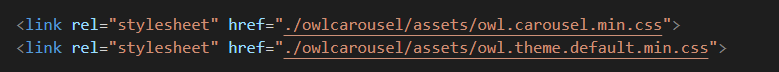
- Cách 1: Dowload owl

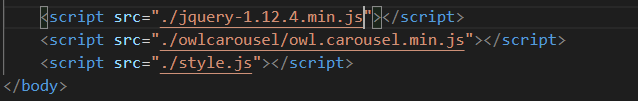
Bước 1: Download jQuery >1.7 tại <https://blog.jquery.com/2016/05/20/jquery-1-12-4-and-2-2-4-released/>

Bước 2: Download owl tại <https://owlcarousel2.github.io/OwlCarousel2/>

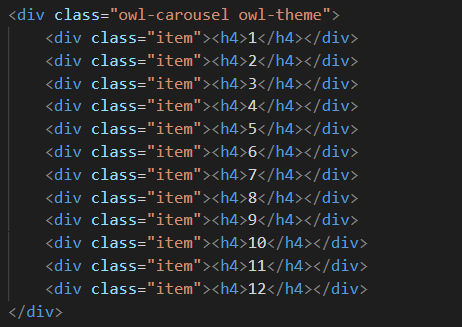
Bước 3: Giải nén và vào OwlCarousel/docs/assets và copy thư mục owlcarousel vào project

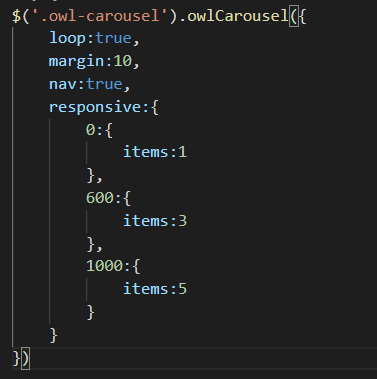
Bước 4: Sau đó làm theo hướng dẫn sau: <https://owlcarousel2.github.io/OwlCarousel2/docs/started-installation.html>





Bước 5: Kiểm tra owl xem đã nhận chưa



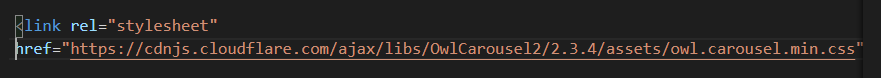


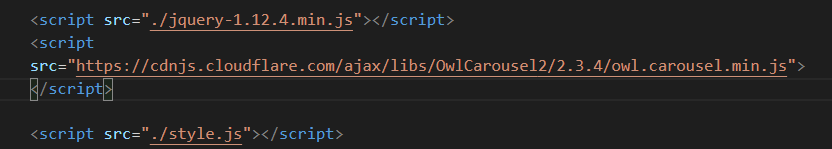


- Cách 2: Nhúng CDN

Bước 1: Truy cập vào trang website <https://cdnjs.com/libraries/OwlCarousel2> để lấy CDN

Bước 2: Dán các link vào thẻ link và script





## Cách sử dụng slick

### Các hàm thường dùng trong slick

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Mục đích sử dụng** | **Kiểu giá trị** | **Ví dụ : (mặc định)** |
| slidesToShow | Số lượng phần tử được nhìn thấy | int | slidesToShow: 4 |
| slidesToScroll | Số phần tử được lướt sang | int | slidesToScroll: 1 |
| accessibility | Cho phép qua lại phần tử bằng bàn phím qua lại | bool | accessibility: true |
| adaptiveHeight | Nếu kích thước item lớn thì sẽ tự động giãn chiều cao | bool | adaptiveHeight:false |
| autoplaySpeed | Tự đông chạy cho slide | int(ms) | autoplaySpeed :3000 |
| arrows | Cho phép hiển thị phím mũi tên chạy | bool | arrows:true |
| centerMode | Hiển thị slide ở trung tâm, bao gồm các slide trước và tiếp theo. slidesToShow chỉ định cả hai với một số lượng item là số lẻ. | bool | centerMode: true |
| centerPadding | Khoảng cách của phần hiển thị bị che một item chưa show (dùng chung với centerMode) | String | centerPadding :’50px’ or ‘50%’ |
| cssEase | Hiệu ứng chuyển trang dạng animation | String | Tìm hiểu thêm về CSS3 Animation Easing  VD: cssEase: 'linear' |
| customPaging | Biến những dấu chấm phân silde thành những con số. Thuộc tính này đi kèm với thuộc tính dot: true | Funtion | <https://jsfiddle.net/AdrienBe/L0v2s26t/> |
| dots | Bật, tắt dấu chấm định danh của mỗi item | bool | dots: false |
| draggable | Cho phép kéo ảnh sang hai bên bằng chuột | bool | draggable: true |
| fade | Ảnh mờ dần khi chuyển slide | bool | fade: false |
| focusOnSelect | Khi click vào slide con bên dưới thì slide chính được show | bool | focusOnSelect: false |
| infinite | Vòng lăp vô tận | bool | infinite:true |
| initialSlide | Thứ tự xuất hiện lần đầu của item show | int | initialSlide: 0 |
| rows | Show số hàng item trên một slile | int | rows: 1 |
| responsive | Reponsive cho từng kích thước view | Object | responsive: [  {  breakpoint: 991,  settings: {  infinite:true,  slidesToShow: 3,  slidesToScroll: 1  }  },  {  breakpoint: 479,  settings: {  infinite:true,  slidesToShow: 1,  slidesToScroll: 1  }  }  ]  }); |
| pauseOnFocus | Tạm dừng auto slide đang chạy khi click vào item show | bool | pauseOnFocus: true |
| pauseOnDotsHover | Tạm dừng auto slide đang chạy khi hover vào dot show | bool | pauseOnDotsHover: true |
| pauseOnHover | Tạm dừng auto slide đang chạy khi hover vào item show | Bool | pauseOnHover: true |
| speed | Tốc độ chuyển slide | Int | Speed:300 |
| variableWidth | Vô hiệu hóa một phần vùng nhìn của slide bên góc | Bool | variableWidth: false |
| vertical | Slide di chuyển lên xuống thay vì trái phải hoặc dùng verticalSwiping | bool | vertical: false, |

### Style hay gặp khi dùng slick

#### Cho ảnh ở giữa slider nhô cao hơn so với các ảnh còn lại



#### Thay đổi vị trí của dot

#### Thay đổi và hiện thị arrow

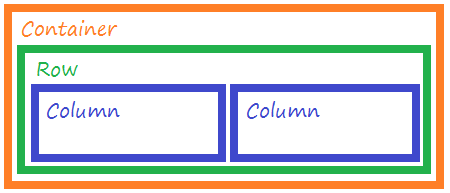
## Cách sử dụng Owl

### Các hàm thường dùng

### Style hay gặp khi dùng owl

# Gird và Responsive

## Gird



### Grid

Tạo grid là một layout bao quanh toàn bộ trang website

.grid {

    width: 100%;

    display: block;

    padding: 0;

}

### Container

Tạo thêm một layout container sẽ làm nhiệm vụ là layout bao quanh phần content của trang website đảm bảo thích ứng với các thiết bị nhỏ hơn.

.container {

    max-width: 1140px;

    margin: 0 auto;

}

### Row

Row có nhiệm vụ dàn hàng ngang cho các col để các col không bị vỡ và nhảy khỏi hàng một cách không mong muốn khi chúng ta responsive website

.row {

    display: flex;

    flex-wrap: wrap;

    margin-left: -15px;

    margin-right: -15px;

}

### Col

Col là cách chia theo cột thường thì ta sẽ chia ra làm 12 mức độ col khác nhau tương ứng với mức độ rộng của cột.

Ví dụ: col-1 có width bằng 100% / 12 = 8,33333%,

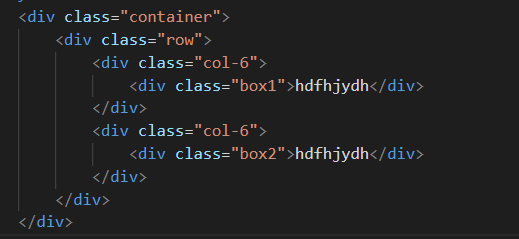
col-12 có width bằng 100%,

col-6 có width bằng 100% / 2 = 50%

Với các thư viện ta cần xây dựng đầy đủ 12 col

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 8,333% | 16,667% | 25% | 33,333% | 41,667% | 50% |
| 7 | 8 | 9 | 10 | 11 | 12 |
| 58,333% | 66,667% | 75% | 83.333% | 91,667% | 100% |

Ta cần phải tạo thêm một thẻ div trong thẻ col để chứa content. Như vậy sẽ dễ dàng điều chỉnh các khối div hơn.



.col-6 {

    width: 50%;

    box-sizing: border-box; /\* Thuộc tính giúp cho các khối col không bị mở

                                rộng bằng padding \*/

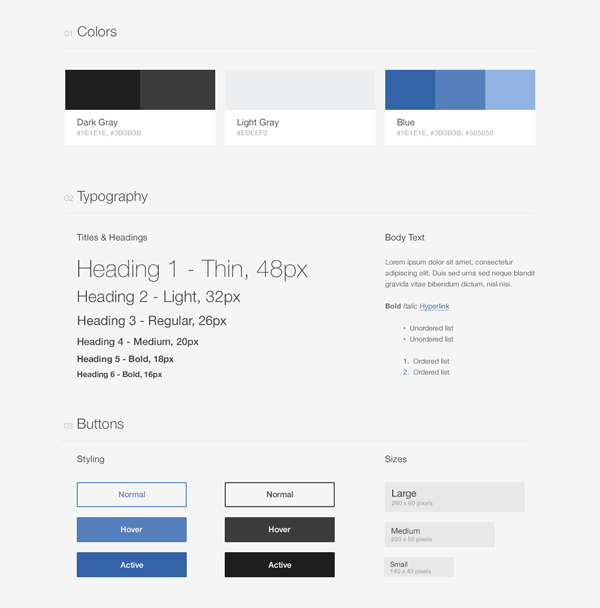
    padding-left: 15px;

    padding-right: 15px;

}

* Và chúng ta sẽ tạo thêm một khối bên trong col để tạo content

## Style guide



- Các tiêu chuẩn để các element khác kế thừa là:

+ Color

+ Typography (H1-H6, Text, link, list)

+ Buttons

+ Icon

+ Form

+ Breadcrumb, Pagination

+ Carts

## Responsive

- Tạo trải nghiệm tốt nhất cho người dùng

- Dùng CSS để style lại phù hợp trên mọi trình duyệt

### Viewport

- Thẻ meta viewport có thể điều chỉnh được khung nhìn

- Cú pháp: <meta name="viewport" content="width=device-width, initial-scale=1.0">

### Media query (@media)

- Cú pháp: @media **not**/**only** mediatype and (mediafeature and/or/not mediafeature){

//CSS code

}

- Keywords:

+ not: loại trừ 1 thằng nào đó

+ only: chỉ với 1 thằng nào đó

+ and: và

+ or: hoặc

- Mediatype:

+ print: Chế đọ máy in

+ screen: Chế độ màn hình

+ speech: Chế độ màn hình nói được

+ all – default: Bao gồm mọi chế độ

- Mediafeature:

+ min-witch

+ max-witch

+ …

### Polifill

- Đây là cách giúp cho media query hiểu được các trình duyệt cũ

- Tìm kiếm google để tìm kiếm cdn: responsive polyfill cdn

### Câu lệnh điều kiện sử dụng cho comment

- Ví dụ câu lệnh:

<!--[if lt IE 9]>

<script src="//cdnjs.cloudflare.com/ajax/libs/html5shiv/r29/html5.min.js">

</script>

<![endif]-->

Với:

lt < (less than)

gt > (greater than)

eq == (equal to)

ne != (not equal to)

gte >= (greater than or equal to)

lte <= (less than or equal to)

- Điều này sẽ giúp các phiên bản trình duyệt cao hơn có hỗ trợ responsive sẽ không cần phải áp dụng thuộc tính này nữa giúp cho trình duyệt được tối ưu và nhẹ hơn vì chúng không phải load những thứ không cần thiết

### Breakpoints

- Khi sử dụng breakpoints ta nên dùng em vì em sẽ không bị gặp vấn đề nhiều trên các trình duyệt

- Khi sử dụng em trong breakpoint thì nó sẽ không phụ thuộc vào element nào cả mà 1 em luôn bằng 16px

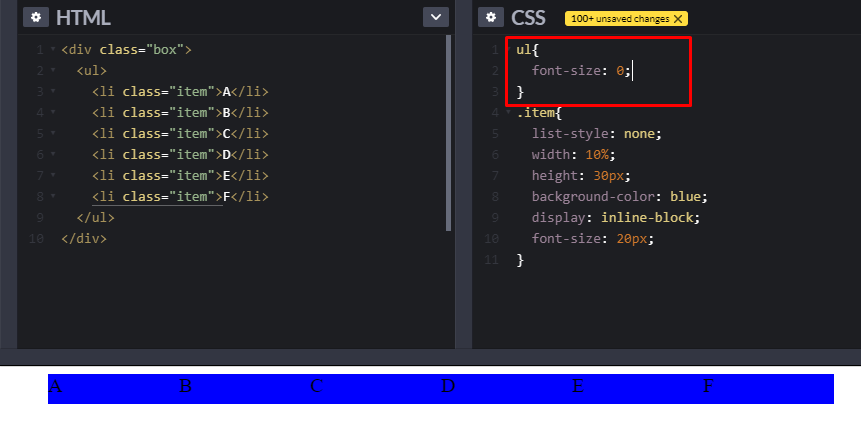
- Các điểm breakpoints chuẩn:

| **Breakpoint** | **Class infix** | **Dimensions** |
| --- | --- | --- |
| X-Small | *None* | <576px |
| Small | sm | ≥576px |
| Medium | md | ≥768px |
| Large | lg | ≥992px |
| Extra large | xl | ≥1200px |
| Extra extra large | xxl | ≥1400px |

# Một số kỹ thuật dàn trang trong Front-end

## Inline-Block

* Ta có thể dùng thuộc tính display: inline-block để cho các thẻ block lên cùng một hàng.
* Thuộc tính display: inline-block sẽ làm cho các thẻ được áp dụng nó có dạng vừa là inline vừa là block



* Nhưng khi sử dụng ta thấy các thẻ cách nhau một dấu space nên để sửa lỗi ta cho font-size = 0 vào thẻ ul và các thẻ li ta lại đặt font-size phù hợp cho text

## Float

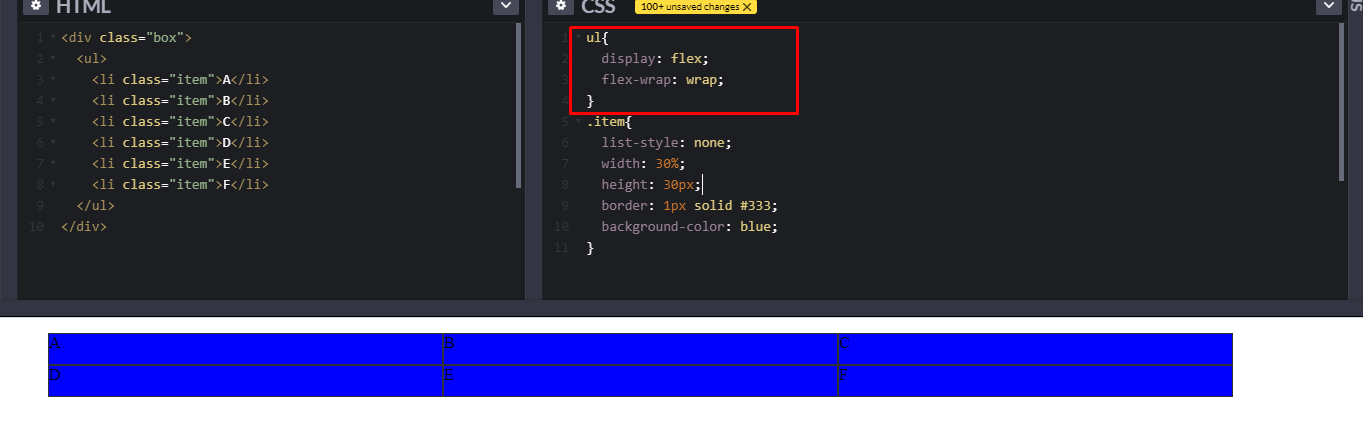
* Thuộc thính float cũng được dùng để dàn trang.



* Nhưng do float hay bị vỡ giao diện trang web nên ta phải đặt thuộc tính clear: both; vào thẻ div ở dưới thẻ áp dụng float

## Flex

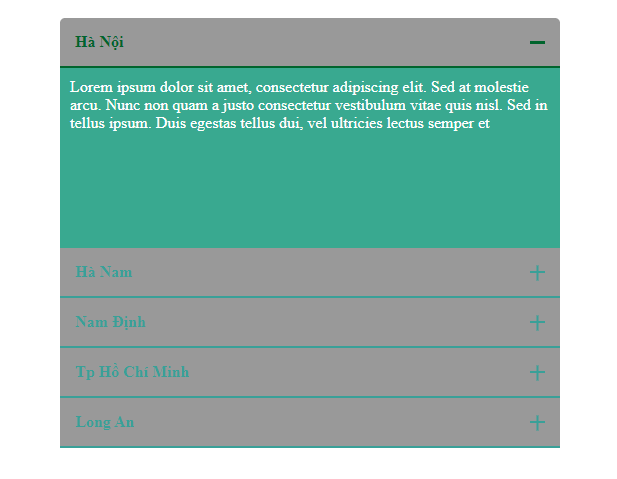
* Hiện tại thuộc tính display: flex được sử dụng nhiều hơn cả vì sự tiện dụng và hữu ích.



# Components

Với các hướng dẫn ở dưới chỉ là hướng dẫn cơ bản. Nếu muốn tìm hiểu sâu hơn thì tham khảo: <https://www.w3schools.com/howto/default.asp>

## Accordion



* HTML:

<div class="accordion\_\_wapper">

        <div class="accordion">

            <div class="accordion\_\_title">Hà Nội</div>

            <div class="accordion\_\_content"><p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed at molestie arcu. Nunc non quam a justo consectetur vestibulum vitae quis nisl. Sed in tellus ipsum. Duis egestas tellus dui, vel ultricies lectus semper et</p></div>

        </div>

        <div class="accordion">

            <div class="accordion\_\_title">Hà Nam</div>

            <div class="accordion\_\_content"><p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed at molestie arcu. Nunc non quam a justo consectetur vestibulum vitae quis nisl. Sed in tellus ipsum. Duis egestas tellus dui, vel ultricies lectus semper et</p></div>

        </div>

        <div class="accordion">

            <div class="accordion\_\_title">Nam Định</div>

            <div class="accordion\_\_content"><p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed at molestie arcu. Nunc non quam a justo consectetur vestibulum vitae quis nisl. Sed in tellus ipsum. Duis egestas tellus dui, vel ultricies lectus semper et</p></div>

        </div>

    </div>

    <script src="./js/jquery-3.6.0.js"></script>

    <script src="./js/jsstyle.js"></script>

* CSS

.accordion\_\_wapper{

    width: 500px;

    margin: 50px auto;

}

.accordion\_\_title{

    width: 100%;

    height: 50px;

    background-color: #999;

    padding: 15px;

    font-weight: 800;

    color: #39a098;

    border-bottom: 2px solid #39a098;

    cursor: pointer;

    position: relative;

}

.accordion\_\_title:hover{

    color: #01642d;

    border-bottom-color: #01642d;

}

.accordion\_\_title:hover::before, /\* Tao hover cho ::before \*/

.accordion\_\_title:hover::after{

    background-color: #01642d;

}

.accordion\_\_title::before,

.accordion\_\_title::after{  /\* Tao nút điều hướng (Khi đóng thì cả after và before đều nằm ngang) \*/

    content: "";

    position: absolute;

    top: 50%;

    right: 15px;

    width: 15px;

    height: 2px;

    background-color: #39a098;

    transform: translateY(-50%);

}

/\* Nut dieu huong thanh dau + \*/

.accordion\_\_title::after{

    transform: rotate(-90deg);

    transition: all 0.3s ease;

}

.accordion:first-child .accordion\_\_title{

    border-top-left-radius: 5px;

    border-top-right-radius: 5px;

}

.accordion\_\_content{

    width: 100%;

    height: 0;

    transition: all 0.3s ease;

    background: #39a990;

    overflow: hidden; /\*test\*/

}

.accordion\_\_content p{

    padding: 10px;

    color: #fff;

}

/\* Style khi class được áp dụng jquery \*/

.accordion\_\_title.active{

    color: #01642d;

    border-bottom-color: #01642d;

}

.accordion\_\_title.active::before, /\* Tao hover cho ::before \*/

.accordion\_\_title.active::after{

    background-color: #01642d;

}

/\* Nut dieu huong thanh dau - \*/

.accordion\_\_title.active::after{

    transform: rotate(0);

}

.accordion\_\_title.active + .accordion\_\_content{

    height: 180px;

}

* Jquery

$(document).ready(function() {

   $(".accordion\_\_title").click(function() {

      $(this).toggleClass("active");

   });

});

## Breadcrumb

* Sử dụng ul li



* Với HTML:

<ul class="breadcrumbs">

        <li class="breadcrumbs\_\_item"><a href="#" class="breadcrumbs\_\_link">Home</a></li>

        <li class="breadcrumbs\_\_item"><a href="#" class="breadcrumbs\_\_link">Product</a></li>

       <li class="breadcrumbs\_\_item"><a href="#" class="breadcrumbs\_\_link">Apple</a></li>

        <li class="breadcrumbs\_\_item"><a href="#" class="breadcrumbs\_\_link">Iphone</a></li>

        <li class="breadcrumbs\_\_item"><a href="#" class="breadcrumbs\_\_link breadcrumbs\_\_link--active">XS Pro Max</a></li>

    </ul>

* Với CSS:

.breadcrumbs{

    margin: 50px;

}

.breadcrumbs\_\_item{

    list-style: none;

    display: inline-block;

}

.breadcrumbs\_\_item:not(:last-of-type)::after{ /\* Ap dung cho cac item dau tien cua list tru item cuoi cung \*/

    content: '/';

    margin: 0 5px;

    color: #ccc;

}

.breadcrumbs\_\_link{

    text-decoration: none;

    color: #999;

}

.breadcrumbs\_\_link:hover{

    text-decoration: underline;

}

.breadcrumbs\_\_link--active{

    color: #009578;

    font-weight: 800;

}

## Badges



* Với HTML

<div class="badges">

        <i class="badges\_\_cart fas fa-shopping-cart"></i>

        <span class="badges\_\_notice">2</span>

    </div>

* Với CSS

.badges{

    margin: 30px;

    position: relative;

}

.badges\_\_cart{

    font-size: 40px;

    color: #b9b9b9;

}

.badges\_\_notice{

    position: absolute;

    padding: 3px 6px;

    background-color: #fff;

    color: #030303;

    font-size: 15px;

    line-height: 12px;

    border-radius: 10px;

    border: 2px solid #b9b9b9;

    left: 37px;

    top: -4px;

}

## Buttons

* Các nút được tạo bằng phần tử BUTTON có chức năng giống như các nút được tạo bằng phần tử INPUT, nhưng chúng cung cấp khả năng hiển thị phong phú hơn: phần tử BUTTON có thể có nội dung.
* Ví dụ: phần tử BUTTON có chứa hình ảnh có chức năng giống như và có thể giống phần tử INPUT có kiểu được đặt thành "hình ảnh", nhưng loại phần tử BUTTON cho phép nội dung.
* Thay vì sử dụng <input type = "button" ..> thì hãy sử dụng <button></button>

## Progress



* HTML:

<div class="progress\_\_bar" style="--width: 8;" data-label="Loading..."></div>

* CSS:

\*{

    padding: 0;

    margin: 0;

    outline: none;

    box-sizing: border-box;

}

\*::before{

    box-sizing: border-box;

}

/\* Progress \*/

.progress\_\_bar{

    position: relative;

    max-width: 500px;

    margin: 50px auto 0;

    height: 3em;

    background: #111;

    border-radius: 1.5em;

    color: #fff;

}

.progress\_\_bar::before{

    content: attr(data-label);

    display: flex;

    align-items: center;

    position: absolute;

    left: .5em;

    top: .5em;

    bottom: .5em;

    width: calc(var(--width, 0) \* 1%);

    min-width: 2rem;

    max-width: calc(100% - 1em);

    background: cornflowerblue;

    border-radius: 1em;

    padding-left: 1em;

}

* Javascript

const progressBar = document.getElementsByClassName('progress\_\_bar')[0]

console.log(progressBar);

setInterval(()=>{

   const computerStyle = getComputedStyle(progressBar);

   const witdh = parseFloat(computerStyle.getPropertyValue('--width')) || 0

   progressBar.style.setProperty('--width', witdh+ .1)

}, 10);

## Spinners



* HTML:

<div class="spinner">

        <div></div>

        <div></div>

    </div>

* CSS:

.spinner{

    width: 100px;

    height: 100px;

    position: relative;

    margin: 50px auto 0;

}

.spinner div{

    box-sizing: border-box;

    position: absolute;

    width: 100%;

    height: 100%;

    border: 10px solid transparent;

    border-top-color: #ad60f5;

    border-radius: 50%;

    animation: spinnerOne 1.2s linear infinite;

}

.spinner div:nth-child(2){

    border: 10px solid transparent;

    border-bottom-color: #ad60f5;

    animation: spinnerTwo 1.2s linear infinite;

}

@keyframes spinnerOne{

    0% {transform: rotate(0deg); border-width: 10px;}

    50% {transform: rotate(180deg); border-width: 1px;}

    100% {transform: rotate(360deg); border-width: 10px;}

}

@keyframes spinnerTwo{

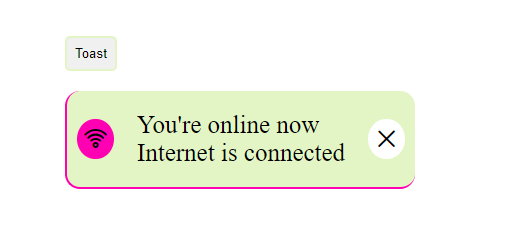
    0% {transform: rotate(0deg); border-width: 1px;}

    50% {transform: rotate(180deg); border-width: 10px;}

    100% {transform: rotate(360deg); border-width: 1px;}

}

## Toasts



* HTML:

<div class="toast\_\_wapper">

        <button class="toast\_\_button">Toast</button>

        <div class="toast">

            <div class="toast\_\_content">

                <div class="icon"><i class="uil uil-wifi"></i></div>

                <div class="details">

                    <span>You're online now</span>

                    <p>Internet is connected</p>

                </div>

                <div class="icon--close"><i class="uil uil-multiply"></i></div>

            </div>

        </div>

    </div>

* CSS:

.toast\_\_wapper{

    max-width: 500px;

    margin: 200px auto 0;

}

.toast\_\_button{

    padding: 8px;

    border: 2px solid #e3f5c4;

    border-radius: 5px;

}

.toast{

    margin-top: 20px;

    width: 100%;

    margin-bottom: 30px;

}

.toast\_\_content{

    width: 70%;

    display: flex;

    background: #e3f5c4;

    flex-wrap: wrap;

    padding: 20px 40px;

    font-size: 25px;

    position: relative;

    border-radius: 15px;

    border-left: 2px solid #ff00b3;

    border-bottom: 2px solid #ff00b3;

}

.toast\_\_content .icon{

    position: absolute;

    left: 10px;

    top: 50%;

    transform: translateY(-50%);

    padding: 6px;

    background-color: #ff00b3;

    border-radius: 100%;

}

.toast\_\_content .details{

    margin: 0 auto;

}

.toast\_\_content .icon--close{

    position: absolute;

    right: 10px;

    top: 50%;

    transform: translateY(-50%);

    padding: 6px;

    background-color: #ffffff;

    border-radius: 100%;

}

* JS:

$(document).ready(function(){

    $('.icon--close').click(function() {

        $('.toast\_\_content').hide();

    });

    $('.toast\_\_button').click(function() {

        $('.toast\_\_content').toggle();

    });

});

## Tooltips



* HTML:

<div class="tooltip\_\_wapper">

        <div class="icon facebook">

            <div class="tooltip">Facebook</div>

            <a href="https://www.facebook.com"><span><i class="fab fa-facebook-f"></i></span></a>

        </div>

        <div class="icon youtube">

            <div class="tooltip">Youtube</div>

            <a href="https://www.youtube.com/"><span><i class="fab fa-youtube"></i></span></a>

        </div>

        <div class="icon skype">

            <div class="tooltip">Skype</div>

            <a href="https://www.skype.com"><span><i class="fab fa-skype"></i></span></a>

        </div>

    </div>

* CSS:

.tooltip\_\_wapper{

    max-width: 1000px;

    margin: 50px auto 0;

    display: flex;

    flex-wrap: wrap;

}

.tooltip\_\_wapper .icon{

    margin: 0 auto;

    position: relative;

}

.tooltip\_\_wapper .icon .tooltip{

    background-color: #fff;

    box-shadow: 0 10px 10px rgba(0, 0, 0, 0.1);

    position: absolute;

    top: 3px;

    left: -10px;

    padding: 10px 20px;

    border-radius: 20px;

    display: none;

}

.tooltip\_\_wapper .icon .tooltip::before{

    content: "";

    height: 15px;

    width: 15px;

    background: #fff;

    position: absolute;

    top: 30px;

    left: 50%;

    transform: translateX(-50%) rotate(45deg);

}

.tooltip\_\_wapper .icon:hover .tooltip{

    display: block;

}

.tooltip\_\_wapper a{

    text-decoration: none;

    color: #333;

    position: absolute;

    top: 50px;

}

.tooltip\_\_wapper span{

    width: 60px;

    height: 60px;

    background-color:#fff;

    display: block;

    line-height: 60px;

    text-align: center;

    border-radius: 50%;

    box-shadow: 0 10px 10px rgba(0, 0, 0, 0.1);

    font-size: 25px;

}

.tooltip\_\_wapper span .fa-facebook-f{

    color: #1877F2;

}

.tooltip\_\_wapper span .fa-youtube{

    color: #FF0000;

}

.tooltip\_\_wapper span .fa-skype{

    color: #0C65A4;

}

.tooltip\_\_wapper .facebook:hover .tooltip,

.tooltip\_\_wapper .facebook:hover .tooltip::before,

.tooltip\_\_wapper .facebook:hover span{

    background-color: #1877F2;

    color: #fff;

}

.tooltip\_\_wapper .facebook:hover span .fa-facebook-f{

    color: #fff;

}

.tooltip\_\_wapper .youtube:hover .tooltip,

.tooltip\_\_wapper .youtube:hover .tooltip::before,

.tooltip\_\_wapper .youtube:hover span{

    background-color: #FF0000;

    color: #fff;

}

.tooltip\_\_wapper .youtube:hover span .fa-youtube{

    color: #fff;

}

.tooltip\_\_wapper .skype:hover .tooltip,

.tooltip\_\_wapper .skype:hover .tooltip::before,

.tooltip\_\_wapper .skype:hover span{

    background-color: #0C65A4;

   color: #fff;

}

.tooltip\_\_wapper .skype:hover span .fa-skype{

    color: #fff;

}

## Scrollspy



* HTML:

<div class="scrollspy\_\_wapper">

        <div class="section\_\_scro" id="one">

            One

        </div>

        <div class="section\_\_scro" id="two">

            Two

        </div>

        <div class="section\_\_scro" id="three">

            Three

        </div>

        <div class="section\_\_scro" id="four">

            Four

        </div>

        <div class="section\_\_scro" id="five">

            Five

        </div>

        <nav>

            <a href="#one">One</a>

            <a href="#two">Two</a>

            <a href="#three">Three</a>

            <a href="#four">Four</a>

            <a href="#five">Five</a>

        </nav>

    </div>

* CSS:

.section\_\_scro{

    position: relative;

    width: 100%;

    height: 100vh;

    display: flex;

    justify-content: center;

    align-items: center;

    font-size: 6em;

    font-weight: bold;

}

.section\_\_scro:nth-child(even){

    background: #efefef;

}

nav{

    position: fixed;

    top: 0;

    width: 100%;

    background: #000;

    text-align: center;

    z-index: 1;

}

nav a{

    position: relative;

    color: #fff;

    text-decoration: none;

    padding: 10px 20px;

    font-size: 1.6em;

    display: inline-block;

}

nav a:hover,

.section\_\_scro#one:hover ~ nav a[href="#one"],

.section\_\_scro#two:hover ~ nav a[href="#two"],

.section\_\_scro#three:hover ~ nav a[href="#three"],

.section\_\_scro#four:hover ~ nav a[href="#four"],

.section\_\_scro#five:hover ~ nav a[href="#five"]{

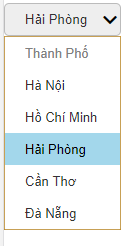
    background: #fff;

    color: #000;

}

### Dropdowns

* Sử dụng select – Option



* HTML:

<div class="dropdown">

        <select class="select\_\_wapper">

            <option disabled selected>Thành Phố</option>

            <option value="Hà Nội">Hà Nội</option>

            <option value="Hồ Chí Minh">Hồ Chí Minh</option>

            <option value="Hải Phòng">Hải Phòng</option>

            <option value="Cần Thơ">Cần Thơ</option>

            <option value="Đà Nẵng">Đà Nẵng</option>

        </select>

    </div>

* CSS:

select{

    appearance: none;

    -webkit-appearance: none;

    -moz-appearance: none;

    -o-appearance: none;

    -ms-appearance: none;

}

.dropdown{

    position: relative;

    margin: 20px 0;

    display: inline-block;

}

.select\_\_wapper{

    padding: 8px 20px;

    color: #333;

    background-color: #eee;

    border: 1px solid #ddd;

    cursor: pointer;

    border-radius: 5px;

    position: relative;

    top: 0;

    left: 0;

}

.dropdown::after{

    content: "\f078";

    font-family: "Font Awesome 5 Free";

    font-weight:600;

    position: absolute;

    right: 4px;

    top: 50%;

    transform: translateY(-50%);

}

.select\_\_wapper:focus,

.select\_\_wapper:hover{

    outline: none;

    border: 1px solid #bbb;

}

.select\_\_wapper option{

    background-color: #fff;

}

### Xóa thanh cuộn trong Dropdowns

* Ta thêm đoạn CSS sau vào style:

select{

    appearance: none;

    -webkit-appearance: none;

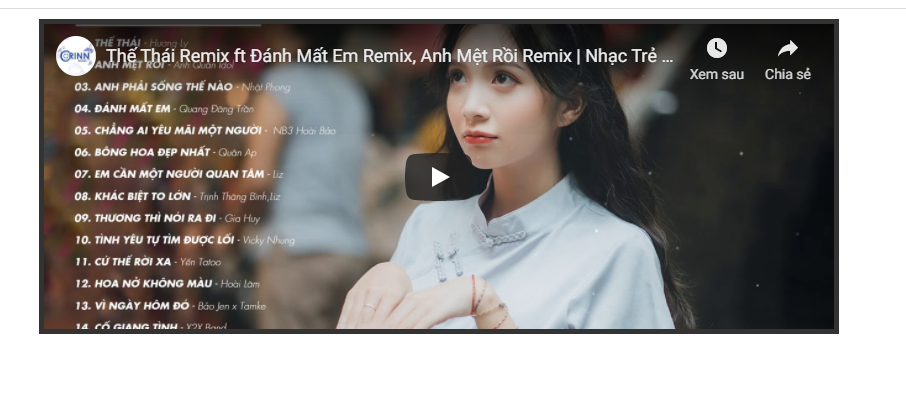
    -moz-appearance: none;

    -o-appearance: none;

    -ms-appearance: none;

}

## Iframe



* HTML:

<div class="iframe\_\_wapper">

        <iframe width="560" height="315" src="https://www.youtube.com/embed/0YgMdHE196k" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

    </div>

* CSS:

.iframe\_\_wapper{

    max-width: 800px;

    margin: 10px auto;

}

.iframe\_\_wapper iframe{

    width: 100%;

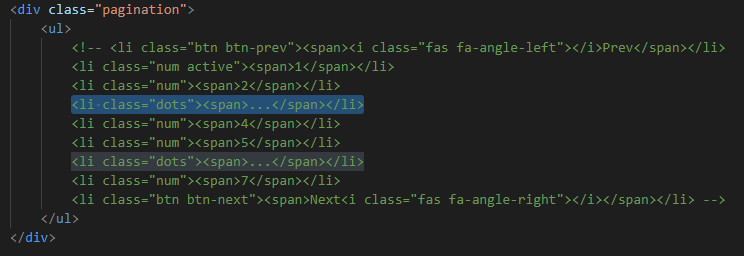
    border: 5px solid #333;

}

## Pagination



* HTML:



* CSS:

.pagination ul {

    display: flex;

    background: #fff;

    padding: 8px;

    border-radius: 50px;

    color: #20b2aa;

}

.pagination ul li {

    list-style: none;

    height: 45px;

    line-height: 45px;

    text-align: center;

    font-size: 18px;

    font-weight: 500;

    cursor: pointer;

    border-radius: 50%;

    transition: all 0.3s ease;

}

.pagination ul li.dots {

    font-size: 22px;

    cursor: default;

}

.pagination ul li.num {

    width: 45px;

    height: 45px;

    margin: 0px 3px;

}

.pagination ul li.active,

.pagination ul li.num:hover,

.pagination ul li.btn:hover {

    background: #20b2aa;

    color: #fff;

}

.pagination ul li i.fas {

    padding: 0px 7px;

}

.pagination ul li.btn {

    padding: 0 20px;

}

.pagination ul li.btn-prev {

    border-radius: 25px 5px 5px 25px;

}

.pagination ul li.btn-next {

    border-radius: 5px 25px 25px 5px;

}

* JS:

const ulTag = document.querySelector("ul");

let totalPage = 20;

function element(totalPage, page) {

    let liTag = '';

    let activeLi;

    let beforePage = page - 1; //beforePage = 5-1 = 4

    let afterPage = page + 1; //afterPage = 5+1 = 6

    if (page > 1) {

        liTag += '<li class="btn btn-prev" onclick=\"element(totalPage, ' + (page - 1) + ')"><span><i class="fas fa-angle-left"></i>Prev</span></li>';

    }

    if (page > 2) {

        liTag += '<li class="num" onclick=\"element(totalPage, 1)"><span>1</span></li>';

        if (page > 3) {

            liTag += '<li class="dots"><span>...</span></li>';

        }

    }

    for (let pageLength = beforePage; pageLength <= afterPage; pageLength++) {

        // ẩn số 0 và số 21

        if (pageLength > totalPage) {

            continue;

        }

        if (pageLength == 0) {

            pageLength = pageLength + 1;

        }

        // hiện thị active

        if (page == pageLength) {

            activeLi = "active";

        } else {

            activeLi = "";

        }

        liTag += '<li class="num ' + activeLi + '" onclick=\"element(totalPage, ' + pageLength + ')"><span>' + pageLength + '</span></li>';

    }

    if (page < totalPage - 1) {

        if (page < totalPage - 2) {

            liTag += '<li class="num"><span>...</span></li>';

        }

        liTag += '<li class="dots" onclick=\"element(totalPage, ' + totalPage + ')"><span>' + totalPage + '</span></li>'

    }

    if (page < totalPage) {

        liTag += '<li class="btn btn-next"  onclick=\"element(totalPage, ' + (page + 1) + ')"><span>Next<i class="fas fa-angle-right"></i></span></li>';

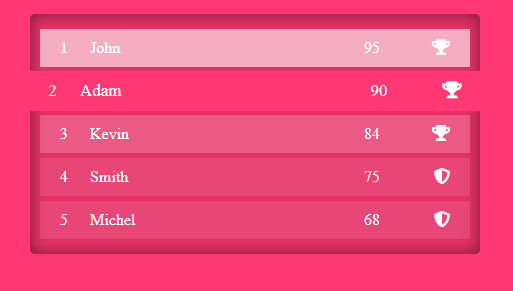
    }

    ulTag.innerHTML = liTag;

}

element(totalPage, 5);

## List group



* HTML:

<ul>

        <li>

            <span class="number">1</span>

            <span class="name">John</span>

            <span class="points">95</span>

            <span class="badge"><i class="fas fa-trophy"></i></span>

        </li>

        <li>

            <span class="number">2</span>

            <span class="name">Adam</span>

            <span class="points">90</span>

            <span class="badge"><i class="fas fa-trophy"></i></span>

        </li>

        <li>

            <span class="number">3</span>

            <span class="name">Kevin</span>

            <span class="points">84</span>

            <span class="badge"><i class="fas fa-trophy"></i></span>

        </li>

        <li>

            <span class="number">4</span>

            <span class="name">Smith</span>

            <span class="points">75</span>

            <span class="badge"><i class="fas fa-shield-alt"></i></span>

        </li>

        <li>

            <span class="number">5</span>

            <span class="name">Michel</span>

            <span class="points">68</span>

            <span class="badge"><i class="fas fa-shield-alt"></i></span>

        </li>

    </ul>

* CSS:

ul{

    position: relative;

    max-width: 450px;

    margin: 100px auto 0;

    padding: 10px;

    box-sizing: border-box;

    background: rgba(0, 0, 0, .1);

    box-shadow: inset 0 0 15px rgba(0, 0, 0, .5);

    border-radius: 5px;

    overflow: hidden;

}

ul li {

    display: flex;

    background: rgba(255, 255, 255, .1);

    padding: 10px 20px;

    color: #fff;

    margin: 5px 0;

    transition: .5s;

}

ul li:nth-child(1){

    background: rgba(255, 255, 255, .6);

}

ul li:nth-child(2){

    background: rgba(255, 255, 255, .4);

}

ul li:nth-child(3){

    background: rgba(255, 255, 255, .2);

}

ul li span:nth-child(1){

    max-width: 30px;

    width: 100%;

}

ul li span:nth-child(2){

    max-width: 200px;

    width: 100%;

}

ul li span:nth-child(3){

    max-width: 100px;

    width: 100%;

    text-align: right;

}

ul li span:nth-child(4){

    max-width: 70px;

    width: 100%;

    text-align: right;

}

ul li:hover{

    transform: scale(1.06);

    background: #ff3772;

}

## Simple Star Rating\*



* HTML:

<div class="rating">

        <input type="radio" name="star" id="star1"><label for="star1"></label>

        <input type="radio" name="star" id="star2"><label for="star2"></label>

        <input type="radio" name="star" id="star3"><label for="star3"></label>

        <input type="radio" name="star" id="star4"><label for="star4"></label>

        <input type="radio" name="star" id="star5"><label for="star5"></label>

    </div>

* CSS:

.rating{

    position: absolute;

    top: 50%;

    left: 50%;

    transform: translate(-50%, -50%) rotateY(180deg);

    display: flex;

}

.rating input{

    display: none;

}

.rating label{

    display: block;

    cursor: pointer;

    width: 50px;

    margin-right: 8px;

}

.rating label::before{

    content: "\f005";

    font-family: "Font Awesome 5 Free";

    font-weight: 600;

    position: relative;

    display: block;

    font-size: 50px;

    color: #101010;

}

/\* Màu của ngôi sao đè lên ngôi sao  \*/

.rating label::after{

    content: "\f005";

    font-family: "Font Awesome 5 Free";

    font-weight: 600;

    position: absolute;

    top: 0;

    display: block;

    font-size: 50px;

    color: #1f9cff;

    opacity: 0;

    transition: .5s;

    text-shadow: 0 2px 5px rgba(0, 0, 0, .5);

}

.rating label:hover::after,

.rating label:hover ~ label::after,

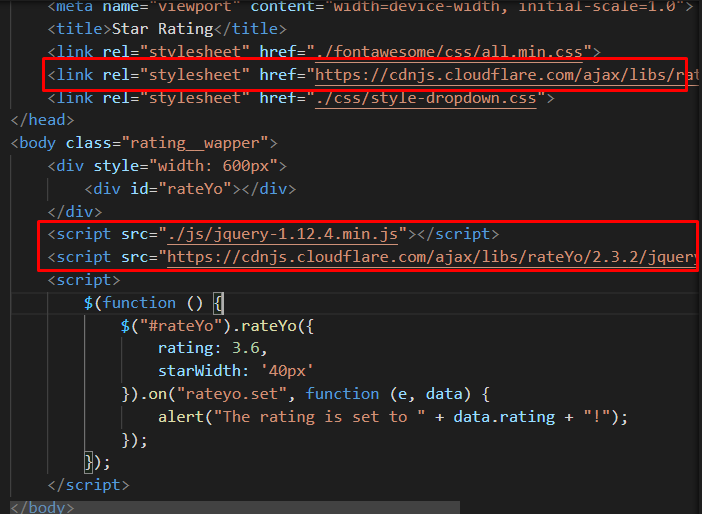
.rating input:checked ~ label::after{

    opacity: 1;

}

### Sử dụng thư viện RateYo

* Truy cập thư viện RateYo: [https://rateyo.fundoocode.ninja/#](https://rateyo.fundoocode.ninja/) và tìm hiểu theo hướng dẫn



# Form

## Form Validation

## Search Bar

## Social Login Form

## Modal Login Form

## Form with Icons

# Image

## Responsive image

.responsive {  
  width: 100%;  
  max-width: 400px;  
  height: auto;  
}

## Hover ảnh



* HTML:

<div class="image\_\_wapper">

        <div class="box\_\_gird">

            <div class="box">

                <div class="box\_\_image">

                    <img src="./img/img-1.png" alt="">

                </div>

                <div class="box\_\_intro">

                    <h3 class="title">

                        Hoa hong

                    </h3>

                    <p>Sed sed orci nunc. Nullam ac dolor a ex sagittis vehicula. Phasellus feugiat erat quis placerat laoreet. Etiam id ligula a libero ornare suscipit sit amet vel mauris. Fusce non turpis in lorem placerat bibendum vitae eget libero. Cras sodales tempor libero. Vivamus sed mi eros. Maecenas ornare eleifend dolor, non tincidunt eros vulputate sit amet</p>

                </div>

            </div>

        </div>

        <div class="box\_\_gird">

            <div class="box">

                <div class="box\_\_image">

                    <img src="./img/img-1.png" alt="">

                </div>

                <div class="box\_\_intro">

                    <h3 class="title">

                        Hoa hong

                    </h3>

                    <p>Sed sed orci nunc. Nullam ac dolor a ex sagittis vehicula. Phasellus feugiat erat quis placerat laoreet. Etiam id ligula a libero ornare suscipit sit amet vel mauris. Fusce non turpis in lorem placerat bibendum vitae eget libero. Cras sodales tempor libero. Vivamus sed mi eros. Maecenas ornare eleifend dolor, non tincidunt eros vulputate sit amet</p>

                </div>

            </div>

        </div>

            </div>

* CSS:

.image\_\_wapper{

    max-width: 1200px;

    margin: 10px auto 0;

    display: flex;

    flex-wrap: wrap;

}

.box\_\_gird{

    width: 25%;

}

.box{

    max-width: 280px;

    border: 1px solid #333;

    margin: 10px;

    overflow: hidden;

    position: relative;

}

.box\_\_image{

    width: 100%;

    max-height: 350px;

}

.box\_\_image img{

    width: 100%;

    height: 100%;

    transition: transform 1s;

}

.box:hover .box\_\_image img{

    transform: scale(1.2);

}

.box\_\_intro{

    position: absolute;

    top: 10px;

    left: 10px;

    bottom: 10px;

    right: 10px;

    background: rgba(0,0,0,.8);

    display: flex;

    text-align: center;

    flex-direction: column;

    justify-content: center;

    padding: 10px;

    color: darkred;

    font-weight: 600;

    transform: scaleY(0);

    transition: transform 0.3s;

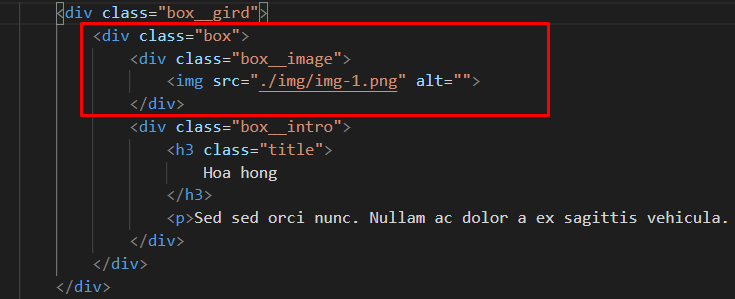
}

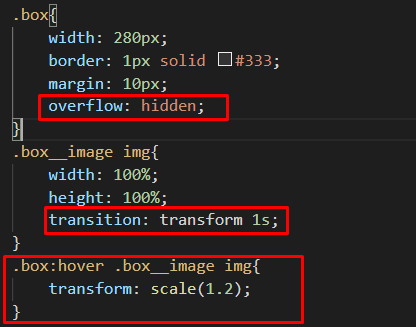
.box:hover .box\_\_intro{

    transform: scaleY(1);

}

### Khi hover đến ảnh thì ảnh sẽ chuyển động giãn ra hay co lại



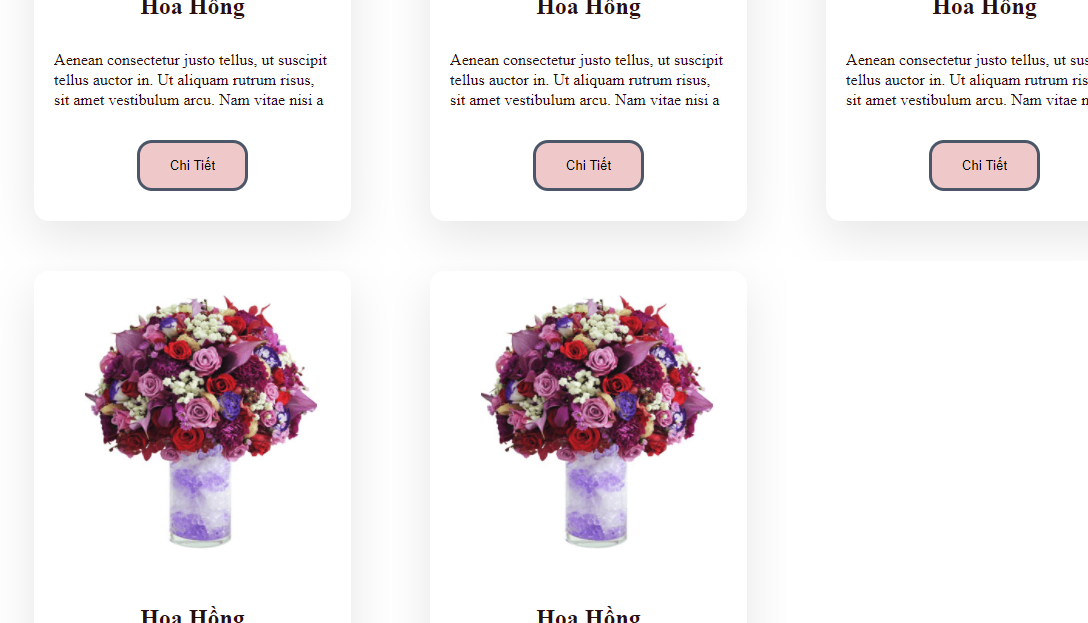




## Thumbnail

## Zoom image

# Cards



* HTML:

<div class="card\_\_wapper">

        <a href="#" class="card\_\_gird">

            <div class="car">

                <img class="card\_\_img" src="./img/img-1.png" alt="Hinh anh binh hoa">

                <div class="card\_\_content">

                    <h2 class="card\_\_content--title">Hoa Hồng</h2>

                    <p class="card\_\_content--cmd">Aenean consectetur justo tellus, ut suscipit tellus auctor in. Ut aliquam rutrum risus, sit amet vestibulum arcu. Nam vitae nisi a arcu vulputate rhoncus. Aliquam varius libero quis mi porta eleifend. Nulla ultricies nunc erat, et rhoncus sem convallis non. Nullam vitae dui ac mauris pharetra ullamcorper. Nullam tortor libero, tempus quis erat ac, tincidunt tincidunt ex.</p>

                    <button class="card\_\_content--btn">Chi Tiết</button>

                </div>

            </div>

        </a>

        <a href="#" class="card\_\_gird">

            <div class="car">

                <img class="card\_\_img" src="./img/img-1.png" alt="Hinh anh binh hoa">

                <div class="card\_\_content">

                    <h2 class="card\_\_content--title">Hoa Hồng</h2>

                    <p class="card\_\_content--cmd">Aenean consectetur justo tellus, ut suscipit tellus auctor in. Ut aliquam rutrum risus, sit amet vestibulum arcu. Nam vitae nisi a arcu vulputate rhoncus. Aliquam varius libero quis mi porta eleifend. Nulla ultricies nunc erat, et rhoncus sem convallis non. Nullam vitae dui ac mauris pharetra ullamcorper. Nullam tortor libero, tempus quis erat ac, tincidunt tincidunt ex.</p>

                    <button class="card\_\_content--btn">Chi Tiết</button>

                </div>

            </div>

        </a>

        <a href="#" class="card\_\_gird">

            <div class="car">

                <img class="card\_\_img" src="./img/img-1.png" alt="Hinh anh binh hoa">

                <div class="card\_\_content">

                    <h2 class="card\_\_content--title">Hoa Hồng</h2>

                    <p class="card\_\_content--cmd">Aenean consectetur justo tellus, ut suscipit tellus auctor in. Ut aliquam rutrum risus, sit amet vestibulum arcu. Nam vitae nisi a arcu vulputate rhoncus. Aliquam varius libero quis mi porta eleifend. Nulla ultricies nunc erat, et rhoncus sem convallis non. Nullam vitae dui ac mauris pharetra ullamcorper. Nullam tortor libero, tempus quis erat ac, tincidunt tincidunt ex.</p>

                    <button class="card\_\_content--btn">Chi Tiết</button>

                </div>

            </div>

        </a>

</div>

* CSS:

.card\_\_wapper{

    max-width: 1200px;

    height: auto;

    display: flex;

    flex-wrap: wrap;

    margin: 50px auto 0;

}

.card\_\_gird{

    width: 33%;

    overflow: hidden;

    text-decoration: none;

    color: #2c0e0e;

}

.car{

    width: 80%;

    margin: 10px auto 40px;

    box-shadow: 0 1rem 3rem rgba(0, 0, 0, 0.1);

    border-radius: 15px;

}

.card\_\_img{

    width: 100%;

    max-height: 300px;

    border-top-left-radius: 15px;

    border-top-right-radius: 15px;

}

.card\_\_content{

    padding: 30px 20px;

}

.card\_\_content--title{

    padding-bottom: 30px;

    text-align: center;

}

.card\_\_content--cmd{

    max-height: 60px;

    line-height: 20px;

    overflow: hidden;

}

.card\_\_content--btn{

    margin: 30px auto 0;

    padding: 15px 30px;

    display: flex;

    justify-items: center;

    background: rgba(201, 76, 76, 0.3);

    border: 3px solid rgba(39,62,84,0.82);

    border-radius: 15px;

}

@media only screen and (max-width:575px){

    .card\_\_gird{

        width: 80%;

        margin: 0 auto;

    }

}

@media only screen and (min-width:576px) and (max-width:991px){

    .card\_\_gird{

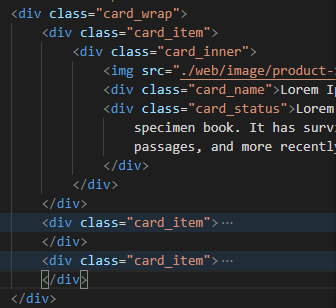
        width: 50%;

    }

}

## Responsive card

HTML:



CSS:

.card\_wrap{

    padding: 20px;

    width: 100%;

    display: flex;

    flex-wrap: wrap;

    justify-content: space-between;

    .card\_item{

        padding: 15px 25px;

        width: 25%;

&:nth-child(4n+4){

padding-right: 0;

         }

        .card\_inner{

            background: rgb(153, 153, 153);

            border-radius: 5px;

            padding: 35px 20px;

            min-width: 225px;

            min-height: 315px;

            width: 100%;

            height: 100%;

            img{

                width: 100%;

                min-height: 265px;

            }

            .card\_name{

                color: #e36686;

                font-weight: 900;

                letter-spacing: 2px;

                text-transform: uppercase;

                font-size: 20px;

                white-space: nowrap;

                overflow: hidden;

                text-overflow: ellipsis;

            }

            .card\_status{

                font-size: 24px;

                line-height: 24px;

                overflow: hidden;

                max-height: 313px;

            }

        }

    }

}

@media (max-width: 1024px) {

    .card\_wrap{

        .card\_item{

            width: 33%;

        }

    }

}

# Căn giữa thẻ

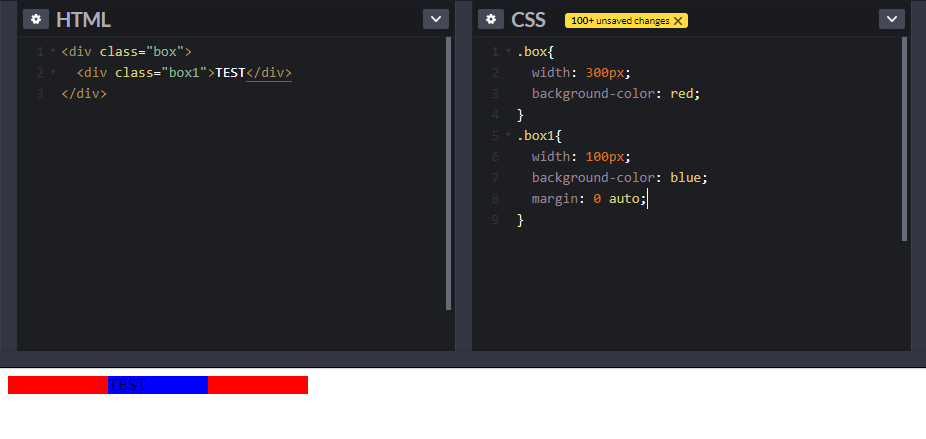
## Theo chiều ngang

### Inline

* Với các phần tử inline hoặc inline-\* thì ta dùng text-align: center;

### Block

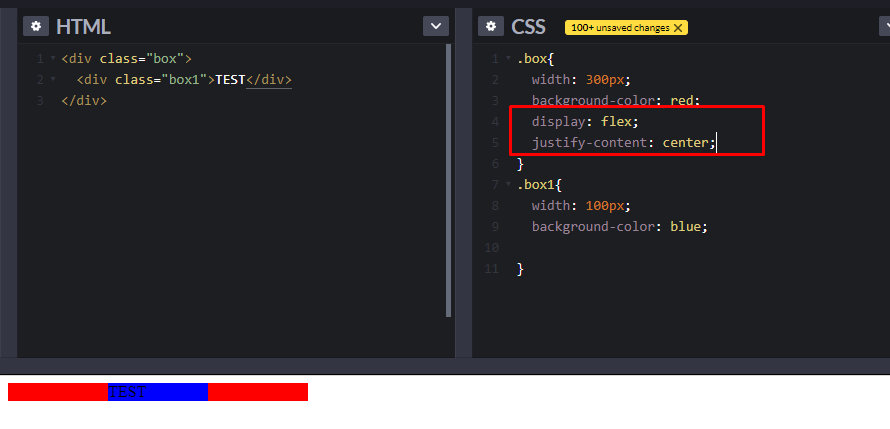
* Với các phần tử block có set width ta dùng margin: 0 auto;



* Ta cũng có thể dùng thuộc tính display: flex để căn giữa cho các khối

display: flex;

justify-content: center;



## Theo chiều dọc

### Nếu không biết height của thẻ cha

* Nếu không biết height thì ta có thể sử dụng padding để căn giữa:

padding-top: Apx;

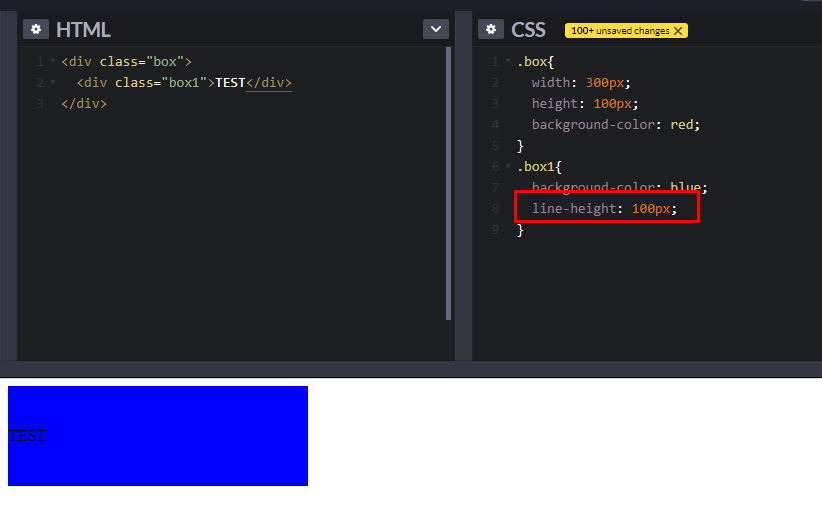
padding-bottom: Apx;



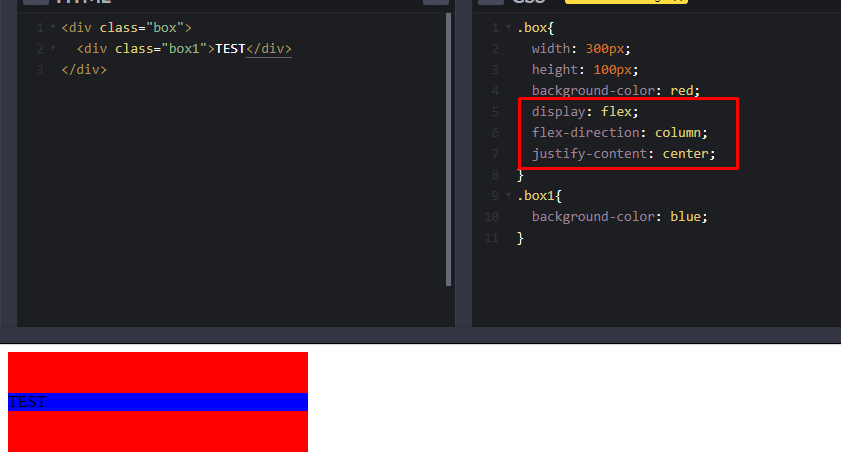
### Nếu biết height của thẻ cha

* Nếu biết height của khối cha thì ta cũng có thể sử dụng thuộc tính line-height:

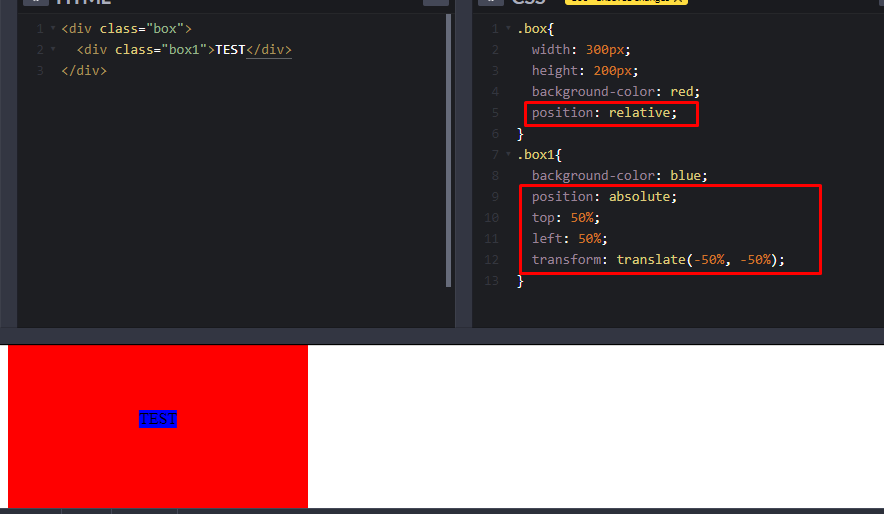
line-height: 100px;



* Ta cũng có thể dùng thuộc tính display: flex và thay đổi thuộc tính flex-direction thành column để nó có thể sắp xếp theo chiều dọc. Cộng với áp dụng thuộc tính justify-content: center;



* Sử dụng thuộc tính position để căn giữa.



# Các thuộc tính của style guide

## Style text

* Các thuộc tính để style text gồm có:

font-family:

color:

font-weight:

font-size:

letter-spacing:

line-height:

## Style ảnh

* Các thuộc tính để cho ảnh full màn hình gồm có:

background-image: url(url\_img);

background-repeat: no-repeat;

background-position: center;

background-size: cover;

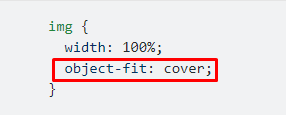
width: 100%;

height: 100%;

* Vát ảnh hình chéo

Ta vào trang web sau để vát ảnh: <http://www.cssplant.com/clip-path-generator>

* Sửa lỗi ảnh bị biến dạng khi thay đổi kích thước



Các giá trị Object-fit có thể nhận được là:

* + fill (default): Chiều rộng và chiều cao của hình ảnh đầy box. Hầu hết các cái này sẽ mess lên tỉ lệ.
  + cover : Những hình ảnh giữ tỉ lệ của nó và lấp đầy toàn bộ box, các bộ phận của nó thừa ra sẽ được cắt đi và sẽ không được hiển thị.
  + contain : Những hình ảnh giữ nó tỉ lệ và phóng to / thu nhỏ lại để phù hợp với bên trong box.
  + none : Lấy kích thước gốc và tỉ lệ.
  + scale-down : hiển thị như 1 trong số những cái ở trên (fill, cover, contai...), phụ thuộc vào cái nào tạo ra size ảnh nhỏ nhất.

## Style button

display: inline-block;

font-weight:

line-height:

color:

text-align: center;

text-decoration: none;

vertical-align: middle;

cursor: pointer;

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none; (Thuộc tính con trỏ chuột)

background-color:

border:

padding:

font-size:

border-radius:

transition:

## Style field

display: block;

width: 100%;

min-height:

padding:

font-size:

font-weight:

line-height:

color:

background-color:

background-clip: padding-box;

border:

border-radius:

transition:

## Style màu chuyển đổi

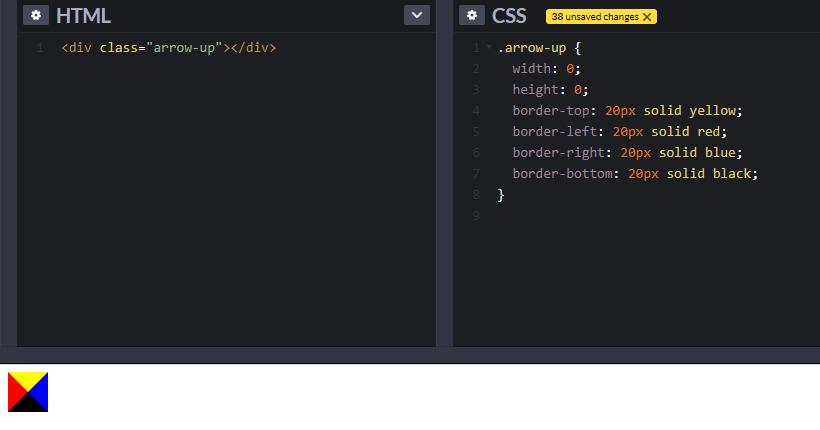
background-color: linear-gradient(#rad, #color1, #color2);

## Tạo dấu gạch dọc ngăn cách



* Ta dùng lớp giả ::after hoặc ::before

## Làm mũi tên css



* Nếu để border-width thì mũi tên có hình chữ nhật.

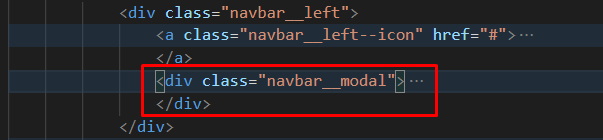
# Tạo hiệu ứng animation



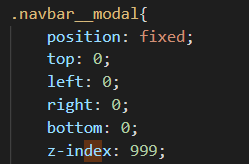
# Tạo modal box

**Bước 1**: Tạo lớp overlay

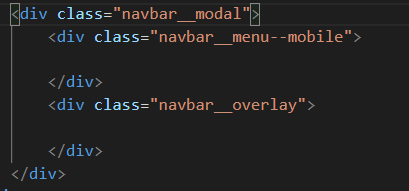
* Tạo khối modal nằm trong khối chứa thanh menu



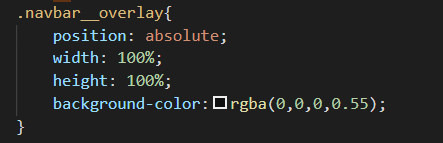
* Style khối modal sao cho nó có thể che hết màn hình



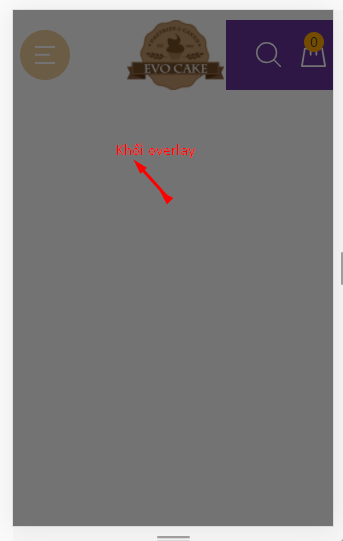
* Trong khối modal có các khối menu dành cho mobile và khối overlay.



* Khối overlay có nhiệm vụ che hết toàn bộ trang web và có một lớp phủ đen mờ. Style như sau:

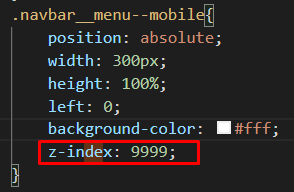


* Vì ta khai báo thuộc tính position: fixed cho khối modal nên khi đặt absolute cho khối overlay thì nó sẽ nhận khối modal là thẻ cha nên ta chỉ cần style cho width và height là 100% là khối này đã full màn hình. Kết quả

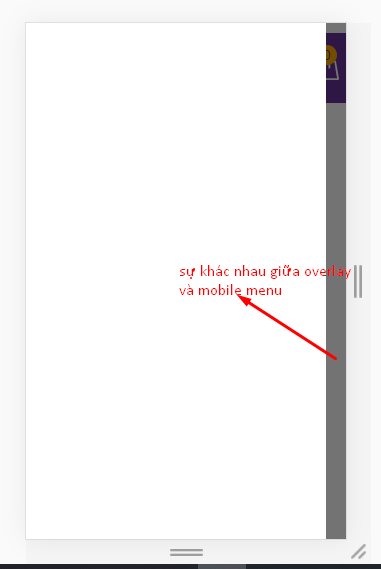


**Bước 2**: Tạo khối menu mobile

* Tạo khối menu mobile nằm trên khối overlay với z-index cao hơn z-index của khối modal



* Kết quả:

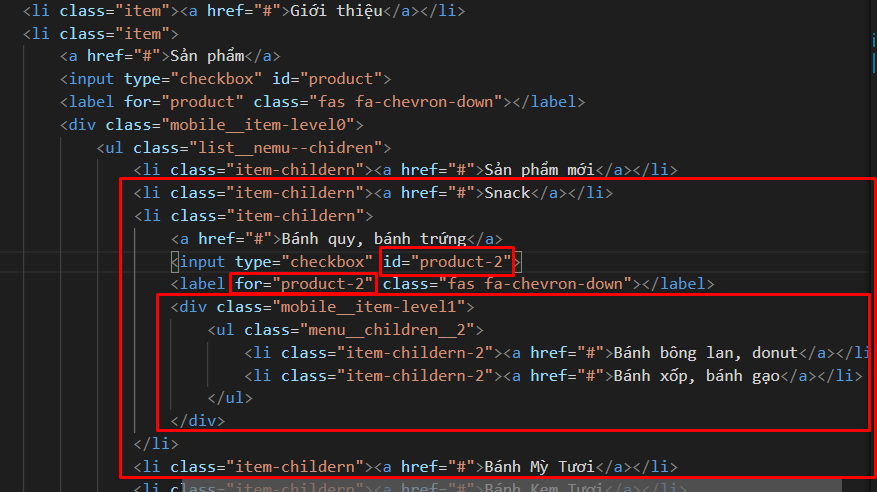


* Ta tiếp tục style theo yêu cầu của website là tạo thanh navbar dọc đa cấp.

**Bước 3**: jhfgiy

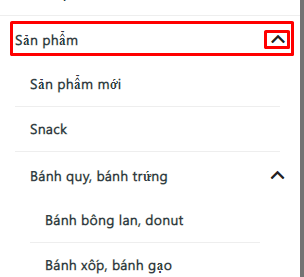
# Tạo menu dọc đa cấp

**Bước 1**: Tạo khung html cho menu đa cấp

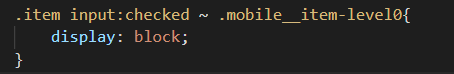


**Bước 2**: Style cho các html

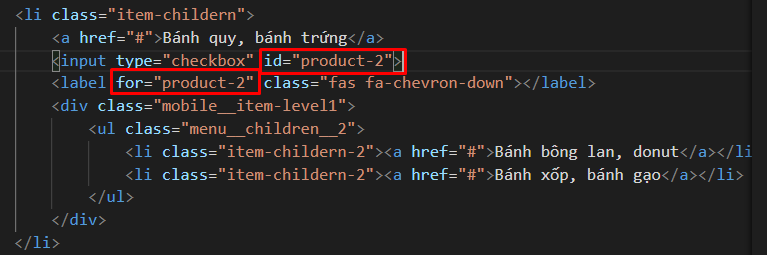
**Bước 3**: Style để cho các khối cấp thấp ẩn, hiện khi click vào các khối cấp cao hơn



* Khi click vào Sản phẩm thì khối con sẽ hiện ra.
* Để làm được như vậy ta style như sau

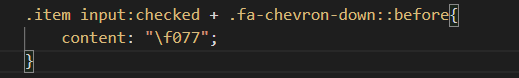


* Lưu ý ta đặt for của label giống với id của input thì input mới hoạt động. Và thẻ input phải lên trước thẻ label thì ta mới cho mũi tên thay đổi khi click vào được

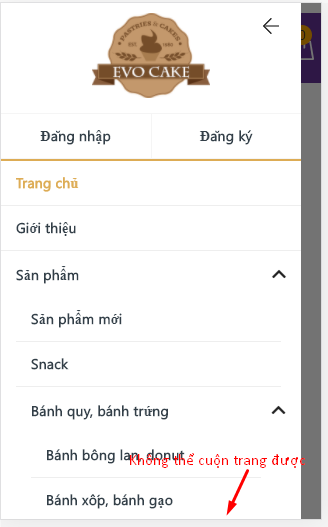


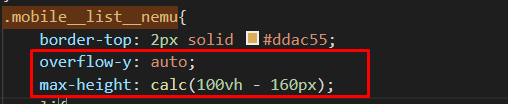
**Bước 4**: Tạo mũi tên khi click vào và mũi tên thay đổi

* Khi đã tạo được các thẻ như các bước trên ta chỉ cần style để thay đổi mũi tên khi click vào như sau:



* Khi có nhiều mục trong nav sẽ dẫn đến các mục sẽ bị tràn ra khỏi khung nhìn và ta không thể cuộn lại được. Để xử lý lỗi ta làm như sau:





* Thêm các thuộc tính trên vào khối
* Chú ý: Ở hướng dẫn này ta chỉ mới làm để khi click vào mũi tên thì nó mới hiện ra các li con. Để click vào đâu trên thanh li nó cũng hiện ra thanh li con thì ta xem hướng dẫn bằng từ khoá “Accordion”.

# Tạo Carousel

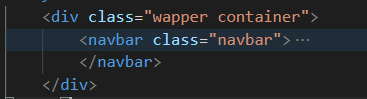
# Bố cục của một website

## Header

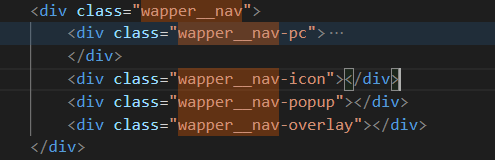
### Navbar

Bước 1: Tạo khung html

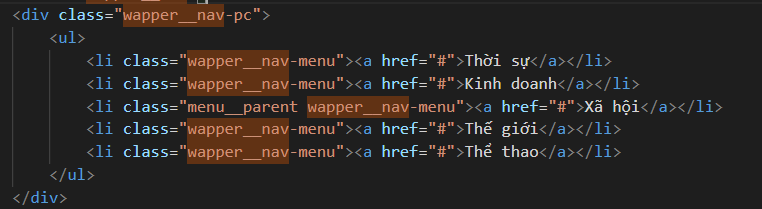
* Tạo có một div bao quanh toàn bộ trang web có khai báo thêm class container (**wapper**).



* Trong thẻ **wapper** là tạo một thẻ navbar (**navbar**), thẻ navbar này giống thẻ [row](#_Row)
* Trong thẻ navbar tạo 3 thẻ ngang hàng là thẻ **navbar\_\_left, navbar\_\_middle** và **navbar\_\_right** các thẻ này giống thẻ [col](#_Col)
* (1) Trong thẻ **navbar\_\_left** có chứa các thẻ ul li để style navbar,
* (2) Trong thẻ **navbar** có chứa 4 thẻ div, một thẻ chứa các danh mục của navbar (**nav-pc**) , một thẻ chứa icon(**nav-icon**), một thẻ chứa các danh mục của navbar khi được co về màn hình mobile (**nav-mb**) và một thẻ sẽ bao quanh toàn bộ trang web làm màn hình mờ đi, phục vụ cho việc tạo modal box (**nav-overlay**)



* + Trong thẻ **nav-pc** có chứa thẻ ul > li > a > Nội dung thẻ



* + Trong thẻ **nav-icon** có thể không chứ gì để nhúng icon vào thẻ
  + Trong thẻ **nav-mb** sẽgiống y hệt như thẻ **nav-icon** nhưng chỉnh nó thành hàng dọc
  + Trong thẻ **nav-overlay** cũng là thẻ trống và ta sẽ style cho thẻ này bao quanh toàn bộ trang website để làm thuộc tính che mờ khi ta bật **nav-mb**
* (3) Trong thẻ **icon** thường có chứ các icon giỏ hàng, đăng xuất, đăng nhập, tìm kiếm,…

Bước 2: Style html

1. Style thẻ text
2. Style thẻ link
3. Căn khối ra giữa theo chiều ngang
4. Căn khối ra giữa theo chiều dọc
5. Style giỏ hàng
6. Tạo animation
7. Tạo menu cấp 2

Bước 3: Responsive

* Responsive trên các thiết bị
* Nhúng icon

Bước 4: Tạo modal box

Bước 5: Tạo hiệu ứng JS với navbar

1. Hiệu ứng navbar bám dính

Hiệu ứng icon back-to-top

## Content

### Carousel

### Card

### Forms

## Footer