포트폴리오 클론코딩

김영일

By 엘리, 2020. 7. 11 - 2020. 7. 21

나만의 포트폴리오 사이트를 만들어보자

어떻게 기획을 할까?

- 한눈에 보여 줄 수있는 포트폴리오를 목적으로.
- 싱글페이지와 최소한의 클릭을 할수있게 단순하게 만들자.
- 깔끔하지만 나의 개성을 살려보자.
- 개발의 대한 나의 열정 보여줄 수 있게
- 나의 기술 스택은?
- 프로젝트(카테고리, 우선순위)
- 링크(깃허브)

프론트엔드 페이지는 뭘 보여줄꺼지?

구성

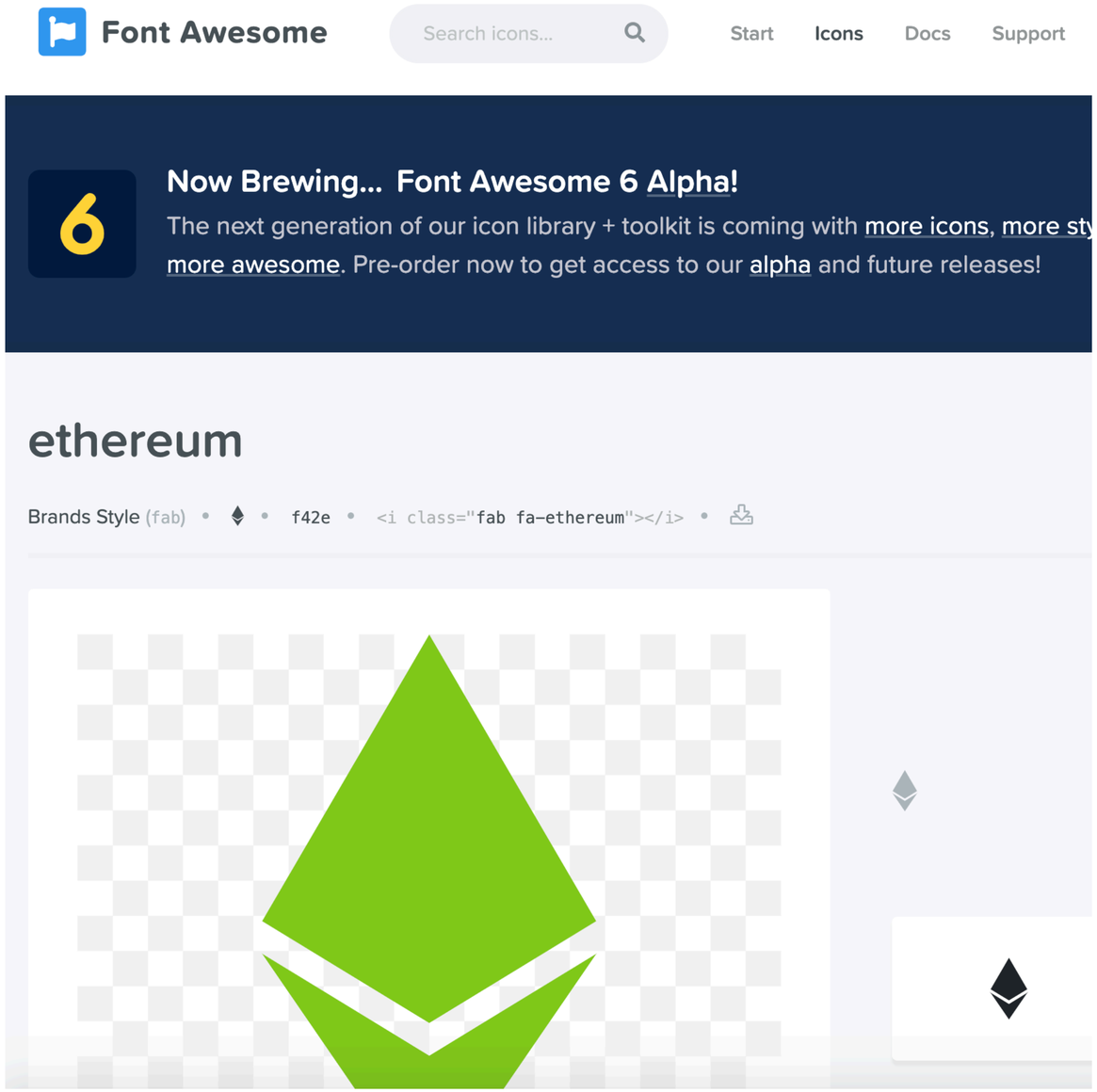
- 소개
  - 학교, 학원에서 어떤 공부를 했는지 혹은 열정
- 주요 스택
  - 프론트엔드,백엔드,모바일
- 경력
  -회사, 기간
- 프로젝트들
  - 뭔지? 주요기능은?
  - 깃허브 사이트
- 연락처
  -이메일,깃허브 등등

디자인하기

Font awesome

- 나의 로고를 만들어 보자.
  - 폰트어썸은 굉장히 편리한 라이브러리임 .필요한 아이콘에 클래스 네임만 가지고 와서 붙여넣기를 해주면 알아서 아이콘으로 만들어주었기 때문에 일일히 다운받고 관리할 필요가 없어졌음.

- 단점이라고 하기엔 뭐하지만 그만큼 폰트어썸의 의존도가 높아졌다는건 불안함.

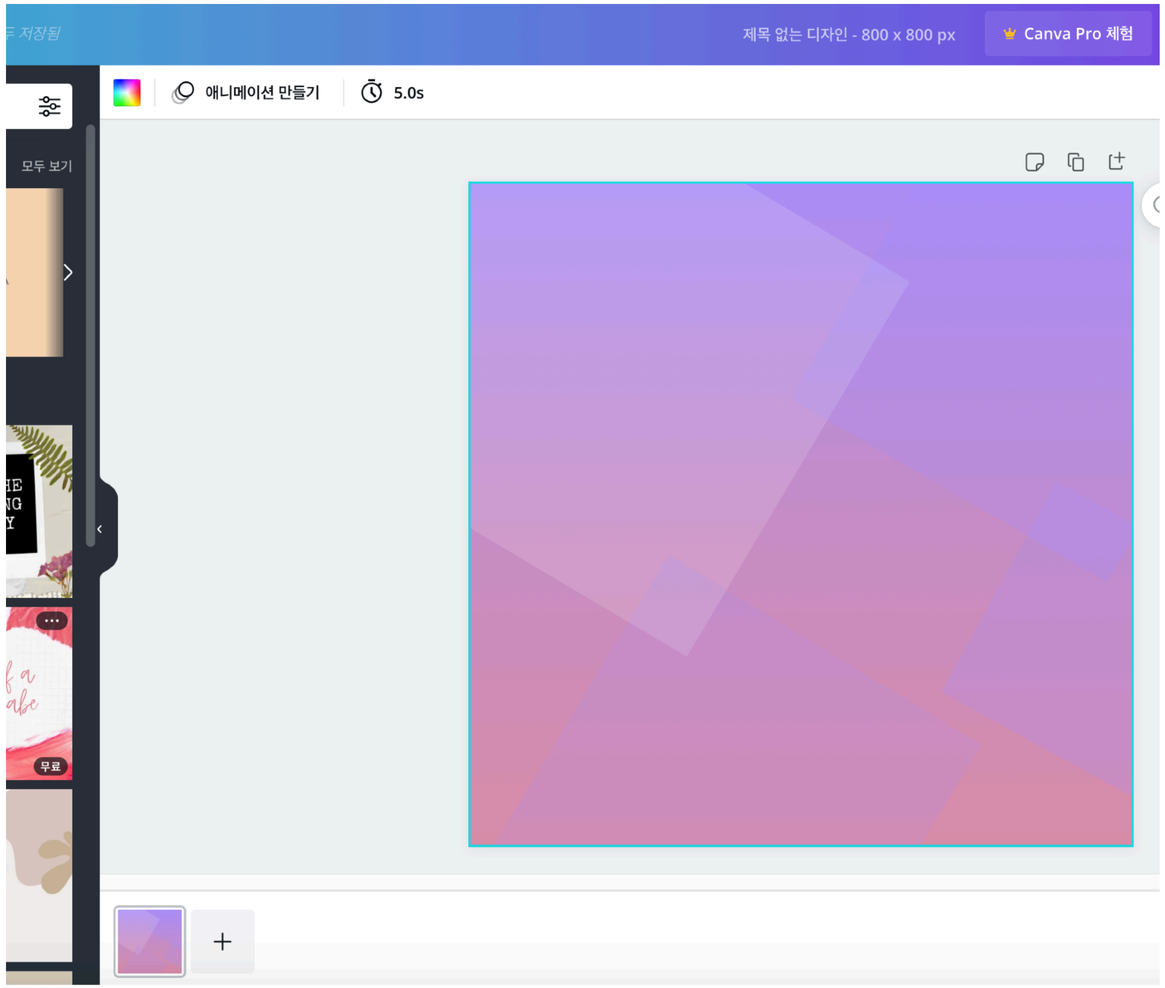


디자인하기

Canva

- 정말 좋은 홈페이지인거 같음.
  비록 부분 유료화여서 좋은 이미지는 무료로 사용 할 수 없지만 그것을 제외한 나머지는 내 입맛에 맞게 커스텀을 집적 할수있다는게 가장 좋음.
  나의 개성을 녹일수 있었음.

- 단점은 크게 없을정도로 굉장히 ui/ux 도 좋았다.

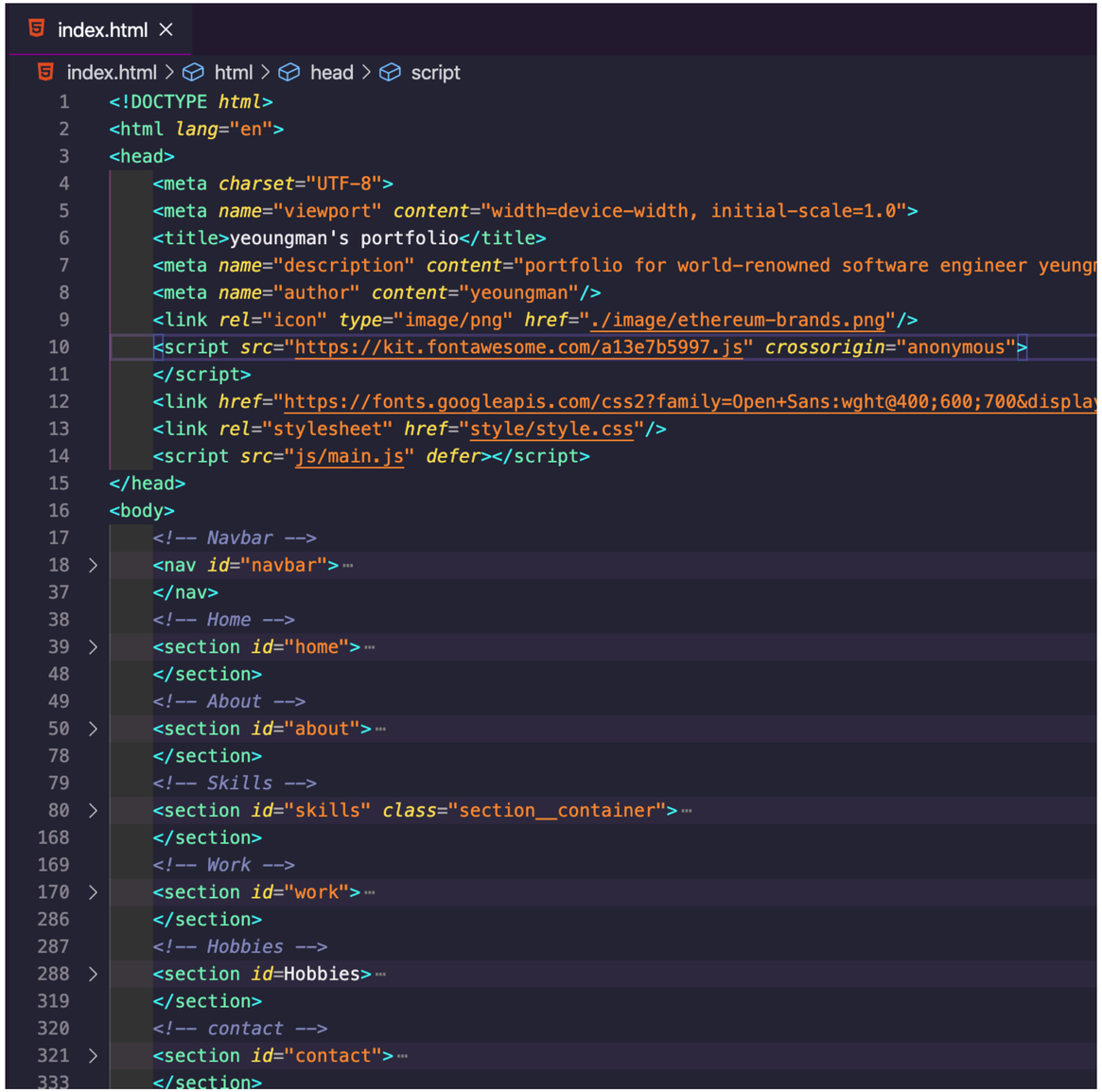


HTML 작성

Metadata

- 우선 HTML 안에 기본적으로 사용할 폰트 어썸과 구글 폰트를 적용하기 위해서 <script>태그 안에 CDN(콘텐츠 전송 네트워크)를 추가하도록 함.

- CDN을 제공받을려면 회원가입을 필수적으로 해야하기때문에 조금은 아쉬움.



HTML

What is the BEM??

Block Element modifier

- HTML 클래스 이름을 작성할때 개발자간의 암묵적인 규칙을 정하는 방법중 하나.
- 블럭과 엘리멘트, 그리고 모디파이어로 나누어서 이름을 작성
- 오직 클래스네임만 할수 있음.
- 예)
- .cards
  .cards__card
  .cards__card__title
  .cards__card__discription

HTML 작성

- 가독성을 위해 주석문을 이용하여 섹션을 분리
- 꼭 태그는 의미있게 씀.
- CSS 를 제외한 실제 필요한 부분만 빠르게 추가하고 다음 섹션으로 넘어감.
- 팁:기본적으로 디자인 구성하기 쉽게 노트에 수기로 대강적인 구도를 그려놓고 해놓으면 이해하기더 쉬움.

```
html
YPE html>
lang="en">
…
>

-- Navbar -->
av id="navbar">
  <div class="navbar__logo">
      <i class="fab fa-ethereum"></i>
      <a href="#">yeoung</a>
  </div>
  <div class="navbar__menu">
      <ul class="navbar__menu">
          <li class="navbar__menu__item active" data-link="#home">Home</
          <li class="navbar__menu__item" data-link="#about">About</li>
          <li class="navbar__menu__item" data-link="#skills">Skills</li>
          <li class="navbar__menu__item" data-link="#work">My work</li>
          <li class="navbar__menu__item" data-link="#Hobbies">Hobbies</l
          <li class="navbar__menu__item" data-link="#contact">Contact</l
      </ul>
  </div>
  <!-- Toggle button -->
  <button class="navbar__toggle-btn">
      <i class="fas fa-dice-d20"></i>
  </button>
nav>
-- Home -->
```

CSS

CSS

Global set up &typography

- 디자인의 통일성을 주기 위해 변수처럼 CSS props 를 셋업함.

- 일관성을 같기위해 폰트의 사이즈 굵기도 글로벌로 통일함

- Property를 일일이 다 외울필요 없음. 그냥 필요할때 찾아보면서 익숙해지면 됨.

```css
/* Global */

:root{
    /* Color */
  --color-right-pupple: #ddc0eb;
  --color-pupple: #7732a8;

  --color-white: #ffffff;

  --color-light-white: #eeeeee;

  --color-dark-white: #bdbdbd;

  --color-pink: #fe918d;

  --color-dark-pink: #ff6863;

  --color-dark-grey: #4d4d4d;

  --color-grey: #616161;

  --color-light-grey: #7c7979;

  --color-blue: #73aace;

  --color-yellow: #fff7d1;

  --color-orange: #feb546;

  --color-black: #000000;

  /* Font size */

  --font-large: 48px;

  --font-medium: 28px;

  --font-regular: 18px;

  --font-small: 16px;

  --font-micro: 14px;

  /* Font weight */

  --weight-bold: 700;
```

CSS

반응형 스크린 설정

- 모바일로도 볼수 있게 반응형으로 제작
- Max-width를 설정했는데. 갤럭시 노트같은 큰 사이즈의 기종은 조금 규격이 안맞는 문제가 생김. 구조 단계에서 px단위를 사용했기에 조금 보수하기 어려움.
- 다음부턴 rem,em을 사용해보자.

```
@media screen and (max-width:768px){
  .navbar__toggle-btn{
    display: block;
  }

  #navbar{
    padding: 16px 16px 16px 0px;
    flex-direction: column;
    align-items: flex-start;
    background-color: var(--color-pupple);
  }

  .navbar__menu ul{
    margin:0px;
  }

  .navbar__menu{
    flex-direction: column;
    text-align: center;
    width:100%;
    display:block;
  }

  .navbar__menu.open{
    display:none;
  }

  .about__majors{
    flex-direction: column;
  }

  .major {
    margin-bottom: 35px;
  }

  .skilltable {
    flex-direction: column;
  }

  .project {
```

Javascript

Java script

- 동적인 요소를 구현하는 .js 파일안에 작성함.
- 자바스크립트를 동적으로 사용하기 이전에 CSS로 충분히 동적인 요소를 간접적으로 만들었기 때문에 많은 기능이 필요하지 않았음.

```
"use strict";

// Make navbar transparent when it is on the top
const navbar = document.querySelector("#navbar");
const navbarHeight = navbar.getBoundingClientRect().height;
document.addEventListener("scroll", function(){
    // console.log(window.scrollY);
    // console.log(`navbarHeight: ${navbarHeight}`);
    if (window.scrollY > navbarHeight) {
        navbar.classList.add("navbar--dark");
    } else {
        navbar.classList.remove("navbar--dark");
    }
});

// Arrow Button to Scroll Up
const arrowUpBtn = document.querySelector(".arrow-up");

document.addEventListener("scroll", function(){
    if(window.scrollY > navbarHeight) {
        arrowUpBtn.classList.add("visible");
    } else {
        arrowUpBtn.classList.remove("visible");
    }
});

// project
const projectBtnContainer = document.querySelector(".work__categories");
const projectContainer = document.querySelector(".work__projects");
const projects = document.querySelectorAll(".project");

projectBtnContainer.addEventListener("click", function(e){
    const filter = e.target.dataset.filter || e.target.parentNode.dataset.filter;
    if(filter == null) {
        return;
    }

    // Cycle effects with selected buttons

    const active = document.querySelector(".categories__btn.selected");
    active.classList.remove("selected");
    const target = e.target.nodeName === "BUTTON" ? e.target :e.target.parentNode;
    target.classList.add("selected");
```

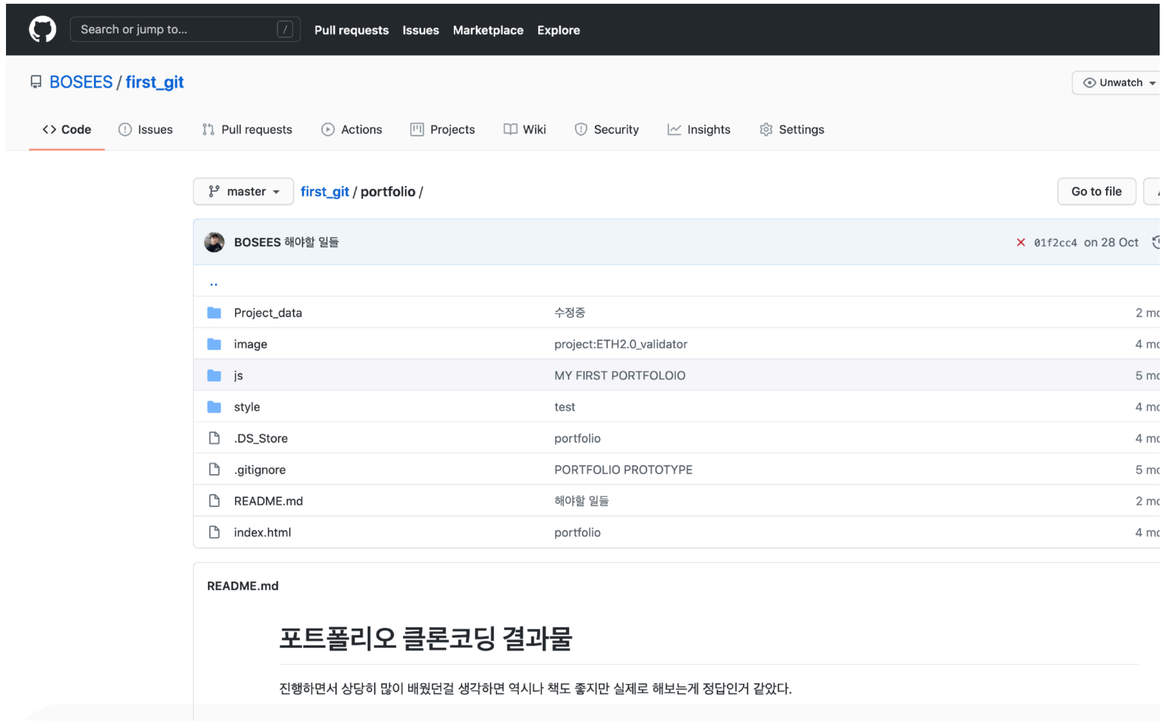
Git & Github

Git & Github

도메인 개설

- 레포지토리 생성 및 git push.
- 가비아 도메인 호스팅 업체에서 도메인 구매
- CNAME을 설정하여 연동

Https:bosees.site



마치며

포트폴리오 클론코딩 결과물

진행하면서 상당히 많이 배웠던걸 생각하면 역시나 책도 좋지만 실제로 해보는게 정답인거 같았다.

진행하면서 가장 기억남는것들

- BEM
- 반응형 웹
- 미리 구상한 섹션을 설계하고 구현하는 법
- css는 외우지말고 찾자...
- flex box
- 주석문 쓰기

항상 생각하는거지만 이름짓는게 가장 난해한거같다. 개발자들 사이에서 편리하게 이름을 지울수 있는 암묵적인 룰이 있다면 사용하는게 나도 편하고 다른사람도 편하니가 최대한 지키자. 작은차이가 큰차이를 만든다. 그리고 꼭 주석문을 달아 내 생각을 정리 및 요약 하자.