23-11-15

* 테스트 예제 완성하기

처음부터 다시 진행

**<1차 진행사항>**

인덱스 html

부트스트랩

글꼴 링크 복사

Css backgroundcolor 설정

Css 글꼴 링크 css부분 복사

Css html과 연결

Css main부분 형태 설정 p태그 부분 div 부분 생성

Html 부트스트랩 컨테이너(반응형 웹 만들 때 컨테이너 안에 넣는 것 중요)

Html 섹션별로 나누어 html 작성

Html id생성

Html 자바 스크립트 연결 생성 (언제든해도됨)

Html 이미지 생성, 부트스트랩 이미지 반응형 설정 ~~(묶어서 위치설정\_col 부트스트랩 그리드)~~

Html 이미지 외 글자와 버튼 생성

Html 전체 위치설정

Css p태그 글꼴 설정 및 보이는 것 설정

Default, 끝 이제 start.js하기

JS : begin : 메인 화면 다 만들면 버튼 눌러서 넘어가는 것 만들기  
Js 외부변수 가지고오기 main과 qna 나중엔 result까지(전역변수이용)  
버튼을 누르면 qna 부분이 보이게 설정(html 함수지정, js 함수설정)

Css qna부분이 메인화면에 보이지 않게 설정(display)

JS : begin : 애니메이션 ~~(webkitanimation, @keyframes) 모질라도큐먼트 사이트 참고~~Js 애니메이션 설정~~(setTimeout,들어오는것나가는 것 설정 시간설정)~~Css 애니메이션 설정 ~~(0->1 1->0 설정)~~

**<2차 진행사항>**

그 전에 버튼 만들어서 실행시키는 부분 만들어보기

-qna부분

Html qna 부분 세부분으로 나누기 ~~(div, class로 나누기 마진과 패딩설정)~~

상태부분해야되고 색깔 넣기 식넣고

질문부분 1 css도 설정 배열로 가지고 오기

대답부분 3개 querySelectorAll로 설정해서 3개로 가지고 와야 함

For 부분 설정하기

누르면 넘어가는 부분이 goNext 함수부분

JS : var a를 도큐먼트로 불러옴 대답리스트 버튼형식으로 만들어서 -> addAnswer 함수만들기  
 (CREATEELEMENT,querySelector)

addAnswer 함수 -> setTimeout부분하고 변수선언했다

appendChild

addEventListener -> click -> function -> 괄호설정

for

setTimeout -> for

js goNext 추가하기

css 부분

최대호환성부분

패딩은 내부?

마진은 외부?

const main=document.querySelector("#main");

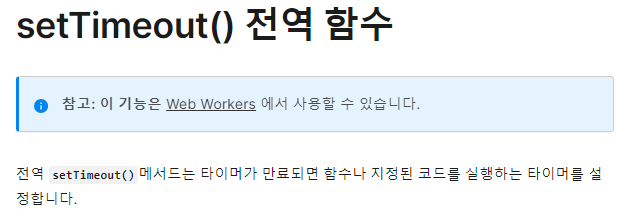
전역변수로 외부 변수 가지고올 때 const 사용

<section id="main" class="mx-auto my-5 py-5 px-3">

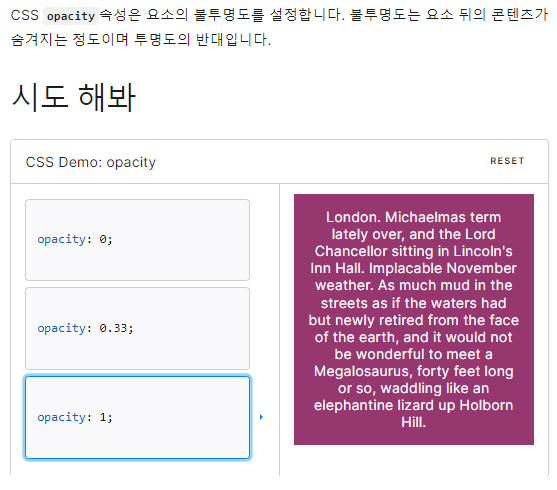
Mt-5는 top으로 위에만 적용, my-5는 위 아래 모두 5씩 적용  
mx-auto하면 원래 x가 한쪽으로 치우쳐져 있는 것이 가운데로 모이고 자동으로 정렬됨

setTimeout(()=>{});

JS : 코드 실행 전 타이머가 기다려야 하는 시간, 문자열은 보안 위험성을 이유로 권장하지 않음  
타임딜레이가 없으면 나중에 fadeIn,fadeOut을 설정했을 때 시간이 지나고 보여지는 것을 할수 없음, 시간이 지나고 애니메이션과 그 main none되고 qna사라지는 것은 그 다음에 하도록 설정



@keyframes fadeIn{

    from {opacity: 0;}

    to {opacity: 1;}

}

CSS : fadeIn은 들어갈 때 어떻게 변하게 할건지  
opacity는 불투명도 설정이다 0에서 1로 되게 설정한 것  
즉, 없다가 나타나는 것 (fadeOut은 있다가 없어지는 것)

@-webkit-keyframes fadeIn{

Keyframes은 모든 브라우저에서 지원됨  
webkit-keyframes는 크롬 사파리 브러우저에서지원  
=> 둘다 사용해야 안보이는 부분이 없어짐

**FadeIn, fadeOut할 때 keyframe을 css에서 설정해줘야 작동한다.**

main.style.webkitAnimation="fadeOut 1s"

main.style.animation="fadeOut 1s"

qna.style.webkitAnimation="fadeIn 1s"

qna.style.animation="fadeIn 1s"

JS: main부분은 나가기가 될거고(버튼누르면) qna부분은 나오게 할거다(나가는것,들어가는 것 설정)  
css에 적용한 부분 불러와서 1초동안

#qna{

    display: none;

CSS : 디스플레이 보이지 않겠다고 설정한 것

var a=document.querySelector('.answerBox');

var answer=document.createElement('button');

answer.classList.add('answerList');

answer list추가

var answer=document.createElement('button'); //이게 있어서 addEventListener이 있어야 된다

이제 qna부분 완성했다 이제는 결과를 더하는 부분 만들어야 한다

goResult -> setResult

var imgURL='img/image-'+ point +'.png'

0~11번의 이미지를 하나씩 써주지 않고 point로 번호 받아서 가져오는 것!

**Js > start.js**

function부분 6개 있다.

begin함수는 버튼 클릭했을 시 애니메이션 효과 주면서 두번째 qna보이게 하는 부분이다.

goNext함수는 begin함수 클릭시 효과부분 준 다음에 이 goNext로 온다.

질문과 대답 배열로 출력하는 부분

if(qIdx===endPoint){

qIdx를 파라미터로 주었는데 이것이 endPoint(즉,12)가 되면은 goResult()부분으로 가라고 하는 것  
그리고 값을 반환하라고 하고 있다.

var q=document.querySelector('.qBox');

q.innerHTML=qnaList[qIdx].q;

qIdx가 endPoint와 같지 않으면 변수 q에 qBox라는 class를 불러온다   
그리고 q를 HTML에 보이게 할건데 data.js에 qnaList의 배열 부분중 .q부분을 불러와서 보여줄것

for(let i in qnaList[qIdx].a){

addAnswer(qnaList[qIdx].a[i].answer, qIdx);

for문으로 qnaList부분을 배열로 가지고와서 a(대답)부분을 addAnswer함수로 전달한다, 그 안에 넣어져 있는 것은 파라미터이고, 두개를 줘서 addAnswer함수에서는 두개를 입력받았다.

var status=document.querySelector('.statusBar');

status.style.width = (100/endPoint)\*(qIdx+1)+"%";

status 변수를 선언했는데 이거는 statusBar라는 class를 가지고 온 것.  
그 변수에 값을 주었는데 qIdx는 qnaList부분이고 12개(0~11)를 가지고 있다.  
상태를 표시해주는 부분이고 개수에 따라 진행되는 것에 따라 상태바의 색깔이 올라갈 것.

addAnswer함수는 goNext에서 파라미터로 보냈던 부분들을 answerText와 qIdx로 각각 받는다.

function addAnswer(answerText, qIdx){

    var a=document.querySelector('.answerBox');

a라는 변수를 answerBox라는 class로 받고,

var answer=document.createElement('button');

//이게 있어서 addEventListener이 있어야 된다

    answer.classList.add('answerList');

// 클래스 네임이다. 그래서 알아볼 수 있는 것으로 설정 가능

    answer.classList.add('my-3');

//이 부분은 환경설정하는 부분이고 이 아래 모두를 answerList로 준 것이다.

    answer.classList.add('py-3');

    answer.classList.add('mx-auto');

    answer.classList.add('fadeIn');

answer라는 변수를 선언하고 이것을 button은 태그 자체를 받았다  
(button이라는 것이 type으로 지정한 부분인데 id나 class를 주지 않았는데도 지정해서 준 것.)

answer에 classList를 추가해준 것인데 처음에 class네임처럼 지정해준 것(다른 곳에 선언되지 않은 부분이지만 나중에 부르기 위해 이 곳에서 지정해준 것이다.

<div class="col-lg-6 col-md-8 col-ms-10 col-12 mx-auto">

html에서 이렇게 지정해준 것처럼 JS에서는 위에처럼 지정해줘야 하는 것이다.

* 버튼을 누르면 classList 추가해준 answerList부분이 작동되는 것이다.