МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ОДЕССКИЙ НАЦИОНАЛЬНЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт компьютерных систем

Кафедра информационных систем

Лабораторная работа №8

По предмету: “Веб-технологии и веб-дизайн”

**Тема: «Создание и форматирование таблиц. Создание закладок и гиперссылок. Мета данные. Метки и карты изображений в html документах»**

Выполнил:

студент группы АИ-183

Новиков Д.Д.

Проверил:

доц. Защолкин К.В.

Одесса 2020

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.

**Код программы**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Laba 8</title>

<link rel="stylesheet" href="tic-tac-toe.css">

<link rel="stylesheet" href="index.css">

</head>

<body>

<form class="registration-form" name="registration-form" novalidate>

<h1>Регистрация</h1

<label class="required">

<span class="label-text">Полное имя</span>

<input type="text" placeholder="Вася Пупкин" name="name" required>

</label>

<label>

<span class="label-text">Организация</span>

<input type="text" placeholder="Eroe consulting">

</label>

<label>

<span class="label-text">Телефон</span>

<input type="text" placeholder="0968667332" name="telephone">

</label>

<label class="required">

<span class="label-text">E-mail</span>

<input type="text" name="email" placeholder="example@gmail.com" required>

</label>

<button type="submit">Зарегистрироваться</button>

</form>

<section class="tic-tac-toe-wrapper">

<h3 id="timer"><time>00:00:00</time></h3>

<div class="tic-tac-toe">

<h2>Tic Tac Toe</h2>

<button class="box"></button><button class="box"></button>button class="box"></button> <button class="box"></button><button class="box"></button>button class="box"></button><button class="box"></button><button class="box"></button><button class="box"></button><button class="box"></button><button class="box"></button><button class="box"></button><button class="box"></button><button class="box"></button><button class="box"></button><button class="box"></button>

</div>

<button id="reset" class="button reset">Сбросить</button>

</section>

<script src="index.js"></script>

<script src="tic-tac-toe.js"></script>

</body>

</html>

const form = document.getElementsByClassName("registration-form")[0]

form.addEventListener("submit", onFormSubmit)

function onFormSubmit(event) {

event.preventDefault()

const email = document.forms["registration-form"]["email"].value

const name = document.forms["registration-form"]["name"].value

const telephone = document.forms["registration-form"]["telephone"].value

let emailRegExp = /^[\.\-\_A-Za-z0-9]+?@[\.\-A-Za-z0-9]+?\.[A-Za-z0-9]{2,6}$/

let isEmailValid = emailRegExp.test(email)

let nameRegExp = /^[a-zA-Zа-яёА-ЯЁ\s\-]+$/u

let isNameValid = nameRegExp.test(name)

let telephoneRegExp = /^(\s\*)?(\+)?([- \_():=+]?\d[- \_():=+]?){10,14}(\s\*)?$/

let isTelephoneValid = telephoneRegExp.test(telephone)

if (isEmailValid === false) {

alert('Email введен неверно')

return

}

if (isNameValid === false) {

alert('Имя введено неверно')

return

}

if (isTelephoneValid === false) {

alert('Teлефон введен неверно')

return

}

if (name.length === 0 || email.length === 0) {

alert('Заподните обязательные поля')

return

}

let confirmAnswer = confirm("Вы уверены, что хотите отправить введенные данные ?")

if (confirmAnswer) submit()

}

.tic-tac-toe {

display: grid;

grid-template-columns: var(--cell-size) var(--cell-size) var(--cell-size) var(--cell-size);

grid-template-rows: var(--reset-height) var(--cell-size) var(--cell-size) var(--cell-size) var(--reset-height) var(--reset-height);

grid-gap: var(--cell-spacing);

justify-content: center;

align-content: center;

}

h2, h3 {

font-size: 28px;

text-align: center;

width: 100%;

height: var(--title-height);

line-height: var(--title-height);

grid-column-start: 1;

grid-column-end: 5;

}

.box {

font-family: Raleway, sans-serif;

font-size: 20px;

background-color: var(--tiles-bg-color);

width: var(--cell-size);

height: var(--cell-size);

display: block;

cursor: pointer;

padding: 0;

border-radius: 5px;

border: 3px solid #eee;

box-sizing: border-box;

transition: 0.1s box-shadow ease-out;

position: relative;

}

.box:active,

.box.ai-active {

box-shadow: 0 0 0 rgba(0, 0, 0, 0.5), 0 2px 5px rgba(0, 0, 0, 0.5) inset;

}

.box:hover {

background-color: var(--tiles-bg-color-hover);

}

.box.blue:after {

content: '✕';

color: var(--tiles-blue-bg-color);

font-size: 72px;

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

display: block;

}

Вывод: cоздали и форматировали таблицы. Создали закладки и гиперссылки, мета данные, метки и карты изображений в html документах