

웹 프로그래밍 3주차

폰트, CSS 셀렉터, 레이아웃 이해하기

0. 구글 웹폰트

- 사용자의 PC에 설치되어 있지 않은 폰트를 웹사이트에서 표현하기 위해서는 직접 폰트를 웹사이트 안에 탑재하여야 함
 - 사용자가 직접 폰트를 제공하거나, 웹 폰트를 제공할 수 있음
- 구글에서는 폰트의 재활용, 로딩 속도를 적절하게 고려한 구글 웹 폰트 (<https://fonts.google.com>)를 제공함.
- 실습과제에도 구글 웹 폰트를 사용하여 내 폰트를 작성해볼 것

0. 구글 웹 폰트

Styles

Type here to preview text

27px

Thin 100

모든 인류 구성원의 천부의 존엄성과 동등하고 양도할 수 없는 권리를 인정하는

Remove Thin 100

Light 300

모든 인류 구성원의 천부의 존엄성과 동등하고 양도할 수 없는 권리를 인정하는

Remove Light 300

Regular 400

모든 인류 구성원의 천부의 존엄성과 동등하고 양도할 수 없는 권리를 인정하는

Remove Regular 400

Medium 500

모든 인류 구성원의 천부의 존엄성과 동등하고 양도할 수 없는 권리를 인정하는

Select Medium 500

Noto Sans Korean

Thin 100

Light 300

Regular 400

Add more styles Remove all

Use on the web

To embed a font, copy the code into the <head> of your html

<link> @import

```

<link rel="preconnect" href="https://
fonts.googleapis.com">
<link rel="preconnect" href="https://
fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.
com/css2?family=Noto+Sans+KR:wght@10
0;300;400&display=swap" rel="styleshe
et">

```

1. FOUT / FOIT

- Flash of unstyled text: 웹 폰트를 사용하는 경우 웹 폰트를 다운로드 받는 과정에서 처음 웹 화면이 로딩 되었을 때 기본 폰트에서, 다운로드가 완료된 후 다시 폰트를 적용하면서 (렌더링) 화면이 깜빡거리는 문제
- Flash of invisible text: 웹 폰트가 다운로드 되기 전까지는 화면에서 글자를 보여주지 않고 있다가, 다운로드가 되면 한 번에 텍스트를 보여주는 방식

1. FOUT / FOIT

- FOUT을 대응하기 위한 font-display 속성
 - auto: 사용자의 브라우저에 정의된 기본 동작에 맡김 (기본값)
 - IE는 FOUT, 크롬은 FOIT
 - block: 짧은 시간 동안 텍스트를 안 보이게 해두었다가, 화면에 노출함
 - swap: 우선 기본 폰트로 렌더링 하고, 로딩 다 되면 바꿔줌
 - fallback: 우선 텍스트를 안 보이게 해두었다가, 일정 시간 안에 웹 폰트가 다운되면 바꿔준다. 그 시간안에 실패하면 기본 폰트를 유지함
 - optional: 우선 안 보이게 해두었다가, 네트워크 상황 봐서 fallback 실패처럼 동작하거나, 웹 폰트를 로딩한다.

1. FOUT / FOIT

Specifying fallback font-display

```
@font-face {  
  font-family: ExampleFont;  
  src: url(/path/to/fonts/examplefont.woff) format("woff"), url(/path/to/fonts/examplefont.eot)  
       format("eot");  
  font-weight: 400;  
  font-style: normal;  
  font-display: fallback;  
}
```

font-display 적용 방법. 출처: MDN

2. 경량 / 서브셋 웹폰트

- 웹 폰트 최적화를 위해 폰트 파일의 크기를 최적화 함
 - woff, woff2 포맷을 권장함
 - eot 포맷은 IE 전용 포맷
- 서브셋 웹 폰트: 완성형 한글 총 11,172자 중에서 불필요한 글자 '값' 같은 글자는 줄인 2,350 자의 폰트 파일을 사용할 수 있다

3. CSS selector

- 실습과제 리뷰
 - 80% 이상의 HTML 태그에 각각 스타일을 부여하기
- 문제점
 1. 작성해야 하는 코드의 양이 많아짐
 2. 비슷비슷한 코드를 복사 붙여넣기 해야하는 경우가 많음
 3. 복사 붙여넣기 한 코드들을 일괄적으로 바꾸기 어려움

3. CSS selector

- CSS selector: CSS 스타일을 일괄적으로 적용할 태그들을 선택할 수 있는 CSS 문법. 일괄적으로 적용할 수 있는 다양한 문법적 장치를 활용하여, 복잡한 규칙도 적용할 수 있다.
- <https://flukeout.github.io/> 같은 웹 페이지에서 실습 추천

3. CSS selector

- HTML 문서에 CSS 를 적용하기
 - style 태그를 head 태그 안에 생성. text/css 는 예전 브라우저를 위한 정보
 - `TAG` 자리에 selector 를 넣고
 - `스타일` 자리에 CSS 속성을 넣어주면 됨

```
<html lang="ko">
  <head>
    <meta charset="utf-8" />
    <style type="text/css">
      TAG {
        스타일
      }
    </style>
    <title>레시피</title>
  </head>
```

3. CSS Selector

- 태그 선택기
 - 태그의 이름을 적으면 해당 태그에 전부 적용됨
 - 여러 개의 태그에 적용하고 싶으면 콤마로 연결
- `h1, h2`: h1 태그와 h2 태그에 적용

```
<style type="text/css">
  li {
    color: blue;
  }
  h1,
  h2 {
    font-style: italic;
  }
</style>
```

3. CSS Selector

- `oi li`: 이 태그 밑에 있는 li 태그 선택
 - 띄어쓰기 단위는 하위에 있는 태그를 선택
 - 깊이는 상관 없음
- `oi > li`: 이 태그 바로 밑에 있는 li 태그 선택
 - 바로 아래 단계에 있는 태그만 선택됨
- `oi > li strong`: _____

```
<style type="text/css">
  li {
    color: ■blue;
  }
  h1,
  h2 {
    font-style: italic;
  }

  ol li {
    color: ■green;
  }
</style>
```

3. CSS Selector

- ID 선택자
 - 특정한 **1개**의 태그에 ID 속성을 추가하고, 해당 태그에만 스타일을 적용
 - ID 이름은 항상 유일해야 함
 - CSS 문서 안에서는 ID 이름 앞에 # 붙여서 표현

```
<table border="solid">
  <thead>
    <tr>
      <th>요리</th>
      <th>탄수화물</th>
      <th>단백질</th>
      <th id="fat">지방</th>
    </tr>
  </thead>
  <tbody>
```

```
#fat {
  width: 55pt;
}
</style>
<title>레시피</title>
```

3. CSS Selector

- ID 선택자
 - ID 이름에 제한은 없으나, HTML4 표준 정의는 아래 글자들만 허용하였음
 - 알파벳 대소문자
 - 숫자
 - - : 하이픈 (빼기)
 - _ : 언더스코어 (밑줄)
 - ` ` : 콜론
 - ` ` : 온점
 - HTML5 표준에서는 다른 글자들도 많이 포함 시킴
 - 실무에서는 알파벳 대소문자~하이픈 정도 쓰는 것이 보통
 - 콜론과 온점은 다른 표현식에 쓰이는 글자와 겹쳐 피하려고 함

3. CSS Selector

- 클래스 선택자

- 비슷하거나 동일한 속성을 적용하기 위해 분류(클래스)를 태그들에게 적용
- 클래스는 한 태그에 여러 개, 여러 개의 태그에 동일한 클래스 적용 가능
- 클래스 선택자 앞에는 `.` (온점) 을 붙여 표현

```
<h1>고추장찌개 레시피</h1>
<h2 class="ingredient">재료</h2>
<ul>
  <li>삼겹살 200g</li>
  <li>두부 1/4모</li>
  <li>양파 1/4개</li>
  <li>감자 1개</li>
  <li>애호박 1/4개</li>
  <li>청양고추 2개</li>
  <li>대파 적당량</li>
</ul>

<h2 class="ingredient">양념 재료</h2>
<ul>
  <li>고추장 4스푼(또는 고춧가루)</li>
  <li>국간장(또는 멸치액젓) 적당량</li>
```

```
.ingredient {
  font-size: 20pt;
}
</style>
```


3. CSS Selector

- 하나의 태그는 여러 개의 클래스를 적용할 수 있음
 - 여러 개의 클래스가 겹친다면 CSS 우선순위에 따름
 - 우선순위까지 동일시 마지막에 지정한 속성이 적용

```
.ingredient {  
  font-size: 20pt;  
}  
  
.aquablue {  
  color: aqua;  
}  
</style>  
<title>레시피</title>  
</head>  
<body>  
<h1>고추장찌개 레시피</h1>  
<h2 class="ingredient aquablue">재료</h2>
```


3. CSS Selector

- 클래스 선택자
 - 여러 개의 클래스를 다 가지고 있는 태그만 선택하고 싶은 경우
 - . 선택자를 이어서 붙임 (띄어쓰기 하는 경우 주의!)
 - 띄어쓰기가 들어가면? ingredient 클래스 하위에 있는 aquablue 클래스

```
    .ingredient.aquablue {  
        background-color:  darksalmon;  
    }  
</style>
```

3. CSS Selector

- 클래스 선택자
 - 클래스 중에서 태그만 선택하고 싶은 경우
 - li 태그 이면서 aquablue 클래스를 포함하는 태그 선택
 - 다른 클래스를 포함하는 것 까지는 허용 됨

```
li.aquablue {  
  background-color: ■ aquamarine;  
}  
</style>
```

4. CSS 우선순위

- Selector Specificity: 여러 개의 설정이 겹칠 때 적용되는 우선순위
 - Inline Style(태그에 바로 추가한 style 속성)이 항상 가장 우선순위 높게 적용됨
 - ID column: ID를 지정하면 우선순위가 제일 높음
 - Class column: class, attribute selector, pseudo-class는 중간 우선순위
 - Type column: 태그, ::before, ::placeholder 는 가장 낮은 우선순위
- important: 모든 우선순위를 무시하고 가장 높게 적용할 수 있음

```
1 <html lang="ko">
2   <head>
3     <meta charset="UTF-8">
4     <style>
5       li {
6         color: blue;
7       }
8     </style>
```

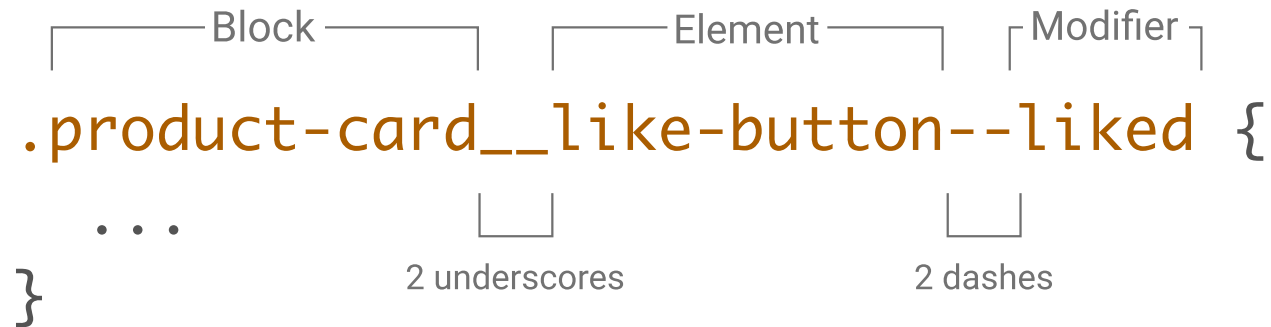
3. CSS Selector

- !important
 - 스타일 마지막에 !important 를 붙여서 적용
 - 인라인 스타일보다도 우선순위 높게 적용
 - important 끼리도 겹치는 경우 css 우선순위를 따름

```
.aquablue {  
  color:  aqua !important;  
}
```

4. CSS BEM 방법론

- 클래스는 여러 개의 태그에 부여할 수 있고 그 역도 성립한다.
 - 클래스 이름 규칙이 없으면 혼란스럽기 쉽다.
- 클래스 이름 규칙을 짓는 방법론 중 하나 BEM 방법론 제시
 - BEM (Block, Element, Modifier) : 세 가지 형식으로 나눔



The diagram illustrates the BEM naming convention for the CSS class `.product-card__like-button--liked`. The class name is broken down into three parts: `product-card` (Block), `__like-button` (Element), and `--liked` (Modifier). Brackets above the text label these parts. Below the class name, a bracket under `__like-button` is labeled "2 underscores", and a bracket under `--liked` is labeled "2 dashes". The class name is followed by an opening curly brace `{`, and an ellipsis `...` is shown below it, indicating that the class is followed by CSS rules. A closing curly brace `}` is shown at the bottom left of the diagram.

4. CSS BEM 방법론

- Block: 재사용 가능한 독립적인 컴포넌트 단위
 - Block 안에는 Block을 넣을 수 있다
 - ex) logo, header, form, card
- Element: 블록에 종속된 구성 단위
 - button, title ...
- Modifier: 일부 성질에 변경이 있을 때 적용하는 단위
 - focused, hover, error...

4. CSS BEM 방법론

- 전체 덩어리를 지정하는 Block
- Block 안에 내용을 지정하면 Element
- 일부 스타일은 Modifier



5. SASS



SASS / SCSS

- CSS의 보완 문법으로 제시된 CSS 전처리기
 - 중첩된 CSS 를 구성하여 속성들을 부여할 수 있음
 - @mixin 통해 미리 정의해놓은 내용을 한 번에 가져올 수 있음

```
1  li {  
2      background-color: blue;  
3      p {  
4          font-size: 14px;  
5      }  
6  }  
7
```


6. position 속성

- 문서 상에 태그를 구성하는 요소를 배치하는 방법을 정의함
 - static: 일반적인 문서 흐름 (inline: 좌->우, block: 상->하)에 따라 배치함(기본)
 - relative: 태그를 일반적인 문서 흐름에 따라 배치한 후, top/left/bottom/right 속성에 따라 오프셋 (간격)을 지정함
 - absolute: 태그를 레이아웃 공간에도 배치하지 않고(태그가 공간을 차지하지 않음) 가장 가까운 위치 지정 기준으로 top/left/bottom/right 속성에 따라 위치가 지정되며, margin-collapse 가 일어나지 않음
 - fixed: 태그를 레이아웃 공간에 배치하지 않고 위치를 viewport 기준으로 top/left/bottom/right 에 배치함. 스크롤 해도 위치가 바뀌지 않음
 - sticky: 태그를 레이아웃에 배치한 후, 상위 태그(조상 태그)중에서 가장 가까운 블록 태그를 기준으로 top/left/bottom/right 에 따라 오프셋 을 지정함.

6. position 속성

- static: 기본값으로, 다른 태그와의 관계에 따라 자동으로 배치됩니다. 위치를 임의로 설정할 수 없습니다.
- relative: 원래 있던 위치를 기준으로 좌표를 지정할 수 있습니다. 레이아웃에는 영향을 주지 않습니다.
- absolute: 절대 좌표와 함께 위치를 지정할 수 있습니다. 부모 요소 중 position 값이 static이 아닌 요소를 기준으로 합니다.
- fixed: 스크롤과 상관없이 항상 문서 최 좌측상단을 기준으로 좌표를 고정합니다.
- sticky: 스크롤 영역의 경계에 도달하면 고정되는 효과를 줍니다.

6. position 실습

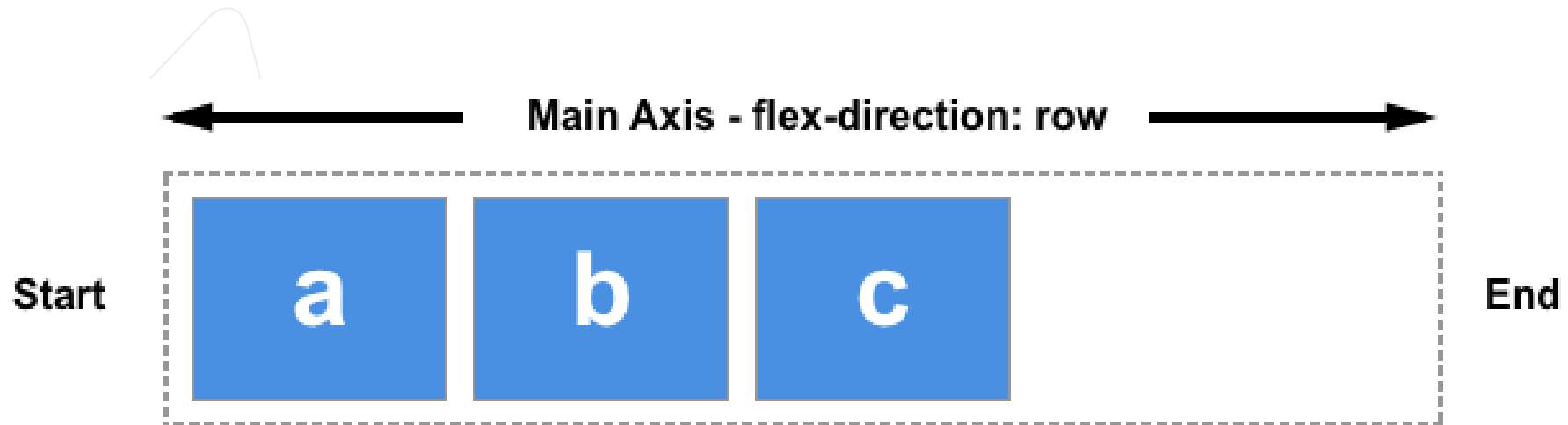
- 주의사항
 1. absolute의 위치 지정은 static 요소가 아닌 가장 가까운 요소를 기준으로 함
 2. sticky 속성은 overflow / scroll 이 존재하는 가장 가까운 요소를 기준으로 함

6. display 속성

- none: 요소를 렌더링하지 않습니다. vs visibility : hidden
- block: 요소를 블록 박스로 표시합니다.
- inline: 요소를 인라인 박스로 표시합니다.
- inline-block: 요소를 인라인 박스로 표시하지만 블록 박스처럼 너비와 높이를 지정할 수 있습니다.
- flex/inline-flex: 요소를 유연한 박스로 표시하고 자식 요소들의 크기와 순서를 조절할 수 있습니다.
- grid: 요소를 그리드 컨테이너로 표시하고 자식 요소들을 그리드 셀에 배치할 수 있습니다.
- table: 요소를 테이블로 표시합니다.
- table-row: 요소를 테이블의 행으로 표시합니다.
- table-cell: 요소를 테이블의 셀로 표시합니다.
- list-item: 요소를 목록 항목으로 표시하고 앞에 마커가 생깁니다.
- run-in: 인라인 박스나 블록 박스 중 부모나 형제 요소에 따라 결정됩니다.
- contents: 자식 요소들을 제외하고 해당 요소는 사라집니다.
- inherit: 부모 요소의 display 값을 상속받습니다.
- initial: display 속성의 기본값인 inline으로 설정됩니다.
- unset: 상속된 값이 있다면 inherit, 없다면 initial로 설정됩니다.

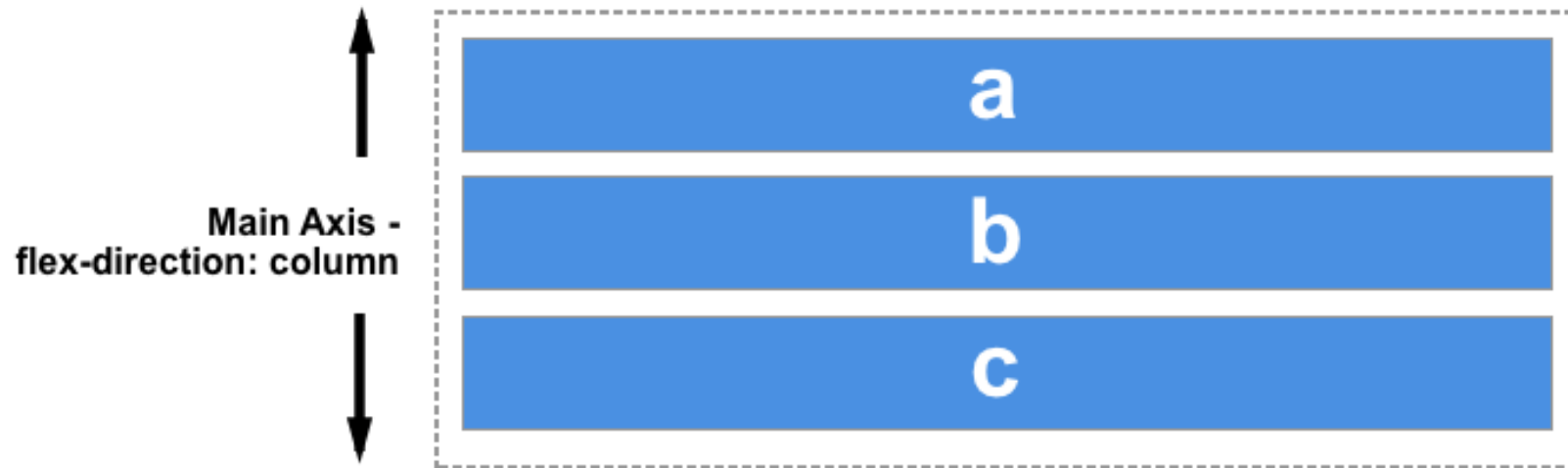
7. flexbox

- flex: flexbox 모델 안에 모든 요소를 추가해준다
 - flex-direction: row/row-reverse 를 적용할 수 있다



7. flexbox

- flex-direction: column/column-reverse
 - 요소들의 배치가 세로로 이뤄진다
 - Main Axis: 요소들이 배치되는 방향
 - Cross Axis: Main Axis 의 회전된 방향



7. flexbox

- flex-wrap: 요소가 공간을 넘었을 때 동작을 정의
 - nowrap: main-axis 방향으로 벗어남 (기본값)
 - wrap: cross-axis 방향으로 벗어남
 - wrap-reverse: cross-axis 반대 방향으로 확장됨
- flex-flow: flex-direction과 flex-wrap이 합쳐진 속성
 - flex-flow: {flex-direction} {flex-wrap} 으로 작성 가능

7. flexbox

- justify-content: Main Axis 방향 정렬
 - flex-start
 - flex-end
 - center
 - space-between
 - space-around
 - space-evenly

flex-start



flex-end



center



space-between



space-around

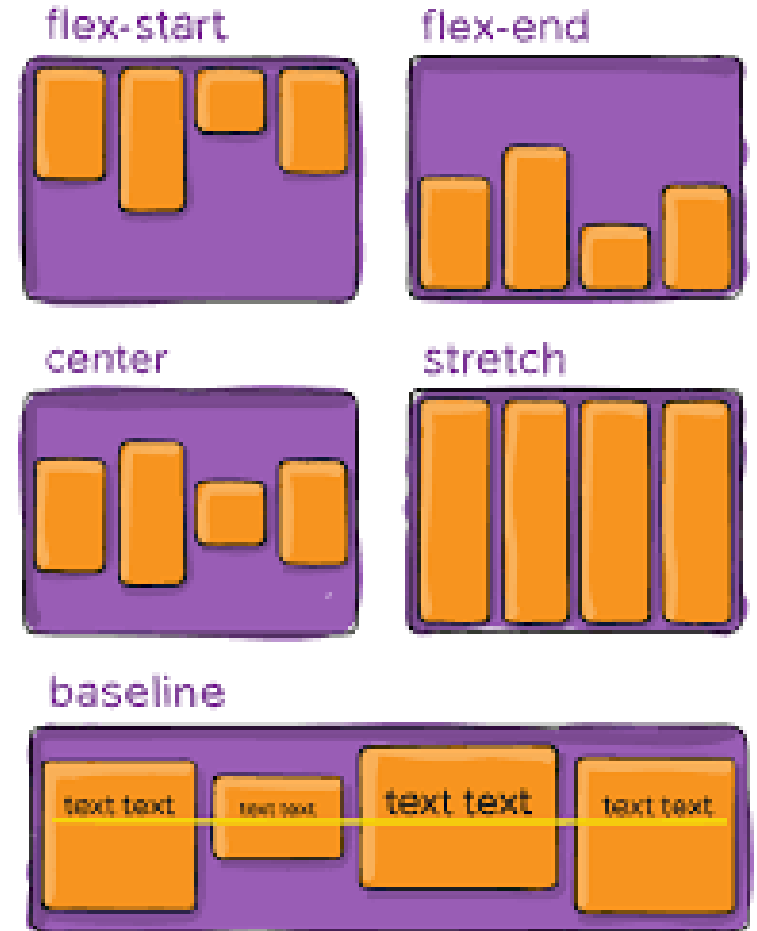


space-evenly



7. flexbox

- align-items: Cross Axis 방향 정렬
 - stretch
 - flex-start
 - flex-end
 - center
 - baseline



7. flexbox

- 태그를 가운데에 중앙에 배치하려면?

```
RecommendResume.tsx  usePreviewIncome.ts
```

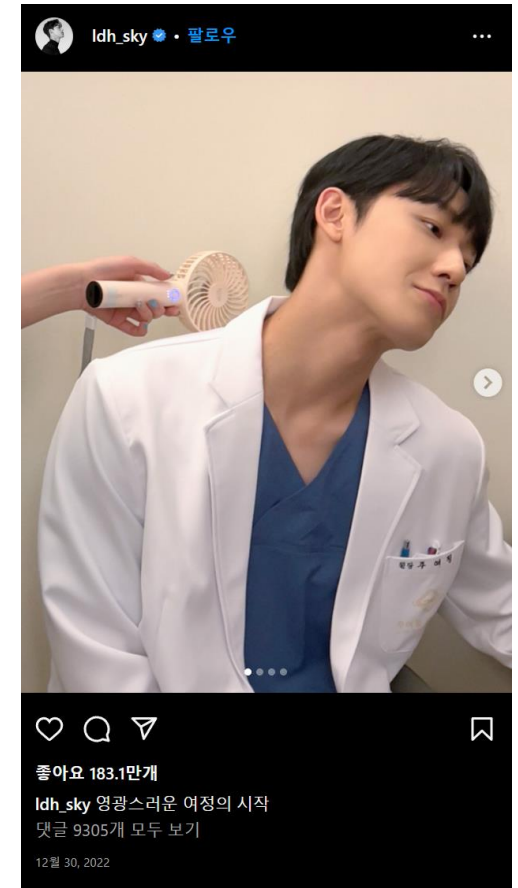
```
{  
  justify-content: center;  
  align-item: center;  
}
```

7. flexbox

- 기타 적용할 수 있는 속성들
 - align-content: cross axis 방향으로 여러줄이 생겼을 때 cross axis 정렬 방향
 - align-self: flex 안의 요소들의 cross axis 정렬 방향
 - flex-basis: flex 안의 요소들의 기본 크기 설정
 - flex-grow: 0/1 flex-basis 보다 유연하게 커지게 할지 여부 선택
 - flex-shrink: 0/1 flex-basis 보다 유연하게 작아지게 할지 여부 선택
 - flex: {flex-grow} {flex-shrink} {flex-basis} 를 한 줄에 적는 속성

실습 과제 1

- 인스타그램 레이아웃을 직접 HTML 에 구성합니다.
 - 이미지는 개인 자유 (정방형 이미지 크기 권장)
 - 이미지를 원형으로 자르는 방법은 리서치 필요
 - carousel / swiper 구현은 X



실습 과제 2

- 레시피 문서에 display 속성을 적용하여 **제시된 레이아웃**으로 구성할 것
 - 단, position: absolute 속성 사용은 지양
- 이력서 HTML의 양식을 각자 스타일 추가해서 개선하기
 - (차주 과제 예정)

다음 시간

- Grid layout
- UI Component 다루기

