![텍스트, 엠블럼, 로고, 원이(가) 표시된 사진

자동 생성된 설명]()

**웹프로그래밍 과제 #1 보고서**

2025. 04. 11.

건국대학교 컴퓨터공학부 202211286 김진영

**목차**

1. 레이아웃 도안
2. 주요 기능
3. 토의 사항
4. **레이아웃 도안**

html 레이아웃을 이미지로 표현하면 다음과 같습니다.

스크린샷, 텍스트, 운영 체제, 직사각형이(가) 표시된 사진

자동 생성된 설명

1. **주요 기능**

**(1) :hover, :active, :checked 등 반응/선택 선택자**

*#category*:*hover* ~ *#sub-category*, *#sub-category*:*hover* {  
 opacity: 1;  
 transform: translateY(-15px);  
 pointer-events: auto;  
}

*#sub-category* > label:*hover* {  
 color: darkgrey;  
 transition: 0.5s;  
 font-size: 20px;  
}  
*#wrapper*:*has*(*#article\_4*:*checked*) *#sub-category* > label[*for*="article\_4"] {  
 color: darkgrey;  
}

img:*hover*, video:*hover* {  
 transform: scale(1.02);  
 transition: 0.3s ease;  
}

*#content* img:*active*, video:*active* {  
 transform: scale(2);  
}

반응 선택자와 선택 선택자를 적절히 사용하여 mouse hover시 해당 태그 혹은 연관 태그의 텍스트의 색, 배경색을 변화시키거나 컨텐츠의 크기를 변화시켜 역동적인 디자인으로 구성했습니다.

**(2) display**

*#main-title* {  
 display: flex;  
 flex-direction: column;  
 justify-content: flex-start;  
 align-items: center;  
}

*#label-container* {  
 display: flex;  
 justify-content: center;  
 align-items: center;  
}

display의 값을 flex 레이아웃으로 적용했습니다. flex는 태그내 컨텐츠를 가운데 정렬, 왼쪽/오른쪽 정렬, 상하 정렬 등 1차원 정렬에 특화된 레이아웃입니다. 기사 형식의 아티클 웹페이지에 적합하다고 판단하여 네비게이션 바, 섹션 컨텐츠 등에 사용하였고, 필요한 경우 flex-direction 속성으로 정렬 방향을 바꾸거나 justify-content, align-items의 속성 값을 space-evenly, flex-start 등으로 설정하여 각 컨텐츠의 간격을 일정하게 하여 정렬 혹은 방향의 시작점에 붙여 정렬했습니다.

**(3) transform**

*#sub-category* {  
 position: absolute;  
 top: 100%; */\* 메뉴 바로 아래 위치 \*/* right: 0;  
 transform: translateY(-10px);  
}

Navigation bar에 특정 컨텐츠(Category)에 mouse hover시 Category list가 적절히 배치되도록 transform 속성을 사용했습니다. 이때 position, top, right 속성을 사용해 부모 태그의 위치로부터 절대적인 값으로 배치 되도록 했습니다.

**(4) :has 조건선택자**

*#wrapper*:*has*(*#article\_4*:*checked*) *#top-content-4* {  
 display: flex;  
}

*#wrapper*:*has*(*#article\_4*:*checked*) *#sub-category* > label[*for*="article\_4"] {  
 color: darkgrey;  
}

:has 선택자를 사용해 특정 태그의 상태에 따라 임의의 태그들의 속성을 변화시킬 수 있도록 했습니다. :has 선택자의 장점은 기존 nth 선택자나 그 외 반응/선택 선택자와 달리 특정 태그와 변화를 주려는 태그 관계가 동위 관계거나 부모/조상 관계일 필요가 없습니다. 오로지 같은 조상 혹은 부모 태그를 가지고 있다면 사용 가능합니다. 모든 태그의 조상 태그는 body 태그이므로 폭넓은 사용 가능한 선택자입니다.

**(5) transition**

*#sub-category* > label:*hover* {  
 transition: 0.5s;  
}

label {  
 transition: 0.5s ease;  
}

transition 속성을 사용해 좀 더 부드러운 디자인 변화를 보일 수 있도록 했습니다. ease 값을 추가해 순간 변화율이 전반에 증가하다 후반에 감소하는 변화로 했습니다.

**(6) z-index**

*#navigation* {  
 z-index: 1000;  
}

img, video {  
 z-index: 999;  
}

z-index를 사용해 부자연스러운 컨텐츠 겹침을 해소했습니다. 높은 수를 가질 수록 낮은 수를 가진 컨텐츠보다 더 위에 위치하게 하여 아티클 컨텐츠가 navigation bar 위에 표시되지 않도록 했습니다.

**(7) box-shadow, text-shadow**

*#main-container* {  
 text-shadow: 2px 2px 5px rgba(0, 0, 0, 0.3);  
}

img, video {  
 box-shadow: 0 4px 8px rgba(0, 0, 0, 0.5);  
}

box-shadow, text-shadow를 사용해 컨텐츠를 입체적으로 표현했습니다.

**(8) letter-spacing, line-height**

.*content-p* {

line-height: 2;  
 letter-spacing: 1px;  
}

letter-spacing 속성으로 글자 간격, line-height 속성으로 적절한 줄간격을 설정해 가독성을 높였습니다.

**(9) cursor**

*#sub-category* > label {  
 cursor: pointer;  
}

a tag에 mouse hover할 때의 상태를 똑같이 구현하기 위해 cursor 속성을 pointer 값으로 label tag에 부여했습니다.

1. **토의 사항**

**(1) viewport에 따른 반응형 웹페이지 필요**

현재 이 웹페이지는 PC 혹은 노트북 화면 크기를 기준으로 제작되었습니다. 모바일 디바이스나 작은 태블릿 디바이스로 이 웹페이지를 방문할 경우 부적절한 컨텐츠 크기, 디자인이 발생되거나 사용자가 웹페이지에 대해 불만족할 수 있습니다.

**(2) javascript 미사용으로 복잡하고 비효율적인 html, css 구성**

과제의 조건에 따라 javascript를 사용하지 않고 웹페이지를 구현했습니다. 이에 html과 css의 구성이 제한되었습니다. (예: 탭바 형식 레이아웃 구성, 불필요한 css 선택자 사용하여 탭바 구현)

이는 추후에 컨텐츠의 수가 무수히 늘어나거나 다양한 형식의 컨텐츠를 삽입해야 할 때 비효율적이고 복잡한 구조로만 제한됩니다.