[스마트웹&콘텐츠개발]

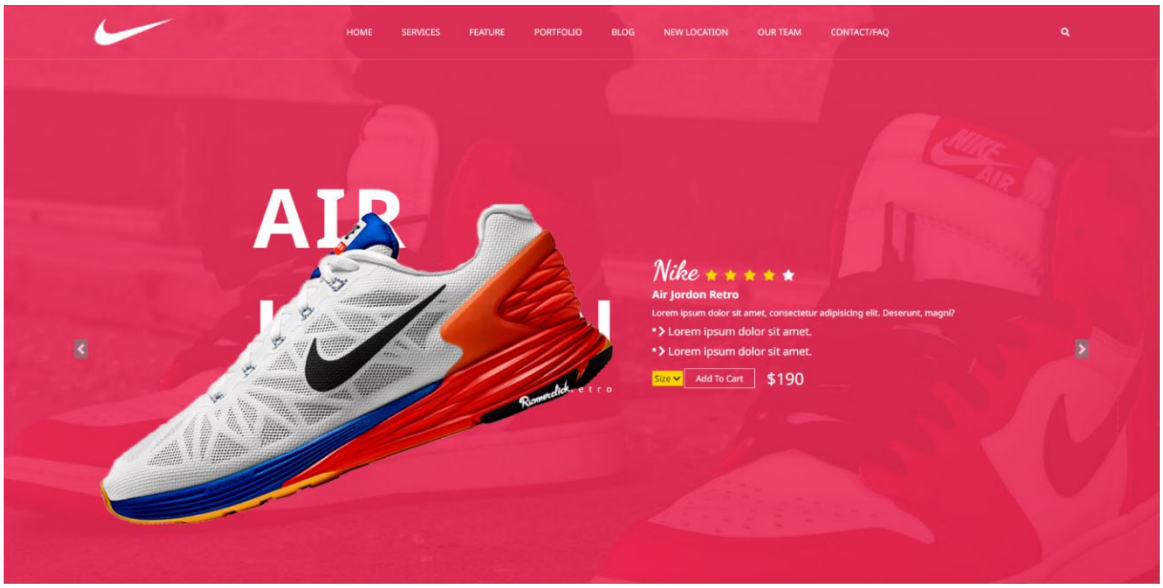
프론트엔드 개발자(자바스크립트, React)

양성과정 22-2

능력단위평가

스마트문화앱구현

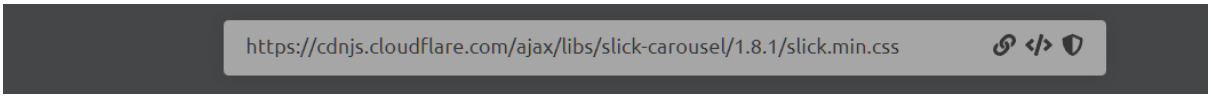
장현정



① Slick 연결하기

(편의상 slick cdn 라이브러리 출처 이미지를 각각의 설명에 맞게 잘게 놓았으며 설명 아래 출처 페이지 전체 사진을 배치해 놓았음.)

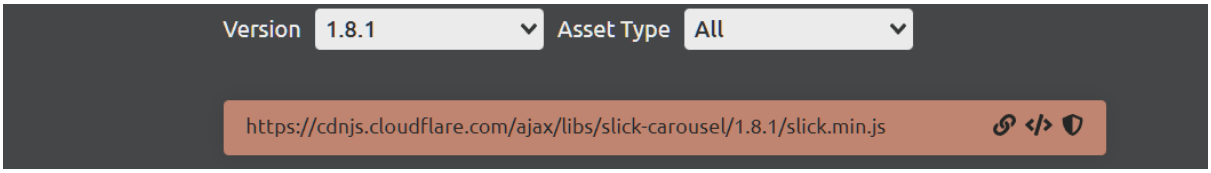
Html의 Head 안에 Slick cdn의 slick.min.css를 연결한다. 이 때 유의해야 할 점은 Slick.min.css가 style.css를 연결한 것보다 반드시 위에 입력되어야 한다는 것이다.




```
<link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/slick-
carousel/1.8.1/slick.min.css"
    />
    <link rel="stylesheet" href="css/style.css" />
```

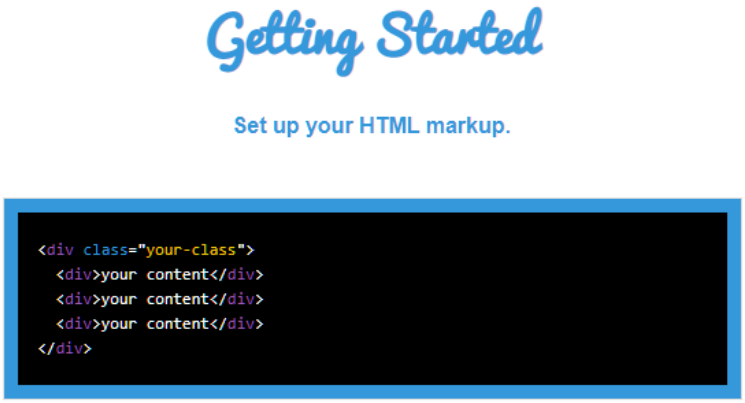
그 다음 body 안에 slick.min.js를 연결해 준다. 이 때 유의해야 할 점은 slick.min.js를 입력한 것보다 위에 jquery(jQuery 1.x)를 연결해야 한다는 것이다.

그리고 위 두 개의 <script></script> 사이에 내용이 입력되면 코드 전체가 먹통이 됨으로 이 점도 유의해야 한다.



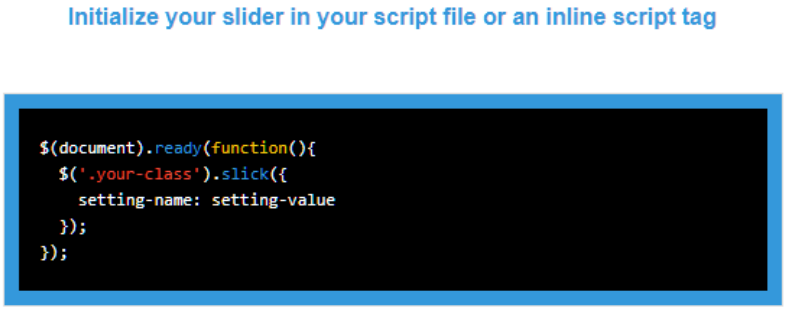

```
<script src="https://code.jquery.com/jquery-
1.12.4.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/slick-
carousel/1.8.1/slick.min.js"></script>
```

위의 연결을 모두 마친 후 body 안의 원하는 위치에 아래의 슬라이드 div를 입력한다.

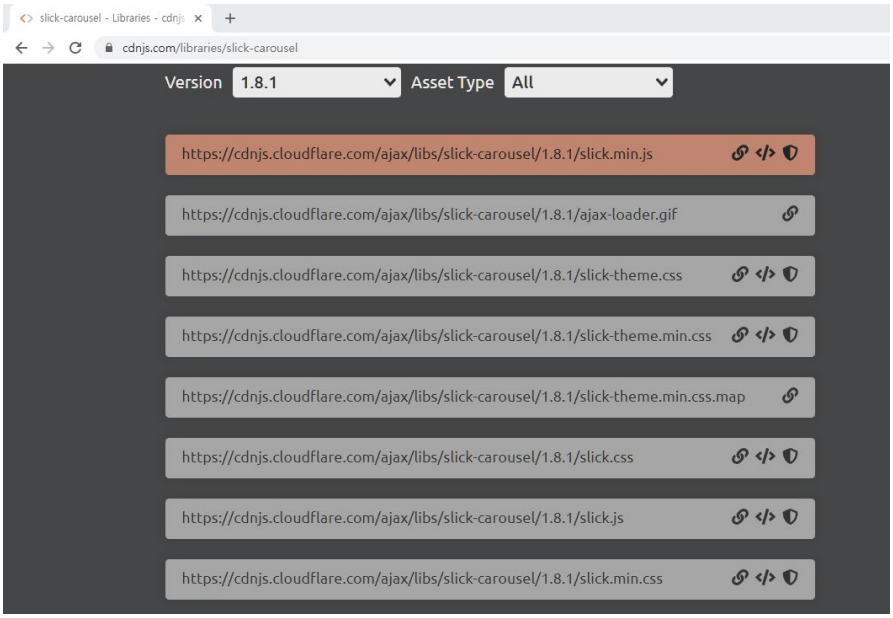


그리고 body 안의 code jquery와 slick cdn을 연결한 부분의 아래에 script tag를 따로 만들어주어 위와 같이 slick 작동을 위한 div를 입력한다(위 사진 속 div). 이 때 유의할 점은 슬라이드 div와 script tag의 class명은 your-class가 아닌 다른 것으로 변경이 가능하지만 이 두 가지는 반드시 일치해야 slick이 제대로 작동한다는 것이다.

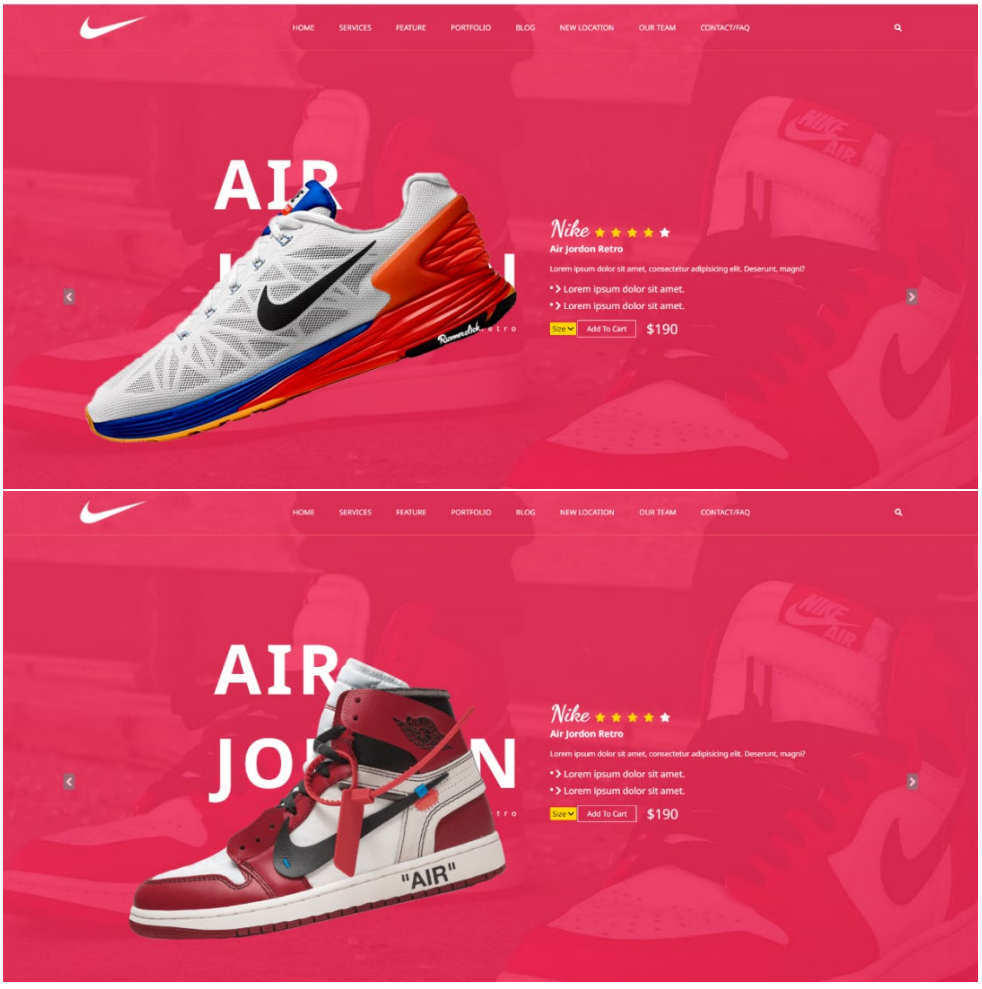
Slick의 원하는 기능을 //setting-name: setting-value를 지운 후 그 부분에 넣어서 사용하면 되고 //setting-name: setting-value 부분은 주석처리하거나 지워야 슬릭이 제대로 생성된다.

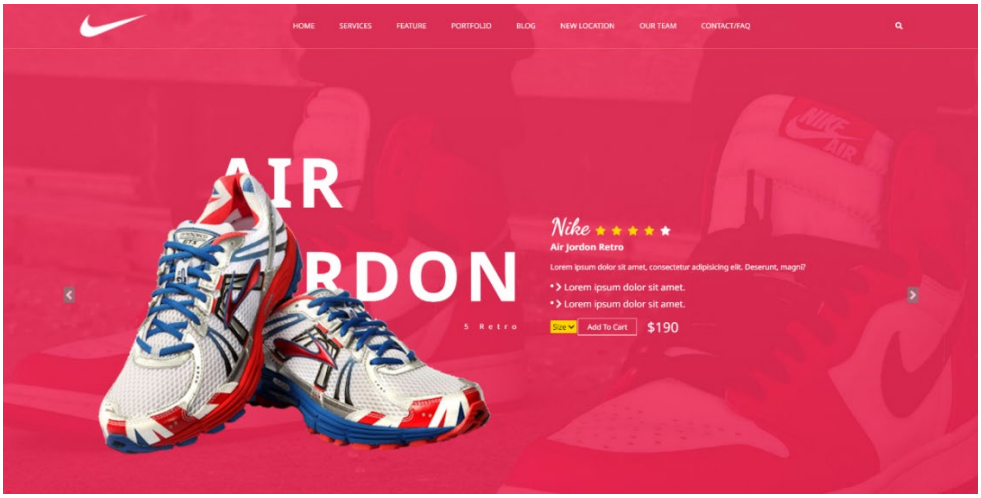



```
<script src="https://code.jquery.com/jquery-1.12.4.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/slick-
carousel/1.8.1/slick.min.js"></script>
    <script>
      $(document).ready(function () {
        $(".slider").slick({
          // setting-name: setting-value
        });
      });
    </script>
```



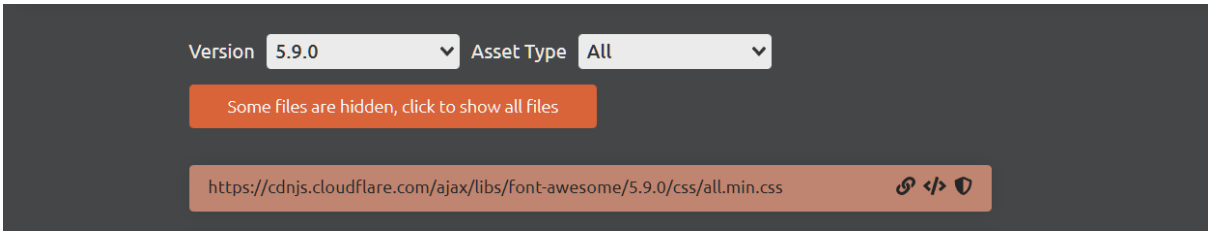
Slick을 이용한 각 슬라이드의 화면





② Fontawesome 연결하기

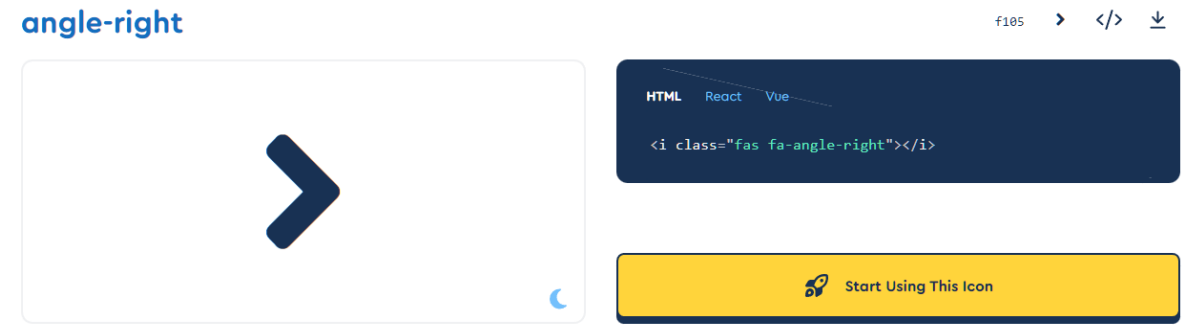
Fontawesome cdn을 html의 head 안에 연결한다. 이 때 위치는 상관 없으나 편의상 style.css 연결한 곳 아래에 배치해 놓았다. 유의할 점은 fontawesome 유료버전을 구매하지 않은 경우 무료 버전 중 가장 최신 버전인 fontawesome 5버전을 연결하여야 사용한 icon이 먹통이 되는 것을 막을 수 있다.(사용하는 icon도 5버전의 무료 icon을 사용해 주어야 한다.)




```
<link rel="stylesheet" href="css/style.css" />
    <link
      rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.9.0/css/all.min.css"
    />
    <title>Document</title>
```

Fontawesome5 가 연결되었다면 css 에서 icon 을 사용할 수 있게 되는데 fontawesome 홈페이지의 개발자 모드에 있는 content, font-family, font-weight 의 값을 모두 적어 주어야 하며 font-family 부분에 pro 라고 적힌 부분의 free 라고 고쳐주어야 icon 이 화면에 제대로 도출된다.

```
content: "\f105";
  font-family: "Font Awesome 5 Free";
  font-weight: 900;
```

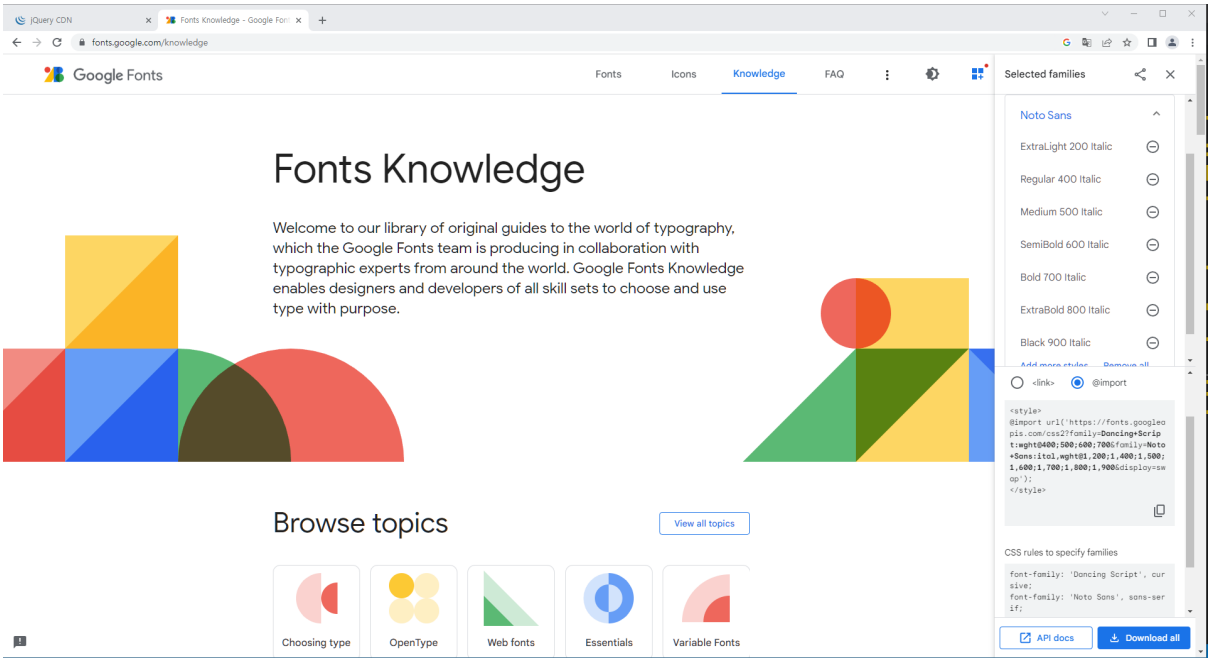


③ Google font 연결하기

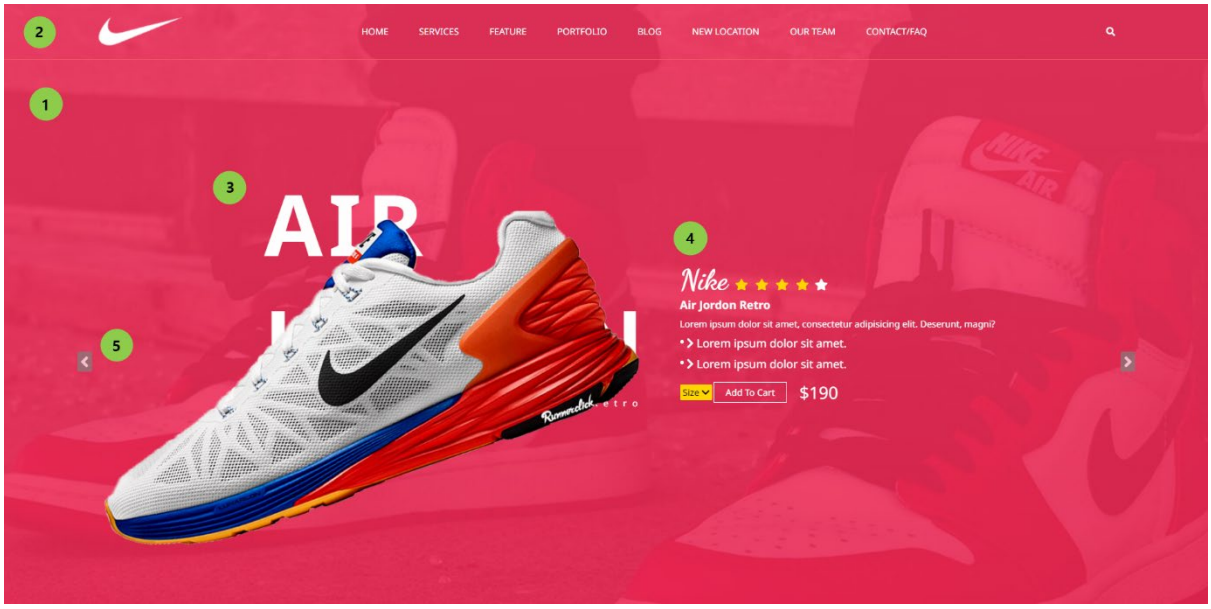
Style.css에 googlefont를 연결한다. 사용할 font-family를 따로 정리해서 주석처리해 놓으면 사용하기 편리하다. 이 때 유의할 점은 연결을 @가 아닌 다른 부분의 아래 부분에 하면 google폰트가 적용되지 않으니 반드시 css의 제일 윗 부분에 입력해야 한다는 것이다.

```
@charset "UTF-8";
@import
url("https://fonts.googleapis.com/css2?family=Dancing+Script:wght@400;500;600;700&family=Noto+Sans:wght@100;300;400;500;600;700;800;900&display=swap");

/* font-family: 'Dancing Script', cursive;
font-family: 'Noto Sans', sans-serif; */
* {
  margin: 0;
  padding: 0;
}
```



④ 코드 구현 설명



1

*{margin:0; padding:0;} - 화면 전체에 빈 공간이 없도록 했다.

li{list-style: none;} - li 태그 앞에 동그라미가 생기는 것을 방지했다.

a{text-decoration: none; color: #fff;} - a태그의 기본값은 글자 밑에 밑줄이 있는 것인데 text-decoration: none;을 해서 밑줄이 생기는 것을 방지했고 a태그의 글자 색깔을 #fff로 통일했다.

body에 다음과 같은 값을 주어 body 전체에 적용되도록 했다.

Body{

font-family: "Noto Sans", sans-serif; - body의 기본 폰트를 Noto Sans로 설정했다.

height: 100vh; - body의 높이를 100포로 설정해 viewport의 100%를 높이로 사용하도록 설정했다.

color: #fff; - body내의 전체 글자색을 #fff로 설정했다. 앞선 a에 설정된 #fff와 별도로 설정해주어야 body 전체의 글자색도 #fff로 설정된다.

background: rgba(190, 32, 238, 0.2) url(../img/bg-main.jpg) no-repeat;

-rgba에 190, 32, 238의 색을 주어 배경 색상을 190, 32, 238으로 설정했고 투명도를 0.2로 해서 사진을 아예 덮어버리진 않게 했다. url(../img/bg-main.jpg)을 연결해서 body의 배경에 사진을 입력했고 no-repeat;를 해주어 사진이 반복되지 않고 하나만 도출되게 했다.

background-size: cover; - background가 화면 전체를 cover하도록 설정했다.

background-blend-mode: color-dodge; } - 배경 이미지와 배경 색상의 섞임 형태를 color-dodge로 해서 주어진 이미지를 다채롭게 꾸며보았다.

.container { max-width: 1800px; margin: 0 auto; } – max-width를 1800px로 설정해 화면을 줄여도 가로로 스크롤바가 생기지 않게 했다. max-width가 아닌 width를 1800px로 하면 화면이 1800px 이하로 줄었을 때 옆으로 스크롤바가 생긴다. margin: 0 auto;을 해주어 위쪽 margin이 0이 되게 했고 양옆으로는 가운데정렬 되게 했다.

.en_dan { font-family: "Dancing Script", cursive; } – class명에 en_dan을 추가할 경우 font가 Dancing Script가 되도록 설정했다.

2

header { border-bottom: 1px solid #f06060; height: 100px;} – header 전체 높이를 100px로 하고 header 아래에 1px의 #f06060 색깔의 실선이 생기게 했다.

header .container {

height: 100%; - header 안의 container의 높이가 부모의 높이의 100%를 가지게 했다.

display: flex; - display에 flex를 주어 가로로 정렬되게 했다. Flex-direction의 기본값이 row(가로)로 설정되어 있기에 그 부분은 따로 설정하지 않았다.

justify-content: space-between; - 가로로 정렬된 컨텐츠들의 배열을 : space-between으로 주어 컨텐츠들이 서로 간격을 두고 퍼지게 만들어 주었다.

align-items: center; - 아이템들의 세로 배열을 center로 설정해서 아이템 각각의 가로의 중앙이

container의 가로의 중앙과 맞게 배치하였다.

header .container .logo { width: 150px; }, header .container .logo a img { width: 100%; } – logo가 위치하는 부분의 너비를 100px을 주었고 이미지의 너비를 100%로 주어 이미지가 logo 부분의 100%를 차지하게 만들었다.

header .container .menu_wrap ul { width: 1000px; display: flex; justify-content: space-between; } - menu_wrap의 ul 너비를 1000px로 주어 display가 flex 될 수 있게 했다. 너비가 주어지지 않으면 flex가 적용되지 않는다. justify-content: space-between; 값을 주어 ul의 각 li들이 서로 간격을 두어 배치될 수 있게 했다.

3 4 공통

.main { margin-top: 5%; } – 3, 4 전체를 잡고 있는 main 부분의 위쪽으로 5%의 공간을 둔다. 이때 5%는 그 부모인 wrap의 5%를 말한다.

.main .slick-slide { display: flex; position: relative; } – main의 left_area와 right_area가 flex 될 수 있게 해준다. position: relative;는 아래의 .left_area .nike_tit와 .main .slick-arrow의 기준이 된다.

3

.left_area { width: 50%; height: 80vh; } - .left_area부분의 너비가 그 부모의 50%가 되게 하고 높이는 viewport의 80%가 되게 한다. 부모의 너비가 설정되지 않은 상태라면 계속해서 상위로 올라가 너비값이 설정된 부분의 50%를 차지하게 된다. 이 경우 .container의 너비값인 1800px의 50%를 차지하게 된다.

.left_area .nike_img { width: 120%; padding-top: 160px; - 신발 이미지를 넣을 공간의 너비를 그 부모의 120%를 차지하게 해서 사진을 크게 키워줬다. Padding-top을 160px 주어서 신발 이미지를 아래쪽으로 내려오게 했다.

opacity: 0; } – slick이 움직일 때 첫 화면의 이미지가 나타났다가 실행되는 것을 막기 위해 이미지의 불투명도를 0으로 설정했다.

.left_area .nike_img img { width: 100%; } – 신발 이미지의 너비를 그 부모의 너비의 100%를 차지하게 했다.

.left_area .nike_tit { position: absolute; top: 80px; left: 295px; - 상위의 main .slick-slide을 기준으로 잡은 상태로 position을 absolute 시켜 주었고 위로부터 80px, 왼쪽으로부터 295px 떨어지게 배치하였다.

z-index: -10; } – z-index의 값이 클수록 컨텐츠의 위쪽에 배치되어 그 값이 작은 컨텐츠들을 가리게 되는데 이 경우에는 그 값을 -10으로 주어 z-index의 기본값을 가진 신발 이미지 밑에 텍스트

가 오게 만들었다.

.left_area .nike_tit h2 { font-size: 150px; letter-spacing: 10px; } – font-size를 150px로 해서 이미지 밑에 텍스트가 깔렸음에도 눈에 들어올 수 있게 해주었고 letter-spacing: 10px;을 주어 자간을 10px로 지정해 그 너비를 넓혀 주었다.

.left_area .nike_tit p { position: absolute; letter-spacing: 10px; right: 10px; } – 상위의 .left_area .nike_tit에 absolute 된 position을 기준으로 잡고 position을 absolute 시켜주었다. right: 10px을 주어 글씨가 오른쪽으로 정렬되게 해주었고 letter-spacing: 10px;을 주어 자간을 10px로 지정해 그 너비를 넓혀 주었다.

4

.right_area { width: 50%; height: 80vh; - .right_area 부분의 너비가 그 부모의 50%가 되게 하고 높이는 viewport의 80%가 되게 한다. 부모의 너비가 설정되지 않은 상태라면 계속해서 상위로 올라가 너비값이 설정된 부분의 50%를 차지하게 된다. 이 경우 .container의 너비값인 1800px의 50%를 차지하게 된다.

padding-top: 14%; padding-left: 130px; - 패딩을 위로 14%, 왼쪽으로 130px주어서 신발 이미지와 적절한 균형을 이루게 배치하였다.

box-sizing: border-box; - 배치해 놓은 영역들이 무너지지 않게 box-sizing: border-box;}를 입력해 주었다.

.right_area .star_rate h2 { font-size: 50px; font-weight: bold; opacity: 0; } – font-size를 50px로 하고 그 두께를 bold로 설정했다. opacity: 0;을 설정하여 slick 구동 시 화면에 미리 표시되었다가 작동되는 것을 방지했다.

.right_area .star_rate h2 i { font-size: 20px; opacity: 0; } - – font-size를 20px로 했고 opacity: 0;을 설정하여 slick 구동 시 화면에 미리 표시되었다가 작동되는 것을 방지했다.

.right_area .star_rate p { font-size: 20px; font-weight: bold; opacity: 0; } - – font-size를 20px로 하고 그 두께를 bold로 설정했다. opacity: 0;을 설정하여 slick 구동 시 화면에 미리 표시되었다가 작동되는 것을 방지했다.

.right_area .desc1 { margin: 20px 0; opacity: 0; } – margin을 위아래로 20px 양옆으로 0을 주어 위아래 텍스트들과의 공간을 주어 구분되게 했다. opacity: 0;을 설정하여 slick 구동 시 화면에 미리 표시되었다가 작동되는 것을 방지했다.

.right_area .desc2 ul li { font-size: 20px; margin-bottom: 10px; - font-size를 20px로 하고 아래쪽에 10px의 공간을 주어 그 다음 컨텐츠들과 구분되게 했다.

position: relative; padding-left: 12px; - 이후에 올 .right_area .desc2 ul li::before의 기준점을 잡아

주기 위해 position: relative;를 주었고 padding-left: 12px;를 해서 .right_area .desc2 ul li::before

의 동그라미들과의 공간을 주었다. 이 때 유의할 점은 :before와 li의 텍스트 사이에 공간을 주려면 :before가 아닌 li에 padding-left를 주어야 한다는 것이다.

opacity: 0; } - opacity: 0;을 설정하여 slick 구동 시 화면에 미리 표시되었다가 작동되는 것을 방지했다.

.right_area .desc2 ul li::before {

content: ""; - :before에는 반드시 content:"";를 써주어야 한다.

position: absolute; left: 0; top: 8px; - .right_area .desc2 ul li를 기준으로 position을 absolute 시켜주었고 왼쪽으로부터 0px, 위로부터 8px; 떨어진 곳에 위치시켰다.

width: 7px; height: 7px; border-radius: 50%; background-color: #fff; } – 너비와 높이가 7px인 #fff 색깔의 점을 만들었다. border-radius을 50%; 으로 지정할 경우 사방 모서리가 깎여 점처럼 만들 수 있다.

.right_area .select { display: flex; width: 300px; - 너비값을 300px로 하고 display를 flex 시켜서 컨텐츠를 가로로 정렬시켰다. 이 때 width 값을 지정하지 않으면 flex되지 않는다.

.justify-content: space-between; - 가로로 정렬된 컨텐츠들의 배열을 : space-between으로 주어 컨텐츠들이 서로 간격을 두고 퍼지게 만들어 주었다.

align-items: center;} - 아이템들의 세로 배열을 center로 설정해서 아이템 각각의 가로의 중앙이 container의 가로의 중앙과 맞게 배치하였다.

.right_area .select .size { padding: 3px 5px; background-color: gold; border-radius: 2px; opacity: 0;} – 텍스트 주위에 padding을 위아래 3px, 좌우 5px씩 주었고, 배경색을 gold로 해서 눈에 띄게 하였다. Border-radius를 2px주어 끝을 덜 뾰족하게 해주었고, opacity: 0;을 설정하여 slick 구동 시 화면에 미리 표시되었다가 작동되는 것을 방지했다.

.right_area .select .size a, .right_area .select .size a i { color: #000; } – 텍스트의 색깔을 #000 으로 지정해 주었다.

.right_area .select .cart { padding: 7px 20px; border: 1px solid #fff; border-radius: 1px; opacity: 0; }

- 텍스트 주위에 padding을 위아래 7px, 좌우 20px씩 주었고, 1px의 #fff의 실선을 border로 설정해 눈에 띄게 하였다. opacity: 0;을 설정하여 slick 구동 시 화면에 미리 표시되었다가 작동되는 것을 방지했다.

.right_area .select .price { font-size: 30px; padding: 7px 20px; opacity: 0; } – 글씨 크기를 30px로 했고, 텍스트 주위에 padding을 위아래 7px, 좌우 20px씩 주었다. opacity: 0;을 설정하여 slick 구동

시 화면에 미리 표시되었다가 작동되는 것을 방지했다.

5

.main .slick-arrow { position: absolute; left: -2%; top: 50%; - slcik-slide를 기준으로 해서 absolute 된다. 좌측으로부터 -2%, 위로부터 50%인 지점에 위치시킨다.

transform: translateY(-50%); - 위에서 위로부터 50%인 지점에 위치시킬 때 slick-arrow 본체의 제일 윗 부분을 기준으로 한 것이기 때문에 Y축을 기준으로 -50%를 해주어 본체의 길이 50%만큼 위로 올려준다.

z-index: 10; - 다른 컨텐츠 아래로 가려지지 않게 해주었다.

font-size: 0; visibility: hidden; - font-size를 0으로 하고 visibility: hidden;를 해서 이후에 입력할 fontawesome의icon을 제외한 다른 부분은 화면에서 보이지 않게 했다.

.main .slick-arrow.slick-prev:after {

content: "₩f104"; font-family: "Font Awesome 5 Free"; font-weight: 900; - fontawesome5 버전의 ₩f104버튼을 가져왔다. "; font-family와 font-weight는 반드시 함께 넣어주어야 하며 font-family: "Font Awesome 5 Pro";로 된 것을 font-family: "Font Awesome 5 Free";로 고쳐주어야 화면에 제대로 도출된다.

visibility: visible; - .main .slick-arrow에서 visibility: hidden;이라고 설정되었기 때문에 따로 보이게 바꾸어 주었다.

font-size: 30px; color: #fff; background-color: gray; opacity: 0.7; padding: 3px 5px; border-radius: 3px; } – 크기 30px, 색깔 #fff, 배경색 gray, 불투명도 0.7을 속성값으로 주었으며, padding은 상하 3px, 좌우 5px을 주었고, border-radius는 3px을 주어 배경의 모서리를 다듬어 주었다.

.main .slick-arrow.slick-next { left: auto; right: -2%; } – left: auto;를 해주면 컨텐츠가 오른쪽으로 정렬되고 나머지 왼쪽 공간은 다른 컨텐츠에 활용될 수 있게 해준다. 오른쪽으로부터 -2%의 지점에 위치시켰다.

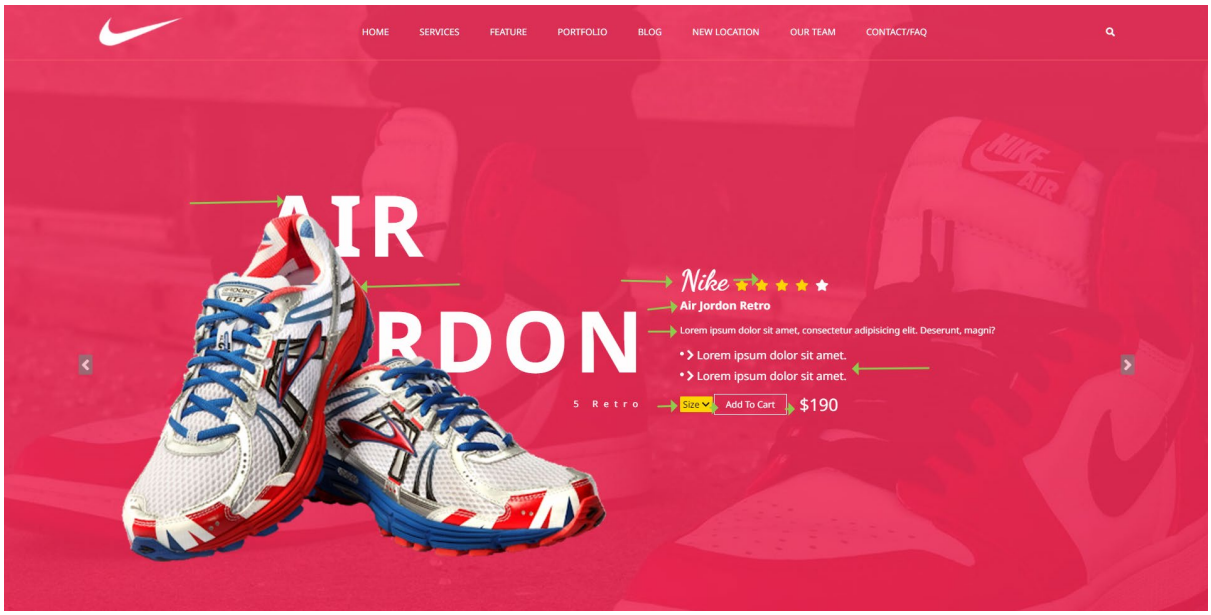
.main .slick-arrow.slick-next:after {

content: "₩f105"; font-family: "Font Awesome 5 Free"; font-weight: 900; - fontawesome5 버전의 ₩f105버튼을 가져왔다. "; font-family와 font-weight는 반드시 함께 넣어주어야 하며 font-family: "Font Awesome 5 Pro";로 된 것을 font-family: "Font Awesome 5 Free";로 고쳐주어야 화면에 제대로 도출된다.

visibility: visible; - .main .slick-arrow에서 visibility: hidden;이라고 설정되었기 때문에 따로 보이게 바꾸어 주었다.

font-size: 30px; color: #fff; background-color: gray; opacity: 0.7; padding: 3px 5px; border-radius: 3px; } – 크기 30px, 색깔 #fff, 배경색 gray, 불투명도 0.7을 속성값으로 주었으며, padding은 상하 3px, 좌우 5px을 주었고, border-radius는 3px을 주어 배경의 모서리를 다듬어 주었다.

⑤ 애니메이션 구현



Keyframes 설정

@keyframes fadeInR {

0% { transform: translateX(150%); opacity: 0; }

100% { transform: translateX(0%); opacity: 1; }

} – fadeInR은 0%일 때는 x축 150%인 지점(화면을 넘어서서 보이지 않는 오른쪽 부분)에 있다가 100%일 때는 x축 0%인 지점까지 오게 하는 애니메이션이다. 0%일 때는 보이지 않게 하기 위해 불투명도를 0으로 했고 100%일 때는 화면에 나타나게 하기 위해서 불투명도를 1로 설정했다.

@keyframes fadeInL {

0% { transform: translateX(-150%); opacity: 0; }

100% { transform: translateX(0%); opacity: 1; }

} – fadeInL은 0%일 때는 x축 -150%인 지점(화면을 넘어서서 보이지 않는 왼쪽 부분)에 있다가 100%일 때는 x축 0%인 지점까지 오게 하는 애니메이션이다. 0%일 때는 보이지 않게 하기 위해 불투명도를 0으로 했고 100%일 때는 화면에 나타나게 하기 위해서 불투명도를 1로 설정했다.

애니메이션 설정

.main .slick-active .left_area .nike_img { animation: fadeInR 0.5s 0.5s both; }

– fadeInR이 0.5초간 동작하며 동작 개시 후 0.5초 후 작동하게 딜레이 시켰다. Both를 주어 애니메이션이 끝난 후 처음과 끝에 지정된 css 스타일을 모두 유지하게 해 주었다.

.main .slick-active .left_area .nike_tit { animation: fadeInL 0.5s 0.7s both; }

– fadeInL이 0.5초간 동작하며 동작 개시 후 0.7초 후 작동하게 딜레이 시켰다. Both를 주어 애니메이션이 끝난 후 처음과 끝에 지정된 css 스타일을 모두 유지하게 해 주었다.

.main .slick-active .right_area .star_rate h2, .main .slick-active .right_area .star_rate p

{ animation: fadeInL 0.5s 0.5s both; }

– fadeInL이 0.5초간 동작하며 동작 개시 후 0.5초 후 작동하게 딜레이 시켰다. Both를 주어 애니메이션이 끝난 후 처음과 끝에 지정된 css 스타일을 모두 유지하게 해 주었다.

.main .slick-active .right_area .star_rate h2 i { animation: fadeInL 0.5s 1.5s both; }

– fadeInL이 0.5초간 동작하며 동작 개시 후 1.5초 후 작동하게 딜레이 시켰다. Both를 주어 애니메이션이 끝난 후 처음과 끝에 지정된 css 스타일을 모두 유지하게 해 주었다.

.main .slick-active .right_area .desc1 { animation: fadeInL 0.5s 0.5s both; }

– fadeInL이 0.5초간 동작하며 동작 개시 후 0.5초 후 작동하게 딜레이 시켰다. Both를 주어 애니메이션이 끝난 후 처음과 끝에 지정된 css 스타일을 모두 유지하게 해 주었다.

.main .slick-active .right_area .desc2 ul li { animation: fadeInR 0.5s 0.6s both; }

– fadeInR이 0.5초간 동작하며 동작 개시 후 0.6초 후 작동하게 딜레이 시켰다. Both를 주어 애니메이션이 끝난 후 처음과 끝에 지정된 css 스타일을 모두 유지하게 해 주었다.

lick-active .right_area .select .size { animation: fadeInL 0.5s 0.5s both; }

– fadeInL이 0.5초간 동작하며 동작 개시 후 0.5초 후 작동하게 딜레이 시켰다. Both를 주어 애니메이션이 끝난 후 처음과 끝에 지정된 css 스타일을 모두 유지하게 해 주었다.

.main .slick-active .right_area .select .cart { animation: fadeInL 0.5s 0.6s both; }

– fadeInL이 0.5초간 동작하며 동작 개시 후 0.6초 후 작동하게 딜레이 시켰다. Both를 주어 애니메이션이 끝난 후 처음과 끝에 지정된 css 스타일을 모두 유지하게 해 주었다.

.main .slick-active .right_area .select .price { animation: fadeInL 0.5s 0.7s both; }– fadeInL이 0.5초간 동작하며 동작 개시 후 0.7초 후 작동하게 딜레이 시켰다. Both를 주어 애니메이션이 끝난 후 처음과 끝에 지정된 css 스타일을 모두 유지하게 해 주었다.