

HTML 출현 이후 여러가지 용량 ↑ ex) 문서의 글꼴, 크기, 색상 변경, 가운데 정렬, 배경 등 다정 다단 → CSS

기본 HTML tag → 필요는 4타법

CSS → 디자인은 4타법

<style>

<a> 태그의 내용은 변경사항으로 표현하고 싶어!

HTML → .. → <a> 태그가 있는 곳에 다 인식이 되어야 함

CSS → <style>

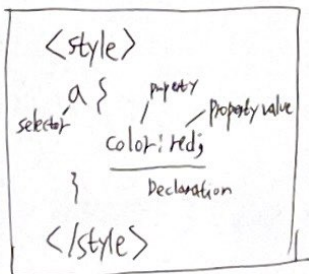
```
a {  
  color: red;  
}
```

→ 한번만 작성하면 됨 (중복 제거) 유지보수, 가독성 ↑

• 가독성 디자인은 분리할 수 있다 by <style> . 효율성 ↑

웹페이지 안에 CSS를 삽입하는 2가지 방법

① <style> 태그 사용



- 선택자(selector): CSS 효과를 적용시킬 대상 ex) a {}

- 효과(Declaration): 선택자에 적용되어 나타내게 될 CSS 효과

CSS 문법을 통해 표현한 것

ex) color: red;

text-decoration: none;

효과는 여러개를 선언하고 지정해 줄 수 있다.

여러개의 효과는 세미콜론(;)으로

각 효과는 구분해 준다.

② style 속성을 쓴다

INTRO

• style 속성은 직접 사용하는 것, 바로 style 속성 위치하고 있는 태그에게 효과는 줄 것이기 때문에 선택자는 사실상 필요가 없다

style=" " : HTML 속성
CSS의 효과

CSS 속성 검색 property

ex) 글씨 크기는 조절하고 싶어! → CSS text size property

↳ font-size;

가운데 정렬을 하고 싶어! → CSS text center property

↳ text-align;

* 모든 property는 라우어 다지 않고 검색기능을 적극적으로 활용하자

CSS 선택자 selector

| 선택자 | 예제 | 예제설명 |
|---------|------------|--|
| .class | .intro | 원래 클래스 class="intro"인 태그들은 선택한다 |
| #id | #firstname | id="firstname"인 태그 1개는 선택한다
id는 유일무이하다 (중복 x) |
| element | p | <p> 태그들은 모두 선택한다
element = 태그 |

우선순위 : 구체적인 것 > 포괄적인 것, 선택자가 같은 시, 가장 마지막에 등장하는 선택자가 우선순위가 높다

* ID 선택자 > class 선택자 > 태그 선택자

선택자들은 조합하여 원하는 효과는 정교하게 어떠한 태그에 타겟팅할 수 있고

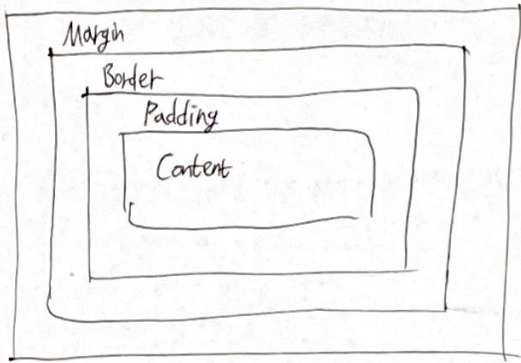
이런 통해 코드의 양은 줄이고, 어떤 한 부분만 수정하면 나머지가 동시에 바뀌는 효율성은 높일 수 있다.

박스 모델 box model

HTML 요소 등은 화면 전체를 쓰는 block level element와 자기 자신의 부피, 콘텐츠 크기를 갖는 inline element가 있다

block level element와 inline element는 display라는 속성의 기본값은 `block`과 `inline` 뿐, 그 기본값은 CSS를 통해 바꿀 수 있다.

필요에 따라 태그를 안보이게 하고 싶다 → `display: none;`



검사: F12 (Ctrl+Shift+I)

검사가은 쪽 활용하자

그리드 grid

디자인을 하기 위해 어떤 의미도 존재하지 않는 태그를 사용한다.

`<div>` , ``
block level element inline element

2개 이상의 태그를 원하는 대로 배치하려면 그 태그들은 감싸는 부모태그가 필요함!

`display: grid;`

ex) 2개 나란히 배치 → `grid-template-columns: 150px;`

미디어 쿼리 media query

반응형 디자인 or 반응형 웹 (responsive web). 화면 크기에 따라 웹사이트의 각 요소들이 반응해서 동작하게 된다.

미디어 쿼리 (media query): 반응형 디자인을 웹(CSS)을 통해 구현하는 방법

ex) `if (screen=width > 800px) → div { display: none; }`



```
@media (min-width: 800px) {  
  div {  
    display: none;  
  }  
}
```

CSS 코드 재사용

모든 웹사이트의 CSS코드는 다 바뀌기 힘들 → 중복제거 by `<link>`, `.css` `<link rel="stylesheet" href="style.css">`

`<link>`: 외부 리소스 연결 요소

변동 파일 다운 아 웹사이트 안에 CSS 코드 내장? → 네트워크적 측면에서 후자가 더 효율적. but Caching 때문에 전자가 실용적으로 더 효율

Caching: 한번 저장하면 파일의 내용이 바뀌기 전까지는 우리의 컴퓨터에 저장해놔다가 파일은 요청하면 저장된 결과를 가져와 속도는 높일 수 있고 (네트워크 X) 네트워크 트래픽을 낮춰 유지보수가 편리하다.

∴ 중복제거 → 재사용성 ↑ 가독성 ↑ 유지보수 편리