**Лабораторна робота №12**

**Форми**

**Перевірка даних, введених у форму**

Інформація, введена в форму, часто відправляється на сервер або відсилається електронною поштою на певну адресу. Проблема в тому, щоб впвнитися, що введені користувачем в форму дані коректні. Скористаємося можливостями JS.

Нехай HTML-сторінка містить два елемента для вводу тексту. В перший з них користувач має вписати своє ім’я, в другий елемент – адресу електронної пошти.

Дані в першому елементі перевірятимемо на пустоту. Звісно, це не захистить нас від імен типу '17'. В другому елементі контролюватимемо наявність символа @:

<html>

<head>

<script language="JavaScript">

function test1(form) {

if (form.text1.value == "")

alert("Не залишайте поле пустим!")

else {

alert("Hi "+form.text1.value+"! Форму заповнено коректно!");

}

}

function test2(form) {

if (form.text2.value == "" ||

form.text2.value.indexOf('@', 0) == -1)

alert("Неправильно введена адреса e-mail!");

else alert("OK!");

}

</script>

</head>

<body>

<form name="first">

Вкажіть ваше ім’я:<br>

<input type="text" name="text1">

<input type="button" name="button1" value="Перевірка"

onClick="test1(this.form)">

<P>

Вкажіть вашу адресу e-mail:<br>

<input type="text" name="text2">

<input type="button" name="button2" value="Перевірка"

onClick="test2(this.form)">

</body>

</html>

**Перевірка наявності певних символів**

Іноді потрібно обмежити інформацію, що вноситься у форму - згадаємо про телефонні номери – надана інформація має містити лише цифри. Нам необхідно перевіряти, чи є введені дані числом. Складність ситуації в тому, що більшість людей вставляють в номер телефона ще і різні символи - например: 01234-56789, 01234/56789 або 01234 56789 (з пробілом всередині). Не варто заставляти користувача відмовлятися від таких символів у телефонному номері, тому ми маємо доповнити програму процедурою перевірки цифр і окремих символів.

Начало формы

Конец формы

<html>

<head>

<script language="JavaScript">

function check(input) {

var ok = true;

for (var i = 0; i < input.length; i++) {

var chr = input.charAt(i);

var found = false;

for (var j = 1; j < check.length; j++) {

if (chr == check[j]) found = true;

}

if (!found) ok = false;

}

return ok;

}

function test(input) {

if (!check(input, "1", "2", "3", "4",

"5", "6", "7", "8", "9", "0", "/", "-", " ")) {

alert("Номер не коректний.");

}

else {

alert("Номер коректний!");

}

}

</script>

</head>

<body>

<form>

Номер:

<input type="text" name="telephone" value="">

<input type="button" value="Перевірити"

onClick="test(this.form.telephone.value)">

</form>

</body>

</html>

**Контроль коректності до відправки форми**

Для скасування відправки некоректних даних користуйтеся обробкою події onSubmit:

function validate() {

// перевірка коректності вводу

// ...

if (inputOK) return true

else return false;

}

...

<form ... onSubmit="return validate()">

...

Дані в такій формі не будуть відправлені в мережу, якщо JS-функція виявить некоректність даних.

**Фокусування на певному елементі форми**

З допомогою метода focus() ви можете зробити форму дружелюбнішою. Наприклад можна вказати, який елемент буде виділений першочергово. Або ви можете наказати броузеру виділити той елемент, де виявлено некоректні дані. Сделать это Вы можете с помощью следующего фрагмента скрипта:

function setfocus() {

document.first.text1.focus();

}

Для фокусування на елементі одразу по завантаженню сторінки причепіть дану функцію до події

load:

<body onLoad="setfocus()">

Залишається доповнити приклад так:

function setfocus() {

document.first.text1.focus();

document.first.text1.select();

}

Так окрім фокусування на елементі ми ще й виділяємо текст в елементі.

Завдання

1. Перевірити приклади із лабораторної роботи.

2. Уточнити контроль коректності імені користувача на ваш розсуд.

3. Уточнити контроль коректності адреси електронної пошти на ваш розсуд.

4. Уточнити контроль коректності телефонного номеру на ваш розсуд.