**CAT-CERT 2022 Web Study #4**

**202221162 김민재**

**목차**

1. 5.2 스터디 실습
2. 조사과제
3. save/load
4. node.js 스마트 출석
5. **5.2 실습**

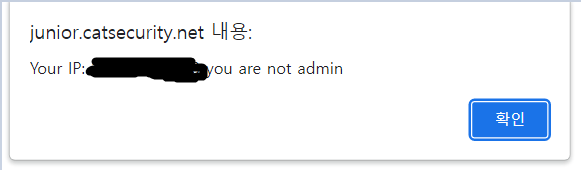
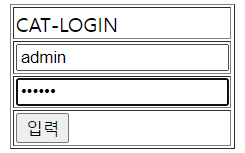
**#1 로그인 인증 페이지**

**compare.php**

|  |
| --- |
| <?php  if ($\_GET['psw'] == "wnsldj") {//주니어  if ($\_GET['id'] == "admin") {  if ($\_SERVER['REMOTE\_ADDR'] == "192.168.2.254") {  setcookie("login", "Hello CAT-CERT Admin", time() + 3600);  }  else {  setcookie("login", "notAdmin", time() + 3600); }  }  else setcookie("login", "Hello CAT-CERT", time() + 3600);  }  else {  setcookie("login", "wrong password!", time() + 3600);  }  header('Location: ./main.php');  ?> |

**main.php**

|  |
| --- |
| <?php  if ($\_COOKIE['login'] == 'notAdmin') echo "<script>alert('Your IP: ".$\_SERVER['REMOTE\_ADDR']." you are not admin');</script>";  else echo $\_COOKIE['login'];  ?>  <html>  <head>  <title> web4 tr1 22김민재 </title>  </head>  <body>  <form method="get" action="./compare.php">  <table border="1px">  <tr>  <td>CAT-LOGIN  </td>  </tr>  <tr>  <td>  <input type="text" name="id"/>  </td>  </tr>  <tr>  <td>  <input type="password" name="psw"/>  </td>  </tr>  <tr>  <td>  <input type="submit" value="입력"/>  </td>  </tr>  </table>  </form>  </body>  </html> |

****

**#2 node.js 와 express 를 사용해 웹페이지 띄워보기**

**main.js.**

|  |
| --- |
| var express = require("express");  var app = express();  app.listen(3000, function () {  console.log("listening on port 3000");  });  app.get("/", function (req, res) {  res.sendFile(\_\_dirname + "/index.html");  }); |

**index.php**

|  |
| --- |
| <html>  <head>  <meta charset="utf-8">  <title> Web4 training#2 </title>  </head>  <body>  <h1>2022 node.js study<h1>  </body>  </html> |

****

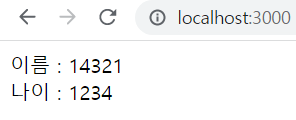
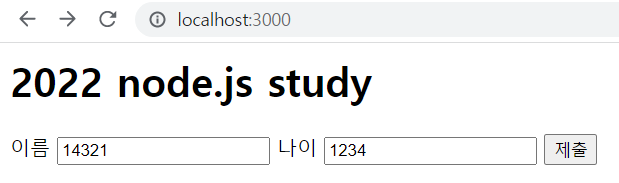
**#3**

**main.js**

|  |
| --- |
| var express = require("express");  var app = express();  app.use(express.urlencoded({ extended: true }));  app.listen(3000, () => {  console.log("listening on port 3000");  });  app.get("/", (req, res) => {  res.sendFile(\_\_dirname + "/req.html");  });  app.post("/", (req, res) => {  res.send("이름 : " + req.body.name + "<br>나이 : " + req.body.age);  }); |

**req.html**

|  |
| --- |
| <html>  <head>  <meta method="utf-8">  <title> Web4 training#2 </title>  </head>  <body>  <h1>2022 node.js study</h1>  <form method="POST" action="">  이름 <input type="text" name="name"/>  나이 <input type="text" name="age"/>  <input type="submit" value="제출"/>  </form>  </body>  </html> |

****

1. **조사 과제**

**#1 Web Storage**

**- Web Storage**

클라이언트에 데이터를 저장할 수 있도록 지원하는 HTML5의 새로운 기능. 기능은 쿠키와 유사하지만 쿠키는 저장 공간이 약 4KB이고, 웹 스토리지는 최소 5MB 이상의 저장 공간을 이용할 수 있다. 웹 스토리지 API는 sessionStorage 객체 와 localStorage 객체를 제공한다.

**- sessionStorage 객체**

하나의 세션(session)만을 위한 데이터를 저장하는 객체. 사용자가 브라우저 탭이나 창을 닫으면 이 객체에 저장된 데이터는 사라진다.

**- localStorage 객체**

보관 기한이 없는 데이터를 저장하는 객체. 브라우저 탭이나 창이 닫히거나, 컴퓨터를 재부팅해도 저장된 데이터는 없어지지 않는다.

**- 예제**

**WebStorage.html**

|  |
| --- |
| <html>  <head>  <meta charset="utf-8">  <title> Web Study #4 - Web Storage </title>  </head>  <body>  <h1> Web Study #4 - Web Storage </h1>  <button onclick="clickSessionCounter()" type="button">Session Counter</button>  <div id="SessionCounter"></div><br>  <button onclick="clickLocalCounter()" type="button">Local Counter</button>  <div id="LocalCounter"></div>  </body>  </html>  <script>  function clickSessionCounter() { //창을 닫으면 카운터가 초기화됨  if (sessionStorage.clickcount) {  sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1;  }  else {  sessionStorage.clickcount = 1;  }  document.getElementById("SessionCounter").innerHTML = sessionStorage.clickcount;  }    function clickLocalCounter() { //창을 닫아도 카운터가 초기화되지 않음  if (localStorage.clickcount) {  localStorage.clickcount = Number(localStorage.clickcount) + 1;  }  else {  localStorage.clickcount = 1;  }  document.getElementById("LocalCounter").innerHTML = localStorage.clickcount;  }  </script> |

**#2 정규표현식**

**- 정규표현식(regular expression)**

특정 패턴의 문자열을 찾기 위한 표현 방식.

**- JavaScript 정규표현식**

**형식 : /*<pattern>* /*<flag>***

- pattern : 매칭시킬 패턴

- flag : 옵션을 설정하는 플래그

**매칭 패턴**

|  |  |
| --- | --- |
| **패턴** | **의미** |
| **a-zA-Z** | 영어 알파벳(범위지정) |
| **ㄱ-g가-힣** | 한글 문자(범위지정) |
| **0-9** | 숫자(범위지정) |
| **.** | 모든 문자열 (줄 넘김 제외) |
| **\d** | 숫자 |
| **\D** | 숫자가 아닌 것 |
| **\w** | 영어 알파벳, 숫자, 언더스코어(\_) |
| **\W** | 영어 알파벳, 숫자, 언더스코어(\_)가 아닌 것 |
| **\s** | Space |
| **\S** | Space 가 아닌 것 |
| **\특수기호** | 특수 기호 |

**검색 패턴**

|  |  |
| --- | --- |
| **기호** | **의미** |
| **|** | OR |
| **[]** | 괄호 안의 문자들 중 하나 |
| **[^문자]** | 괄호 안의 문자들 제외 |
| **^문자열** | 특정 문자열로 시작 |
| **문자열$** | 특정 문자열로 끝 |
| **()** | 그룹 검색 및 분류 |
| **(?: 패턴)** | 그룹 검색 (분류X) |
| **\b** | 단어의 처음/끝 |
| **\B** | 단어의 처음/끝 이 아님 |

**수량 패턴**

|  |  |
| --- | --- |
| **기호** | **의미** |
| **?** | 최대 한 개 |
| **\*** | 있거나 없거나 |
| **+** | 최소 한 개 |
| **{n}** | n개 |
| **{Min,}** | 최소 Min개 이상 |
| **{Min, Max}** | 최소 Min개 이상, 최대 Max개 이하 |

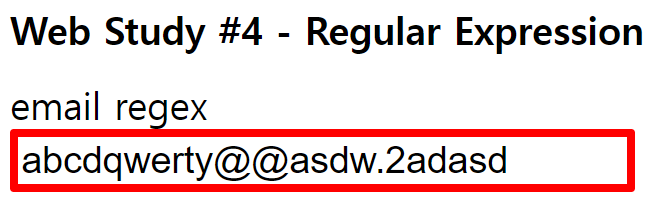
**플래그**

|  |  |
| --- | --- |
| **기호** | **의미** |
| **g (Global)** | 모든 문자 검색 |
| **i (Ignore Case)** | 대소문자 구분 X |
| **m (Multi line)** | 여러 행의 문자열에 대해 검색 |

**- 예제**

**regex.html**

|  |
| --- |
| <html>  <head>  <meta charset="utf-8">  <title> Web Study #4 - Regular Expression </title>  <style>\*{font-size:30px;}input{border-radius:3px;width:500px;}</style>  </head>  <body>  <h1> Web Study #4 - Regular Expression </h1>  email regex<br><input type="text" id="text" style="border: 7px solid red;"/>  </body>  </html>  <script>  function check() {  var text = document.getElementById("text");  if (/^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/.test(text.value)) {  text.style = "border: 7px solid green;";  }  else {  text.style = "border: 7px solid red;";  }  }  setInterval(check, 100);  </script> |

****

****

**#3 Node.js, npm, express**

**- Node.js**

확장성이 있는 네트워크어플리케이션 개발에 사용되는 소프트웨어 플랫폼. Javascript를 활용하며, Non-blocking I/O와 단일 스레드 이벤트 루프를 통한 높은 처리 성능을 가지고 있다. 내장 HTTP 서버 라이브러리를 포함하고 있어 웹 서버에서 아파치 등의 별도 소프트웨어 없이 동작하는 것이 가능하다.

**- npm (Node Package Manager)**

자바스크립트 프로그래밍 언어를 위한 패키지 관리자이자 자바스크립트 런타임 환경 Node.js의 기본 패키지 관리자. 명령 줄 클라이언트(npm), 그리고 공개 패키지와 지불 방식의 개인 패키지의 온라인 데이터베이스(npm 레지스트리)로 이루어져 있다. 이 레지스트리는 클라이언트를 통해 접근되며 사용 가능한 패키지들은 npm 웹사이트를 통해 찾아보고 검색할 수 있다.

**- express (Node Package Manager)**

Node.js의 핵심 모듈인 http와 Connect 컴포넌트를 기반으로 하는 웹 프레임워크. 다음과 같은 메커니즘을 제공한다.

- HTTP 통신 요청(Request; GET, POST, DELETE 등)에 대한 핸들러를 만든다.

- 템플릿에 데이터를 넣어 응답(response)을 만들기 위해 view의 렌더링 엔진과 결합(integrate)한다.

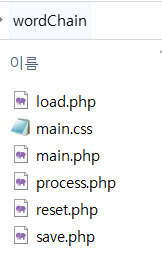
- 접속을 위한 포트나 응답 렌더링을 위한 템플릿 위치 같은 공통 웹 어플리케이션 세팅을 한다.

- 핸들링 파이프라인(reqest handling pipeline) 중 필요한 곳에 추가적인 미들웨어 처리 요청을 추가한다.

1. **save/load**

**#1 끝말잇기**

**파일구조**

****

**main.php**

|  |
| --- |
| <?php  session\_start();  if(!isset($\_SESSION['start'])) { //default setting  $\_SESSION['start'] = 1;  $\_SESSION['words'] = array('시작');  $\_SESSION['score'] = 0;  if(!isset($\_SESSION['save'])) $\_SESSION['save'] = array(array(), 0, 0); //words, score, time  }  ?>  <html>  <head>  <meta charset="utf-8">  <title>끝말잇기</title>  <link rel="stylesheet" href="./main.css">  <head>  <body>  <form method="post" action="./reset.php">  <input id="resetbtn" class="btn" type="submit" value="Restart">  </form>  <form method="post" action="./save.php">  <input id="savebtn" class="btn" type="submit" value="save">  </form>  <?php if (time() - $\_SESSION['save'][2] <= 600) { ?>  <form method="post" action="./load.php">  <input id="loadbtn" class="btn" type="submit" value="<?php echo 'load: '.($\_SESSION['save'][1] + 1).' '.$\_SESSION['save'][0][$\_SESSION['save'][1]]; ?>">  </form>  <?php } ?>  <center><br><br>  <h1>끝말잇기</h1>  <div id='div'>  <table><?php //tr을 생성하고 $\_SESSION['words']의 단어들을 마지막 인덱스부터 자례대로 집어넣는다  for ($i = $\_SESSION['score']; $i >= 0; $i--) {  echo "<tr><td>".($i + 1)."</td><td>".$\_SESSION['words'][$i]."</td></tr>";  }  ?></table>  </div>  <br><br>  <div><?php echo mb\_substr($\_SESSION['words'][$\_SESSION['score']], 0, -1, 'UTF-8')."<span>".mb\_substr($\_SESSION['words'][$\_SESSION['score']], -1, 1, 'UTF-8')."</span>"; //last word?> ->  <form id="input" method="post" action="./process.php">  <input type="text" id="word" name="word" maxlength="20" autocomplete="off" autofocus <?php echo 'value="'.$\_SESSION['word'].'"'; //if wrong, keep input?>>  </form><br><br>  </div>  </center></body>  </html> |

**main.css**

|  |
| --- |
| \* {  font-size: 20px;  }  #div {  display: inline-block;  overflow-y: scroll;  height: 500px;  border: 5px solid #000;  border-radius: 10px;  }  #div::-webkit-scrollbar {  width: 5px;  cursor: pointer;  }  #div::-webkit-scrollbar-stack {  display: none;  }  #div::-webkit-scrollbar-thumb {  width: 10px;  background-color: #ccc;  border-radius: 100px;  }  #div::-webkit-scrollbar-thumb:hover {  background-color: #aaa;  }  #div::-webkit-scrollbar-thumb:active {  background-color: #888;  }  table {  border-collapse: collapse;  width: 600px;  }  td:nth-child(odd) {  text-align: right;  width: 60px;  padding: 8px;  font-weight: bold;  }  td:nth-child(even) {  text-align: left;  padding: 8px;  }  tr:nth-child(odd) {  background-color: #f2f2f2;  }  input[type=text] {  width: 425px;  border: 0px;  border-bottom: 2px solid #000;  }  input[type=text]:focus {  outline: 0;  border-bottom: 4px solid #000;  }  #input {  display: inline-block;  }  .btn {  background-color: #000000;  border: 4px solid #000000;  color: white;  padding: 14px 20px;  cursor: pointer;  height: 60px;  }  .btn:focus {  border: 4px solid #999999;  }  .btn:hover {  background-color: #555555;  border: 4px solid #555555;  }  .btn:active {  background-color: #000000;  border: 4px solid #FFFFFF;  font-size: 17px;  }  #resetbtn {  position: absolute;  right: 10px;  top: 10px;  }  #savebtn {  position: absolute;  left: 10px;  top: 10px;  }  #loadbtn {  position: absolute;  left: 110px;  top: 10px;  }  span {  font-size: 25px;  color: red;  font-weight: bold;  } |

**process.php**

|  |
| --- |
| <?php  session\_start();  $s = $\_SESSION['score'];//마지막 단어의 끝글자와 입력한 단어의 첫글자가 같다면, 또는 한글만 입력되었다면 통과  if (mb\_substr($\_SESSION['words'][$s], -1, 1, 'UTF-8') == mb\_substr($\_POST['word'], 0, 1, 'UTF-8') && preg\_match('/^[가-힣]{6,60}$/', $\_POST['word'], $m)) {  $\_SESSION['words'][] = $\_POST['word'];  $\_SESSION['score']++;  $\_SESSION['word'] = "";  }  else $\_SESSION['word'] = $\_POST['word'];  header('Location: ./main.php');  ?> |

**save.php**

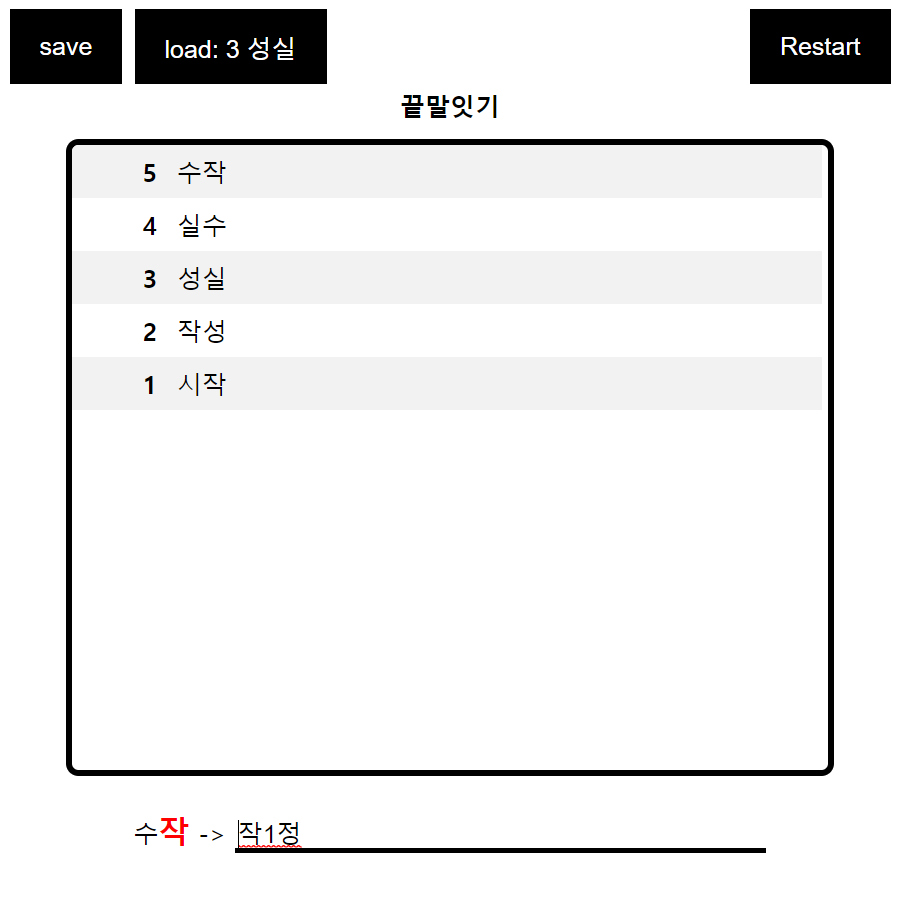
|  |
| --- |
| <?php  session\_start();  $\_SESSION['save'][0] = $\_SESSION['words'];  $\_SESSION['save'][1] = $\_SESSION['score'];  $\_SESSION['save'][2] = time();  header('Location: ./main.php');  ?> |

**load.php**

|  |
| --- |
| <?php  session\_start();  $\_SESSION['words'] = $\_SESSION['save'][0];  $\_SESSION['score'] = $\_SESSION['save'][1];  header('Location: ./main.php');  ?> |

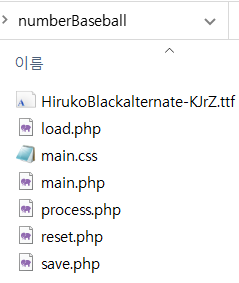
**reset.php**

|  |
| --- |
| <?php  session\_start();  $save = $\_SESSION['save'];  session\_destroy();  session\_start();  $\_SESSION['save'] = $save;  header('Location: ./main.php');  ?> |

****

**#2 숫자야구**

**파일구조**

****

**main.php**

|  |
| --- |
| <?php  session\_start();  if (!isset($\_SESSION['start'])) { //default setting  $\_SESSION['start'] = 0;  $\_SESSION['nd'] = array(); //number data  $\_SESSION['ad'] = array(); //Strike Ball Out (SBO) data  $\_SESSION['count'] = 0;  $\_SESSION['s'] = 0;  $arr = array(0,1,2,3,4,5,6,7,8,9);  shuffle($arr);  $\_SESSION['answer'] = array($arr[0], $arr[1], $arr[2]); //answer  if (!isset($\_SESSION['save'])) $\_SESSION['save'] = array(array(), array(), 0, 0, array(), 0); //nd, ad, count, s, answer, time  }  $c = $\_SESSION['count'];  $s = $\_SESSION['s'];  ?>  <html>  <head>  <meta charset="utf-8">  <title>Number Baseball</title>  <link rel="stylesheet" href="./main.css">  </head>  <body>  <form method='post' action='./reset.php'>  <input type='submit' id='resetbtn' class='btn' value='Reset'>  </form>  <form method='post' action='./save.php'>  <input id='savebtn' class='btn' type='submit' value='save'>  </form>  <?php if (time() - $\_SESSION['save'][5] <= 600) { ?>  <form method='post' action='./load.php'>  <input id='loadbtn' class='btn' type='submit' value='load'>  </form>  <?php } ?>  <center>  <?php if ($s == 3 || $c == 10) {  ?> <div id='alert'><?php  echo ($s == 3) ? 'YOU WIN!' : 'GAME OVER';  ?><span><br><?php  echo ($s == 3) ? 'attemps : '.$c : (($c == 10) ? 'answer : '.$\_SESSION['answer'][0].$\_SESSION['answer'][1].$\_SESSION['answer'][2] : '');  ?></span></div><?php } ?>  <div id='title'>Number Baseball</div>  <div id='nums'><?php $np = 0; $ap = 0;  for ($i = 0; $i < $c - 1; $i++) { //already done divs  for ($j = 0; $j < 3; $j++)  echo '<div class=\''.$\_SESSION['ad'][$ap++].'\'>'.$\_SESSION['nd'][$np++].'</div>';  echo '<br>';  }  if ($c > 0) { //current input div  echo '<span id=\'ans\'>';  for ($i = 0; $i < 3; $i++)  echo '<div>'.$\_SESSION['nd'][$np++].'</div>';  echo '<br></span>';  }  if ($s < 3 && $c < 10) echo '<span id=\'input\'><div></div><div></div><div></div><br></span>'; //to recieve div  else if ($c < 10) echo '<div></div><div></div><div></div><br>';  for ($i = 0; $i < 9 - $c; $i++) //blank divs  echo '<div></div><div></div><div></div><br>';  ?></div>  </center></body>  <?php //recieve post only if game is not over  if ($s < 3 && $c < 10) { ?>  <form method='post' action='./process.php' id='sub'><input type='hidden' name='1'><input type='hidden' name='2'><input type='hidden' name='3'></form>  <?php }  ?>  </html>  <script>  function dec(div, tmp, c) { //flip, decreasing  div.style = 'transform: scale(1, ' + tmp + ')';  if (tmp > 0) setTimeout(dec, 5, div, tmp - 0.05, c);  else {  div.className = c;  setTimeout(inc, 5, div, tmp + 0.05);  }  }  function inc(div, tmp) { //flip, increasing  div.style = 'transform: scale(1, ' + tmp + ')';  if (tmp < 1) setTimeout(inc, 5, div, tmp + 0.05);  }  <?php if ($c > 0) { ?> //set className current input  var ans = document.getElementById('ans');  <?php  for ($i = 0; $i < 3; $i++)  echo 'setTimeout(dec, 0, ans.childNodes['.$i.'], 1.0, \''.$\_SESSION['ad'][$ap++].'\');';  }  ?>  <?php if ($s < 3 && $c < 10) { ?>  var ptr = 0;  var list = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]; //not typed number list  window.addEventListener("keydown", e => { //detect keyboard input  if (isFinite(e.key)) { //if number, write number  if (ptr < 3 && list[e.key]) document.getElementById('input').childNodes[ptr++].innerHTML = e.key;  list[e.key] = 0;  }  else if (e.key == 'Backspace' && ptr > 0) { //if backspace, remove current input  list[parseInt(document.getElementById('input').childNodes[--ptr].innerHTML)] = 1;  document.getElementById('input').childNodes[ptr].innerHTML = '';  }  else if (e.key == 'Enter' && ptr == 3) { //if enter, send POST  for (var i = 0; i < 3; i++)  document.getElementById('sub').childNodes[i].value = document.getElementById('input').childNodes[i].innerHTML;  document.getElementById('sub').submit();  }  if (ptr == 0) list = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]; //reset list if input is empty  });  <?php } else { ?>  function end(tmp) {  document.getElementById('alert').style.color = 'rgba(255, 255, 255, ' + tmp + ')';  document.getElementById('alert').childNodes[1].style.color = 'rgba(255, 255, 255, ' + tmp + ')';  document.getElementById('alert').style.backgroundColor = 'rgba(155, 155, 155, ' + (tmp / 2) + ')';  document.getElementById('alert').style.bottom = (320 + tmp \* 50) + 'px';  if (tmp < 1) setTimeout(end, 10, tmp + 0.03);  }  setTimeout(setTimeout, 500, end, 10, 0.01);  <?php } ?>  </script> |

**main.css**

|  |
| --- |
| @font-face {  src : url("./HirukoBlackalternate-KJrZ.ttf");  font-family: "cir";  }  \* {  color: #fff;  overflow-x: hidden;  overflow-y: hidden;  z-index: 1;  }  body {  background-color: #111;  }  form {  margin: 0;  }  .btn {  background-color: #ddd;  border: 4px solid #ddd;  color: #000;  padding: 12px 16px;  cursor: pointer;  font-size: 20px;  font-weight: bold;  }  .btn:hover {  background-color: #fff;  border: 4px solid #fff;  }  .btn:active {  background-color: #ddd;  border: 4px solid #000;  font-size: 17px;  }  #resetbtn {  position: absolute;  right: 10px;  bottom: 10px;  }  #savebtn {  position: absolute;  left: 10px;  bottom: 10px;  }  #loadbtn {  position: absolute;  left: 110px;  bottom: 10px;  }  #title {  font-size: 40px;  font-weight: bold;  }  #nums div {  font-family: "cir";  border: 2px solid #444;  height: 50px;  width: 50px;  display: inline-block;  font-size: 40px;  line-height: 50px;  margin: 2px;  }  .s {  background-color: #c9b458;  border: 2px solid #c9b458 !important;  }  .b {  background-color: #6aaa64;  border: 2px solid #6aaa64 !important;  }  .o {  background-color: #f53916;  border: 2px solid #f53916 !important;  }  #alert {  font-family: "cir";  margin: 0 50%;  position: absolute;  left: -170px;  width: 300px;  background-color: rgba(155, 155, 155, 0);  color: rgba(255, 255, 255, 0);  padding: 20px;  border-radius: 5px;  font-size: 50px;  z-index: 2;  }  #alert span {  font-size: 20px;  color: rgba(255, 255, 255, 0);  } |

**process.php**

|  |
| --- |
| <?php  session\_start();  $c = ++$\_SESSION['count'];  $s = 0; $b = 0;  $p = array($\_POST['1'], $\_POST['2'], $\_POST['3']);  for ($i = 0; $i < 3; $i++) {  $\_SESSION['nd'][] = $p[$i]; //append POSTs to nd  for ($j = 0; $j < 3; $j++) { //calculate SBO  if ($\_SESSION['answer'][$i] == $p[$j]) {  if ($i == $j) $s++;  else $b++;  }  }  }  for ($i = 0; $i < $s; $i++) //append SBO to ad  $\_SESSION['ad'][] = 's';  for ($i = 0; $i < $b; $i++)  $\_SESSION['ad'][] = 'b';  for ($i = 0; $i < 3 - $s - $b; $i++)  $\_SESSION['ad'][] = 'o';  $\_SESSION['s'] = $s;  header('Location: ./main.php');  ?> |

**save.php**

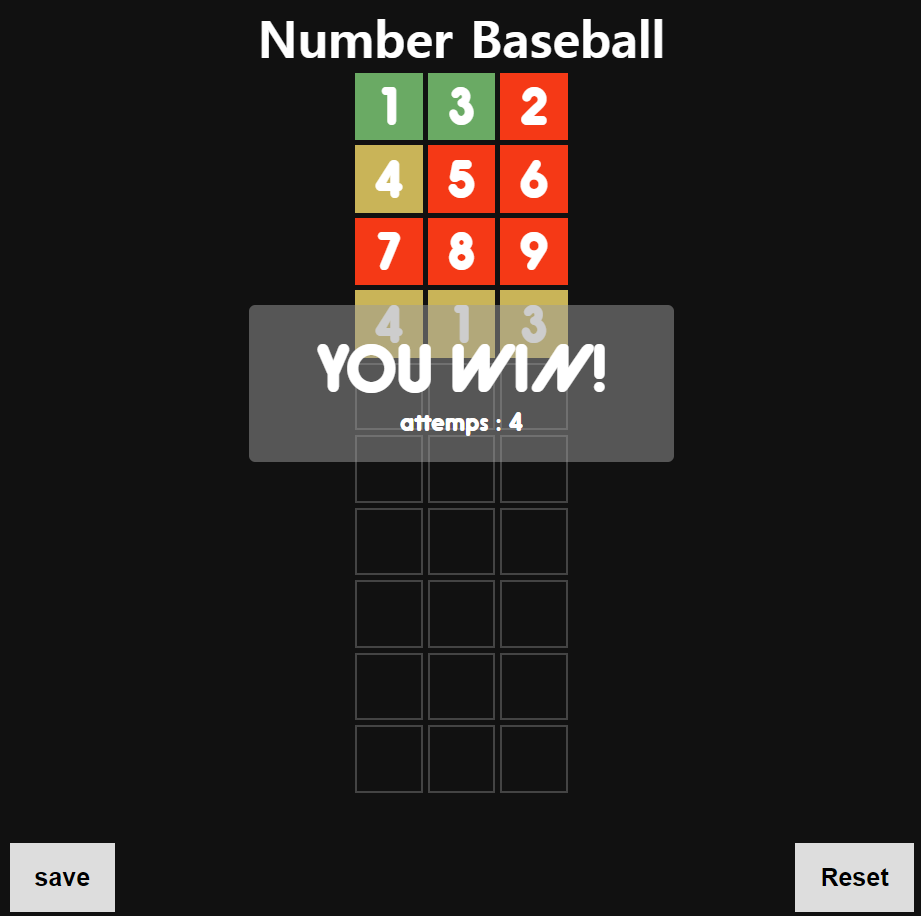
|  |
| --- |
| <?php  session\_start();  $\_SESSION['save'][0] = $\_SESSION['nd'];  $\_SESSION['save'][1] = $\_SESSION['ad'];  $\_SESSION['save'][2] = $\_SESSION['count'];  $\_SESSION['save'][3] = $\_SESSION['s'];  $\_SESSION['save'][4] = $\_SESSION['answer'];  $\_SESSION['save'][5] = time();  header('Location: ./main.php');  ?> |

**load.php**

|  |
| --- |
| <?php  session\_start();  $\_SESSION['nd'] = $\_SESSION['save'][0];  $\_SESSION['ad'] = $\_SESSION['save'][1];  $\_SESSION['count'] = $\_SESSION['save'][2];  $\_SESSION['s'] = $\_SESSION['save'][3];  $\_SESSION['answer'] = $\_SESSION['save'][4];  header('Location: ./main.php');  ?> |

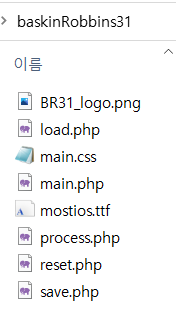
**reset.php**

|  |
| --- |
| <?php  session\_start();  $save = $\_SESSION['save'];  session\_destroy();  session\_start();  $\_SESSION['save'] = $save;  header('Location: ./main.php');  ?> |

****

**#3 Baskin Robbins 31**

**파일구조**

****

**main.php**

|  |
| --- |
| <?php  session\_start();  if (!isset($\_SESSION['start'])) { //default setting  $\_SESSION['start'] = 0;  $\_SESSION['count'] = 0;  $\_SESSION['ans'] = 0; //cpu answer  $\_SESSION['f'] = 0; //flag  if (!isset($\_SESSION['save'])) $\_SESSION['save'] = array(0, 0, 0, 0); //count, ans, f, time  }  $count = $\_SESSION['count'];  $ans = $\_SESSION['ans'];  $f = $\_SESSION['f'];  ?>  <html>  <head>  <meta charset="utf-8">  <title>Baskin Robbins 31</title>  <link rel="stylesheet" href="./main.css">  <head>  <body>  <form method="post" action="./reset.php">  <input id="resetbtn" class="btn" type="submit" value="Reset">  </form>  <form method="post" action="./save.php">  <input id="savebtn" class="btn" type="submit" value="Save">  </form>  <?php if (time() - $\_SESSION['save'][3] <= 600) { ?>  <form method="post" action="./load.php">  <input id="loadbtn" class="btn" type="submit" value="<?php echo 'Load: '.($\_SESSION['save'][0] + $\_SESSION['save'][1]); ?>">  </form>  <?php } ?>  <center><div id="board">  <img id="logo" src="./BR31\_logo.png">  <form method="post" action="./process.php">  <?php //output  echo 'count: '.$count.'<br>'; //current count value  echo '<span>cpu</span>: '; //cpu answer  if ($count) { //if game is in progress  for ($i = 0; $i < $ans; $i++) {  if ($i) echo ', ';  echo ($count + $i + 1);  }  echo '<br>';  echo '<br>count: '.($count += $ans).'<br>'; //fixed count value  }  else { //if game started now  echo 'Your turn.<br>';  }  if (!$f) { //if game didn't end  echo '<button class="nbtn" name="input" value="1">'.($count + 1).'</button>';  if ($count < 30) echo '<button class="nbtn" name="input" value="2">'.($count + 2).'</button>';  if ($count < 29) echo '<button class="nbtn" name="input" value="3">'.($count + 3).'</button>';  }  else if ($f == 1) { //if win  echo 'You Win!';  }  else { //if lose  echo 'You Lose!';  }  ?>  </form>  </div></center>  </body>  </html> |

**main.css**

|  |
| --- |
| @font-face {  src : url("./mostios.ttf");  font-family: "br";  }  \* {  font-family: "br";  overflow-x: hidden;  overflow-y: hidden;  z-index: 1;  }  body {  background-color: rgb(240,77,163);  }  .wrapper, body {  display: flex;  align-items: center;  justify-content: center;  }  #board {  display: inline-block;  width: 450px;  height: 300px;  border: 15px solid rgb(64,32,34);  border-radius: 30px;  background-color: rgb(187,249,248);  font-size: 30px;  color: rgb(64,32,34);  }  #logo {  height: 100px;  width: 400px;  }  span {  color: rgb(240,77,163);  font-weight: bold;  }  form {  margin: 0;  }  .nbtn {  width: 40px;  height: 40px;  padding: 5px;  background-color: rgb(94,62,64);  border: none;  border-radius: 10px;  color: #fff;  cursor: pointer;  }  .btn {  background-color: rgb(64,32,34);  border: 4px solid rgb(64,32,34);  padding: 12px 16px;  cursor: pointer;  color: #fff;  font-size: 20px;  font-weight: bold;  }  .btn:hover {  background-color: rgb(94,62,64);  border: 4px solid rgb(94,62,64);  }  .btn:active {  background-color: #100;  border: 4px solid #100;  font-size: 17px;  }  #resetbtn {  position: absolute;  right: 10px;  bottom: 10px;  }  #savebtn {  position: absolute;  left: 10px;  bottom: 10px;  }  #loadbtn {  position: absolute;  left: 110px;  bottom: 10px;  } |

**process.php**

|  |
| --- |
| <?php  session\_start();  $\_SESSION['count'] += $\_SESSION['ans'] + $\_POST['input'];  if ($\_SESSION['count'] > 30) { //if player lose  $\_SESSION['f'] = 2;  $\_SESSION['ans'] = 0;  }  else { //if player not lose  if ($\_SESSION['count'] % 4 == 2) { //claculate cpu answer  $\_SESSION['ans'] = $\_SESSION['count'] % 3 + 1;  }  else {  $\_SESSION['ans'] = 4 - ($\_SESSION['count'] + 2) % 4;  }  if ($\_SESSION['count'] + $\_SESSION['ans'] > 30) { //if player win  $\_SESSION['f'] = 1;  }  }  header('Location: ./main.php');  ?> |

**save.php**

|  |
| --- |
| <?php  session\_start();  $\_SESSION['save'][0] = $\_SESSION['count'];  $\_SESSION['save'][1] = $\_SESSION['ans'];  $\_SESSION['save'][2] = $\_SESSION['f'];  $\_SESSION['save'][3] = time();  header('Location: ./main.php');  ?> |

**load.php**

|  |
| --- |
| <?php  session\_start();  $\_SESSION['count'] = $\_SESSION['save'][0];  $\_SESSION['ans'] = $\_SESSION['save'][1];  $\_SESSION['f'] = $\_SESSION['save'][2];  header('Location: ./main.php');  ?> |

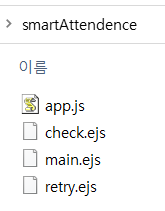
**reset.php**

|  |
| --- |
| <?php  session\_start();  $save = $\_SESSION['save'];  session\_destroy();  session\_start();  $\_SESSION['save'] = $save;  header('Location: ./main.php');  ?> |

****

1. **node.js 스마트 출석**

**파일구조**

****

**app.js**

|  |
| --- |
| var express = require("express");  const path = require("path");  var app = express();  app.use(express.urlencoded({ extended: true }));  app.set("view engine", "ejs");  app.set("views", path.join(\_\_dirname, "/"));  app.listen(3000, () => {  console.log("listening on port 3000");  });  const num = Math.floor(Math.random() \* 1000);  global.num = num;  console.log(num);  app.get("/", (req, res) => {  res.render("main", { number: num });  });  app.post("/", (req, res) => {  if (req.body.input == num) res.render("check");  else res.render("retry");  }); |

**main.ejs**

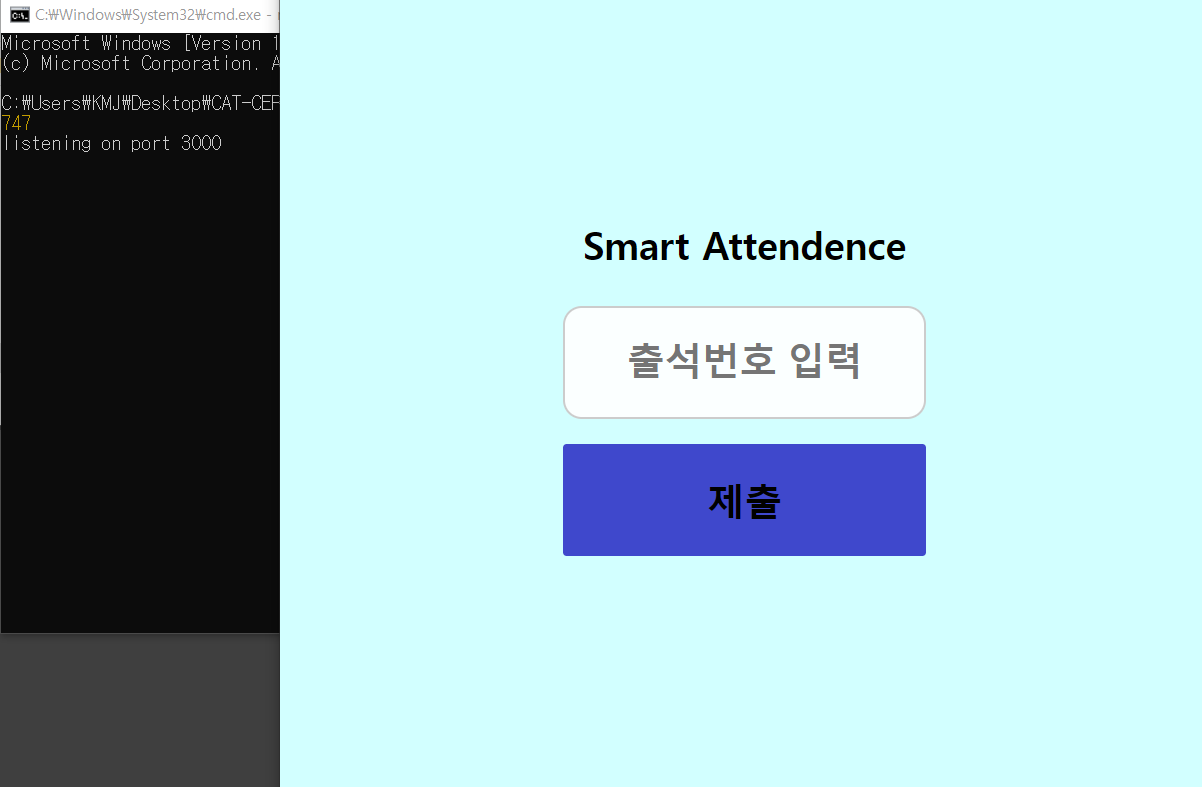
|  |
| --- |
| <script> alert('출석번호: <%= number %>'); </script>  <html>  <head>  <meta charset="utf-8">  <title> Web4 smartAttendence </title>  <style>  \* {  font-size: 30px;  font-weight: bold;  }  body {  display: flex;  justify-content: center;  align-items: center;  background-color: rgb(210,255,255);  }  div {  display: inline-block;  }  input {  width: 290px;  height: 90px;  padding: 20px 30px;  margin: 10px;  }  input[type=text] {  background-color: rgba(255,255,255,0.9);  border: 2px solid #ccc;  border-radius: 15px;  text-align: center;  }  input[type=submit] {  background-color: rgb(63,72,204);  border-radius: 3px;  border: none;  }  </style>  </head>  <body><center>  <h1>Smart Attendence</h1>  <div>  <form method="post" action="">  <input type="text" name="input" placeholder="출석번호 입력"/><br>  <input type="submit" value="제출"/>  </form>  </div>  </center></body>  </html> |

**check.ejs**

|  |
| --- |
| <script type="text/javascript">  alert("출석 완료");  history.back();  </script> |

**retry.ejs**

|  |
| --- |
| <script>  alert("출석번호를 다시 확인하세요");  history.back();  </script> |

****