****

**REPORT**

**웹프로그래밍 과제 리포트**

제출일 : 2023년 4월 10일

학번 : 202011372

이름 : 조성현

1. **레이아웃 도안**

위 홈페이지는 div class=”total-page”인 태그 안에 4개의 div class=”bookcover” 기사형태로 이루어져 있다. 각각의 기사마다 nav-bar를 위치시켰는데 1,3 기사는 화면 오른쪽(빨간색 영역)에 위치시켰으며 2,4 기사는 기사 젤 윗부분(갈색)에 위치시켜 레이아웃의 약간의 변화를 두었다.



첫 번째 기사에 레이아웃은 아래 그림과 같다. 가장 바깥영역(검은색 테두리 부분) class = “bookcover” – 초록색 영역인 class=”book-dot – 노란색 영역인 class=”page”로 이루어져 있으며 class=”page”은 영역은 하늘색 영역인 class=”profile-container” – 남색 영역인 class=”content-container” – 빨간 영역인 class=”menu-container”로 구성되어 있다.



두 번째 기사의 레이아웃은 첫번째와 약간 다르다. 가장 바깥영역(검은색 테두리 부분) class = “bookcover” 위에 class=”nav-bar”영역이 추가하여 기존 빨간 영역의 네비게이션 역할을 수행한다.

빨간 영역의 부분은 남색과 하늘색 영역이 추가로 차지하며 메인 컨텐츠인 class=”content-container”에 하위 태그인 class=”miniroom” 파트에 스크롤 바 기능을 통해 20문 20답을 진행했다.

세 번째 기사의 레이아웃은 첫 번째 기사와 유사하다. 가장 바깥영역(검은색 테두리 부분) class = “bookcover” – 초록색 영역인 class=”book-dot – 노란색 영역인 class=”page”로 이루어져 있으며 차이점은 class=”page”은 영역은 하늘색 영역을 남색 영역 class=”content-container”에 추가시켰고 네비게이션 바 역할을 다시 빨간 영역인 class=”menu-container”로 구성했다.

네 번째 기사의 레이아웃은 두 번째 기사와 유사하다. 가장 바깥영역(검은색 테두리 부분) class = “bookcover” 위에 class=”nav-bar”영역이 추가하여 기존 빨간 영역의 네비게이션 역할을 수행하며 노란색 영역에 class=”page”영역에 남색 영역인 box-content-box를 추가하였으며 하위 태그인 class=”miniroom” 파트에 스크롤 바 기능을 통해 블로그 형식으로 찍은 사진과 스토리들을 전개했다.

ㅈㅈㅈㅈㅈㅈㅈㅈㅈㅈㅈ

1. **CSS 주요 기능**

Body 태그에 대한 기본적인 위치 및 패턴 형태의 배경 이미지를 설정했다. 그리고 네 가지 형태의 기사를 스크롤 바를 내리면서 볼 수 있도록 overflow: scroll를 추가했다

텍스트이(가) 표시된 사진

자동 생성된 설명

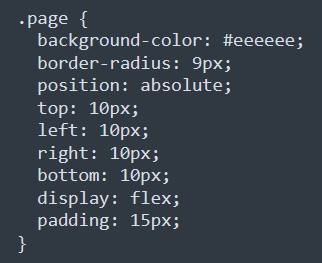
가장 외곽을 차지하는 bookcover에는 색깔, 크기, margin는 기본적인 설정을 하였다. 특히 자식 태그인 bookdot에 position: absolute를 통한 상대적인 위치 설정을 하기 위해 position: relative를 설정하였다. Margin 값을 30px auto를 두었다가 margin-bottom만 추가한 이유는 2, 4번째 기사에 네비게이션 바와 간격을 두기 위해 추가적으로 설정하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

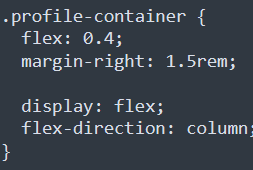
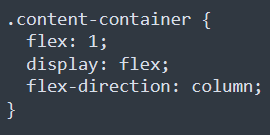
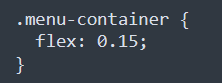
그 다음은 영역인 class = “page”에는 bookdot 기준으로 10px 안쪽에 위치시켰으며 새롭게 배운 display: flex 속성을 설정하여 flexbox를 통해 자식 요소들이 수평으로 정렬이 되도록 하였다.



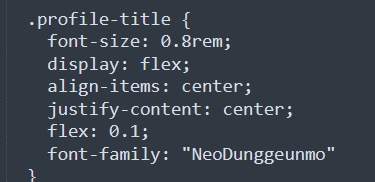
부모 태그인 class=”page” 영역에 display: flex 설정하여 자식 태그들인 3가지 요소들이 수평 정렬이 되었고 특히 수평 정렬이 컨텐츠 부분인 content-container 부분을 기준으로 상대적인 크기를 설정하기 위해 각기 다른 flex 값을 지정하여 메인 컨텐츠를 큰 영역을 프로필과 네비게이션 부분을 상대적으로 적은 영역을 배치했다.

또한 profile, content 컨테이너에는 추가적인 내용이 나오므로 display: flex 설정을 해주었고

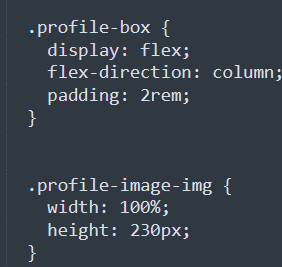
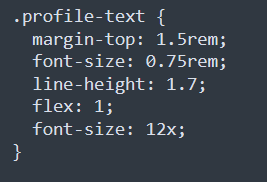
Flex-direction: column으로 설정하여 수평이 아닌 수직 정렬이 되도록 설정했다.

Profile-container 하위 요소인 profile-title에는 크기 flex: 0.1 차지하도록 했고 차지하는 공간안에서 중앙으로 정렬되도록 하기 위해 align-items, justify-content 값을 center로 두었다.



Profile-box 영역에 이미지와 좋아하는 노래가사, 이름, 버튼 등 다양한 요소들이 수직 정렬되도록 display: flex, direction은 column 설정했다. 각 요소들의 크기 값들은 어림잡아 설정하고 크기를 바꿔 나가면서 최적의 크기를 설정했다.

  텍스트이(가) 표시된 사진

자동 생성된 설명

프로필 영역에서 Click Me! 버튼에 마우스가 hover 상태가 되면 인스타그램, 블로그 해당 사이트로 이동할 수 있도록 display: block 형태를 주어 2줄로 출력되게 하였으며 box-shadow값과 text-decoration 값을 none으로 주어 심미성을 높였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

폰트를 적용하기 위해서는 html의 head영역에 링크를 걸고 CSS에 font-family를 설정했다.

하지만 CSS에서 @font-face 속성을 이용하여 사용할 폰트의 이름 및 해당 폰트를 다운받을 수 있는 위치를 브라우저에게 알리고 font-weight, font-style 등의 기본 값들을 설정 한 채 적용시킬 수 있어 편리했다.

텍스트이(가) 표시된 사진

자동 생성된 설명

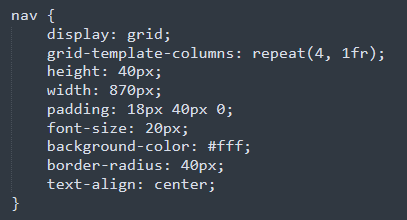
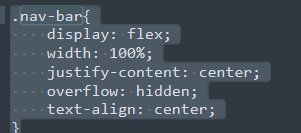
메인 컨텐츠 부분은 프로필 영역과 flex 크기 설정, margin등 위치 관계 설정 중복되는 내용이므로 넘어가서 1,2 기사에 사진 3장이 배치되는 부분에서 원래는 div 태그안에 사진과 소감 부분을 같이 넣기 위해 flex-wrap: wrap으로 넣어서 div 끝에 다다르면 자동으로 줄이 바뀌도록 설정했지만 코드의 단순화를 위해 사진을 담당하는 div 소감부분을 담당하는 div로 나누어 코드를 단순화하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

2,4 번째 기사 부분에 윗부분에 네비게이션 바에 CSS 속성이다. Text-align 값을 center로 주어 가운데에 위치하도록 하였고, border-radius를 통해 직사각형 테두리를 곡선으로 바꾸었다. Height, width, padding 값들은 실제로 바꿔 보면서 설정하였다. 새롭게 배운 display: grid, grid-template-columns 값을 설정하여 navbar 안에 값들인 자기 소개 / 짝꿍 소개 / 올림픽 공원 / 추천 경로! 값들이 일정한 간격을 두고 배치되도록 하였다. << 네비게이션 바 안의 값과 레이아웃 설정

이제는 네비게이션 바 외관상의 레이아웃을 설정하기 위해 width 값을 주고 justify-content, text-align 값을 center로 설정하여 가운데로 정렬되도록 하였다. 또한 overflow: hidden으로 두어 크기를 넘어가는 부분을 자동으로 감춰지도록 설정했다.

navbar 안의 값들을 class=”item”으로 주어 a태그의 기본 decoration none변경하고 hover 되었을 때 색상이 변하도록 설정하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

3번째 기사의 올림픽 공원에 대한 소개 부분 사진을 넣기 위해 부모 태그 class=”introducing”에

Width = 100% 가로 길이 설정 및 display: flex, direction를 설정해 가로로 배치되도록 하였고

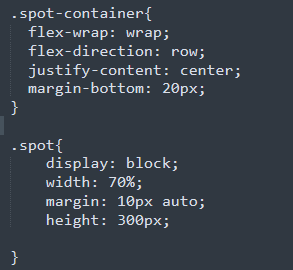
자식인 picture2 태그에는 inline-block 설정과 height 값을 주어 높이 값을 가지지만 inline으로 배치되도록 하였다. Margin: 10px auto: 상하 값 설정 및 좌우는 가운데 정렬되도록 하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

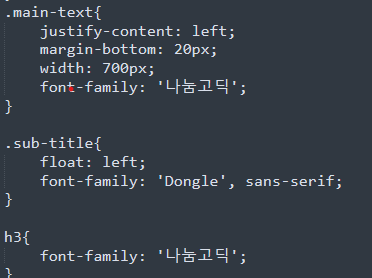
마지막 4번째 기사에 블로그 형식의 포스팅 CSS값 설정이다. 기사를 담은 큰 div 태그에 class=”spot-container”를 주었고 크기 및 가운데 정렬 방식을 설정했다.

Class = “spot”은 각각의 코스별 사진들인데 줄 단위로 출력되기 위해 display = “block”으로 설정했고 margin: 10px auto; 통한 가운데 정렬 width, height 값은 바꿔가면서 값을 설정했다.



4번째 기사의 본문 부분에 대한 CSS값 설정이다.

텍스트의 크기를 700px로 설정하였고 소제목과 테스트에 대해 justify-content 값을 left 설정하여 왼쪽으로 설명을 붙였다. 또한 아래 내용과 거리를 두기 위해 margin-bottom 값을 20px 설정하였고 font-family 값을 통해 각각의 글꼴 값들을 지정해 주었다.



1. **토의사항**

과제를 진행하면서 크게 총 3가지에 어려움에 직면했다.

1. 레이아웃의 부재

지금까지 교수님이 짜 주신 레이아웃을 토대로 디자인 적인 요소들만 추가하는 코딩을 해 왔던 나는 아무것도 없이 텅 빈 html 화면속에서 레이아웃을 잡아 나갈 때 정말 막막했다. 좋은 아이디어가 떠 오르지 않아 여러 사이트를 돌아다니면서 페이지 검사를 통해 이 페이지는 어떻게 구성되어 있나 찾아보기도 하고 강의 영상도 여러 번 반복해 보았지만 영감이 잘 떠오르지 않았다. 그러던 중 부모님이 예전에 하시던 싸이월드 사진을 옮겨 달라 요청하셨고 그렇게 싸이월드 미니홈피 느낌으로 레이아웃을 정하게 되었다. 레이아웃을 정한 후 안에 들어갈 내용들을 배치하면서 우여곡절 끝에 4개의 기사 중 첫 번째 기사를 완성했을 때 “나머지도 비슷한 형식으로 하면 되겠다”는 확신이 들었고 과제의 윤곽이 잡힌 것 같아 동굴 속에 한 줄기의 빛을 발견한 기분이었다.

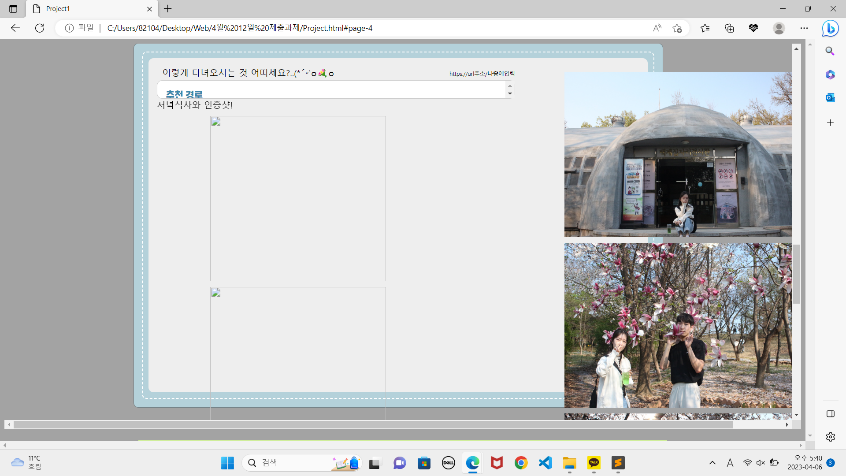
2. a 태그 이용한 페이지 이동 및 컨텐츠 정렬

네비게이션 바에 a 태그를 이용해 외부 html 연결은 많이 해 보았지만 같은 페이지 내 이동은 처음이었다. 어쩔 수 없이 구글링을 해야 했는데 마땅히 검색해 볼 키워드가 잘 생각나지 않아 <a> 태그에 관한 내용을 무작정 읽기 시작했고 마침내 한 페이지 내 <a> 태그로 이동하기 위해서는 <a name=””>이란 속성을 지정해주면 쉽게 해결할 수 있음을 알게 되었다. 찾아보면 몇 줄 안되어서 쉽게 알 수 있는 내용이었지만 검색할 키워드가 잘 떠올리지 않아 꽤나 고생했던 것 같다.

다음으로는 부모 div 안에 글, 이미지 등 컨텐츠를 넣었을 때 자꾸 부모 영역을 벗어났다. 이는 컨텐츠 정렬이라는 키워드가 금방 떠 올라 구글링을 통해 display: flex, direction 통해 수평, 수직 정렬을 할 수 있었고 flex을 변경하여 배치할 수 있었다. 이를 통해 자료를 찾을 때 키워드 검색의 중요성을 다시 한번 깨닫게 되었다.

3. div 태그의 중첩

하나의 html 안에 총 4개의 기사를 넣다 보니 수 없이 많은 div 태그들이 생겨나기 시작했고 div태그가 어디서 열려서 닫히는지 파악하는데 어려움이 있었다. 전 보다 좋은 아이디어가 떠 올라 코드를 바꿀 때 div태그를 잘못 지워 닫혀야 할 태그가 닫히지 않아 그림과 같이 페이지 레이아웃이 아예 엉망이 됐다. 또한 div태그가 너무 많아 어디를 잘못 건드려서 에러가 발생했는데 찾는데도 많은 고생을 했다. 이를 해결하기 하기 위해 코드를 최대한 단순화하고 들여쓰기를 통해 구분을 해보고 있으며 과제가 마무리될 때까지 계속 시도해 볼 예정이다.



추가적으로 과제를 하면서 아쉬웠던 점은 카메라로 사진을 찍다 보니 사진파일 하나의 용량이 1MB에서 많게는 10MB까지도 나왔다. 용량이 30MB 밖에 되지 않는 과제 제출함에 맞추어 올리기 위해 원본사진이 아닌 압축을 할 수밖에 없었고 압축한 사진들로 금방 30MB가 되어서 올림픽공원 가서 찍은 더 많은 아름다운 사진들을 넣지 못해 아쉬웠다.