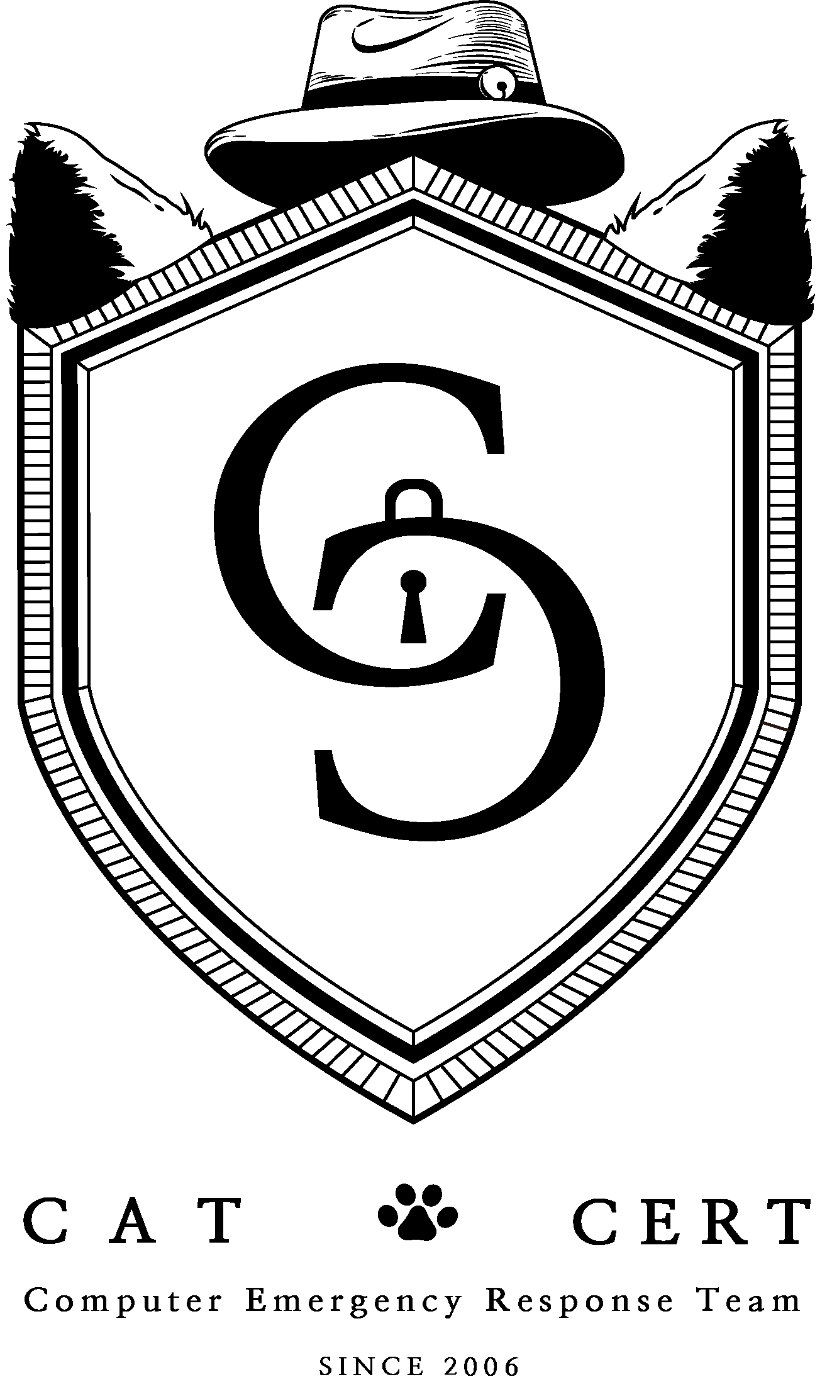
**캣서트 보고서**



202221649 인공지능학과 김민영

Web\_Study4

3.

**실습 #1**

|  |
| --- |
| <html>  <head>  <title>계산기</title>  </head>  <body>  <form method="post" action="">  <table border="1" width="200px" height="100px">  <tr>  <td> CAT-LOGIN </td>  </tr>  <tr>  <td> <input style="border:0" type="text" name="num1"> </td>  </tr>  <tr>  <td > <input style="border:0" type="text" name="num2"> </td>  </tr>  <tr>  <td> <input type="submit" value="입력"> </td>  </tr>  </form>  </body>  </html>  <?php  $ip\_list = "your IP: ".$\_SERVER['REMOTE\_ADDR']." you are not admin";  if($\_POST['num1']== 'gdjung'){  setcookie("gdjung", "cookieValue", time() + (60\*60\*24));  echo "Hello CAT-CERT";  }  else if($\_POST['num1']== 'admin'){  echo "<script>alert('$ip\_list')</script>";  }  else if (isset($\_POST['num1'])) {  echo "wrong password!" ;  }  ?> |

텍스트이(가) 표시된 사진

자동 생성된 설명

**#1 Web Storage 조사 및 실습**

Web storage : 서버가 아닌, 클라이언트에 데이터를 저장할 수 있도록 지원하는 HTML5의 새로운 기능

키(Key)와 값(Value)의 쌍 형태로 데이터를 저장

|  |  |  |
| --- | --- | --- |
|  | **Web storage** | **Cookie** |
| 용량제한 | 4KB | 5MB |
| 서버 | 전송되지 X | 전송 O |
| 유효기간 | X | O |

용량제한 : 5MB (쿠키는 4KB)

**1) Local storage :**

반영구적으로 데이터를 저장, 브라우저를 종료해도 데이터가 유지

브라우저 자체에 반영구적으로 데이터가 유지 → 도메인이 다른 경우에는 로컬 스토리지에 접근 못함

특정 웹사이트를 새탭이나 새창으로 띄워도 동일한 데이터를 공유

**오리진(= 프로토콜 + 도메인 + 포트)별**로 생성

|  |
| --- |
| <!DOCTYPE html>  <html lang="ko">  <head>  <meta charset="UTF-8">  <title>HTML5 API Web Storage</title>  </head>  <body>  <h1>localStorage 객체를 이용한 데이터의 저장</h1>  <div id="counter"></div>  <p><button onclick="clickCounter()" type="button">카운터 증가!!</button></p>  <p>브라우저 탭이나 창을 닫아도 카운터의 횟수는 초기화되지 않을 것입니다.</p>  <script>  function clickCounter() {  if(typeof(Storage) !== "undefined") {  if (localStorage.clickcount) {  localStorage.clickcount = Number(localStorage.clickcount) + 1;  } else {  localStorage.clickcount = 1;  }  document.getElementById("counter").innerHTML = "카운터의 현재 횟수는 " + localStorage.clickcount + "입니다!";  } else {  document.getElementById("counter").innerHTML = "이 문장은 사용자의 웹 브라우저가 Web Storage API를 지원하지 않을 때 나타납니다!";  }  }  </script>  </body>  </html> |

크롬 브라우저에서 클릭을 15번 하고 새롭게 크롬 창을 만들었을 때 15번이 유지되어 있다는 것을 확인함 (크롬이 아니라 다른 브라우저를 사용하면 데이터 유지 X)

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**2) Session storage :**

각 세션마다 데이터가 개별적으로 저장

세션을 종료하면 데이터가 자동으로 제거 → 같은 도메인이라도 세션이 다르면 데이터에 접근 X

동일한 탭/윈도우라도 **오리진(= 프로토콜 + 도메인 + 포트)라면 또 다른 세션 스토리지가 생성**

|  |
| --- |
| <!DOCTYPE html>  <html lang="ko">  <head>  <meta charset="UTF-8">  <title>HTML5 API Web Storage</title>  </head>  <body>  <h1>sessionStorage 객체를 이용한 데이터의 저장</h1>  <div id="counter"></div>  <p><button onclick="clickCounter()" type="button">카운터 증가!!</button></p>  <p>브라우저 탭이나 창을 닫으면 카운터의 횟수는 초기화될 것입니다.</p>  <p>하지만 결과보기를 누르거나 F5를 누르면 초기화되지 않을 것입니다.</p>  <script>  function clickCounter() {  if(typeof(Storage) !== "undefined") {  if (sessionStorage.clickcount) {  sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1;  } else {  sessionStorage.clickcount = 1;  }  document.getElementById("counter").innerHTML = "카운터의 현재 횟수는 " + sessionStorage.clickcount + "입니다!";  } else {  document.getElementById("counter").innerHTML = "이 문장은 사용자의 웹 브라우저가 Web Storage API를 지원하지 않을 때 나타납니다!";  }  }  </script>  </body>  </html> |

텍스트이(가) 표시된 사진

자동 생성된 설명

**#2 정규표현식 조사 및 실습**

**정규표현식 :** 문자열을 처리하는 방법 중 하나 / 특정한 조건의 문자를 ‘검색’하거나 ‘치환’하는 과정을 매우 간편하게 처리할 수 있도록 하는 수단

**구성요소 :** 메타문자와 추출하고 싶은 문자를 합친 것 / 패턴의 양식에 따라 문자열에서 문자를 추출

**메타문자 :** 문자를 설명하기 위한 문자, 문자의 구성을 설명하기 위해 원래의 의미가 아닌 다른 의미로 쓰이는 문자를 말함

텍스트, 게이지이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

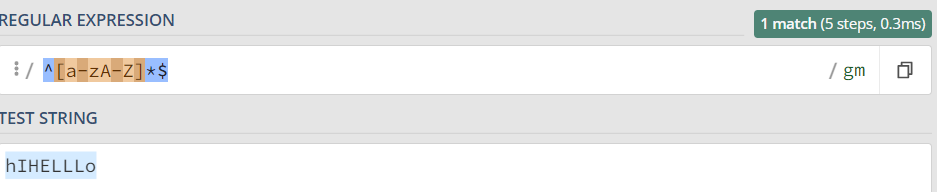
**^** - 입력 문자열의 시작에서 그 다음 나오는 문자나 부분식과 일치하는지 검사  
**$** - 문자열의 끝과 일지

\* - 앞의 문자나 부분식이 0개 이상 탐욕적으로 찾기

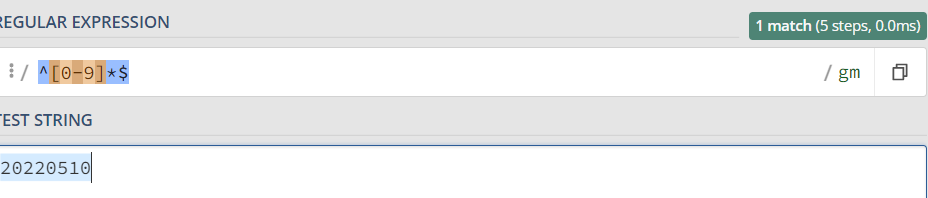
[] – “[] 사이의 문자들과 매치”라는 의미를 가짐 [] 사이에는 어떤 문자도 들어갈 수 있음

a.b – “a + 모든 문자 + b”와 같음

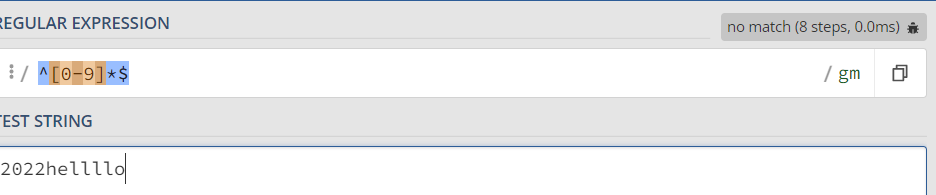
\* / + 의 차이 : \*는 없는 경우가 포함, +는 무조건 하나는 있어야 함



알파벳으로만 되어 있기에 1match가 뜸



숫자로만 되어 있기에 1match가 뜸



숫자로만 되어 있어야 하는데 문자도 섞여 있기에 no match가 뜸

**#3 Node.js 와 npm, 추가로 express 에 대해 조사해오기**

**Node.js :** Javascript를 서버에서도 사용할 수 있도록 만든 프로그램 (스크립트 언어 x, 프로그램 환경임)

웹서버와 같이 확장성 있는 네트워크 프로그램을 제작하기 위해 만들어졌음

**웹 서버에서 아파치 등의 별도 소프트웨어 없이 동작하는 것이 가능**

**장점:** 자바스크립트를 동일하게 사용해서 서버단 로직을 처리할 수 있음

새로운 언어 습득 X, 자바스크립트를 활용해서 서버기술 개발,응용 가능

이벤트 기반 비동기방식이라 서버 무리 적음

Npm 을 통한 다양한 모듈 제공

**단점:** 이벤트 기반 비동기방식이라 서버단 로직이 복잡한 경우 콜백함수 발생 가능

(콜백함수 – 다른 함수의 인자로써 넘겨진 후 특정 이벤트에 의해 호출되는 함수)

코드를 순차적으로 실행 X, 비동기 방식으로 이벤트 보내고, 응답이 오면 처리

→ JAVA 개발을 했던 방식으로 설계

단일 쓰레드이기 때문에 하나의 작업 자체가 많이 걸리는 웹서비스에는 어울리지 않음

(게시판 같은 I/O가 많은 웹서비스 GOOD)

**NPM : (Node Package Manager)** https://www.npmjs.com/

Node.js의 기본 패키지 관리자 역할을 함

Npm을 통해 패키지 또는 모듈이라고 불리는 자바스크립트 소프트웨어를 자신의 프로젝트에 설치 할 수 있음

**구성**

**Npm 웹사이트** : 패키지를 찾을 수 있음 npm<패키지 이름> 검색시 해당 패키지의 저장소 페이지가 나옴

Package – npm에 배포한 코드를 말함 ex. Angular, react

**CLI(Command Line Interface) :** 터미널에서 cli 작동 가능

**저장소 :** 패키지들을 모아놓은 공개 저장소가 존재함. 계정 업그레이드 시 비공개 저장소도 사용 가능

**Express:** Node.js를 사용하여 쉽게 서버를 구성할 수 있게 만든 클래스와 라이브러리의 집합체 ( Node.js를 위한 빠르고 개방적인 간결한 웹 프레임워크)

프레임워크이므로 웹 애플리케이션을 만들기 위한 각종 라이브러리와 미들웨어 등이 내장되어 있어 개발이 편함.

수많은 개발자들에게 개발 규칙을 강제하여 코드 및 구조의 통일성을 향상 시킬 수 있음

HTTP 통신 요청 (Request, GET, POST 등)에 대한 핸들러를 만듦

(핸들러 – 자바 핸들러는 자바 프로그램 진행 중 발생한 로그를 출력하는 방법을 다루는 클래스)

Node.js에서 많이 사용하는 템플릿 엔진인 **EJS**를 이용 가능

EJS – (Embedded JavaScript) 자바스크립트가 내장되어 있는 html 파일