PA1:

(웹프로그래밍 과제)

2024 봄 –

서울의 길을 걷다!

보고서

(제출일: 2024년 4월 12일)

202111248 김나영

레이아웃 도안

body

<div id=”out-line”>

<div id=”page-wrapper”>

<header id=”main-header”>

<div id=”title”>

<div id=”day-block”>

<nav>

<div id=”content”>

<div class=”tab-me”>

<div class=”tab-friend”>

<div class=”tab-street”>

<div class=”tab-inf”>

주요기능

1. 탭바 -> hover시 그림자, 커서 변경 / checked시 이미지 변경 후 content에 관련 내용 등장. (기본값 unchecked시 content 없음)

일단 네개의 주제에 각각 접근 가능한 탭바를 구성하기위해서 input태그에 대한 label을 텍스트와 이미지를 동시에 만들었습니다.

input태그가 checked 되면 이미지를 바꾸고 싶었는데 찾아보니 그것은 자바스크립트를 써서 src 속성을 바꾸는게 일반적이었습니다. 하지만 자바스크립트는 쓰면 안되므로 다른 방법을 사용하여 구현했습니다. 방법은 다음과 같습니다. Label로 선택된 이미지와 똑같은 위치에 checked 시 바뀔 이미지를 겹쳐서 배치합니다. 겹쳐서 이미지를 배치해야하기 때문에 position: absolute를 사용하였습니다. 또한 겹쳐진 이미지는 투명도를 0으로 설정했습니다. 하지만 이미지가 겹쳐져있어서 label을 클릭을 해도 checked가 되지 않는 문제가 발생했습니다. 따라서 겹쳐져있는 이미지를 평소엔 display:none을 했다가 checked시에만 display:block으로 바꾸는 방식으로 해결했습니다. 또한 input의 label이미지 hover시 filter:drop-shadow()라는 css속성을 사용하여 그림자가 생기도록 했고, 좀 자연스러운 동작을 위해 천천히 나오도록 transition: filter 0.2s ease-out라는 css속성,값을 사용했습니다. 커서를 변경을 하기위해선 cursor:pointer라는 css속성,값을 사용했습니다. 각 탭이 checked되었을 때만 content가 등장해야하므로 평소엔 <div id=”content”>가 display:none이다가 input이 checked되었을때만 각 input에 연결된 content의 탭class와 content의 내용이 display:block으로 등장하도록 했습니다.

1. 웹페이지 오른쪽 fixed된 home버튼

네개의 주제 탭이 선택된 후에는 웹페이지를 초기화하지 않는 이상, 아무것도 안 선택된 상태로 만들 수가 없었습니다. 따라서 페이지 오른쪽 중간에 고정된 home탭을 만들어 home버튼을 누르면 네개의 주제 모두 선택되지 않은 상태, 즉 content가 없는 상태로 만들도록 하였습니다. Home탭의 name속성을 네개의 주제탭의 name속성과 똑같이 하면 home탭이 어디에 위치하든 checked가 되면 네개의 주제탭은 선택되지 않은 상태가 될 것입니다. 주제 탭(input의 label)은 page-wrapper안의 header의 nav안에 위치해 있어고 home탭은 page-wrapper바깥쪽, out-line내부에 위치하도록 했습니다. 이때 position:fixed로 설정하고, top:60%, right:10%에 고정되도록 했습니다. Home탭은 웹페이지의 크기가 변해도 그 위치에 고정될 테니까 다른 영역에 침범하더라도 잘 보이도록 z-index를 5로 설정하여 다른 내용과 겹치더라도 맨 위에 위치하도록 했습니다.

1. 애니메이션으로 떨어지는 벚꽃잎

계속 비춰지는 헤더부분을 조금 더 예쁘게 꾸며보고 싶어서 애니메이션 속성을 추가로 찾아봐서 벚꽃 잎을 넣어봤습니다. 사실 짝꿍과 산책을 갔을 시기는 벚꽃이 피기 전이고 비까지 왔지만 봄의 산책이라는 컨셉에 맞도록 한번 꾸며보았습니다.

일단 html문서에는 <div class=”blossom1”><div> <div class=”blossom2”><div>. . .를 만들 꽃잎의 개수에 맞게 만들어줍니다. 그리고 css에는 .blossom1 선택자의 속성에서 꽃잎의 크기와 색깔, border-radius로 모양을 다듬어줍니다. Position을 absolute로 하여 처음 벚꽃잎이 시작할 위치를 정해주고 애니메이션 속성을 이용해서 떨어뜨릴 겁니다. 애니메이션 속성을 적용하기 전에 @keyframe을 이용하여 어떤식으로 떨어뜨릴건지 설정을 해주어야 합니다.

@keyframes fall1 {

  0% {

    opacity: 0;

    transform: translateX(100px) translateY(-100px) rotate(360deg); /\* 시작 위치와 각도 \*/

  }

  50% {

    opacity: 0.8;

  }

  100% {

    opacity: 0.2;

    transform: translateX(300px) translateY(400px) rotate(0deg); /\* 끝 위치와 각도 \*/

  }

.blossom1 {

  position: absolute;

  left: 30%;

  width: 25px; /\* 꽃잎의 너비 \*/

  height: 45px; /\* 꽃잎의 높이 \*/

  background: pink; /\* 꽃잎의 색상 \*/

  border-radius: 50%; /\* 꽃잎을 둥글게 \*/

  animation: fall1 7s cubic-bezier(0.25, 0.15, 0.81, 0.94) infinite 5s; /\* fall 애니메이션 적용 \*/

}

위와 같이 fall의 keyframe을 만들어 처음, 중간, 끝의 위치에서 어떻게 움직일 것인지를 설정해주었습니다. 이렇게 설정했으니 이제 아까 언급한 .blossom 선택자에 animation속성을 적용하여 animation속성값에는 어떤 keyframe을 쓸 건지, 몇초동안 떨어질건지, 어떻게 등장하고 사라질건지(ease, ease-in, ease-out, ease-in-out, cubic-bezier( ) ), 반복할건지말건지, 떨어지고 나서 몇초동안 delay를 할건지를 설정해줍니다. 저는 꽃잎5개를 만들었는데 그 5개를 다른 방식으로 무한히 떨어지도록 설정했습니다.

토의사항 (문제점, 새롭게 발견한 사실, 토의사항)

이렇게 웹페이지를 직접 만들어보며 수업시간에 배우지 않은 CSS의 새로운 속성,값들을 알게 되었습니다.

대표적으로 transition, rgba, :has(), flex를 유용하게 사용했고 많이 도움이 되었습니다.

Transition은 특정속성을 지정한대로 은근하게 동작하게 해서 좀더 예쁘고 자연스럽게 만들 수 있었습니다. 예를 들어 transition: color 0.2s ease-out을 사용해서 hover를 하거나 특정한 동작을 할 때 색깔을 스르륵 바꿔 더 예쁘게 동작하도록 했습니다.

Rgba는 색깔과 투명도를 한번에 설정할 수 있도록 하는 기능인데, 이 기능 덕에 웹페이지를 만들 때 겪었던 난관을 해결할 수 있었습니다. home탭의 라벨을 <p>태그 안에 넣어서 라벨이 hover가 되면 <p>태그의 배경을 보라색으로 변하게 하고 싶었는데 투명도도 조금 투명하게 하고싶었습니다. 스르륵 동작하기 위해 transition: background-color 0.2s ease-out을 할 때 색깔은 보라색깔로만 지정하고 투명도는 opacity속성을 따로 쓰면 투명도와 색깔을 함께 스르륵 동작하도록 할 수가 없었습니다. 이때 rgba를 사용해서 한번에 동작하도록 해결했습니다.

:has()는 부모를 선택할 때 사용하는 선택자입니다. css에서 후손(스페이스바), 자식(>),동위(+,~)선택자로 각각의 태그들에 접근할 수 있었습니다. 하지만 input이 체크된 상황에서 동작해야할 것들을 설정할 때 부모를 선택하기가 애매했습니다. 따라서

#main-header:has(#about-friend:checked) + #content .tab-friend

와 같이 :has()를 이용하여 괄호안에 있는 자식을 가진 부모를 선택할 수 가 있었습니다.

Flex는 display속성의 값인데, float : right또는 float : left을 함부로 하면 레이아웃이 깨져버리는 현상이 발생해서 부모속성에 반드시 overflow : hidden을 해줘야 하는 불편함을 해결할 수 있었습니다. 부모요소에 display : flex를 설정함으로써 justify-content, align-items, align-content, flex-direction, flex-wrap 등 다양한 속성을 이용하여 자식들을 원하는대로 정렬할 수 있어 편하게 사용했습니다.