Báo cáo CSS

**1. CSS introduction**

- là ngôn ngữ sử dụng để tạo kiểu cho trang web

- viết tắt của Cascading style sheets

**2. CSS Syntax**

- gồm bộ chọn selectors và khối khai báo

**3. CSS Selectors**

- 1 css selector chọn phần tử html muốn tạo style.

- 5 danh mục:

1. Simple selectors: chọn các phần tử dựa trên #id, name, .class
2. Combinator selectors: chọn các phần tử dựa trên mqh cụ thể giữa chúng
3. Pseudo-class selectors: chọn các phần tử dựa trên 1 trạng thái nhất định
4. Pseudo-element selectors: tạo và tạo kiểu cho 1 phần của phần tử
5. Attribute selectors: chọn phần tử dựa trên thuộc tính và giá trị thuộc tính.

**4. Có 3 cách add CSS vào tài liệu html**

* external: là 1 file với đuôi .css, import lại vào file html thông qua thẻ link trong phần head.
* internal: viết tại file html, nằm trong cặp thẻ style đặt ở phần head
* inline: viết trực tiếp vào thẻ html qua thuộc tính style
* Độ ưu tiên: inline -> internal, external (gọi sau sẽ đc dùng) ->default

**5. Comments**

* Sử dụng để giải thích mã, comment đoạn mã k sử dụng
* Đặt trong <style>, bắt đầu /\* kết thúc \*/

**6. Colors**

Sử dụng: tên màu, RGB, HEX, HSL, RGBA, HSLA

* RGB (red, green, blue), giá trị 0->255
* RGBA (red, green, blue, alpha)

alpha: độ trong suốt 0->1

* HEX: màu thập lục phân #rrggbb, #rgb, giá trị 00->ff (đen->trắng)
* HSL (hue, saturation, lightness)

hue: màu sắc 0->360o (red->blue)

saturation: bão hòa 0->100% (gray->full)

lightness: độ sáng 0->100% (black->white)

* HSLA (hue, saturation, lightness, alpha)

**7. Backgrounds**

* background-color
* background-image: url(“ ”);

Default: ảnh đc lặp lại để bao phủ toàn bộ phần tử

* background-repeat: lặp lại ảnh

+ repeat-x: lặp lại theo chiều ngang

+ repeat-y: lặp lại theo chiều dọc

+ no-repeat: k lặp lại

* background-position: chỉ định vị trí ảnh
* background-attachment: xđ hình nền cuộn hay cố định vs trang

+ fixed: cố định

+ scroll: cuộn

* background-shorthand: kết hợp của background-color, image, repeat, position.

background: color image repeat attachment position

**8. Border**

* border-style: kiểu đường viền: dotted, dashed, solid, double, …
* border-width: độ rộng đg viền, tính bằng px, pt, cm, em, …
* border-color
* shorthand border: border: width style color
* border-radius: bo góc

**9. Margins**

* Khoảng cách ngoài đường viền
* Giá trị:

+ auto

+ length: chỉ định bằng px, pt, cm, …

+ %: % chiều rộng phần tử chứa

+ inherit: đc kế thừa từ phần tử mẹ

* Cho phép giá trị âm

**10. Paddings**

* Tạo khoảng cách trong đường viền

padding: top right bottom left

* Không cho phép giá trị âm
* Padding và element width: khi padding thì element width sẽ đc cộng thêm padding

Giải quyết: sd box-sizing: border-box; để giữ đc chiều rộng thực của element

**11. height, width**

* Đặt chiều cao và ngang cho element là vùng content, k bao gồm padding, border, margin.
* Giá trị: auto (default), length, %, initial (set to default), inherit

max-width: chiều rộng tối đa.

**12. Box model**

* Là hộp bao bọc quanh mọi phần tử html, sử dụng khi nói về thiết kế và bố cục.
* Gồm:

+ content: nội dung

+ padding: khu vực xung quanh nội dung

+ border: đường viền bao quanh padding và content

+ margin: khu vực ngoài đg viền

**13. Outline**

* Là 1 đường đc vẽ bên ngoài border, có thể chồng lên nd khác, k thuộc kích thước của phần tử.
* Làm phần tử nổi bật
* Thuộc tính:

+ outline-style

+ outline-color

+ outline-width

+ outline-offset: thêm khoảng cách giữa outline và border

+ outline: width style(required) color

**14. Text**

* text-color: màu mặc định đc thiết lập trong body
* text-align: căn lề ngang

+ left

+ right

+ center

+ justify: căn lề trái và phải đều thẳng

text-align-last: căn chỉnh dòng cuối cùng

* direction và unicode-bidi: thay đổi hướng vb của 1 phần tử

+ rtl: right to left

+ ltr

+ bidi-override

* vertical-align: căn theo chiều dọc của phần tử

+ baseline

+ text-top

+ text-bottom

+ sub

+ super

* text-decoration: thiết lập/ gỡ hiệu ứng

+ none: bỏ gạch chân

+ overline: gạch ngang trên text

+ line-through: gạch giữa text

+ underline: gạch chân

* text-transform

+ uppercase: chữ hoa

+ lowercase: chữ thường

+ capitalize: viết hoa đầu mỗi từ

* text spacing

+ text-indent: thụt đầu dòng vb

+ letter-spacing: k/c các ký tự trog 1 từ

+ line-height: k/c các dòng vb

+ work-spacing: k/c các từ trog vb

+ white-space: nowrap (k xuống dòng)

* text-shadow: thêm bóng

text-shadow: x y blue color;

**15. Fronts**

* Chủng font:

+ serif: nét nhỏ ở các cạnh của mỗi chữ cái.

+ sans-serif: đg nét rõ ràng, k có nét nhỏ

+ monospace: các chữ cái có chiều rộng cố định

+ cursive: bắt trước chữ viết tay

+ fantasy: font trang trí

* font-family: font, font dự phòng, chủng font
* font an toàn: arial (sans-serif), verdana, Helvetica, Tahoma, …
* font-style: normal/ italic (nghiêng)/ oblique
* font-weight: độ dày
* font-variant: small-caps; chữ hoa nhỏ
* font-size: kích thước chữ

default: 16px =1em

* font: style variant weight size/line-height (require) family (required)
* Sd Google font, thêm liên kết vào <head>

**16. Icons**

* Sd Font awesome, thêm lk vào <head>:

<script src="https://kit.fontawesome.com/yourcode.js" crossorigin="anonymous"></script>

* Add Bootstrap icons vào <head>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

* Add Google icons vào <head>

<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">

**17. Links**

* state:

a:link –normal

a:visited –liên kết đã truy cập

a:hover –khi di chuột vào

a:active –khi đc click

**18. Lists**

* list-style-type: loại điểm đánh dấu mục danh sách

+ circle, square, upper-case, upper-roman, lower-case

+ none (margin:0, padding:0 trong ul/ol) để remote default

* list-style-image:url(‘’); chỉ định hình ảnh làm điểm đánh dấu.
* list-style-position: vị trí các điểm đánh dấu (dấu đầu dòng)

+ outside, inside

* list-style: type position image

**19. Tables**

* table border

+ border: table, th, td

+ border-collapse: collapse; đặt đường viền thu gọn là 1

* table size: width, height
* text-align: căn chỉnh text theo chiều ngang

vertical-align: căn chỉnh text theo chiều dọc

* table padding: sd trong th, td

hover

* tr:nth-child(even): thêm màu riêng cho tất cả hàng chẵn

**20. Display**

* block: bắt đầu ở dòng mới, chiếm toàn bộ độ rộng còn lại.

div, h1->h6, p, form, header, li, …

* inline: k bắt đầu ở dòng mới và chiếm độ rộng cần thiết

span, a, img, …

* display:none / visibility:hidden; ẩn element

<script> default display:none;

- inline-block: có tính chất khối, k bắt đầu ở dòng mới, set đc width, height.

**21. Layout**

1. Using width,

max-width: sử lý vs các cửa sổ nhỏ.

margin-auto: phần tử sẽ chiếm width đc chỉ định, k gian còn lại sẽ đc chia đều cho 2 lề.

1. Position

+ static: default, k bị ảnh hưởng bởi top, right, bottom, left.

+ relative: đc định vị so vs vị trí bình thường của nó vs top, right, bottom, left

+ fixed: hiển thị cố định

+ absolute: đc định vị so với thẻ chứa chúng có position: relative

+ sticky: đc định vị dựa trên vị trí cuộn của người dùng.

* Chuyển đổi giữa relative và fixed, phải chỉ định ít nhất 1 trong các vị trí top, left, right, bottom.
* Internet Explorer: k hỗ trợ
* Safari requires: position: -webkit-sticky;

+ text trong hình ảnh

1. z-index: chỉ định thứ tự ngăn xếp của 1 phần tử, hoạt động trên position element, flex.
2. overflow: kiểm soát vấn đề vs nd quá lớn để vừa vs 1 khu vực, hoạt động vs element block có chiều cao đc chỉ định.

+ visible: default, nd bị tràn

+ hidden: phần tràn bị cắt, nd còn lại sẽ k đc hiển thị

+ scroll: phần tràn bị cắt, thêm 1 thanh cuộn để xem nd còn lại.

+ auto: tương tự như scroll, thêm thanh cuộn khi cần thiết.

+ overflow-x,y: chỉ định nd tràn theo chiều ngang/dọc

1. float, clear

* float: left, right, none, inherit (kế thừa giá trị float của phần tử gốc)
* clear: left, right, none, both (k ảnh hưởng bởi float phía trước nó)
* clearfix: khi phần tử float cao hơn phần tử chứa, sd overflow: auto;

1. align

* căn giữa block element: margin: auto, phải set width
* căn giữa text trog element: text-align: center
* căn giữa ảnh: set display: block cho <img>, margin-left: auto, margin-right: auto;
* căn element: sd position: absolute; right: 0; (căn phải)

sd float, cleadfix hack

* căn giữa theo chiều dọc: padding: 50px 0; or line-height = height

căn cả dọc, ngang: sd padding, text-align

sd position, transform

sd flex

**22. Combinators**

* Giải thích mqh giữa các selector
* 4 loại combinators:
* Space: element là con của 1 phần tử
* > con trực tiếp của element
* + element liền kề
* ~ all element phía sau

**23. Pseudo-classes**

* Xác định 1 trạng thái đặc biệt của 1 phần tử

: hover

: active: chọn liên kết hoạt động khi click vào

: first-child: chỉ định phần tử con đầu tiên

: focus: kích chuột vào element

: checked: nút radio, hộp kiểm, option

: disabled: vô hiệu hóa 1 phần tử, sd trên biểu mẫu

: root: chọn phần tử gốc vủa tài liệu

: nth-child(n): chọn phần tử là con thứ n

…

**24. Pseudo-elements**

* Use: tạo kiểu cho các phần cụ thể của phần tử

:: first-line: thêm kiểu cho dòng đầu tiên

:: first-letter: thêm 1 kiểu vào chữ cái đầu tiên

:: before: chèn nd trước nd của phần tử

:: after: chèn nd sau nd của phần tử

:: marker: chọn các điểm đánh dấu

:: selection: bôi đen từ người dùng

**25. Opacity**

**-** Chỉ định độ mờ đục, trong suốt của phần tử

- Có giá trị 0 -> 1

**26. Units**

* absolute: px, pt, cm, …
* relative:

% phụ thuộc vào thẻ chứa nó

rem: phụ thuộc vào font-size đc định nghĩa ở thẻ html(100%=16px)

em: phụ thuộc vào thẻ chứa nó gần nhất có css font-size

vw: phụ thuộc vào chiều ngang trình duyệt, 1vw=1% trình duyệt

vh: phụ thuộc vào chiều dọc trình duyệt

**27. Specificity**

* Tính độ ưu tiên sẽ đc áp dụng:
* inline styles: 1000
* IDs: 100
* Classes, attributes, pseudo-classes: 10
* Elements, pseudo-elements: 1

Nếu cùng loại thì quy tắc mới nhất đc áp dụng

**28. Box Sizing**

* Gồm padding, border trong tổng width, height của element.

box-sizing: border-box; padding, border đc bao gồm trog width, height.

* Sd trong \*{}: áp dụng all element

**29. Flex box**

* ***Container****:*
* display: flex;
* flex-direction: xác định hướng mà vùng chứa muốn xếp

row(default), column, row-reverse, column-reverse

* flex-wrap: xác định các item có bọc hay k

+ wrap: tràn các item bên trong

+ nowrap, wrap-reverse

* flex-flow: direction + wrap
* justify-content: chỉnh item khi k sd hết k gian (theo chiều ngang)

+ center

+ flex-start

+ flex-end

+ space-around: có khoảng cách trước, giữa, sau

+ space-between: có khoảng cách ở giữa

+ space-evenly: khoảng cách bằng nhau

* align-items: chỉnh items theo chiều dọc

+ center

+ flex-start

+ flex-end

+ stretch: co dãn item để lấp đầy container

+ baseline: đường cơ sở căn chỉnh các items

* align-content: căn chỉnh các dòng khi sd vs wrap

+ align-between: khoảng trắng giữa các dòng

+ space-around: có khoảng cách trước, giữa, sau

+ stretch: co dãn items

+ center

+ flex-start

+ flex-end

* ***Items***
* order: thứ tự của flex items
* flex-grow: mức độ mà 1 item phát triển so với các item còn lại.
* flex-shrink: mức độ mà 1 item sẽ co lại so với các item còn lại.
* flex-basic: độ dài ban đầu của 1 mục
* flex: grow shrink basics
* align-self: chỉnh mục đã chọn bên trong container, ghi đè align-items
* ***Respnsive***
* Sd flex-direction: row-> column
* Sd % vs item flex, flex-wrap trên container

**30. Rounded Corners**

* Bo góc các phần tử: border-radius

border-image: url(); đặt hình ảnh làm đường viền

**31. Shadow**

* Tạo bóng cho text: text-shadow: x y blue color
* Tạo bóng cho element: box-shadow: x y blue spread color inset

spread: giá trị dương làm tăng kích thước bóng, gtri âm là giảm

inset: đổi bóng từ ngoài vào trong

**32. Animations**

* Tạo hoạt ảnh cho element

@keyframes: thay đổi từ kiểu hiện tại sang kiểu mới tại 1 số thời điểm nhất định.

@keyframe abc {

0% {} …

100% {}

}

+ animation-name: tên @keyframes

+ animation-duration: thời lượng hoạt ảnh sẽ hoàn thành

+ animation-delay: thời gian trễ để bắt đầu hoạt ảnh, cho phép giá trị âm

+ animation-iteration-count: số lần hoạt ảnh sẽ chạy, infinite: lặp lại mãi mãi

+ animation-direction: chạy hoạt ảnh theo hướng tiến lùi hoặc chu kỳ thay thế

* normal: mặc định
* reverse: phát hướng ngược lại
* alternate: phát về phía trước, sau đó phát ngc lại
* alternate-reverse: phát ngược, sau đó chuyển tiếp.

+ animation-timing-function: đg cong tốc độ của hoạt ảnh

* ease: bắt đầu chậm, rồi nhanh, và kết thúc chậm
* linear: tốc độ k đổi
* ease-in: bắt đầu chậm
* ease-out: kết thúc chậm
* ease-in-out: bắt đầu, kết thúc chậm
* cubic-bezier(n,n,n,n)

+ animation-fill-mode: chỉ định kiểu cho element khi hoạt ảnh k phát (trước khi bắt đầu, sau khi kết thúc)

* none: áp dụng bất kỳ
* forwards: giữ lại giá trị đc đặt bởi khung nhìn chính cuối cùng
* backwards: nhận các giá trị đc đặt bởi khung nhìn chính đầu tiên
* both

+ animation: name duration timing-function delay iteration-count direction

**33. Transitions**

* Tạo hiệu ứng chuyển tiếp, thay đổi giá trị thuộc tính trong 1 khoảng thời gian nhất định

+ transition: thuộc tính - thời lượng …

+ transition-delay

+ transition-duration

+ transition-property: tên thuộc tính

+ transition-timing-function: giống animation

* transition + transform

**34. Transforms 2D**

* Cho phép di chuyển, xoay, chia tỉ lệ, làm nghiêng phần tử.

transform:

+ translate (X, Y): di chuyển 1 phần tử khỏi vị trí hiện tại của nó.

+ rotate (): xoay theo kim đồng hồ hoặc ngược lại

+ scaleX (): tăng or giảm chiều rộng của 1 phần tử

+ scaleY ()

+ scale (): tăng or giảm kích thước của 1 phần tử theo width, height.

+ skewX (): xiên 1 phần tử theo trục X 1 góc cho trước

+ skewY ()

+ skew (): xiên 1 phần tử dọc theo trục X, Y các góc đã cho

+ matrix (scaleX- scaleY- skewX- skewY- translateX- translateY)

transform-origin: thay đổi vị trí trên các phần tử đã chuyển đổi

**35. Transforms 3D**

+ rotateX (): xoay 1 phần tử quanh trục x

+ rotateY (): xoay 1 phần tử quanh trục y

+ rotateZ (): xoay 1 phần tử quanh trục z

**36. Image**

* border-radius
* border
* responsive: max-width: 100%, height: auto; giảm tỉ lệ, k đc tăng
* center: display: block, margin-left: auto, margin-right: auto
* opacity: 0 -> 1
* text: sd position: relative ở container, position: absolute ở item
* filters: thêm các hiêu ứng mờ, bão hòa, …
* hover overlay: sd posiion
* flip: lật ảnh
* modal
* Phản chiếu hình ảnh: -webkit-box-reflect: vị trí – kcach- gradient

+ below, above, left, right