

| KWEB 준회원 스터디

Cascading Style Sheets, 2nd

@KWEB

목차

1. 다양한 CSS 선택자
2. “display” 속성
3. position
4. float
5. Deep dive to CSS
6. 3주차 과제

다양한 CSS 선택자

■ 지난 시간에 다 못 배운 CSS 선택자들을 소개합니다.

복합 선택자 (Combinator)

- 하위 선택자

부모 요소에 포함된 모든 하위 요소를 선택합니다.

Ex) `ul li { /* 속성 선언 */ }`

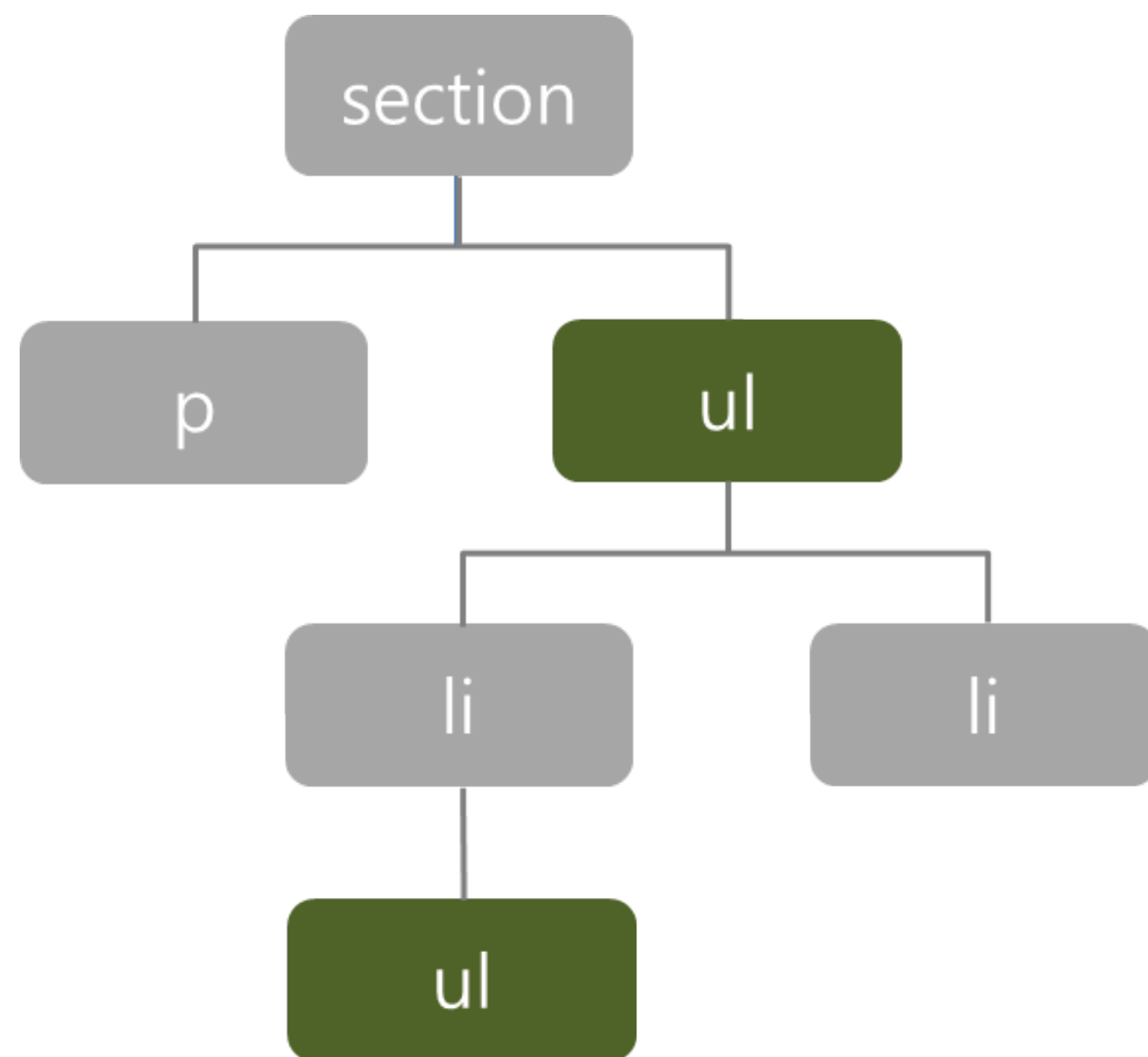
- 자식 선택자

부모의 바로 아래 자식 요소만 선택합니다.

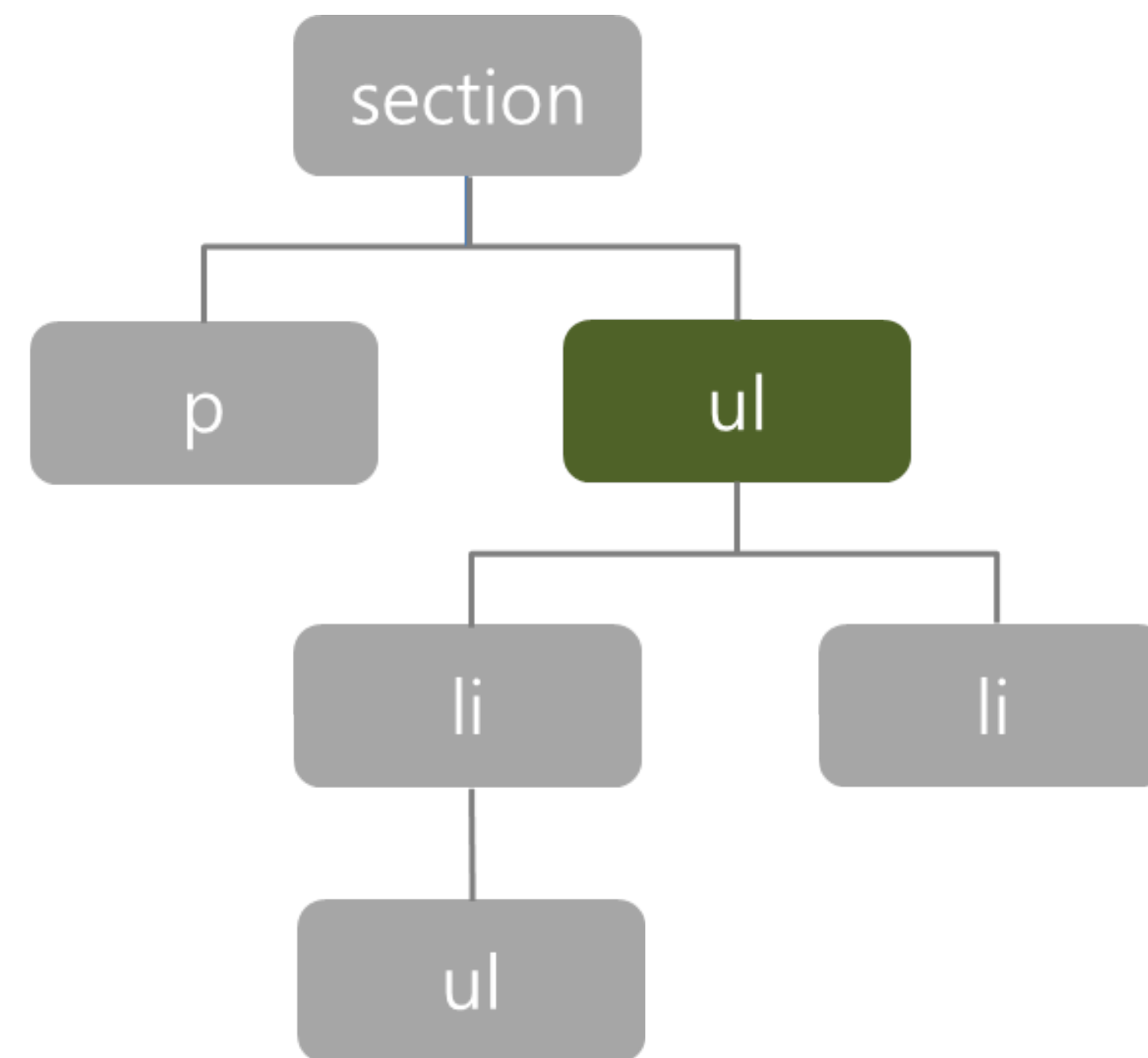
Ex) `div > span { /* 속성 선언 */ }`

```
/* CSS */  
/* 하위 선택자 */  
section ul { border: 1px dotted black; }  
  
/* 자식 선택자 */  
section>ul { border: 1px dotted black; }
```

<하위 선택자>



<자식 선택자>



가상 클래스 선택자 (Combinator)

CSS 가상 클래스는 선택될 요소의 특별한 상태를 지정할 수 있는 키워드입니다.

- **:active**

selector 요소에 마우스 클릭 또는 키보드 엔터가 눌린 동안 해당 요소를 선택합니다.

Ex) `button:active { /* 속성 선언 */ }`

- **:hover**

selector 요소에 마우스가 올라가 있는 동안 해당 요소를 선택합니다.

Ex) `a:hover { /* 속성 선언 */ }`

가상 클래스 선택자 (Combinator)

- **:first-child, :last-child**

첫 번째, 마지막에 등장하는 요소가 selector라면 선택합니다.

Ex) `p:first-child { /* 속성 선언 */ }, li:last-child { /* 속성 선언 */ }`

- **:not(selector)**

selector에 해당하지 않는 요소를 선택합니다.

Ex) `:not(p) { /* 속성 선언 */ }`

그 외의 수많은 CSS 선택자들은 다음 글들을 참고합니다.

- **w3schools**

https://www.w3schools.com/cssref/css_selectors.asp

- **MDN**

https://developer.mozilla.org/ko/docs/Web/CSS/CSS_%EC%84%A0%ED%83%9D%EC%9E%90

- **CSS: 선택자(Selector) 이해**

<http://www.nextree.co.kr/p8468/>

“display” 속성

■ CSS에서 레이아웃을 제어하기 위한 가장 중요한 속성을 배워봅시다.

“display” 속성값

모든 요소에는 요소의 유형에 따라 기본 표시값이 있습니다

- **block**

새 줄에서 시작해 좌우로 최대한 늘어납니다.

여러 블록들을 배치하면 수직으로 배열됩니다.

- **inline**

텍스트 문장이 쓰이듯 단락의 흐름을 방해하지 않고 텍스트를 감쌀 수 있습니다.

여러 블록들을 배치하면 수평으로 배열됩니다.

너비와 높이를 지정할 수 없습니다.

“display” 속성값

- **inline-block**

inline처럼 작동하지만 너비와 높이 지정이 가능합니다.

- **none**

요소를 마치 존재하지 않는 것처럼 렌더링합니다.

공간을 차지하진 않지만 실제로 요소는 존재합니다.

예제1. Block과 Inline

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8" />
5      <style>
21  </style>
22 </head>
23 <body>
24     <!-- Block -->
25     <section id="block-elements">
26         <div>Block1</div>
27         <div>Block2</div>
28     </section>
29
30     <!-- Inline -->
31     <section id="inline-elements">
32         <span>Inline1</span>
33         <span>Inline2</span>
34     </section>
35 </body>
36 </html>
37
```

Block1

Block2

Inline1

Inline2

<결과 화면>

position

요소를 다양한 곳에 위치 시켜 봅시다.

“position” 속성값

- **static**

기본값입니다.

- **relative**

별도의 프로퍼티를 지정하지 않으면 static과 동일하게 작동합니다.

상대 위치가 지정된 요소에 top, left 등을 지정하면 기본 위치와 다르게 위치가 조정됩니다.

“position” 속성값

- **fixed**

뷰포트(화면)에 상대적으로 위치가 지정됩니다.

즉, 페이지가 스크롤 되더라도 똑같은 곳에 위치합니다.

- **absolute**

가장 가까운 위치에 있는 부모 요소에 상대적으로 위치가 지정됩니다.

만약 위치를 기준으로 삼을 부모 요소가 없는 경우 body를 기준으로 위치를 지정합니다.

예제2. position

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8" />
5      <link rel="stylesheet" href="position.css" />
6      <style>
7          * { box-sizing: border-box; }
8
9          nav {
10             padding: 8px;
11             border: solid #D64078 3px;
12         }
13
14         section {
15             padding: 8px;
16             border: solid #FDC72F 3px;
17         }
18
19         footer {
20             padding: 8px;
21             border: solid #96C02E 3px;
22         }
23     </style>
24 </head>
25 <body>
26     <div class="container">
27         <nav>네비게이션</nav>
28         <section>"margin-left"값을 주어 네비게이션이 들어갈 공간을 줍니다.</section>
29         <section>
30             여러 콘텐츠들이 여기 들어갈 것입니다.<br />
31             여러 내용들을 입력해 봅시다.<br />
32         </section>
33     </div>
34
35     <footer>고정 헤더입니다.</footer>
36 </body>
37 </html>
```

네비게이션

"margin-left"값을 주어 네비게이션이 들어갈 공간을 줍니다.

여러 콘텐츠들이 여기 들어갈 것입니다.
여러 내용들을 입력해 봅시다.

고정 헤더입니다.

float

요소를 공중에 띄워 봅시다.

“float” 속성

요소를 왼쪽 또는 오른쪽 부터 배치 시키기 위한 속성.

`float: left;` -> 왼쪽부터 정렬

`float: right;` -> 오른쪽부터 정렬

웹 사이트의 레이아웃을 잡는데 float 속성을 많이 사용합니다. 중요한 속성이니 많이 사용해 보고 숙지하셨으면 합니다.

```
<div class="box">...</div>
<section>...</section>
```

html

```
.box {
  float: left;
  width: 200px;
  height: 100px;
  margin: 1em;
}
```

CSS

<section>

<div class="box">

마치 떠 있는 것만 같아요!

</div>

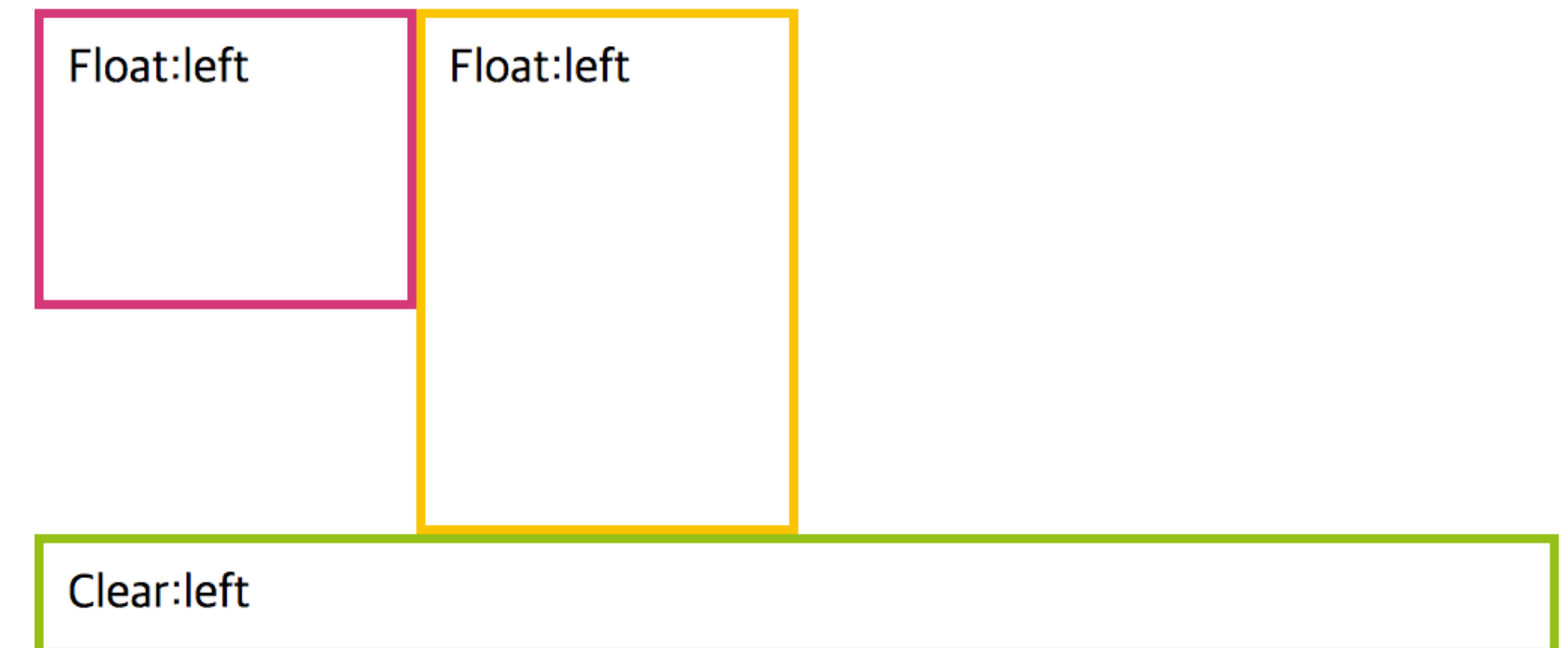
이 경우 `section` 엘리먼트는 실제로 `div` 다음에 있습니다. 하지만 `div`가 왼쪽으로 떠 있기 때문에 이런 결과가 나타난 것입니다. 즉, `section` 안의 텍스트가 `div` 주위에 떠 있고 `section`이 전체를 감싸고 있습니다. `section`을 실제로 떠 있는 엘리먼트 다음에 나타나게 하려면 어떻게 해야 할까요?

</section>

clear

요소가 선행 부동 요소의 다음일 수 있는지 또는 그 아래로 내려가야 하는지를 결정합니다.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8" />
5      <style> ...
26  </style>
27  </head>
28  <body>
29  <div id="wrapper">
30      <nav style="float: left;">Float:left</nav>
31      <section style="float: left;">Float:left</section>
32      <footer style="clear: left;">Clear:left</footer>
33  </div>
34  </body>
35  </html>
36
```



clearfix 핵

가끔 부유된 자식 요소가 부모 요소보다 높이가 커 부모 요소를 넘쳐 밖으로 빠져 나올 수 있습니다.

이럴 때 해결하는 방법이 있는데 이를 **clearfix 핵**이라 합니다.

여러 방법들이 있는데, 다음과 같이 부모 요소의 대체 가상 요소에 clear를 쓰는 방식이 있습니다.

```
1  #container::after {  
2      content: "";  
3      display: block;  
4      clear: both;  
5  }
```

float로 수 많은 레이아웃을 표현하기에는 많은 문제들이 따릅니다.
이 난해하고 복잡하고 말도 잘 안 듣는 속성을 사람들이 어떻게 다루는지 알아보시다.

- **Position: float**

<http://www.beautifulcss.com/archives/787>

- **All About Floats**

<https://css-tricks.com/all-about-floats/>

- **w3schools**

https://www.w3schools.com/css/css_float.asp

Deep dive to CSS

■ CSS is awesome



중앙 정렬 기법 완벽 가이드

CSS로 중앙 정렬하는 것은 정말 어렵습니다.

다행히도 많은 사람들이 오랜 시간 동안 연구한 끝에 다양한 기법들이 발견되었습니다.

- **Centering in CSS: A Complete Guide**

<https://css-tricks.com/centering-css-complete-guide/>

미디어 쿼리

"반응형 디자인"은 사이트가 표시되는 브라우저와 디바이스에 "반응하는" 사이트를 만드는 전략입니다.

미디어 쿼리를 사용하면 브라우저와 디바이스에 반응하여 각각 다른 CSS를 적용할 수 있습니다.

- **MDN: CSS media queries**

https://developer.mozilla.org/ko/docs/Web/Guide/CSS/Media_queries

- **w3schools**

https://www.w3schools.com/cssref/css3_pr_mediaquery.asp

flexbox

CSS3 명세에 반영된 새로운 레이아웃을 잡는 방법입니다.

최근에 나온 기술이기 때문에 지원 브라우저가 많지 않은 편입니다. (IE 10+)

flex를 사용하면 요소를 배치, 정렬, 분산하는 것이 간단해집니다.

- **A Complete Guide to Flexbox**

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

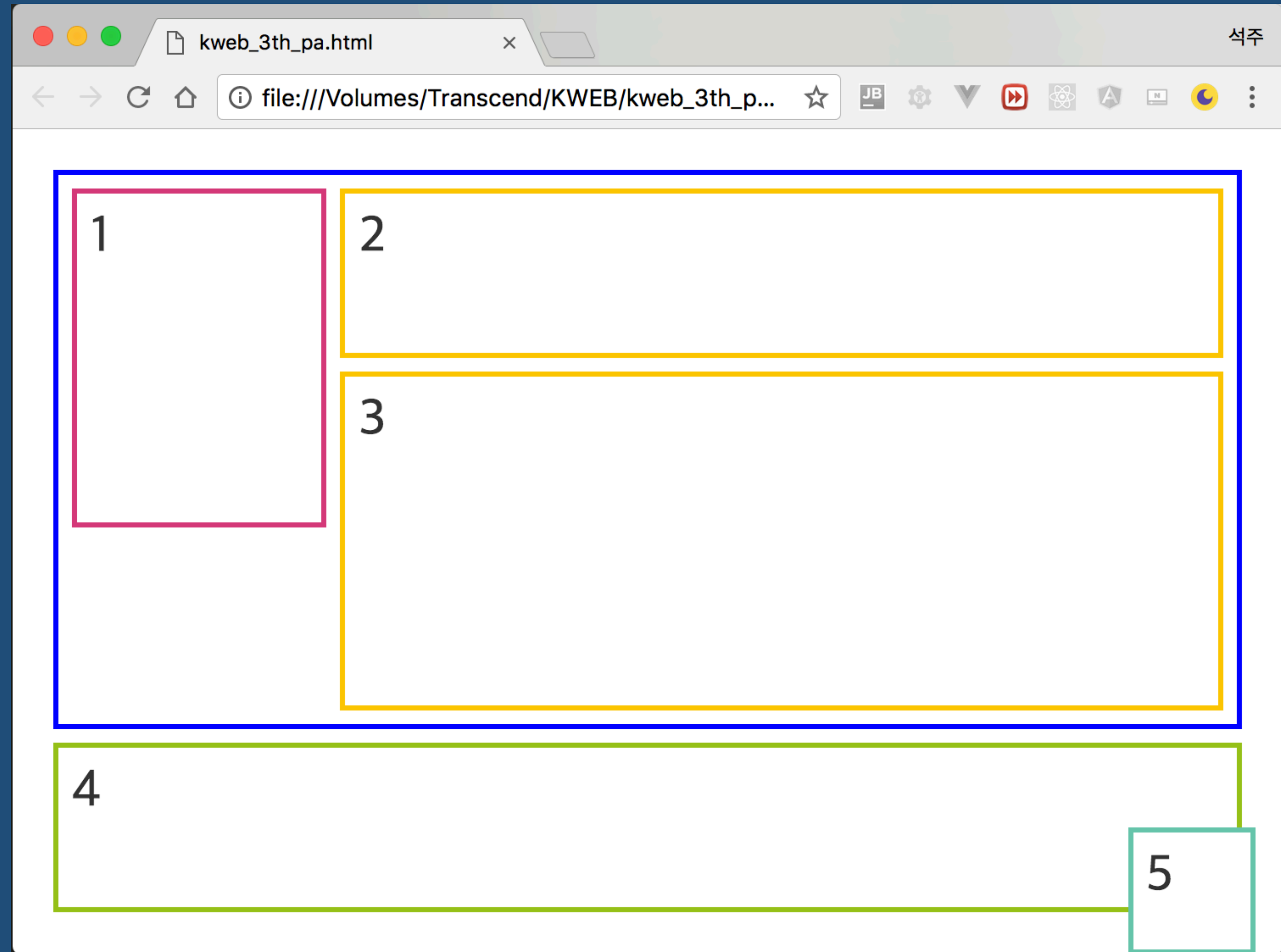
- **MDN: Advanced layouts with flexbox**

https://developer.mozilla.org/ko/docs/Web/CSS/CSS_Flexible_Box_Layout/Advanced_layouts_with_flexbox

3주차 과제

이번 주 과제입니다. 간단한 레이아웃을 만들어 보아요 :)

과제 예시



과제 설명

- index.html, beauty.css, layout.css 세 가지의 파일이 주어집니다.
- **layout.css 파일을 고쳐 예시에 보이는 것처럼 레이아웃을 만들어 주세요.**
- position, float 등 속성을 이용하시면 됩니다.
- 과제 제출할 때 꼭 파일명 바꿔 주세요.
- (Hint) float으로 1, 2, 3을 정렬하는 경우 해당 요소들의 부모인 파란색 박스는 cleafix 핵을 사용하셔야 합니다.

과제 제출

- **제출 기한** : 각 요일별 다음 스터디 전까지
- **제출 방법** : html과 css 파일을 제출합니다.
 - A. 월요일 - 16 이승열 (josephlsy777@gmail.com)
 - B. 화요일 - 15 나석주 (seokmaTD@gmail.com)
 - C. 목요일 - 14 전용석 (ysjeon741@gmail.com)
- **제출 양식** : 메일 제목과 파일 이름 -> [KWEB 준스 3주차]학번_이름
- 복붙하지마세요!
- 궁금하신 점은 꼭 스터디장께 물어보세요.

참고하면 좋은 사이트

- CSS 레이아웃을 배워봅시다

<http://ko.learnlayout.com/>

- w3schools exercise

<https://www.w3schools.com/css/exercise.asp>