23-11-21

* 예제문제로 자바스크립트 전체내용 확인

<문제>

1. Password validation.html의 파일을 받고 여기에 유효성 충족 검사하기

~~유효성 검사가 충족합니다.~~  
~~Disable -> css~~

addEventListener안에 if문을 e의 값을 받아와서 사용하기

#방법1

<js부분>

        const ok = document.querySelector('.passok');

        const no = document.querySelector('.passno');

        if(isValidLowercase(password)==true&&isValidUppercase(password)==true&&isValidNumber(password)==true&&isValidSpecial(password)==true&&isValidCaracterLength(password)==true){

            ok.style.display="block"

            no.style.display="none"

        }

        else{

            ok.style.display="none"

            no.style.display="block"

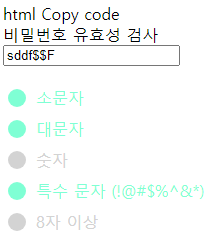
        }

Html에 화면출력할 부분을 만들어놓고, css에 display를 none으로 설정해 놓은 것을

<html부분>

    <div><p class="passok">비밀번호 사용이 가능합니다</p></div>

    <div><p class="passno">비밀번호 조건이 일치하지 않습니다</p></div>

<css부분>

        .passok, .passno{

            display: none;

        }

#방법2

(isValidLowercase(password)==true&&isValidUppercase(password)==true&&

        isValidNumber(password)==true&&isValidSpecial(password)==true&&

        isValidCaracterLength(password)==true)

        ? 지정해주기.classList.add('active') : 지정해주기.classList.remove('active');

이 방법은 소문자 대문자 처럼 효과로 지정이 된다.  
모든 조건들이 true가 되면 add가 되고, false가 되면 remove가 된다.  
효과로 적용한 것이기 때문에 대문자 소문자와 같이 지정이되면 지정이 되는 곳에 효과가 나오고,  
지정이 되지 않으면 지정되지 않음에 효과가 나온다.  
(html은 목록만드는 li에 넣어줘서 작성하기)  
방법1처럼 동일하게 나오지 않음

#방법3

innerHTML로 만드는 방법

<html부분>

<p id="checkPass">check</p>

<js부분>

const checkPass = document.getElementById('checkPass');

id로 잡은 p태그를 새로운 변수로 선언한 것   
getElementByID가 아닌 class로(getElementClass) 불러오려면 아래 innerHTML할 때

checkPass[0].innerHTML = "유효성 검사를 충족합니다.";

이렇게 배열로 불러와야함 왜냐면 class는 배열로 불러오기 때문이다.

             if (isValidCaracterLength(password) && isValidSpecial(password)

&& isValidNumber(password) && isValidUppercase(password) &&

isValidLowercase(password)) {

                    //console.log("T")  
                    checkPass.innerHTML = "유효성 검사를 충족합니다.";

             } else {

                   //console.log("F")  
                    checkPass.innerHTML = "유효성 검사를 충족하지 못합니다.";

             }

1. Event.stop modal 파일 받은 것 아래 주석처리 안된 것 주석달기

정리하기

    const stopPropagation = (e) => {

        e.stopPropagation();

    }

재귀함수구나????

removeChild 주석달기

const modalWrapperElem = document.querySelector('.modal-wrapper');

외부 컨테이너 찾기 위한 도큐먼트 셀렉  
모델 감싸는 외부컨테이너 찾는것

bodyElem.removeChild(modalWrapperElem);

보이지 않게 하는 부분이 이 부분.

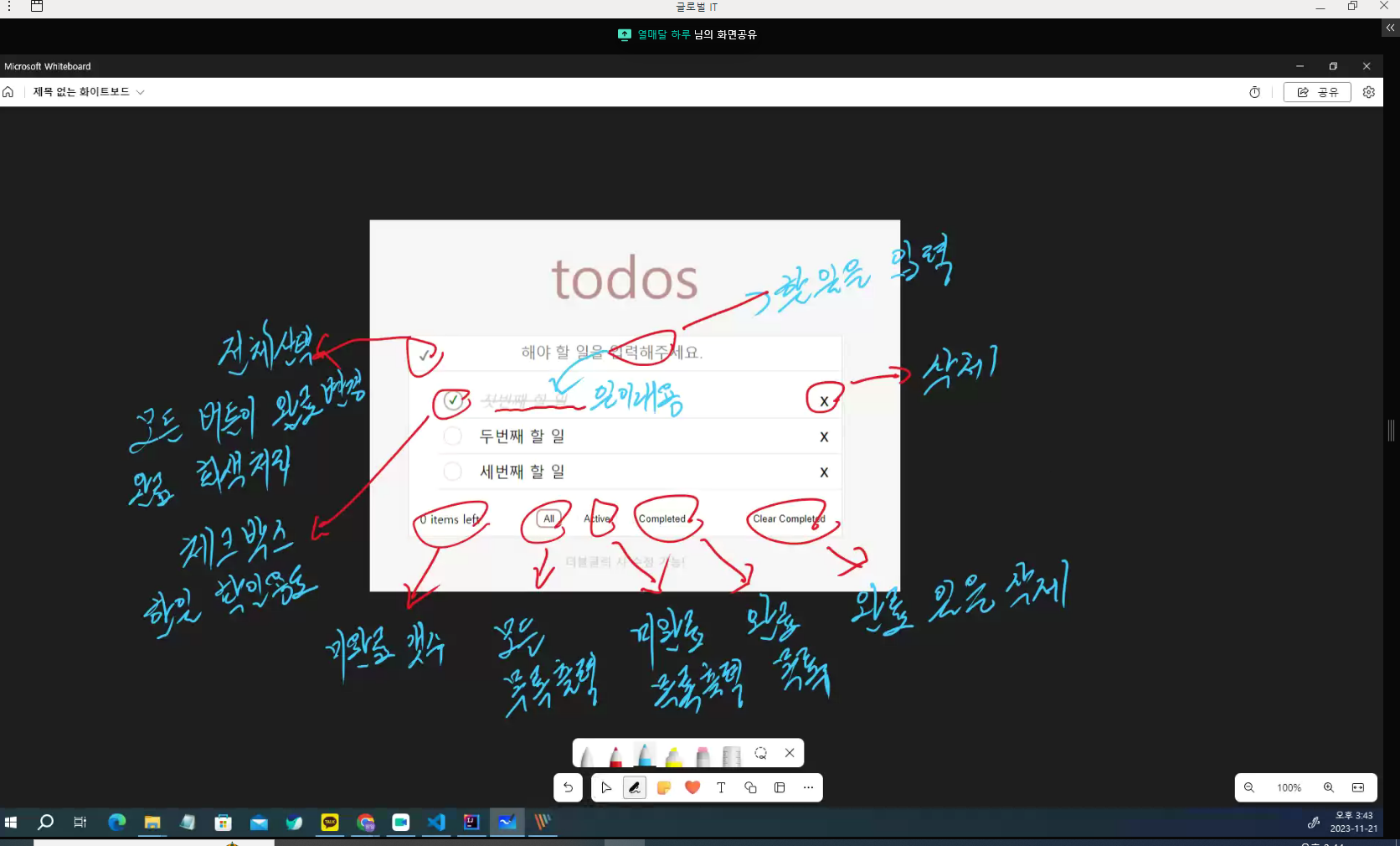
    // 이벤트 전파 중지 함수

    const stopPropagation = (e) => {

        e.stopPropagation();

다른곳으로 데이터 가지고 가지 말라는  
닫기 누를 때 그 부분이 main

**Todo\_javascript 예제**



**Todo.js**

//html class 부분 가지고 와서 선언-> 사용가능

const todoInputElem = document.querySelector('.todo-input');

const todoListElem = document.querySelector('.todo-list');

const completeAllBtnElem = document.querySelector('.complete-all-btn');

const leftItemsElem = document.querySelector('.left-items')

const showAllBtnElem = document.querySelector('.show-all-btn');

const showActiveBtnElem = document.querySelector('.show-active-btn');

const showCompletedBtnElem = document.querySelector('.show-completed-btn');

const clearCompletedBtnElem = document.querySelector('.clear-completed-btn');

class로 만들어준 부분을 가지고 와서 js에서 선언해줌

//id개수 0으로 만들어놓음

let id = 0;

id개수는 0개 -> 앞으로 할일들이 채워지면 그 개수가 늘어나는 것 처음에는 0개

//todos 배열로 설정 -> 새로 들어와야 하니까.

let todos = [];

todos는 배열로 설정하였는데, 새로 들어와야 하기 때문에 배열로 설정한 것

//배열의 길이만큼 함수형태로 처리 (eval의 자바버전이나, 정확하게 모르겠음)

    leftItemsElem.innerHTML = `${leftTodos.length} items left`

$ 이 부분은 정확하게 이해가 가지 않는 부분이다

const onClickCompleteAll = () => {

    //배열의 길이찾는 것이 getAllTodos이고 그 길이가 0이되면 아래로 내려가도 입력할

    값이 없으니까 return하라는 것.

    if(!getAllTodos().length) return; // todos배열의 길이가 0이면 return;

!getAllTodos에서 !는 부정이고 이것의 길이가 0이면 return하라는 것이다.  
0이면 아래 코드가 작동되도 입력할 값이 없으니까 return하라는 것.

//todoId는 할일 생성한 한 부분이 하나의 아이디가 될 것

const deleteTodo = (todoId) => {

옆에 노란색 부분이 todoId가 될 것이다.

       //엔터키 눌러지면 업데이트 하라는 부분

        if(e.key === 'Enter') {

            updateTodo(e.target.value, todoId);

            //기존내용 지워버리고 새로운 내용으로 변경되라는 것

            document.body.removeEventListener('click', onClickBody );

        }

todo부분은 엔터누르면 할일이 추가된다 그 부분이고, 엔터 눌러지면 업데이트 하라는 부분이다  
기존내용이 있었는데 새로운 내용으로 변경되라는 것.

//작업 실행하게 되면 화면 내용 바뀌는 부분(addEventListener 사용)

const paintTodo = (todo) => {

    const todoItemElem = document.createElement('li');

    todoItemElem.classList.add('todo-item');

작업 실행하게 되면 화면 내용 바뀌는 부분이다 아래는 createElement를 여러곳 사용했다.

const paintTodos = () => {

    todoListElem.innerHTML = '';

    //조건 들어가는 자리, 조건에 변수명만 들어가면 된다(if와 같은 조건문)

    //요일(월,화,수,목,금) 할 때 switch 가장 많이 사용된다

    switch (currentShowType) {

        case 'all':

            const allTodos = getAllTodos();

            allTodos.forEach(todo => { paintTodo(todo);});

            break;

        case 'active':

            const activeTodos = getActiveTodos();

            //forEach

            activeTodos.forEach(todo => { paintTodo(todo);});

            break;

        case 'completed':

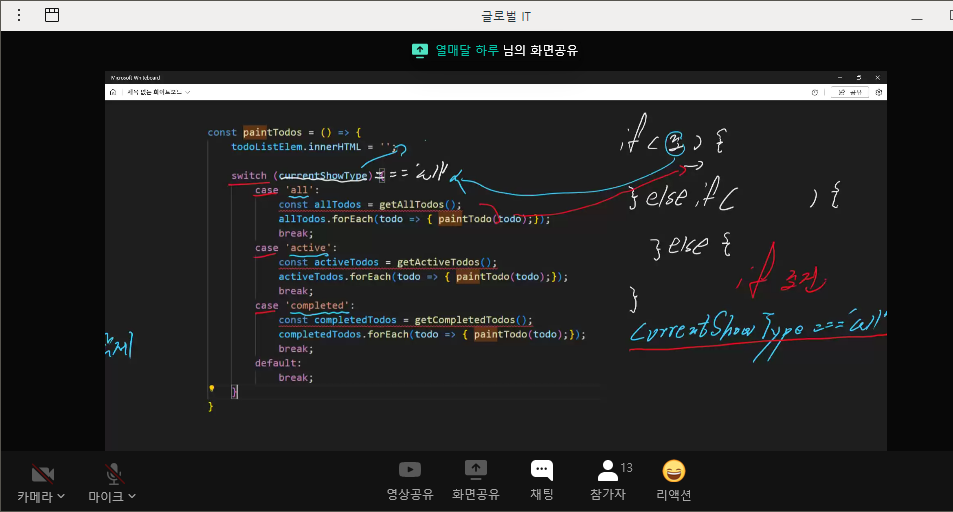
            const completedTodos = getCompletedTodos();

            // => 화살표 쓰는 것 이것이 편한 사람들이 있음 그럼 다른방법은 뭐가있을까? -> 일반함수로도 구현 가능

            // => 를 쓰면 외부에 있던 this를 사용가능함

            completedTodos.forEach(todo => { paintTodo(todo);});

            //break 를 쓰지 않으면 무한루프 됨(계속작동)

            break;

        default:

            break; }

Switch , case  
Select , case

 switch (currentShowType) {

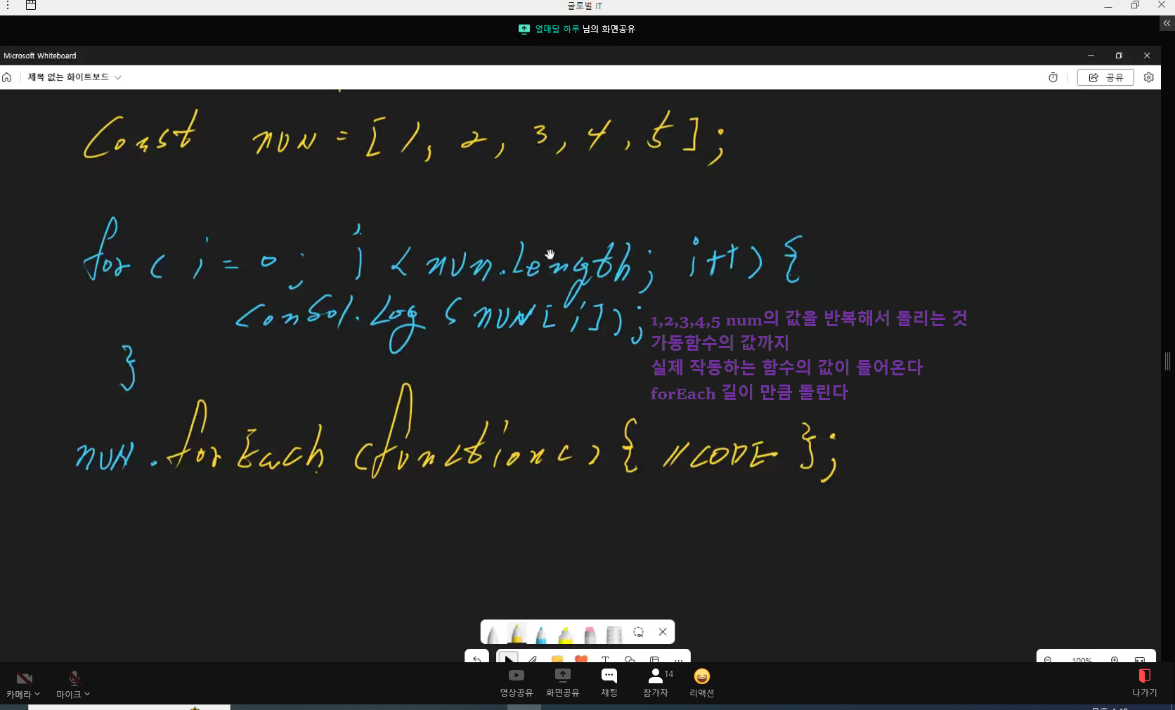
        case 'all':

            const allTodos = getAllTodos();

            allTodos.forEach(todo => { paintTodo(todo);});

            break;

forEach 예제 내일 할 것  
forEach는 조건문 if와 같은 역할을 함, 자세한 것은 문제 풀면서 알아갈 것.



//최초 실행될 내용 가지고 올 것임을 알림

const init = () => {

    todoInputElem.addEventListener('keypress', (e) =>{

        //해야할일을 입력하고 엔터 할 때 작동하는 부분(엔터하면 할일로 들어간다)

        if( e.key === 'Enter' ){

  //value 빈공간 처리 되어 있는 것은 입력받은 값을 e.target.value로 처리하라는 것

            appendTodos(e.target.value); todoInputElem.value ='';

최초 실행될 내용을 가지고 올 것임을 알리는 것  
해야할 일을 입력하고 엔터 누르면 작동하는 부분이다.

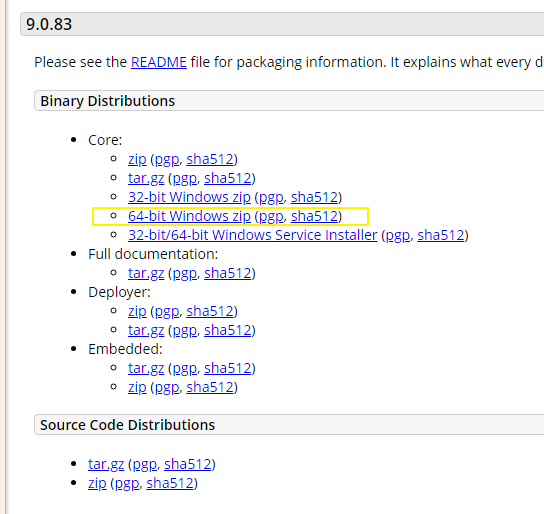
//모든 기능으로 만들어진 스크립트

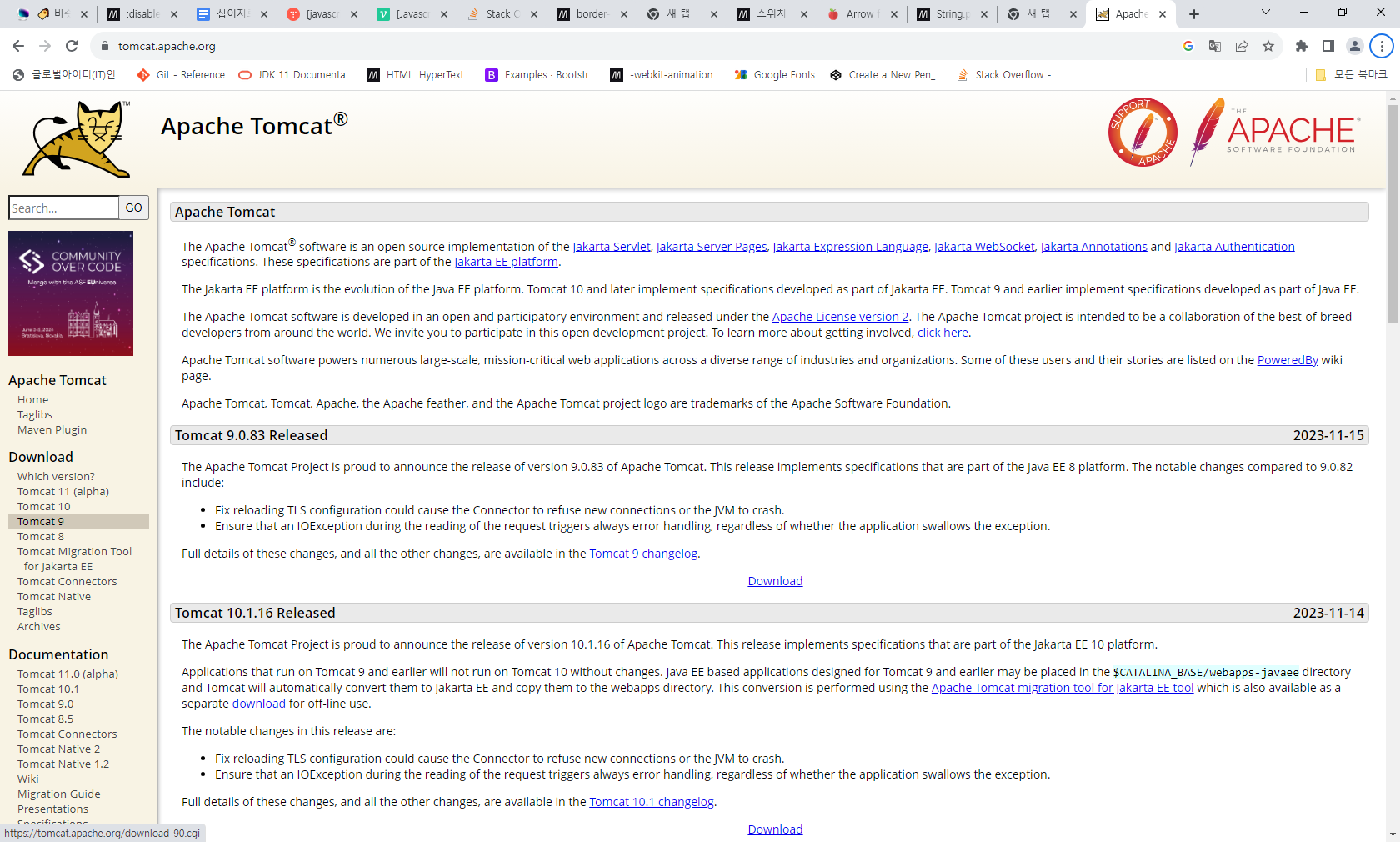
//이 코드상에서 작동되는 것이 아닌 기능정의 되는 함수 찾는것

//어떻게 흘러가는지가 이 코드에서 중요한 것이 아니라

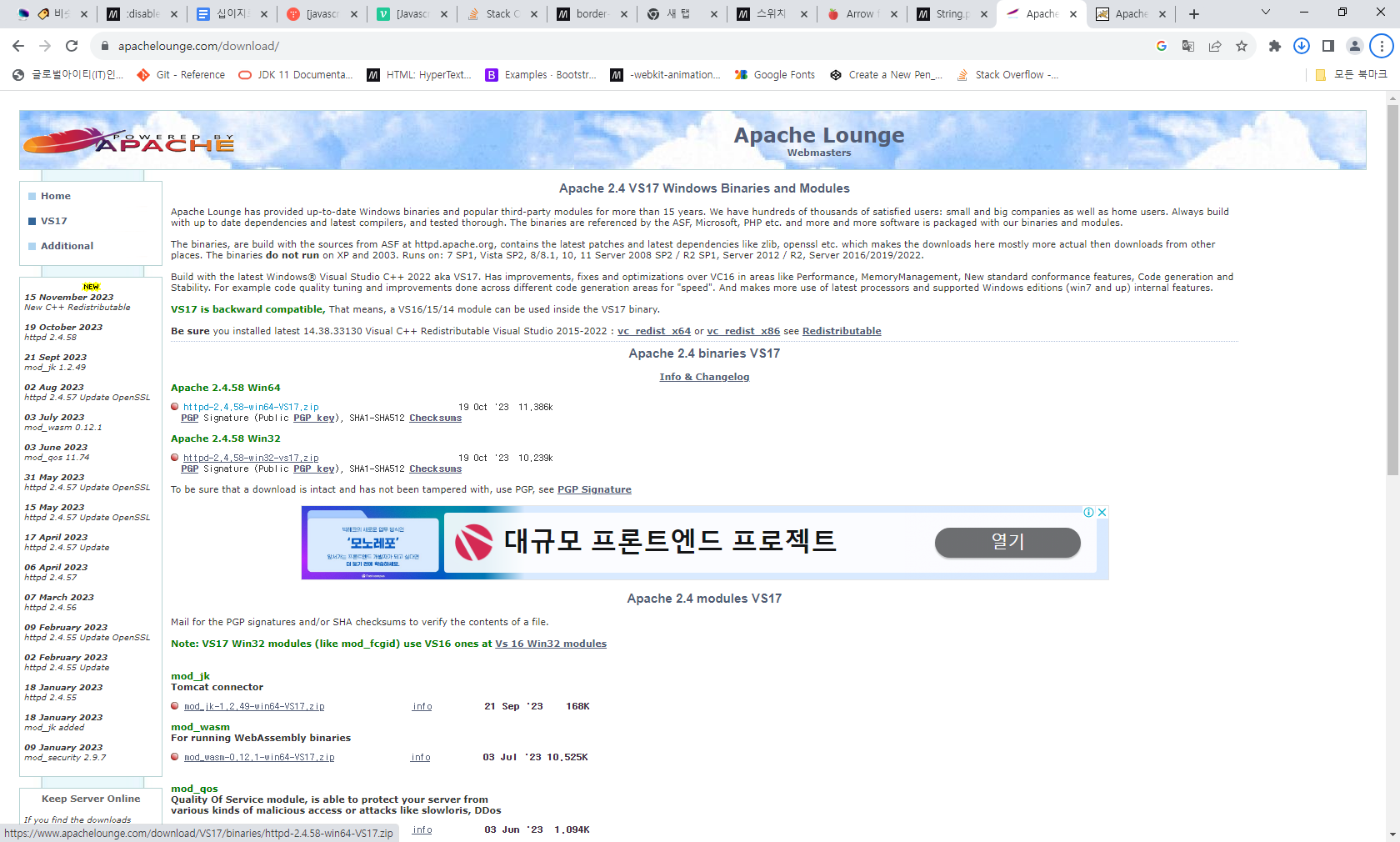
//함수 어떤 기능, 함수의 역할 연결이 중요

<다운로드>

아파치 톰캣 9.0.83



아파치 라운지 다운로드 2.4.58 win64 와



Additional에 15버전에 64win 다운로드 아파치 2.4.54 VC15



이클립스 다운로드에 download Packages선택해서  
오른쪽 아래 버전 더보기에서 버전 2021-06 win64 다운받기

* 이클립스는 일주일 동안만 사용한다(이후 사용하지 않음)

