HTML TAG

: 각 쓰임새에 대해 정확한 의미를 가지고 있다 = Semantic 하다

=) 오류가 나지 않더라도 태그를 이용해 수행되는 다 작업들에 영향을 미칠수 있어 정확한 태그사용이 요구된다

- 링크 <a> ,
- 이미지 <img>
- 목록 <ul> , <li>
  (unordered list)
- 제목 <h1> ,<h2>. . .

  + <div> : block element

layout Tags

HTML 내부에서는 div 와같은 block element

- header
- section
- nav
- aside
- footer

+ PC에서는 주로 div 에 클래스로 지정하고, 모바일에서는 header . . 를 사용한다

section ~ aside: 본문영역 => div. 나 container 등으로도 표현

class : 비슷한 스타일을 여기저기에 같이 표현하기 위해서

CSS 클래스를 어떤 하나를 만들어두고 그 이름을 문어항으로 해서 여기저기

같은 클래스이름을 가진 것끼리 같은 스타일을 갖게됨

고유할때 : id (권장)

여러군데 적용할때 : class

span { color : red; }

(span: 선택자 selector, color: property, red: value)

- style로 HTML에 적용하는 방법
  - inline : HTML 태그안에 넣는 방법 (최우선 적용) =) 유지보수, 관리가 힘들다
  - internal : head 안에 바로 style을 넣는 방법
    + 별도의 CSS파일을 관리하지 않아도된다 => 브라우저가 CSS 파일을 부르기위한 별도의 요청을 보내지 않아도된다
  - external : 별도 파일로 분리해서 관리하는 방법

- CSS 상속 : 상위에서 설정된 스타일을 하위에서도 쓴다
  => 재사용 측면에서 이득.
  + 배치와 같은 속성(padding, border etc..) 등은 상속을 받지 않는다

- CSS selector
  : HTML의 요소를 tag, id, class. html 태그속성등을 통해 쉽게 찾아주는 방법
  3가지 기본 선택자
  - tag
  - id : javascript에서는 하나만 찾아내지만 CSS는 해당 id를 가진 요소에 적용된다
  - class
    + 그룹을 콤마(,)로 선택
    + 하위 요소를 공백/ > 로 선택
    + n번째 자식요소를 선택
      nth-child (숫자)
      ex) #json > p:nth-child(2){color: red}

- CSS는 여러가지 스타일 정보를 기반으로 최종적으로 '경쟁'에 의해서 적절한 스타일이 반영된다
  ↳ 선택자간 우선순위를 규정 (cascading)

  ※ 우선순위 (참고: CSS specificity)
  inline > internal > external
  - 동일하면 나중의 것 or 구체적으로 표현된 것을 우선
  - id 값이 우선됨 (id > class > element)

- 폰트와 백그라운드 속성

  색
  - red , green . . .
  - rgb(255,0,0)
  - rgba(255,0,0,0.5) #알파값
  - #ff0000; #16진수

  font-size
  - 1 em (상대크기 , 배수)
  - 16px

  - background-color
  - font-family
    ↳ 폰트 설정

layout 작업, Rendering 작업

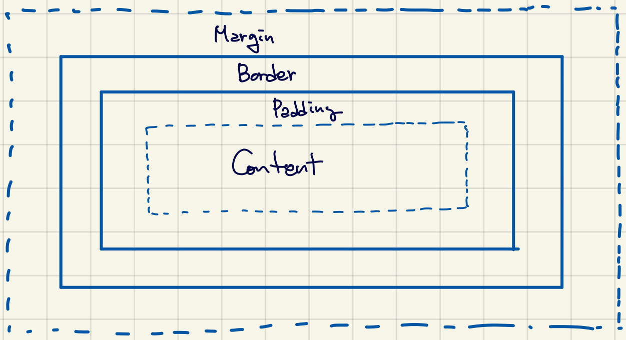
엘리먼트가 배치되는 방식

- 엘리먼트는 위에서 아래로 순서대로 블록을 이루면서 배치되는 것이 기본
  - => 표현의 제약 존재 → CSS에서는 추가적인 속성을 제공
  - -display (block, inline, inline-block)
    - inline: 내용영역을 가진 기본 속성 / <span><a> 등, 블록을 가지고 있지 않아서 좌에서 우로 흐름
  - - position (static, absolute, relative, fixed)
  - - float (left, right) => 기본 엘리먼트 배치에서 벗어나서 하늘에 떠있는 느낌으로 배치된다

- display : block : 위에서 아래로 쌓임
- display : inline : 좌에서 우로 흐르기를, 꽉차면 오른쪽 아래로 빈자리를 차지하면서 흐르게 된다
- position 속성 (상대적, 절대적인 위치지정)
  - → 기본 : 순서대로
  - → absolute : 상위엘리먼트 "position" 기준 기준점따라 위치 (top / left / right / bottom) // position 없으면 body 기준
  - → relative : 원래 자신이 위치할 곳을 기준으로 상대적인 값을 부여
  - → fixed : viewport 기준, 절대위치
- 간격 : margin
- float : 기본 배치에서 벗어나서 떠있기
  - ⇒ block 엘리먼트가 중첩되어서 배치된다 → 실제로 layout 구성할 때 좌우 영역으로 많이 쓰이게 된다

\+ margin : 엘리먼트간 간격
  border : 테두리 정보로 두께를 조절
  padding : 엘리먼트 안에 콘텐츠와 그 엘리먼트가 가진 원래 크기간 간격

<The CSS Box Model>



엘리먼트의 크기는 부모크기가 기준

box-sizing

padding ↑ → div ↑

=> box-sizing을 border-box로 바꿔줌 (기본 = box-sizing : content-box)

\+ float를 활용해 다단 layout 구현
  => css-grid 나 flex 속성 사용도 가능

\+ 자식이 float인 경우에는 자기의 자식으로 생각하지 않는다
  => overflow 속성으로 float를 인식하도록

\+ 스타일 정보에는 브라우저가 CSS를 파악한 다음에 어떻게 파악 했는지에 대한 정보를 나타냄