<svg>Nội dung</svg>:

* fill:màu sắc bên trong ,
* stroke-width:kich thươc px;
* stroke-dasharray:vi du 100 chỉnh kích thước;
* stroke-dashoffset:400 tốc độ(sài animation);

position:sticky : đến vị trí đó thì sẽ cố định lại

* Thuộc tính đi kèm;
* Top:0;
* Min-width:50%;
* Height:100vh;

background: linear-gradient(0deg,#fff 50%,#2196f3)

Làm nhòe ảnh, ví dụ .myfilter { filter: blur(3px);}

1. **BEM VÀ CÁCH ĐẶT TÊN CLASS**

BEM : Block Element Modifier:

+ Block ( khối ) : là tên của components ( các thành phần ) mô tả tổng thể.

Ví dụ : heading-primary.

+ Element ( phần tử ) : là đặt tên các thành phần từ bổ trợ cho Block.

Ví dụ : heading-primary\_\_main,heading-primary\_\_sub.

+ Modifier ( bộ điều chỉnh ) : là các dạng khác biến thể của Block và Element , thay đổi trạng thái của Block.

Ví dụ : heading-primary—big.

Extends: kế thừa không bị lặp đi lặp lại code,

%CSSlink{

Color:red;

Text-decoration:none;

}

Kế thừa:

.list\_\_link{

@extend %CSSlink;

}

Lệnh CMD:

- cd chuyển thư mục.

+ cd .. :trở về thư mục trước nó , cha của nó.

+ cd users :đi vô thư mục users.

- mkdir test: tạo thư mục test.

- echo test\test.html : tạo file test.html trong thư mục test.

- del test\test.html : xóa file test.html trong thư mục test.

- rmdir test : xóa thư mục test đi.

Setup Node-SASS:

Lệnh CMD: ctrl + c : để thoát.

-npm init: vô sử npm.

-npm install node-sass –save: đê tạo package chuyển đổi sang sass.

-npm uninstall node-sass: để xóa package.

Cách 2:

-npm install –save-dev node-sass: cài package vào ở trạng thái là dev, khi code chúng ta mới có.

Node\_modules: chứa package mặc định.

***Chuyển đổi scss sang css***: package.json => tìm dong scripts :

“sasstocss” : “node-sass sass/main.scss scc/style.css -w”

* Tạo 1 file complie cass to css.bat : npm run sastocss

**FODER TRONG SASS:**

**- Abstracts/.**

**+ \_functions.scss.**

**+ \_mixin.scss.**

**+ \_variable.scss :** Lưu biến css.

**- Base/.**

**+ \_utilities.scss:** Lưu một số class có chức năng tắt ngắn gọn sử dụng lại text-left,..

**+ \_animation.scss:** Lưu keyframes.

**+ \_typography.scss:** Setting fonsize và các thuộc tính liên quan đên text element body,heading, tái sử dụng được.

**+ \_base.scss:** Lưu những khởi tạo css ban đầu như \*,body,html,…

**- Components/:** Thành phần nào có thể tái sử dụng được thì sài components.

**+ \_button.scss:** Lưu các nút button.

**- Layout/: chứ các phần tạo nên layout, header, footer, navigation, section, …**

**+ \_header.scss:** header của trang bỏ vô.

**- Pages/: chứa các phần css vào tưng trang home,…**

**- Themes/ :** Ưng dụng có nhiều themes khác nhau ( admin , v.v.. ) từ bên ngoài.

**- Vendors/ :** Nơi chứa tất cả các file không phụ thuộc nếu ứng dụng sử dụng thư viện từ bên ngoài ( fontAwesome, Boostrap ) và các nội dung tương tự.

**- styles.scss :**

**+** import ‘( ./abstracts/variable )’;

**+** import ‘( ./base/\_animation )’;

**+** import ‘( ./base/utilities )’;

Import liên kết…

**CHƯƠNG 2 : GIRD SYSTEM**

Sử dụng biến trong code width: calc((100% - #{$gutter-horizontal})/2);

Các cách truy xuất trong css:

1. [class^=”col-”]{

Tìm tất cả class có col- để thực hiện css

}

1. [class$=”col-”]{

Tìm tất cả class kết thúc col-

}

1. &:not( last-child ){

Tất cả đều có css trừ phần tử cuối

}

1.CSS GRADIENT BORDER: làm border nhiều màu lồng nhau

Các thuộc tính:

- border : 4px solid transparent.

- border-image : linear – gradient (45deg, <mã màu> ,<mã màu>, …);

- border-image-slice(lát mỏng):1;

2. ÁP MÀU GRADIENT LÊN CHỮ ( TEXT ).

Các thuộc tinh:

* Background: linear-gradient(<mã màu>,…);
* Webkit-background-clip: text;
* Color: transparent;

3. background: linear-gradient(0deg , <mã màu> 0%(độ tương ứng),…)

Màu lồng nhau;

* Background-size:40px : chỉnh lớn nhỏ khối gradient;

background: linear-gradient(0deg , <mã màu 1> 0%(độ tương ứng 1),<mã màu 1> 20%);

Màu nào ra màu đấy không bị pha trộn;

4.Tạo border bằng gradient animation: thích hợp chạy hiệu ứng cho border.

Chia nhỏ chỉnh nhỏ to:

* Background-size:40px;

Animation : 0%{ background-position: 0 }

* 100%{ background-position: 40px }

5. Tạo border cho text bằng stroke:

- Color: red;

- Webkit-text-fill-color: transparent;

- Webkit-text-stroke-width:2px;

6: scss: in danh sách các class theo thứ tự tăng dần

$noOfSlider:5; Đặt biến giới hạn.

-Vòng lặp: @for $i from 1 through $noOfSlider{

&-#{$i}{ //nối tiếp vào tên class

Background-image:url(#{$i}.jpg); //lặp ra 1.jpg đến 5.jpg

}

}

7: Convert px sang rem.

Px element = rem element \* root font size của browser.

\*,\*::before,\*::affter{

Margin:0;

Padding:0;

Box-sizing:inherit;

}

Body{

Box-sizing:border-box;

}

8: Thuộc tính reset về nguyên thủy radio

Appearance: none

9: Thuộc tính không cho người dùng bôi đen

user-select: none

**HỌC CSS**

**Bai 1: Color in Css.**

1. RGB Hex (#00FF00)

2. rgb( red , green, blue)

3. rgba( red , green, blue, alpha): alpha là độ trong suốt tự 0 đến 1.

**Bai 2: Styling text**

Text – index: dùng chữ đầu tiên lùi vô một chút.

Word – scpacing: khoảng cách giữa các từ.

Text – shadow: x, y màu.

**Bai 3:Background color:**

Background-color: transparent. /màu trong suốt

**Bai 4:Background image:**

Background-image: url(‘./hinh.png’);

Background-size: kích thước. ( content: hiển thị 100% trong khung hình, cover: làm cho ảnh bao tròn đến hình)

Background-position: x y;

Bacground-repeat: repreat. // lặp lại hình.

Background-attachment:fixed;

**Bai 5: Styling links:**

A:link //Trạng thái chưa được click.

A:visited //Trạng thái được click rồi.

A:hover //Trỏ chuột lên.

A:active //Trỏ chuột vào click chưa thả chuột ra.

**Bai 6: inline vs block:**

Inline : có chiều rộng bằng chiều rộng của content , nó có thể nằm trên một hang với inline.

* Css padding thì có hiệu lực,width , height, margin thì trên dưới không có hiệu lực.

Block: chiều rộng bằng chiều rộng một hàng.

**Bai 7: Flexbox:**

Sinh ra làm layout thuận tiện trong css.

Display: flex; //đặt cho cha.

Flex-direction:row or column// hướng của flex là gì , row con sắp xếp theo chiều dọc, column sắp xếp theo hàng ngang.

Justify-content: hướng sắp xếp con theo cha.

Align-item: stretch; dãn ra bằng chiều cao của cha// align-item chiều vuông góc với cha.

**Con:**

Flex-basis:<number>; chiều dài của con giá trị thay đổi không chính xác.

Flex-grow: <number>; chia khoảng dư của cha.

Order: thay đổi hiển thị.

**Bai 8:Float:**

Thành phần bao thành phần float

Wrapper::after{ content:’’ , display: block ,clear: both }

**Bai 9 Grid:**

Css gird tạo ra một lưới.

Cha: chia theo 3 x 3

* display: gird;
* display-template-columns: 100px 100px 100px; // cột
* display-template-row: 100px 100px 100px; // hàng

Con: Item1:

* grid-column-start: 1;// bắt đầu từ đường kẻ cột 1 đến
* gird-column-end: 4;// kết thúc đường kẻ thứ 4.

**Bai 10: px vs em ,rem**

* 1 em: bằng 1 lần font-size thàng cha.
* 1 rem: bằng 1 lần font-size thằng root là html.