- Cú pháp viết code css

Selector{

Thuộc-tính: giá-trị;

}

- Cách kết nối css vào html

<link href=”style.css rel=”stylesheet”/>

- Định nghĩa Selector là gì?

Selector xác định phần tử html nào sẽ áp dụng code css sẽ được viết trong {...}

Selector có một số quy tắc như chọn theo thẻ-vị trí phần tử, chọn theo class của phần tử, con theo id của phần tử

- Selector toàn cục universal

\*{

/\* code css sẽ áp dụng toàn bộ phần tử trong trang web\*/

/\* thường dùng để reset xóa margin và padding mặc định\*/

/\* có thể ảnh hưởng đến nhiều thuộc tính\*/hết hợp các

/\* độ ưu tiên thấp dễ bị ghi đè bởi các selector cụ thể hơn\*/

/\* có thể kết hợp để sử dụng cùng các selector khác\*/

}

- selector element HTML

/\* chọn tất cả\*/

\*{

}

/\* chọn theo tên thẻ html\*/

a{

}

/\* chọn theo class và id \*/

.text{

/\*tất cả phần tử có class=”text”\*/

}

#username{

/\*phần tử có id =”username”\*/

}

/\* chọn theo thuộc tính phần tử html\*/

[type]{

/\* chọn các phần tử có thuộc tính type\*/

}

[type=”email”]{

/\* chọn các phần tử có thuộc tính type có giá trị là email\*/

}

- selector class

.region{

/\* chọn tất cả các phần tử có class=”region”\*/

}

- selector theo id

#password{

/\* chọn các phần tử có id=”pasword”\*/

}

- selector con cháu, hậu duệ

Selector1 selector2 {

/\* selector 2 phải nằm trong cấu trúc của selector1\*/

/\* ví dụ\*/

}

Div a {

code

}

- selector cha con

/\* chọn phần tử a, a phải là con của p (tức là các thẻ a nằm trong thẻ p)\*/

p > a{

......

}

/\* chọn phần tử a có class=”username”, a là con của p, p là con của phần tử có class là login\*/

.login > p > a .username{

.........

}

/\* chọn phần tử cùng cha phía sau \*/

/\* sử dụng dấu ~ để kết hợp

- selector theo nhóm, theo thuộc tính

/\* sử dụng dấu “,” chọn nhiều phần tử cùng lúc\*/

h1, h2, p {

.....

}

/\* chọn theo thuộc tính\*/

[type] {

......

}

/\* thuộc tính có giá trị cụ thể\*/

[type=”text”] {

...........

}

/\* ‘~=’ thuộc tính chứa từ cụ thể\*/

[class~="btn"] {

.......

}

/\* ‘^=’thuộc tính bắt đầu bằng giá trị cụ thể\*/

[class^="col-"] {

........

}

/\* ‘$=’ thuộc tính kết thúc bằng giá trị cụ thể\*/

[href$=".pdf"] {

..........

}

/\* ‘\*=’ thuộc tính chứa giá trị cụ thể\*/

[class\*="col"] {

..................

}

- selector hover chuột

element:hover {

/\* Styles khi di chuột qua \*/

}

- selector before, after

element::before {

content: "";

/\* Styles cho phần tử giả trước \*/

}

element::after {

content: "";

/\* Styles cho phần tử giả sau \*/

}

- selector kết hợp, khoanh vùng

parent child {

/\* Styles cho phần tử con \*/

}

parent > child {

/\* Styles cho phần tử con trực tiếp \*/

}

- selector định vị trí phần tử con (:first-child,…)

element:first-child {

/\* Styles cho phần tử con đầu tiên \*/

}

element:last-child {

/\* Styles cho phần tử con cuối cùng \*/

}

element:nth-child(n) {

/\* Styles cho phần tử con thứ n \*/

}

- selector loại trừ(:not)

element:not(.class) {

/\* Styles cho phần tử không có class cụ thể \*/

}

- selector phần tử anh chị em liền kề (+)

element + sibling {

/\* Styles cho phần tử liền kề ngay sau \*/

}

- selector phần tử đằng sau cùng cấp (~)

Selector1 ~selector2 các phần tử thứ 2 nằm sau phần tử 1 cùng cha thì phần tử thứ 2 được chọn

\*/

Ví dụ:

Strong ~ span{

.......

}

- Form selector :focus

input:focus {

/\* Styles khi phần tử form được focus \*/

}

- Form selector :checked

input:checked {

/\* Styles cho checkbox hoặc radio button được chọn \*/

}

- Thứ tự ưu tiên và cách thăng hạn style css

Thứ tự ưu tiên từ cao đến thấp:

1. !important
2. Inline styles
3. ID selectors
4. Class selectors, attribute selectors, pseudo-classes
5. Element selectors, pseudo-elements

Cách thăng hạng:

* Sử dụng !important (hạn chế dùng)
* Tăng độ cụ thể của selector
* Đặt rule sau cùng trong file CSS