Одесский Национальный Политехнический Университет

Кафедра информационных систем

Лабораторная работа № 8

по дисциплине: «Веб-технологии и веб-дизайн»

## на тему «Java Script сценарии. Регулярные выражения»

Выполнил:

Ст. группы АИ-166

Дидух Э. Г.

Проверили:

Червоненко П. П.

Панькина А. С.

Одесса, 2018

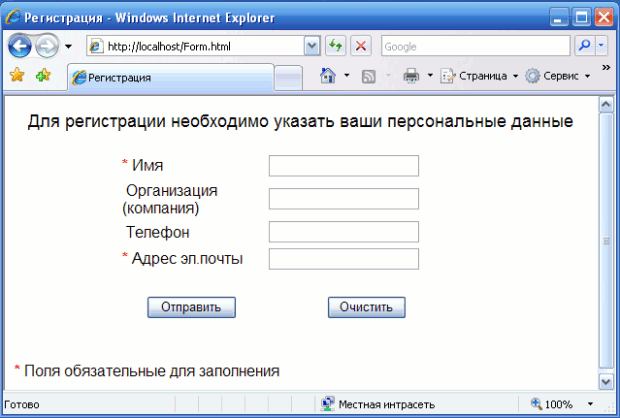
**ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ:**

Цель работы: получить представление об общих принципах обработки в JavaScript событий, связанных с окном веб-браузера, веб-страницей, содержащейся в браузере и элементами документа.

Научиться использовать простейшие элементы регулярных выражений для поиска подстрок, структура и содержание которых описывается нетривиальным шаблоном.

Практические задания

Задание 1:

Проверка значений, введенных пользователем в поля формы для регистрации. Для выполнения лабораторной работы необходимо создать веб-страницу, содержащую форму с полями, следующего вида: ****

В тэге <form> добавьте обработчик события отправки данных вида:

onSubmit = "CheckData(); return false;"

В данном случае указана функция обработчик CheckData(). Оператор return false; предотвращает автоматическую отправку данных после выполнения функции-обработчика. Отправка данных будет выполняться из обработчика.

Добавьте на странице секцию JavaScript кода, описывающего функцию-обработчик:

function CheckData()

{

var ans;

ans = confirm("Вы уверены, что хотите отправить введенные данные ?");

if (ans) submit();

}

Как это видно из кода, функция CheckData() в случае подтверждения со стороны пользователя самостоятельно вызывает метод submit() для передачи данных из формы.

Теперь необходимо добавить проверку значений, введенных в поля формы пользователем.

Прежде всего, необходимо убедиться в том, что заполнены все поля, обязательные для ввода. Для этого можно использовать проверку на равенство нулю длины строки (свойство length ), являющейся значением узла дерева документа, соответствующего полю ввода, например:

document.getElementById("uname").value.length.

Следующая проверка должна контролировать структуру и содержимое полей. Для этого можно использовать объект RegExp, например:

var validEMail, pattn;

pattn = new RegExp("^[\.\-\_A-Za-z0-9]+?@[\.\-A-Za-z0-9]+?\.[A-Za-z0-9]{2,6}$");

validEMail = pattn.test(document.getElementById("email").value));

В данном фрагменте описана проверка структуры электронного адреса из поля формы с идентификатором "email". Для проверки был использован шаблон на основе регулярного выражения.

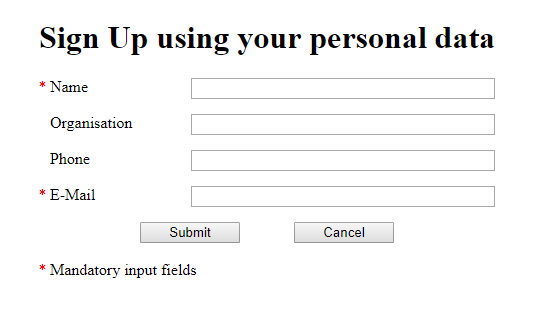
Задание 2:

Самостоятельно постройте регулярное выражение, описывающее шаблон для проверки, адреса, номера телефона, адреса электронной почты, и внесите все необходимые изменения и дополнения в функцию CheckData().

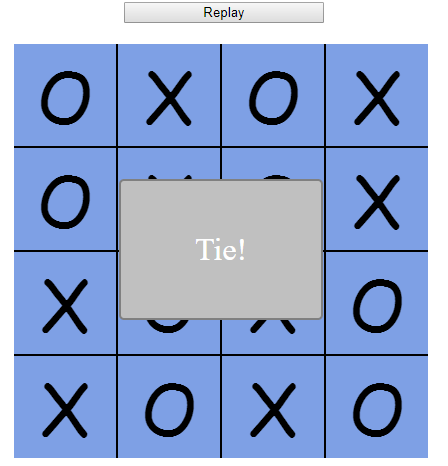
Задание 3:

Написать игру «Крестики-нолики» с использованием атрибутов событий. Минимальное требование: выбор ячейки для размещения крестика или нолика компьютером сделать зависящим от выбора ячейки игроком (по крайней мере осуществить выбор соседней с выбранной игроком, но более приемлемым для сдачи работы будет код, предотвращающий победу игрока). Задать размер поля 4х4, добавить таймер, который будет фиксировать время каждого хода игрока и время игры в целом. Необходимое условие победы – расположение трех крестиков или ноликов рядом по диагонали, горизонтали или вертикали.

Страницы HTML:



|  |
| --- |
| Default.html |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta content="text/html; charset=utf-8" http-equiv="Content-Type" />  <title>JS Event Handling</title>  <script type="text/javascript">  function CheckData()  {  var name, org, phone, email;    var p1 = new RegExp("^[a-zA-Z ]+$");  var p2 = new RegExp("^[0-9]\*$");  var p3 = new RegExp("^[\.\-\_A-Za-z0-9]+?@[\.\-A-Za-z0-9]+?\.[A-Za-z0-9]{2,6}$");    name=document.getElementById("name").value;  org=document.getElementById("org").value;  phone=document.getElementById("phone").value;  email=document.getElementById("email").value;    if(document.getElementById("name").value.length == 0 || document.getElementById("email").value.length == 0)  {  alert("Mandatory fields must be filled in!");  reset();  }  if(!p1.test(name) || !p1.test(org) || !p2.test(phone) || !p3.test(email))  {  alert("Error in one or multiple input fields!");  reset();  }  else  {  var ans;  ans = confirm("Are you sure you want to submit current data?");  if (ans) submit();  }  }  </script>  </head>  <body>  <div style="text-align:center;">  <h1>Sign Up using your personal data</h1>    <div style="width:60%; position:relative; left:20%;">    <form onsubmit="CheckData(); return false;">    <label id="Label1" style="color:red; position:absolute; left: 30%;">\*</label><label id="Label2" style="position:absolute; left: 31%;">Name </label>    <input id="name" type="text" style="position:absolute; right: 30%; width: 300px;"/>  <br/>  <br/>    <label id="Label1" style="color:red; position:absolute; left: 30%;"></label><label id="Label2" style="position:absolute; left: 31%;">Organisation </label>    <input id="org" type="text" style="position:absolute; right: 30%; width: 300px;"/>  <br/>  <br/>    <label id="Label1" style="color:red; position:absolute; left: 30%;"></label><label id="Label2" style="position:absolute; left: 31%;">Phone </label>    <input id="phone" type="text" style="position:absolute; right: 30%; width: 300px;"/>  <br/>  <br/>    <label id="Label1" style="color:red; position:absolute; left: 30%;">\*</label><label id="Label2" style="position:absolute; left: 31%;">E-Mail </label>    <input id="email" type="text" style="position:absolute; right: 30%; width: 300px;"/>  <br/>  <br/>    <input name="submit" type="submit" value="Submit" style="width:100px;margin-right:25px;"/>  <input name="reset" type="reset" value="Cancel" style="width:100px;margin-left:25px;"/>    </form>  <br/>    <label id="Label1" style="color:red; position:absolute; left: 30%;">\*</label><label id="Label2" style="position:absolute; left: 31%; text-weight:bold;">Mandatory input fields </label>    </div>    </div>  </body>  </html> |



|  |
| --- |
| TicTacToe.html |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta content="text/html; charset=utf-8" http-equiv="Content-Type" />  <title>Tic Tac Toe</title>  <link rel="stylesheet" href="style.css" />  </head>  <body>    <table>    <tr>  <td class="cell" id="0"></td>  <td class="cell" id="1"></td>  <td class="cell" id="2"></td>  <td class="cell" id="3"></td>  </tr>    <tr>  <td class="cell" id="4"></td>  <td class="cell" id="5"></td>  <td class="cell" id="6"></td>  <td class="cell" id="7"></td>  </tr>    <tr>  <td class="cell" id="8"></td>  <td class="cell" id="9"></td>  <td class="cell" id="10"></td>  <td class="cell" id="11"></td>  </tr>    <tr>  <td class="cell" id="12"></td>  <td class="cell" id="13"></td>  <td class="cell" id="14"></td>  <td class="cell" id="15"></td>  </tr>    </table>    <div class="endgame">  <div class="text"></div>  </div>    <button onclick="startGame()" style="position:absolute; left:50%; width:200px; margin-left:-100px;">Replay</button>    <script src="TTT\_Script.js"></script>  </body>  </html> |

|  |
| --- |
| Style.css |
| td{  border:2px solid black;  height: 100px;  width: 100px;  text-align: center;  vertical-align: middle;  font-family: "Comic Sans MS";  font-size: 70px;  cursor: pointer;  }  table{  border-collapse: collapse;  position: absolute;  left: 50%;  margin-left: -210px;  top: 50px;  }  table tr:first-child td{  border-top:0;  }  table tr:last-child td{  border-bottom:0;  }  table tr td:first-child{  border-left:0;  }  table tr td:last-child{  border-right:0;  }  .endgame{  display: none;  width: 200px;  top: 185px;  background-color: silver;  border: 2px solid gray;  position: absolute;  left: 50%;  margin-left: -105px;  padding-top: 50px;  padding-bottom: 50px;  text-align: center;  border-radius: 5px;  color: white;  font-size: 2em;  } |

|  |
| --- |
| TT\_Script.js |
| var origBoard;  const huPlayer = 'O';  const aiPlayer = 'X';  const winCombos = [  [0,1,2],  [1,2,3],  [4,5,6],  [5,6,7],  [8,9,10],  [9,10,11],  [12,13,14],  [13,14,15],  [0,5,10],  [5,10,15],  [1,6,11],  [4,9,14],  [2,5,8],  [3,6,9],  [6,9,12],  [7,10,13],  [0,4,8],  [4,8,12],  [1,5,9],  [5,9,13],  [2,6,10],  [6,10,14],  [3,7,11],  [7,11,15]  ]  const cells = document.querySelectorAll('.cell');  startGame();  function startGame()  {  document.querySelector(".endgame").style.display = "none";  origBoard = Array.from(Array(16).keys());  for (var i =0; i < cells.length; i++)  {  cells[i].innerText = '';  cells[i].style.removeProperty('background-color');  cells[i].addEventListener('click', turnClick, false);  }  }  function turnClick(square)  {  if(typeof origBoard[square.target.id] == 'number')  {  turn(square.target.id, huPlayer);  if(!checkTie()) turn(bestSpot(), aiPlayer);  checkTie();  }  }  function turn(squareId, player)  {  origBoard[squareId] = player;  document.getElementById(squareId).innerText = player;  let gameWon = checkWin(origBoard, player);  if(gameWon) gameOver(gameWon);  }  function checkWin(board, player)  {  let plays = board.reduce((a,e,i)=>  (e === player) ? a.concat(i):a,[]);    let gameWon = null;  for(let [index, win] of winCombos.entries())  {  if(win.every(elem => plays.indexOf(elem) > -1))  {  gameWon = {index: index, player: player};  break;  }  }  return gameWon;  }  function gameOver(gameWon)  {  for (let index of winCombos[gameWon.index])  {  document.getElementById(index).style.backgroundColor =  gameWon.player == huPlayer ? "#45b269" : "#931616";  }    for(var i = 0; i < cells.length; i++)  {  cells[i].removeEventListener('click', turnClick, false);  }  declareWinner(gameWon.player == huPlayer ? "Victory!" : "Defeat!");  }  function declareWinner(who)  {  document.querySelector(".endgame").style.display = "block";  document.querySelector(".endgame .text").innerText = who;  }  function emptySquares()  {  return origBoard.filter(s => typeof s == 'number');  }  function bestSpot()  {  return emptySquares()[0];  }  function checkTie()  {  console.log(emptySquares());  if (emptySquares().length == 0)  {  for(var i = 0; i < cells.length; i++)  {  cells[i].style.backgroundColor = "#7ea0e5";  cells[i].removeEventListener('click', turnClick, false);  }  declareWinner("Tie!");  return true;  }  return false;  } |

**Выводы:** В данной лабораторной работе мы ознакомились c обработкой событий и регулярными выражениями в JavaScript и написанием сценариев динамического взаимодействия со страницей. С помощью полученных знаний создали целостный сайт в соответствии с вариантом и требованиям лабораторной работы используя оформление тегов HTML через стили CSS и сценарии JS, следовательно цель работы достигнута.