📌 Q1. 5주 차에 배운 개념을 정리해봅시다. 빈 칸을 채워 봐주세요. 빈 칸에 들어갈 수 있는 키워드는 다음과 같습니다.

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

2. **inline-block** 속성값을 사용하면 해당 요소를 inline level 요소처럼 렌더링(배치)하지만 block level 성질을 가질 수 있게 할 수 있습니다.

3. **display: none;** 을 설정하면 해당 요소가 화면에 렌더링 되지 않는 반면(DOM에 존재하지 않음) **visibility: hidden;** 을 설정하면 요소가 화면에 보이지는 않지만 렌더링 되며 화면에 공간을 가지고 있게 됩니다(DOM에 존재함).

4. 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.

5. 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.

6. position 속성의 기본값은 **static**이고, **absolute, fixed, relative**값을 가질 수 있습니다. 이 중 **absolute**는 해당 요소를 normal-flow에서 벗어나게 합니다.

7. **offset**은 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.

8.요소가 겹칠 때, 어느 요소가 더 위에 올라와야 하는지는 요소의 쌓임 규칙에 따라 결정됩니다. 개발자는 **z-index**를 사용해 쌓임 순서를 바꿀 수 있습니다.

9.**미디어 쿼리**를 사용하면 각 미디어 매체에 따라 다른 스타일(CSS style)을 적용할 수 있습니다.

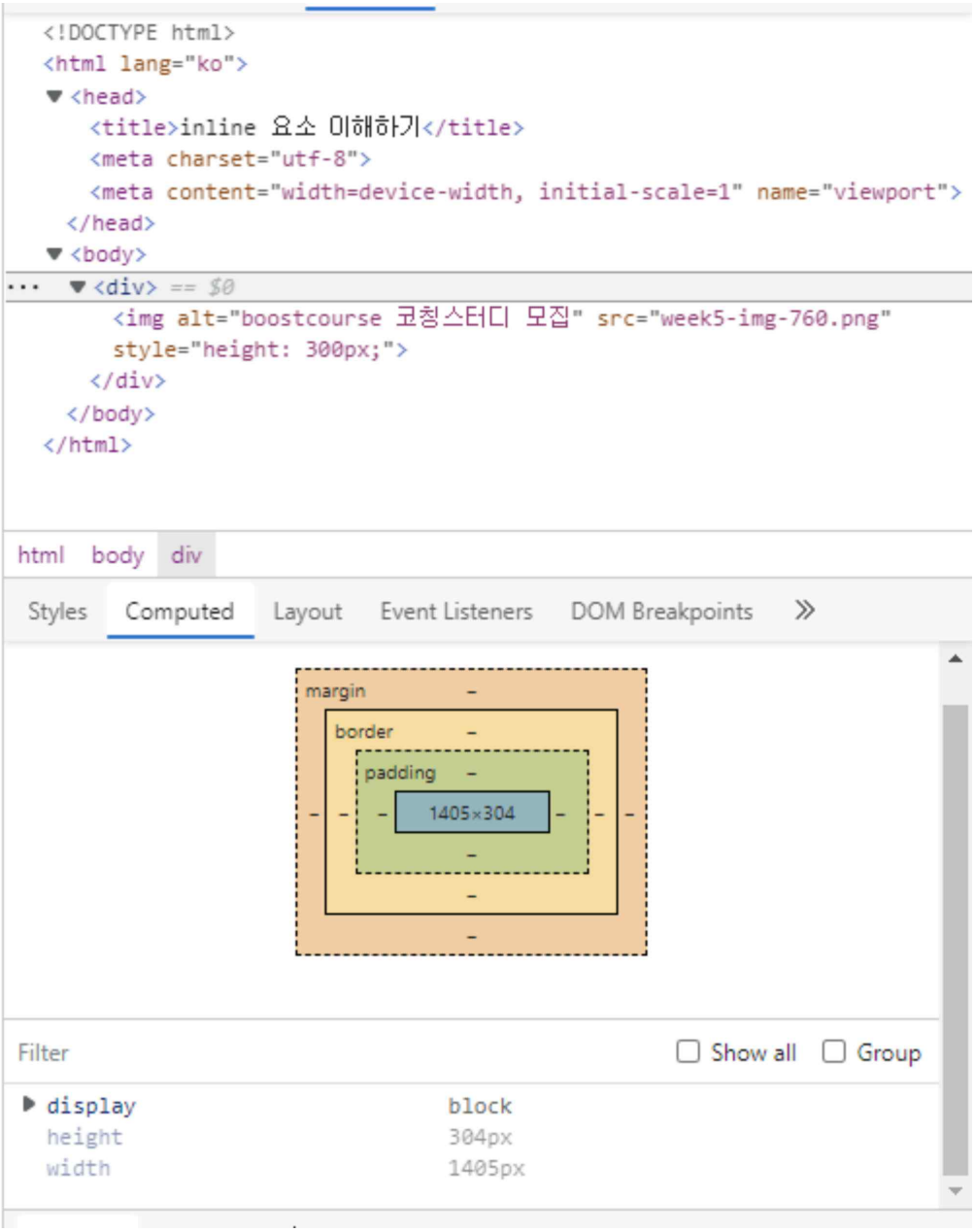
10. **screen**은 미디어 타입에서 가장 많이 쓰이는 타입입니다. 미디어 특성 중 가장 많이 쓰이는 특성은 **width**입니다.

11. 미디어 특성은 이름 앞에 **min-** 또는 **max-** 접두사를 붙일 수 있습니다. 실제로 반응형 사이트를 제작할 때는 보통 접두사를 붙여서 사용합니다.

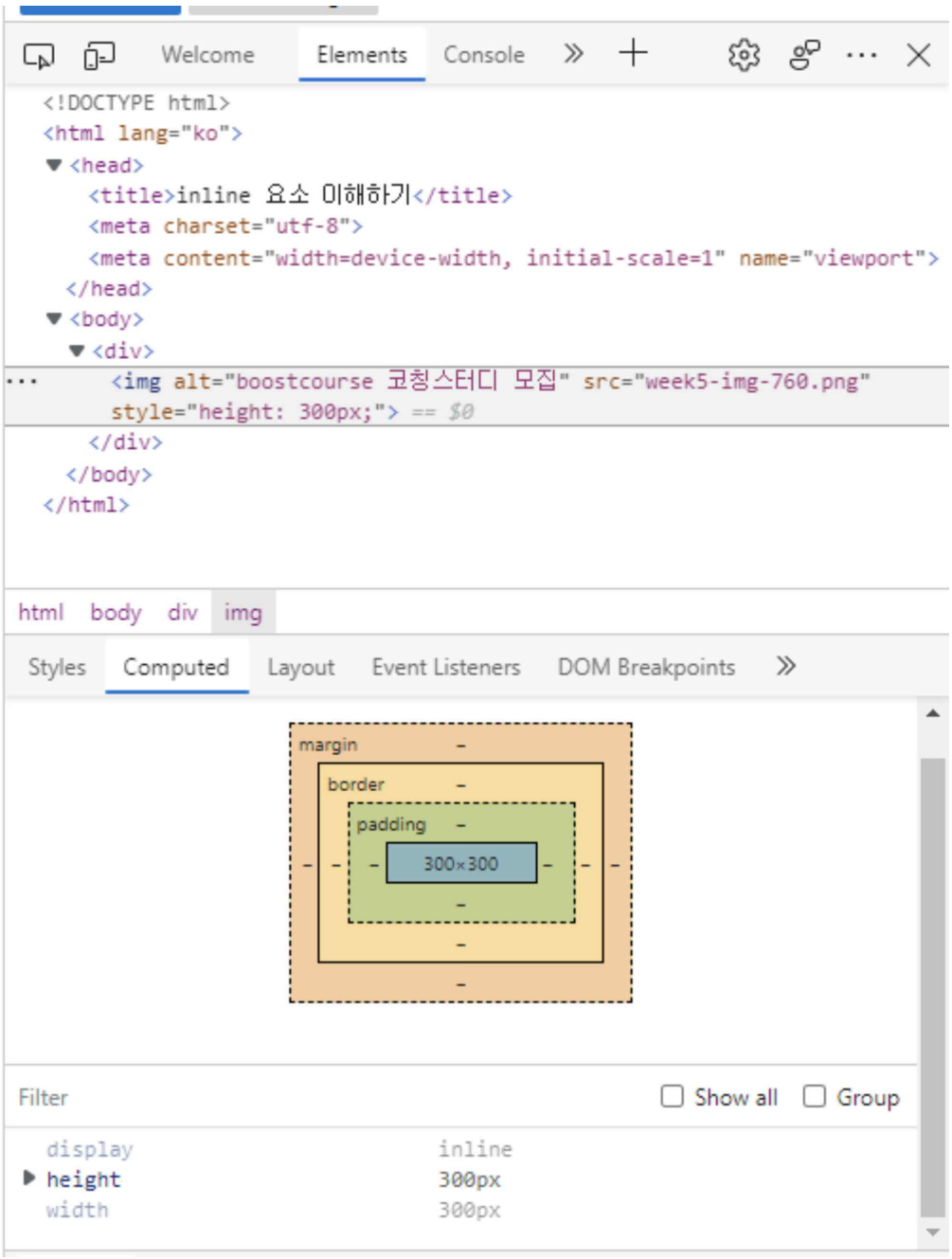
📌 Q2. question.html엔 로 감싼 이미지가 하나 있습니다. 엇 그런데 뭔가 이상하네요. 이미지(<img>)의 크기는 가로 300px, 세로 300px인데 이미지를 감싸는 div의 크기는 이미지의 크기보다 큽니다.

1. 개발자 도구를 열어서 요소와 <img>요소의 computed 값을 캡처해주세요.

- 요소 computed 값



- <img> 요소 computed 값



2. 크기가 <img> 보다 큰 이유를 적어보세요.

: div 태그는 block 요소 태그이므로 가로 영역의 전 범위를 차지하는 반면, img태그는 인라인 요소이기 때문에 지정된 300px 만큼의 크기를 차지하기 때문입니다.

3. 세로 크기를 이미지 크기와 같게 하려면 어떻게 해야 하는지 적어보세요.

1) <img>에 CSS를 다음과 같이 수정하여 이미지 크기를 세로 크기와 동일하게 할 수 있습니다.

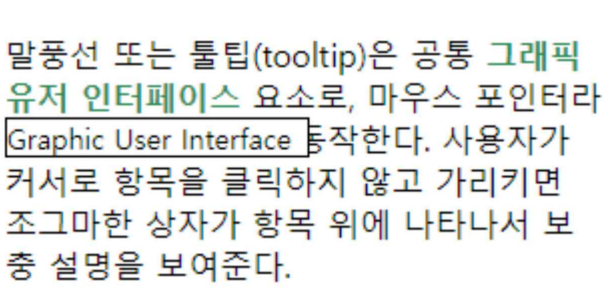
```
<img alt="boostcourse 코칭스터디 모집" src="week5-img-760.png" style="height: 300px; vertical-align: top;">
```

2) <img>의 style에 display: block;를 추가로 선언하면서 세로 크기와 동일하게 할 수 있습니다.

```
<img alt="boostcourse 코칭스터디 모집" src="week5-img-760.png" style="height: 300px; display: block;">
```

📌 Q3. 현재는 다음 그림과 같이 툴팁이 이상하게 출력되고, 툴팁 좌측이 설명을 덧붙이려는 텍스트의 첫 글자 좌측과 맞지 않습니다. 스타일을 수정해 이를 개선해주세요.

먼저, 커서를 올리면 Normal-flow에 흐름에 영향을 받아 다른 요소들이 뒤로 밀려납니다. 따라서 Normal-flow에 영향을 받지 않기 위해 position: absolute;를 선언해줍니다. 또한 나머지 스타일 조건 1~4에 대한 스타일도 추가로 선언해주었습니다.

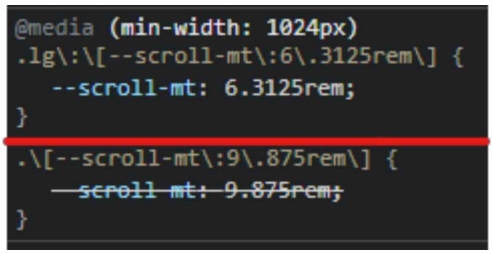



```css
.tooltip {
  /* 처음엔 툴팁이 숨어있습니다. */
  display: none;
  background: white;
  border: 1px solid;
  position: absolute;
  min-width: 150px;
  color: black;
  font-weight: 400;
  font-size: 0.875rem;
}
```

+추가로 미션의 툴팁에 비해 여백이 없는 것 같아 padding을 추가해서 미션 툴팁과 근접하게 구현했습니다.


```css
.tooltip {
  /* 처음엔 툴팁이 숨어있습니다. */
  display: none;
  background: white;
  border: 1px solid;
  position: absolute;
  min-width: 150px;
  color: black;
  font-weight: 400;
  font-size: 0.875rem;
  padding: 10px 15px 10px 10px;
}
```

📌 **Q3. 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 분석해보세요.**



제일 위에 있는 html요소를 클릭하니까 @media라는 미디어 쿼리를 볼 수 있고 해당 미디어 쿼리는 expression Syntax로 되어있습니다. 보이는 것과 같이 min-width: 1024px(태블릿 환경)을 기준으로 해당 조건이 되면 빨간선 기준으로 윗부분이 적용이 되고, 그게 아니면 빨간선 기준 밑부분의 것이 적용될 것이라고 판단했습니다.