

# HTML TAG

: 각 씬임새에 대해 정확한 의미를 가지고 있다 = Semantic 하다

=> 우리가 사용하지 않는 태그를 이용해 수행되는 작업들이 의미를 미칠 수 있어 정확한 태그 사용이 요구된다

- 링크 <a> </a>, <a href="주소"> </a>
- 이미지 <img>
- 목록 <ul> </ul>, <li> </li>  
(unordered list)
- 제목 <h1>, <h2> ...  
+ <div> : block element

## Layout Tags

- HTML 내부에서 div 태그를 block element
  - header + pcom은 큰 div에 클래스로 지정하고, 모바일에서는 header...를 사용한다
  - section
  - nav
  - aside
  - footer
- 본문영역 => div + container 사용도 가능

class : 비슷한 스타일을 여러개에 같이 표현하기 위해서

CSS 클래스를 어떤 하나를 만들어두고 그 이름을 부여함으로써 여러개

같은 클래스이름을 가진 것끼리 같은 스타일을 갖게 됨

적용할 때 : id (권장)

여러군데 적용할 때 : class

span { color : red; }

선택자 selector      property      value

• style로 HTML에 적용하는 방법

- inline : HTML 태그안에 넣는 방법 (최소한 적용) => 유지보수, 연리가 쉽다
- internal : head 안에 바인 스타일을 넣는 방법  
+ 별도의 CSS파일을 관리하지 않아도 된다 => 브라우저 CSS파일을 부하하지만 별도의 요청도 보내지 않아도 된다
- external : 별도 파일을 불러와서 관리하는 방법

• CSS 상속 : 상위에서 설정된 스타일을 하위에서 쓴다

=> 자식들 측면에서 이득.

+ 배치와 같은 속성 (padding, border etc...) 등은 상속을 받지 않는다

• CSS selector

: HTML의 태그, id, class, html 등 다양한 용에 쉽게 찾아주는 방법

가장 기본 선택자

- tag

- id : javascript에서 하나만 찾아내지만 CSS는 해당 id를 가진 모든 것이 적용된다

- class

+ 클래스를 중괄호()로 선택

+ 대괄호를 공백/>로 선택

+ nth에 자식요소로 선택

nth-child (숫자)

예) #div > p:nth-child(2) { color: red; }

• CSS는 여러가지 스타일 정보를 기반으로

최종적으로 '경쟁'에 의해서 적용된 스타일이 반영된다

↳ 선택자의 우선순위를 규정 (cascading)

\* 우선순위 (참고: CSS specificity)

inline > internal > external

• 동일하면 다음의 것 or 관계적으로 표현된 것을 우선

• id 값이 우선됨 (id > class > element)

• 폰트와 배경색 설정

색 { red, green ...  
rgb(255, 0, 0)  
rgb(255, 0, 0, 0.5) # 알파라  
#ff0000; #162f

• background-color

• font-family  
↳ 폰트 설정

font-size

1em (상대크기, 16px)

16px