서니_7조_5주차 미션

첫번째 미션

- inline level 요소는 좌에서 우로 요소가 추가됩니다. 반면 block level 요소는 위에서 아래로 추가됩니다. 이런 요소 배치 로직을 normal-flow라고 합니다.

참고: https://developer.mozilla.org/ko/docs/Web/CSS/CSS_Flow_Layout

- inline-block 속성값을 사용하면 해당 요소를 inline level 요소처럼 렌더링(배치)하지만 block level 성질을 가질 수 있게 할 수 있습니다.
- display: none을 설정하면 해당 요소가 화면에 렌더링 되지 않는 반면(DOM에 존재하지 않음) visibility: hidden을 설정하면 요소가 화면에 보이지는 않지만 렌더링 되며 화면에 공간을 가지고 있게 됩니다(DOM에 존재함).
- 요소를 normal flow에서 벗어나서 띄우고 싶을 땐 float 속성을 사용합니다. 기사 글 서두에 이미지를 넣어주거나 첫 글자를 크게 키워서 보여주는 등의 목적으로 만들어진 이 속성은 Flexible Box Layout과 Grid Layout이 나오기 전에 레이아웃을 짜기 위한 목적으로도 쓰였으나, Flexible Box Layout과 Grid Layout이 나온 이후에는 본래 목적 이외에 레이아웃을 짜기 위한 목적으로는 잘 쓰이지 않습니다.
- 요소를 원하는 위치로 이동시킬 땐 position 속성을 사용합니다. 일반적인 flow 알고리즘은 여러 요소가 한 픽셀을 차지하지 않도록 합니다. 하지만 이 속성을 사용하면 요소(박스)를 겹치게 할 수 있습니다.
- .position 속성의 기본값은 static이고, relative, absolute, fixed 값을 가질 수 있습니다. 이 중 absolute는 해당 요소를 normal-flow에서 벗어나게 합니다.
- offset은 요소가 화면에서 차지하는 영역 전체 크기를 나타내는데, 요소의 너비와 높이에 패딩, 스크롤바, 테두리를 합친 크기이며 마진은 포함되지 않습니다. offsetLeft 값을 지정(CSS에서 left 속성)하면 offsetParent를 기준으로 요소를 오른쪽으로 옮길 수 있고, offsetTop 값을 지정(CSS에서 top 속성)하면 offsetParent를 기준으로 요소를 아래로 옮길 수 있습니다.

참고:https://ko.javascript.info/size-and-scroll에서 기하 프로피터 섹션

- 요소가 겹칠 때, 어느 요소가 더 위에 올라와야 하는지는 요소의 쌓임 규칙에 따라 결정됩니다. 개발자는 z-index를 사용해 쌓임 순서를 바꿀 수 있습니다.
- 미디어 쿼리를 사용하면 각 미디어 매체에 따라 다른 스타일(CSS style)을 적용할 수 있습니다.
- screen은 미디어 타입에서 가장 많이 쓰이는 타입입니다. 미디어 특성 중 가장 많이 쓰이는 특성은 width 입니다.
- 미디어 특성은 이름 앞에 min- 또는 max- 접두사를 붙일 수 있습니다. 실제로 반응형 사이트를 제작할 때는 보통 접두사를 붙여서 사용합니다.

두번째 미션

- question.html에는 로 감싼 이미지가 하나 있습니다. 그런데

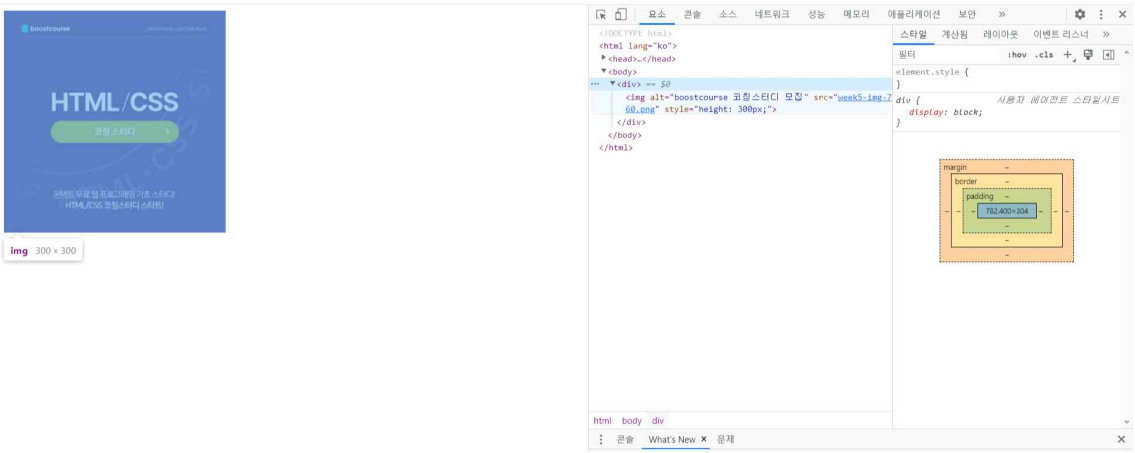
이미지(<img>)의 크기는 가로 300px, 세로 300px인데 이미지를 감싸는 div의 크기는 이미지의 크기보다 크다.

- question.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <title>inline 요소 이해하기</title>
    <meta charset="utf-8">
    <meta content="width=device-width, initial-scale=1" name="viewport">
</head>

<body>
    <div>
        <img alt="boostcourse 코칭스터디 모집" src="week5-img-760.png" style="height: 300px;">
    </div>
</body>
</html>
```

- 개발자 도구를 열어서 요소와 <img> 요소의 computed 값을 캡쳐



- 크기가 <img> 보다 큰 이유

<img> 태그는 글로벌 속성(attributes)이자 replaced 요소이며 inline level 요소입니다. inline 요소는 타이포그래피의 구조 중 baseline을 기준으로 그려집니다. 아래 삐침이 없는 영문자(ex. a, b, c, x…) 하단의 라인인 baseline에 맞춰 그려지지만 아래 삐침이 있는 영문자(ex. g, y…)는 bottom 라인까지 내려갑니다. <img> 태그는 글자가 아니지만 inline 요소이므로 baseline과 bottom 라인 사이의 공간이 생겨 공백이 보이게 됩니다.

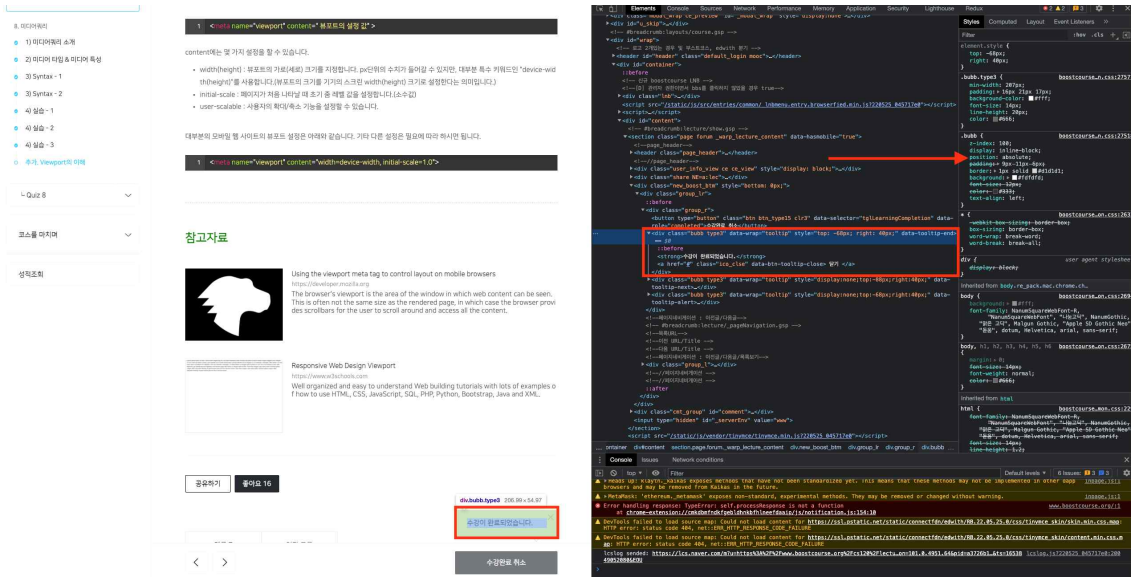
- 가로가 img 태그보다 큰 이유 → div는 block 레벨 요소로서 특정한 width값을 설정해주지 않으면, 해당 컨텐츠의 브라우저 가로의 크기만큼 모두 크기를 차지한다. 그래서 창의 크기를 조절할때마다 div 사이즈는 줄어들거나 늘어난다.
- 세로가 img 태그보다 큰 이유 → img는 inline 요소로서 여백을 가진다. inline 요소h base-line과 bottom-line 사이에 공백이 생긴다. inline 요소가 가로로 쌓이지만 여백은 가로뿐만이 아니라 세로에도 여백이 생긴다

- 세로 크기를 이미지 크기와 같게 하려면 어떻게 해야 하는지

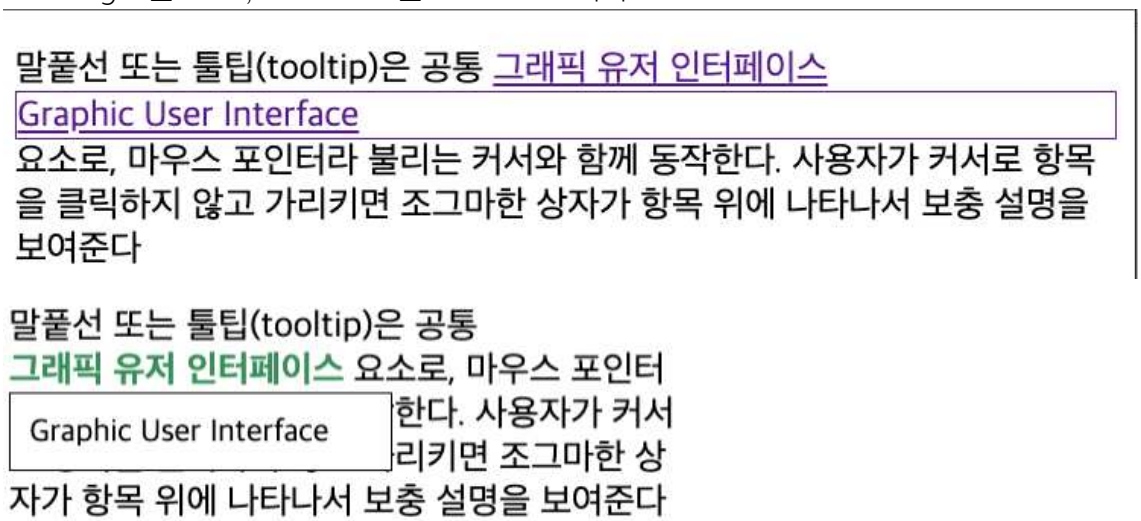
1. 태그(<img> 태그를 감싼)에 이미지의 크기 만큼의 width: 300px; 의 값과 hieght: 300px의 값을 준다.
2. 태그(<img> 태그를 감싼)에 font-size: 0;
3. <img> 태그에 display: block; 의 css 효과를 준다.
4. <img> 태그에 baseline을 제외학기위해 vertical-align : top; 또는 vertical-align : bottom; 의 css 효과를 준다. 그 중에서도 top을 주로 사용합니다.
5. <img> 태그에 margin-bottom: -3px;

세번째 미션

- 툴팁은 아래 그림처럼 웹사이트에서 부연 정보를 선택적으로 보여주기 위한 수단으로 position: absolute; 가 쓰이는 가장 대표적인 예입니다.
- 그림1 : 툴팁을 만들고 있다고 가정해봅시다. 관련 요구사항은 다음과 같습니다. 그런데 현재는 다음 그림과 같이 툴팁이 이상하게 출력되고, 툴팁 좌측이 설명을 덧붙이려는 텍스트의 첫 글자 좌측과 맞지 않습니다. 스타일을 수정해 이를 개선.



- 그림2 : 툴팁 너비는 최소 150px, 툴팁 내 텍스트와 border는 black, font weight는 400, font size는 0.875rem이다.



- question.html 수정 전

```
<!DOCTYPE html>
<html lang="ko">
```

```
<head>
    <title>위치 속성 fixed를 사용해 툴팁 만들어보기</title>
    <meta charset="utf-8">
    <style>
        .tooltip {
          /* 처음엔 툴팁이 숨어있습니다. */
          display: none;
          background: white;
          border: 1px solid;
        }

        /*
            마우스를 오버했을 때, 툴팁을 보여줍니다.
        */
        .tooltip-trigger:hover .tooltip {
          display: block;
        }

        p {
            max-width: 300px;
        }

        .tooltip-trigger {
            font-weight: bold;
            color: seagreen;
            text-decoration: none;
        }
      </style>
</head>
<body>
    <p>
        말풍선 또는 툴팁(tooltip)은 공통
        <a href="/" class="tooltip-trigger">
            그래픽 유저 인터페이스
          <span class="tooltip">
            Graphic User Interface
          </span>
        </a>
        요소로, 마우스 포인터라 불리는 커서와 함께 동작한다. 사용자가 커서로 항목을 클릭하지 않고
가리키면 조그마한 상자가 항목 위에 나타나서 보충 설명을 보여준다.
      </p>
</body>
</html>
```

- question.html 수정 후

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <title>위치 속성 fixed를 사용해 툴팁 만들어보기</title>
    <meta charset="utf-8">
    <style>
         .tooltip {
                    /* 처음엔 툴팁이 숨어있습니다. */
                    display: none;
                    background: white;
                    border: 1px solid;
         }

         /*
            마우스를 오버했을 때, 툴팁을 보여줍니다.
        */
```

```
        .tooltip-trigger:hover .tooltip {
                display: block;
                min-width: 150px;
                color: black;
                font-weight: 400;
                font-size: 0.875rem;
                position: absolute;
        }

        p {
                max-width: 300px;
        }

        .tooltip-trigger {
                font-weight: bold;
                color: seagreen;
                text-decoration: none;
        }
    </style>
</head>
<body>
    <p>
        말풍선 또는 툴팁(tooltip)은 공통
        <a href="/" class="tooltip-trigger">
                그래픽 유저 인터페이스
                <span class="tooltip">
                        Graphic User Interface
                </span>
        </a>
        요소로, 마우스 포인터라 불리는 커서와 함께 동작한다. 사용자가 커서로 항목을 클릭하지 않고
가리키면 조그마한 상자가 항목 위에 나타나서 보충 설명을 보여준다.
    </p>
</body>
</html>
```

- 현재 툴팁은 inline 요소인 <a> 태그에 사용자 동작과 관련된 가상 클래스 hover로 숨겨져 있는 기본 상태(display:none;)에서 보여지는 상태(display:block;)로 바뀌도록 되어 있습니다. normal flow에 속해 있는 상태이기 때문에 <a> 태그에 마우스를 올리거나 올리지 않을 때 본문의 모습에 영향을 줍니다.

1. 툴팁(.tooltip)에 position : absolute; 추가
 본문 위로 툴팁이 뜨기 위해서는 툴팁에 position : absolute를 선언해주어야합니다. 이 때, float를 선언해주면 안되는 이유는 float는 요소를 보통의 흐름에서 벗어나 띄어지게 하지만 주변 텍스트나 인라인 요소가 주위를 감싸기 때문에 안됩니다.
2. 툴팁을 감싼 <a> 태그에 position:relative 추가
 우리는 '그래픽 유저 인터페이스'라는 내용이 담긴 <a> 태그에 마우스를 올리면 툴팁이 보이도록 하고 싶습니다. 툴팁의 부모 요소인 <a> 태그에 position:relative를 선언해 줌으로써 자식 요소인 툴팁의 기준이 되도록 만들어줍니다.
3. 툴팁에 left:0; 추가
 '설명을 덧붙이려는 텍스트의 첫 글자 좌측과' 맞게 하려면 툴팁이 기준이되는 <a> 태그 왼쪽에 붙도록 left값을 0으로 줍니다.
4. 툴팁에 min-width: 150px 추가
 요구사항 1번에 '너비는 최소 150px'이라고 되어있으므로 width에 min 프리픽스를 붙여 적어도 너비 150px을 가지게끔 만들어줍니다. 나중에 툴팁의 내용이 늘어날 수 있는 상황도 커버할 수 있습니다.
5. 요구사항 2번에서 4번까지 스타일 추가적용
 border: 1px solid #000; /border 색 검은색/ color: #000; /글자색 검은색/ font-weight: 400; /글자 굵기 400/ font-size: 0.875rem; /글자 크기 0.875rem(14px)/
6. 툴팁에 padding: 12px 추가

수정완료 이미지를 보니 툴팁 텍스트에 사방으로 여백이 있게 나와있어서 눈대중으로 padding을 사방으로 12px 주었습니다.
7. 툴팁을 감싼 <a> 태그에 display:inline-block 추가
→ <a> 태그는 position: relative를 가지고 있는 inline 요소 상태이기 때문에 위의 두 캡쳐 화면처럼 툴팁이 나옵니다. 그래서 이 때 <a> 태그에 inline 이면서 block 레벨의 성질도 가지는 inline-block 요소로 만들어주면 수정완료 이미지 처럼 나옵니다.

네번째 미션

- Tailwind CSS는 빠르게 시장 점유율을 높여가는 CSS 프레임워크입니다. Tailwind CSS 사이트를 방문해서 디스플레이 사이즈를 변경해가면서 Tailwind CSS는 스크린 사이즈(width) 몇을 기준으로 미디어 쿼리를 분기 처리했는지 분석

사이트 링크: https://tailwindcss.com/
- 총 6개의 브레이크 포인트로 640px, 720px, 768px, 1024px, 1280px, 1536px 을 사용하고 있습니다.

```
/*! purgecss end ignore */
.container{width:100%}
@media (min-width:640px){.container{max-width:640px}}
@media (min-width:720px){.container{max-width:720px}}
@media (min-width:768px){.container{max-width:768px}}
@media (min-width:1024px){.container{max-width:1024px}}
@media (min-width:1280px){.container{max-width:1280px}}
@media (min-width:1536px){.container{max-width:1536px}
```

- Tailwind CSS 에서는 일반적으로 5개의 브레이크 포인트를 설명하지만, 자사 웹 사이트에서는 700px 을 추가하여 제작되어있었습니다. :hover, :focus, :first-child 및 같은 의사 클래스 :required가 포함되어 있고, ::before, ::after, ::placeholder 및 같은 의사 요소 ::selection이 포함되어 있다. 그리고 반응형 중단점, 다크 모드 및 prefers-reduced-mothin, 속성 선택자, like [dir="rtl"] 및 [open]rk 있다. 이러한 modifies는 더 구체적인 상황을 대상으로 하기 위한 스택이 될 수 있다.

Breakpoint prefix	Minimum witdh	CSS
sm	640px	@media (min-width: 640px) { ... }
md	768px	@media (min-width: 768px) { ... }
lg	1024px	@media (min-width: 1024px) { ... }
xl	1280px	@media (min-width: 1280px) { ... }
2xl	1536px	@media (min-width: 1536px) { ... }

→ 위와 같은 분기 처리를 하였다는 것을 알게 되었고 위와 같은 너비로 디스플레이 사이즈를 변경해가면서 실제로 어떻게 변화하는지 확인해 보았습니다. Bootstrap과 유사한 mobile first breakpoint system을 가지고 있다는 것도 알게 되었습니다.

유틸리티 클래스 기반의 CSS

유틸리티에 사전에 만들어진 css로 인해 따로 파일을 작성하지 않아도 미리 정의된 유틸리티를 이용해 HTML에 클래스를 넣어서 레아이웃의 스타일을 만들어주는 프레임워크로서 편리함.

- 적용할 스타일의 클래스 이름을 생각하지 않아도 됩니다.
- HTML 태그안에 바로 사용되므로 CSS 파일을 따로 작성하지 않습니다.
- HTML 태그에 local하게 적용되므로 스타일 변경 시 전체적인 레이아웃에 가는 영향을 크게 생각하지 않아도 됩니다.
- 인라인 스타일링과 달리 호버링, 포커싱, 미디어 쿼리 등의 구현도 추가적인 CSS를 작성하지 않고 구현할 수 있습니다.
- 동일한 속성을 변경하고자 할 때 멀티커서로 동시에 속성을 바꾸는 것을 통해 여러 요소의 스타일을 한 번에 변경할 경우 이점이 있습니다.

모바일 우선 중단점 시스템 (mobile first)

작은 사이즈를 중심으로 먼저 설정한 후 더 큰 하나의 브레이크 포인트를 설정한다. 브레이크 포인트에는 최소 너비만 포함되고 최대 너비는 포함되지 않습니다. 초기의 컨테이너, 즉 브라우저의 가로 사이즈(container)를 사용하도록 100%으로 설정하고 전체의 와이드 값의 최소값이 특정 a일때, 컨테이너(container)의 최대값을 브래이크 포인트로 만들어주어, 그 시점에 스크린의 사이즈가 변화할 수 있도록 만든다.