```
document.getElementsByTagName('Tagname')
                  document.getElementsByClassName('Classname')
                  document.querySelector('CSS Selector')
                  document.querySelectorAll('CSS Selector')
                                                     document.name속성값
                  document.createElement('Tagname');
                       Element.innerHTML = 'content';
                                   Element.getAttribute('속성');
                                    Element.setAttribute('속성', '값');
                                   Element.hasAttribute('속성');
                                    Element.removeAttribute('속성');
 02) 동적 HTML접근
                                 [ Element.classList.add('Classname');
                                 Element.classList.remove('Classname');
                                 Element.classList.toggle('Classname');
                                 Element.classList.contains('Classname');
                             이벤트 리스너 Element.addEventListener('이벤트종류', 콜백함수);
             Event 정의
                             마우스 이벤트
                            키보드 이벤트
              Event 종류
                            form 관련 이벤트
                             브라우저 이벤트
                            트랜지션 이벤트
                      상위 요소 찾기
 04) HTML 탐색하기 하위 요소 찾기
                                        Element.children HTMLCollection Array.from(Element);
                      이전, 다음 요소 찾기
                                            상위Element.appendChild('추가할 Element');
                                             상위Element.insertBefore(추가할 Element, 기준 Element);
 05) HTML 생성하기 HTML 하위요소 추가
                                            상위Element.insertAdjacentHTML('위치값', 쌩 HTML코드);
                                                                                       HTML코드는 innerHTML에 넣듯이
쌩으로 코딩해서 넣어야 함
                              inputElement.value; if문을 이용하여 입력여부 검사
                                       특징 select요소 = 배열 & option = 배열의 요소
                                       selectElement.selectedIndex; if문으로 0인 경우를 선택하지 않은 경우로 처리
                 select 요소(드롭다운)
                                                      selectElement.addEventListener('change', 콜백함수)
                                      change 이벤트 inputElement의 change 이벤트는 입력후 포커스가 빠져나왔을 때 실행
                                  특징 name속성이 동일한 요소끼리 각각 배열로 그룹화 됨 반복문 처리 가능
                                   선택여부 (태문으로 체크여부 조회
                  체크박스, radio
                                                1. 전체 선택 항목들을 반복문을 통해 탐색
                                  전체 선택 2. dataset을 통해 연결된 항목들을 반복문을 통해 탐색
 06) Form 요소
                                                 3. 전체선택 항목의 checked값을 각 항목의 checked값에 대입
                                 formElement.submit(); HTML <input type='submit'/> 요소 클릭함과 동일
                  reset, submit
                                                                                                  콜백함수 첫 줄에 e.preventDefault(); 처리
                                submit 이벤트 document.querySelector('form요소 selector').addEventListener('submit', 콜백함수)
                                                                                                  입력값 검사 후 e.currentTarget.submit(); 으로 preventDefault();로 중지한 submit 실행
                  05-check_value.html 아랫부분 모던 JS 탐색 기법 숙지
                                             파일 추가는 change 이벤트로 감지
                  <input type='file' multiple />선택된 파일(반복문으로 돌아도 배열이 아닌 단일 객체 리턴)
                                             URL.createOnjectURL(fileElement); 파일의 URL을 리턴(HTML에 동적으로 삽입할 때 사용)
                                                                   이름과 값은 URLEncoding처리
                                "이름=값; 유효시간; 유효경로; 유효도메인"
                                                                  유효경로 -> 해당 쿠키값을 읽고 쓸 수 있는 위치
보통 "/"만 지정하여 사이트 전역 사용
                                 도메인 별 유효 .naver.com 등으로 패밀리 사이트간 공유
                                           장점 = 대부분의 브라우저가 지원

        쿠키
        장단점
        단점
        쿠키 용량이 4kb로 작음

                                                   매 http 요청마다 백엔드에 모든 쿠키를 전송하여 서버에 부담
                                                    암호화 불가-> 보안 취약
                                           오늘 본 상품
                               사용 예 오늘 하루 이 창 열지 않음

    ("쿠키이름=값; path=/; max-age=시간")
    (max-age를 0으로 설정하여 쿠키 즉시 삭제)

    서장 방식)
    (document.cookie = STring)
    (저장된 쿠키도 document..cookie를 호출하여 표시)

                                       (Key와 Value형태
                                       로컬에만 저장
(<->쿠키는 로컬+서버에 저장)
  07) 데이터 읽기 쓰기
                                                               클라이언트 정보를 영구적으로 저장
                                                               ex) 자동 로그인 저장
                                                                         localStorage.setItem(keyName, ketValue);
                                                            사용법 (localStorage.removeltem(keyName);
                                                                        localStorage.getItem(keyName);
                       웹 스토리지 종류
                                                                        [localStorage.clear();
                                                               세션 종료시 클라이언트에 대한 정보 삭제
                                                              ex) 입력폼 정보 저장, 비로그인 장바구니
                                                                         sessionStorage.setItem(keyName, ketValue);
                                               세션 스토리지
                                                                         sessionStorage.getItem(keyName);
                                                              사용법
                                                                        sessionStorage.removeItem(keyName);
                                                                         sessionStorage.clear();
                                                서버에 불필요하게 데이터 저장 x = 백엔드 전송x
                                    장점 약 5mb로 저장 가능 용량이 크다.
                                                                               location -> 주소창 / 피싱 사이트 방지를 위해
브라우저 개발사에서 자체적으로 지원 X
                             window.open(url, '창이름', 'width = ..., height = ..., scrollbars = yes/no, toolbar=yes/no, menubar=yes/no, status=yes/no, location=yes/no')
                                                                              창이름을 빈 문자열로 전달할 경우 매번 새로운 팝업창
창이름을 전달할 경우 한 창을 반복 사용
                               웹브라우저의 정보를 저장하고 있는 객체
                                navigator.appName; 브라우저 이름
                               navigator.CodeName; 브라우저 코드명
                               navigator.platform; 플랫폼 정보
                              navigator.appVersion; 브라우저 버전
                                                 사용자 정보
                                                                               edge, chrome, opera, firefox, safari, skipstone, msie or trident, netscape 등을 if문에 indexOf의 값이 -1이 아님을 조건으로 판단하여 브라우저의 종류를 알 수 있다.
                                                 navaigator.userAgent.toLowerCase();
                                                                               iphone, ipad, ipod, android 등을 사용하여
모바일 브라우저인지 판단할 수 있다.
                                                   gps정보가 없을 경우 IP주소 기반 위치 탐색
-> 우리나라는 대부분 서울시청 부근으로 나옴
                                                                      position.coords.latitude;
                                                  위도, 경도만을 리턴 position.coords.longitude; 경도
                                                                       https://www.openstreetmap.org/#map=18/${latitude}/${longitude}
                              (navigator.geolocation;
                                                                                                                              {enableHighAccuracy: true/false, maximumAge: 0, timeout: 30000,}
                                                                                                                             enableHighAccuracy -> 높은 정확도 사용하기
                                                              navigator.geolocation.getCurrentPosition(성공시 호출될 콜백함수,
실패시 호출될 콜백함수, 조회옵션(json형식이며 선택사항));
                                                                                                                             maximumAge -> 캐시에 저장한 위치정보를 대신 반환할 수 있는
최대 시간(단위 ms). 0으로 설정시 매번 gps값을 새로 취득
                                                                                                                            timeout -> 제한시간안에 조회하지 못하면 에러로 간주(단위 ms)
                                                              navigator.geolocation.watchPosition(파라미터는 같음)
                             decodeURIComponent(location.href); 문서의 URL주소
                             location.host; 호스트 이름과 포트
                            location.hostname; 호스트 컴퓨터 이름
                             location.hash; Hash값
                            (decodeURIComponent(location.pathname); ) 디렉토리 이하 경로
                            [location.port;] 포트 번호 부분
                            (location.protocol; 프로토콜 종류
                                             URL 조회부분
                                             substring(1); 을 이용하여 URL 조회부분을 추출 query
                                            추출된 부분을 .split('&')을 이용하여 각 변수 추출 params
                            | Coation.search; | CTA 수출된 부분을 .split('&')을 이용하여 각 면수 수출 | CTA 수출된 부분을 .split('=')을 이용하여 key와 value로 분할
                                             분할된 값을 빈 json에 저장
                                             각 key별 value값을 switch의 case로 나누어
페이지 이동 없이 여러 페이지를 구현
                            location.reload(); 페이지 새로고침
                            history.back();
                            history.forward(); 앞으로 가기
                                                       state -> 백업한 변수 json형식
                            history.pushState(state, title, url); title -> 페이지 제목
                                                      url -> 페이지 주소(주로 뒤에 #을 붙여 해시값을 바꿈)
                            구조를 유지하면서 페이지 전체의 내용을 바꾸고 싶을 때 querystring -> location.search
                             페이지 내용은 그대로 두고 새로운 내용을 바꾸고 싶을 때 history
                                   Module 형태에 상관없이 소스코드를 파일 형태로 분리하여 재사용 가능 하게 만든 단위
                                   Library 모듈 중에서 함수나 클래스 단위로 구성된 형태들의 모임
                                   Package 라이브러리를 모아놓은 형태. 주로 압축파일 형태
                                                 무료로 누구나 사용할 수 있게 공개되어있는 소프트웨어or소스코
                                                무료인 대신 원작자 표기 필수, 수정사항에 대해 원작자에게 알려
줄 의무
                                                 영리, 비영리 목적의 사용 가능 여부에 따라 라이센스 모델이 변
경
 09) 라이브러리 활용
                                                                                                  GNU, MIT, Apache 라이센스
                                          lightbox https://fslightbox.com/
                                          mediabox https://github.com/pinceladasdaweb/mediabox/
                        (https://animations.vercel.app/)
(https://github.com/robertoentringer/animations.js)
                                         aos https://michalsnik.github.io/aos/
                                         chart https://www.chartjs.org/
                                         new XMLHttpRequest();
                            기본 사항 method 백엔드 페이지 접속 방식(CRUD) -> GET, POST, PUT, DELETE
                                         url 접속할 대상 페이지 주소
                                          XMLHttpRequestElement의 현재 상태를 나타냄
                                          UNSET = 0
OPENED = 1
                            readyState
                                         HEADERS_RECEIVED = 2
                                          LOADING = 3
                                         DONE = 4
                                                                            객체.onreadystatechange = e => {
const ajax = e.target;
switch (ajax.readyState){
                                                           XMLHttpRequestElement의 readyState 프로퍼티 값이 변할 때마다
자동으로 호출되는 함수
                             ajaxElement.onreadystatechange
                                                          서버에서 응답이 도착할 때 까지 readyState 프로퍼티 값의 변화
에 따라 총 5번 호출됨
                                                          통신 결과를 리턴받는 것이 아니라, 통신 과정중에 발생하는 이벤
트를 통해 후속처리를 구현
                                                          별도의 모듈로 작성하며, 콜백함수를 통해 결과를 핸들링
                                                       [ajax.responseText => 통신을 통해 읽어온 내용 데이터 타입이 string이므로, JSON.yarse(ajax.responseText); 이후 json 핸들링
 10) XMLHttpRequest(Ajax)
                                                       400 접근권한 X
                                                       403 서버의 접근 거부
                                      서버의 문서 상태
                             status
                                                        (404) NotFound
                                                        (50x) 백엔드 에러
                                                        statusText로 상세 내용이 리턴됨
                                                          이 값이 4 -> 400번대 에러
                                     (parseInt(status/100)) 이 값이 5 -> 500번대 에러
                             XMLHttpRequestElement.open(method, url); 요청 초기화 = 통신준비 마침
                             XMLHttpRequestElement.send(); 백엔드와 통신 시도
                                                                                                                   ajax.status == 200 resolve(json) 호출 .then((json) => { ... }).catch(e) => { ... }).finally{ ... }
                            이외의 경우, 모듈화시킨 onreadystatechange에서 콜백을 리턴받
지 않고, 에러창을 띄움
                                                                                            성공시 모듈을 사용한 바깥의 .then(콜백함수) 호출
                            비동기처리(2)- Promise 모듈 Promise객체를 리턴하는 함수 자체를 모듈화
                                                                                            실패시 모듈을 사용한 바깥의 .catch(콜백함수) 호출
                                                                                          비동기로 promise객체가 리턴된 후 async 함수가 실행됨
                                                     클릭이벤트 콜백함수에 async를 건 후,
try catch 구문 안에서
Promise객체를 리턴하는 모듈을 await시킴
                                                                                          정상적으로 실행시 결과값을 변수에 저장하여 핸들링
                                                                                         비정상 실행시 catch 구문 실행
                                       https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js
                                       1. 각 링크에 dataset을 이용하여 url 뒷부분을 설정
                                       2. 이를 axios.get(url)에 변수 형태로 지정
                                       3. 리턴값이 promise이므로, .then().catch().finallly()를 적용
                                                 1. 각 링크의 클릭이벤트 콜백함수에 async 적용
                                                 2. 일반 axios와 같이 진행
                                                                                                이 때 필요한 params와 headers는 url 뒤에 json형태의 파라미터
로 전달(각각의 params와 headers도 json형태)
                                                 3. try catch 구문에 await axios.get(url)을 삽입하여 변수에 저장
                                                 4. try catch구문 바깥에 해당 변수가 존재할 경우 실행할 코드를
삽입
              Create axios.post(url, {추가할 데이터 json}) 일련번호인 id값은 자체적으로 생성되기 때문에 일련번호를 제외 한 데이터를 전달한다.
              Read axios.get(url)

      Update
      axios.put(url, {수정할 데이터 json});

      수정 전에 get을 이용하여 기존 데이터를 불러온다.

              Delete axios.delete(url)
              Element.keyCode; 키보드의 각 버튼에 할당된 일련번호
                        window.scrollY 스크롤바 상단지점부터 최상위까지의 높이
                        window.screen.availHeight; 웹브라우저의 창 높이
                        document.body.scrollHeight; HTML 문서의 전체 높이
             Width window.innerWidth; 웹브라우저의 넓이
                                       JS로 불러온 HTML요소의 style 프로퍼티 객체는
모든 CSS속성을 프로퍼티로 갖는다.
             Element.style.CSS속성 = 값; (CSS에서 '-'를 사용하던 부분은 카멜표기법으로 변환
                                        값에 단위를 명시해야 할 경우 따옴표로 감싸야 한다.
00) ETCs 
반복문
복습
                      Attribute 태그의 값이 있는 속성 ex) src, href, width, height 등
                                                                                                                   disabled 비활성 = true, 활성 = false
                                                                                                                  readonly 위기전용 = true, 읽기 쓰기 겸용 = false
                      Property 태그의 값이 없는 속성 ex) checked, readonly, selected 등(true, false값을 갖는다)
                                                                                                  체크박스, 라디오 checked 체크됨 = true, 체크 안됨 = false
                                                                                                  드롭다운 selected 선택됨 = true, 선택해제 = false
                                 focus();는 해당 항목에 입력커서를 이동시키는 함수
                                focus는 이벤트로 특정 항목으로
커서가 이동되었을 때 동작하는 이벤트
                                  blur는 이벤트로 특정 항목에서
커서가 빠져나왔을 때 동작하는 이벤트
```

document.getElementById('id')