**Практическая работа 15**

**Создание форм**

**Цель:** Изучить приемы и методы создания форм и работы с ними

**Задания:**

1. Ознакомиться с теоретическими аспектами темы.
2. Создайте документ, содержащий два элемента для ввода текста.
3. Создайте документ, содержащий текстовое поле и кнопку, при нажатии на которую осуществляется перевод курсора в начало текстового поля.
4. Создайте документ, содержащий текстовое поле и кнопку, при нажатии на которую осуществляется перевод курсора в начало текстового поля и выделение содержимого данного текстового поля.

**Необходимые приборы:** ПК, текстовый редактор Блокнот, браузер

**Методические рекомендации к выполнению лабораторной работы:**

**Методические рекомендации к выполнению задания 1**

**Проверка информации, введенной в форму**

Формы широко используются на Интернет. Информация, введенная в форму, часто посылается обратно на сервер или отправляется по электронной почте на некоторый адрес. Проблема состоит в том, чтобы убедиться, что введенная пользователем в форму информация, корректна. Легко проверить ее перед пересылкой в Интернет можно с помощью языка JavaScript.

**Проверка на присутствие определенных символов**

В некоторых случаях нужно ограничивать информацию, вводимую в форму, лишь некоторым набором символов или чисел. Достаточно вспомнить о телефонных номерах - представленная информация должна содержать лишь цифры (предполагается, что номер телефона, как таковой, не содержит никаких символов). Нам необходимо проверять, являются ли введенные данные числом. Сложность ситуации состоит в том, что большинство людей вставляют в номер телефона еще и разные символы - например: 01234-56789, 01234/56789 или 01234 56789 (с символом пробела внутри). Не следует принуждать пользователя отказываться от таких символов в телефонном номере. А потому мы должны дополнить наш скрипт процедурой проверки цифр и некоторых символов.

**Начало формы**

**Предоставление информации, введенной в форму**

Самый простой способ состоит в передаче данных формы по электронной почте. Если нужно, чтобы за вносимыми в форму данными следил сервер, то необходимо использовать интерфейс CGI (Common Gateway Interface). Последнее позволяет автоматически обрабатывать данные. Например, сервер мог бы создавать базу данных со сведениями, доступную для некоторых из клиентов. Другой пример - поисковые страницы, такие как Yahoo. Обычно в них представлена форма, позволяющая создавать запрос для поиска в собственной базе данных. В результате пользователь получает ответ вскоре после того, как нажимает на соответствующую кнопку. Ему не приходится ждать, пока люди, отвечающие за поддержание данного сервера, прочтут указанные им данные и отыщут требуемую информацию. Все это автоматически выполняет сам сервер. JavaScript не позволяет делать таких вещей.

**Выделение определенного элемента формы**

С помощью метода focus() можно сделать форму более дружественной. Так, можно выбрать, который элемент будет выделен в первую очередь. Либо можно приказать браузеру выделить ту форму, куда были введены неверные данные. То есть, что браузер сам установит курсор на указанный Вами элемент формы, так что пользователю не придется щелкать по форме, прежде чем что-либо занести туда.

**Методические рекомендации к выполнению задания 2**

1 Создайте документ, содержащий два элемента для ввода текста.

<html>

<head>

<script language="JavaScript">

<!-- Скрыть

function test1(form) {

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

alert("Пожалуйста, введите строку!")

else {

alert("Hi "+form.text1.value+"! Форма заполнена корректно!"); }}

function test2(form) {

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

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

alert("Неверно введен адрес e-mail!");

else alert("OK!");}

// -->

</script>

</head>

<body>

<form name="first">

ВведитеВашеимя:<br>

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

<input type="button" name="button1" value="Проверка" onClick="test1 (this.form)">

<P>

Введите Ваш адрес e-mail:<br>

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

<input type="button" name="button2" value="Проверка" onClick="test2(this.form)">

</body>

</html>

В первый элемент пользователь должен вписать свое имя, во второй элемент - адрес для электронной почты. Можно ввести туда какую-нибудь информацию и нажать клавишу. Попробуйте также нажать клавишу, не введя в форму никакой информации. Начало формы

Конец формы

Что касается информации, введенной в первый элемент, то будет выводиться сообщение об ошибке, если туда ничего не было введено. Любая представленная в элементе информация будет рассматриваться на предмет корректности. Конечно, это не гарантирует, что пользователь введет не то имя. Браузер даже не будет возражать против чисел. Например, если Вы введете '17', то получите приглашение 'Hi 17!'. Так что эта проверка не может быть идеальна.

Второй элемент формы несколько более сложный. Попробуйте ввести простую строку – например, свое имя. Сделать это не удастся до тех пор, пока вы не укажите @ в Вашем имени. Признаком того, что пользователь правильно ввел адрес электронной почты, служит наличие символа @. Этому условию будет отвечать и одиночный символ @, даже несмотря на то, что это бессмысленно. В Интернет каждый адрес электронной почты содержит символ @, так что проверка на этот символ здесь уместна.

В разделе body создаются лишь два элемента для ввода текста и две кнопки. Кнопки вызывают функции test1(...) или test2(...), в зависимости от того, которая из них была нажата. В качестве аргумента к этим функциям мы передаем комбинацию *this.form*, что позже позволит нам адресоваться в самой функции именно к тем элементам, которые нам нужны.

Функция test1(form) проверяет, является ли данная строка пустой. Это делается посредством *if (form.text1.value == "")...* . Здесь 'form' - это переменная, куда заносится значение, полученное при вызове функции от 'this.form'. Мы можем извлечь строку, введенную в рассматриваемый элемент, если к *form.text1* припишем 'value'. Чтобы убедиться, что строка не является пустой, мы сравниваем ее с "". Если же окажется, что введенная строка соответствует "", то это значит, что на самом деле ничего введено не было. И наш пользователь получит сообщение об ошибке. Если же что-то было введено верно, пользователь получит подтверждение - ok.

Следующая проблема заключается в том, что пользователь может вписать в поле формы одни пробелы. И это будет принято, как корректно введенная информация. Однако к команде if мы добавлен символ ||. Он называется оператором OR (ИЛИ).

Команда if проверяет, чем заканчивается первое или второе сравнения. Если хотя бы одно из них выполняется, то и в целом команда if имеет результатом true, а стало быть будет выполняться следующая команда скрипта. Словом, Вы получите сообщение об ошибке, если либо предоставленная строка пуста, либо в ней отсутствует символ @. (Второй оператор в команде if следит за тем, чтобы введенная строка содержала @).

**Методические рекомендации к выполнению задания 3**

1 Создайте документ, содержащий текстовое поле и кнопку, при нажатии на которую осуществляется перевод курсора в начало текстового поля. Для осуществления перехода курсора используйте функцию:

functionsetfocus() {

document.first.text1.focus();}

Начало формы

Конец формы

Эта запись может выделить первый элемент для ввода текста в скрипте. Вы должны указать имя для всей формы - в данном случае она называется *first* - и имя одного элемента формы - *text1*.

**Методические рекомендации к выполнению задания 4**

1 Создайте документ, содержащий текстовое поле и кнопку, при нажатии на которую осуществляется перевод курсора в начало текстового поля и выделение содержимого данного текстового поля. При выполнении задания используйте функцию:

functionsetfocus() {

document.first.text1.focus();

document.first.text1.select();}

Конец формы

При этом не только будет выделен элемент, но и находящийся в нем текст.

**Вопросы для самоконтроля:**

1. Каким образом осуществляется проверка информации, введенной в формулу?
2. Каким образом производится проверка на отсутствие определенных символов?
3. Для чего предназначен метод focus()?
4. Что нужно сделать для того, чтобы за вносимыми в форму данными следил сервер?