

패스트캠퍼스 HTML & CSS 캠프 4번째 수업 - CSS

Practice (30min) - HTML

1. Visual Studio Code (VS Code)를 실행합니다.
 2. `ctrl + n` 혹은 `cmd + n` 키를 눌러 새 창을 엽니다.
 3. `ctrl + s` 혹은 `cmd + s` 키를 눌러 저장을 합니다.
 1. 유효한 위치에 저장하여야 합니다. (본인이 찾을 수 있는 곳)
 2. 저장할 때 파일명은 `index.html` 입니다.
 4. 기본 HTML 템플릿[아래 참고]에 맞추어 HTML 문서를 만듭니다.
 5. 조별 과제로 내준 사이트의 MAIN PAGE의 HTML을 작성해봅니다.
 1. Heading (`h1` ~ `h6`) 레벨에 유의해가면서 작성합니다.
 2. 콘텐츠의 구성에 대한 충분한 이해가 필요합니다.
- A조 : APPLE Site 개발
 - B조 : Google Play Store 개발
 - C조 : Youtube 개발
 - D조 : APPLE Site 개발
 - E조 : Google Play Store 개발
 - F조 : Youtube 개발

HTML 템플릿

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <title>HELLO WORLD</title>
  </head>
  <body>

  </body>
</html>
```

제출하기

- Google 드라이브 - 모든 파일을 한 곳에서에 제출해주시면 됩니다.
- 현재는 SLACK 채널의 `general` 채널에 링크 공유 예정입니다.

Visual formatting model (2h)

유저 에이전트가 비주얼 미디어를 위해 문서 트리를 어떻게 해석하는가? visual formatting model에서 모든 요소는 0개 혹은 그 이상의 박스를 박스 모델에 따라 생성합니다. 박스의 레이아웃은 다음과 같이 구성 되어있습니다.

- box dimension과 type
- 포지셔닝 스킴 (normal flow, float, absolute positioning)
- 문서 트리 내 요소간의 관계
- 외부 정보 (viewport, 이미지의 고유한 dimension 등)

아래 내용은 codepen에서 수업합니다.

경로 : <https://codepen.io/ChoEun/pen/VzGmaL>

박스 생성 제어하기

inline과 block으로 구분됨.

inline level elements

포맷상 텍스트로써 취급되는 것

Content Model에서 Phrasing Content에 속하는 것들의 display model이 보통 inline
INLINE LEVEL은 콘텐츠 만큼만 영역을 가짐

box의 크기 계산법

너비 : $\text{content width} + \text{padding-left} + \text{padding-right} + \text{border-left} + \text{border-right}$

높이 : line-height

block level elements

포맷상 블록으로써 취급되는 것, 여기서 블럭이란 한 영역 전체를 차지하는 것

BLOCK Level은 영역 전체를 차지함

BLOCK Level은 box dimension의 영향을 받을 때,

height가 콘텐츠의 높이 + padding top + padding-bottom

box의 크기 계산법

가로너비 : $\text{width} + \text{padding-left} + \text{padding-right} + \text{border-left} + \text{border-right}$

세로높이 : $\text{height} + \text{padding-top} + \text{padding-bottom} + \text{border-top} + \text{border-bottom}$

margin은 별도

inline-block

box 자체는 inline처럼 배치

box의 크기 계산법은 block을 따라감
버튼 등을 만들 때 inline block을 활용

HTML:

```
<button>전송하기</button>  
<button>취소</button>
```

CSS:

```
button {  
  display:inline-block;  
  padding:.2em 1em;  
  margin:1em;  
  border:1px solid #2ac1bc;  
  background:#2ac1bc;  
  font:inherit;  
  text-align:center;  
}
```

display 속성

위 값들은 display 속성을 이용해서 제어 가능하다.

1. inline
2. block
3. inline-block (하는 짓은 inline인데 박스 모델은 block)
4. table (표)
5. ruby (루비)
6. list-item (li 요소)
7. flex (flexbox할 때 사용)
8. grid (Grid layout할 때 사용)

포지셔닝 스킴

User agent에서 요소를 어떻게 배치할 지 결정하는 요인들

- Normal flow
- Floats
- Absolute Positioning

Normal flow

- 블록 레벨 엘리먼트는 블록으로 쌓이고, 인라인 레벨 엘리먼트는 인라인으로 쌓인다.

- CSS로 요소의 Flow를 바꾸지 않은 상태

Floats

- 요소를 배치하는 방법 중 하나. normal flow의 첫번째 위치로 레이아웃을 잡게하는 것.
- float 속성을 넣게되는 경우 가능한 한 최대한의 left 혹은 right로 이동
- floating => 띄우는, 부양하는
- float 속성은 특정한 요소를 왼쪽 or 오른쪽으로 부양시키는 역할
- 속성값으로 left 혹은 right 를 가지게 됨

clearfix

- float요소의 containing block이 제대로 높이를 가질 수 있게 하는 테크닉

```
.clearfix::after {
  content: '';
  display: block;
  clear: both;
  /* 형제 요소 중에 float 요소가 있다면, 그 float의 영향을 해제시켜주는 속성 */
  /* clear 속성을 넣어줌으로써 float의 영향을 받지 않게 된다 */
}
```

Absolute Positioning

- Position 속성을 사용하게 될 경우, Containing block을 기준으로 박스를 특정 위치에 배치시키는 것이 가능.
- 이 상태를 Absolute Positioning 상태라 지칭함
- Position : 특정한 요소를 어떻게 배치할 것인가
- 지금까지 다뤘던 것 : static position

Absolute Positioning

- 특정한 위치에 해당 요소를 고정시키고 싶을 때
- 해당 요소를 containing block을 기준으로 특정 위치에 고정시킬 수 있음 (absolute)
- box offset 속성 : top, right, bottom, left
- Absolute Positioning에서 containing block은 조상 요소 중 Position이 정적 Position이 아닌 요소
- out-of-flow 상태
- 콘텐츠 중첩 현상이 발생하기도 함
- 다른 콘텐츠의 변화에 유동적으로 대응하기가 힘들다. (남발하지 마세요)
- 위 케이스에서, Floats나 Absolute Positioning을 이용해서 Flow를 변경하는 상태를 out-of-flow (flow에서 벗어났다)라고 지칭함.

- 그 외의 경우는 in-flow (flow 내에 있다)라고 지칭
- **out-of-flow** 상태에서는 예기치 못할 일이 많이 발생

Practice - CSS (20min)

1. VS Code를 실행합니다.
2. `ctrl + n` 혹은 `cmd + n` 키를 눌러 새 창을 엽니다.
3. `ctrl + s` 혹은 `cmd + s` 키를 눌러 저장을 합니다.
 1. 아까 HTML을 작성하던 곳과 동일한 곳에 저장해야합니다.
 2. 저장할 때 파일명은 `style.css` 입니다.
4. HTML 파일의 `head` 요소 안에 `<link rel="stylesheet" href="style.css">`를 기입합니다.
5. CSS 문법에 맞추어 아래와 같은 일을 수행합니다.
 1. 폰트 사이즈 (`font-size`)를 12px로 변경합니다.
 2. 폰트 패밀리(`font-family`)를 `sans-serif`로 변경합니다.
 3. 행간 (`line-height`)를 18px로 변경합니다. 이 때 단위는 `em`을 사용해주세요.
 4. 본인이 작성하는 사이트에 맞추어 레이아웃을 짭니다.
 1. 아직은 `float`가 많이 사용됩니다.
 2. `box model`에 유의하여 작성합니다.
 3. `containing block`에 유의하여 작성합니다.
6. 필요하다면 HTML을 변경해도 무관합니다.

