

БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ

Кафедра «Автоматизированные системы управления»

ТЕХНОЛОГИЯ ИНТЕРНЕТ- ПРОГРАММИРОВАНИЯ

*Методические указания к лабораторной
работе № 8 для студентов специальности*
**09.03.01 Автоматизированные системы обработки
информации и управления**

Могилев 2020

Лабораторная работа №8

Формы

Цель работы: Разобрать способы выполнения проверки формы, а также возможности для пересылки информации по Интернет

Порядок выполнения работы.

Изучить теоретические сведения.

Выполнить задание к лабораторной работе в соответствии с вариантом.

Оформить отчет.

Требования к отчету.

Цель работы.

Постановка задачи.

Текст программы.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

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.

Сперва нам необходимо создать простой скрипт. Допустим, HTML-страница содержит два элемента для ввода текста. В первый из них пользователь должен вписать свое имя, во второй элемент - адрес для электронной почты. Вы можете ввести туда какую-нибудь информацию и нажать клавишу. Попробуйте также нажать клавишу, не введя в форму никакой информации.

Введите Ваше имя:

Введите Ваш адрес e-mail:

Что касается информации, введенной в первый элемент, то Вы будете получать 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 сложнее. Попробуйте ввести простую строку - например Ваше имя. Сделать это не удастся до тех пор, пока Вы не укажете @ 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.

Как скрипт работает с этими двумя элементами формы и как выглядит проверка? Это происходит следующим образом:

```
<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("ОК!");
}
// -->
</script>
</head>
<body>
<form name="first">
Введите Ваше имя:<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"
nClick="test2(this.form)">
</body>
</html>
```

Рассмотрим сначала HTML-код в разделе body. Здесь мы создаем 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 - ок.

Следующая проблема заключается в том, что пользователь может вписать в поле формы одни пробелы. И это будет принято, как корректно введенная информация! Если есть желание, то Вы конечно можете добавить проверку такой возможности и исключить ее.

Рассмотрим теперь функцию *test2(form)*. Здесь вновь сравнивается введенная строка с пустой - "" (чтобы удостовериться, что что-то действительно было введено читателем). Однако к команде *if* мы добавили еще кое-чего. Комбинация символов `||` 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 Вами строка пуста, либо в ней отсутствует символ @. (Второй оператор в команде *if* следит за тем, чтобы введенная строка содержала @.)

Проверка на присутствие определенных символов

В некоторых случаях Вам понадобится 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 or 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.

Telephone:

Исходный код этого скрипта:

```
<html>
<head>
<script language="JavaScript">
<!-- hide
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("Input not ok.");
    }
    else {
        alert("Input ok!");
    }
}
// -->
</script>
</head>
<body>
<form>
Telephone:
<input type="text" name="telephone" value="">
<input type="button" value="Check"
    onClick="test(this.form.telephone.value)">
</form>
</body>
</html>

```

Функция test() определяет, какие из введенных символов признаются корректными.

Предоставление информации, введенной в форму

Какие существуют возможности для передачи информации, внесенной в форму? Самый простой способ состоит в передаче данных формы по электронной почте (этот метод мы рассмотрим поподробнее).

Если Вы хотите, чтобы за вносимыми в форму данными следил сервер, то Вы должны использовать интерфейс CGI (Common Gateway Interface). Последнее позволяет Вам 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.

С помощью JavaScript Вы не сможете создать книгу читательских отзывов, поскольку JavaScript лишен возможности записывать данные в какой-либо файл на сервере. Делать это Вы можете только через интерфейс CGI. Конечно, Вы можете создать книгу отзывов, для которой пользователи присылали сведения по

электронной почте. Однако в этом случае Вы должны заносить отзывы вручную. Так можно делать, если Вы не предполагаете получать ежедневно по 1000 отзывов.

Соответствующий скрипт будет простым текстом HTML.

```
<form method=post action=mailto:your.address@goes.here
enctype="text/plain">
```

Нравится ли Вам эта страница?

```
<input name="choice" type="radio" value="1">Вовсе нет.<br>
```

```
<input name="choice" type="radio" value="2" CHECKED>Напрасная трата
времени.<br>
```

```
<input name="choice" type="radio" value="3">Самый плохой сайт в
Сети.<br>
```

```
<input name="submit" type="submit" value="Send">
```

```
</form>
```

Параметр `enctype="text/plain"` используется для того, чтобы пересылать именно простой текст без каких-либо кодируемых частей.

Если Вы хотите проверить форму прежде, чем она будет передана в сеть, то для этого можете воспользоваться программой обработки событий `onSubmit`. Вы должны поместить вызов этой программы в тэг `<form>`. Например:

```
function validate() {
    // check if input ok
    // ...
    if (inputOK) return true
    else return false;
}
```

```
...
```

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

```
...
```

Форма, составленная таким образом, не будет послана в Интернет, если в нее внесены некорректные данные.

Выделение определенного элемента формы

С помощью метода `focus()` Вы можете сделать вашу форму более дружелюбной. Так, Вы можете выбрать, который элемент будет выделен в первую очередь. Либо Вы можете приказать браузеру выделить ту форму, куда были введены неверные данные. Сделать это Вы можете с помощью следующего фрагмента скрипта:

```
function setfocus() {
    document.first.text1.focus();
}
```



Эта запись могла бы 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*. Если Вы хотите, чтобы при загрузке страницы данный элемент выделялся, то для этого Вы можете дополнить тэг `<body>` атрибутом `onLoad`. Это будет выглядеть как:

```
<body onLoad="setfocus()">
```

Остается еще дополнить пример следующим образом:

```
function setfocus() {  
    document.first.text1.focus();  
    document.first.text1.select();  
}
```

Попробуйте следующий код:

la la la

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

Задания к лабораторной работе

Вариант 1

Напишите сценарий, который по заданному времени определяет количество минут, прошедших от начала текущего занятия.

В анкете заполняется информация о семи сотрудниках, принятых на работу на основе почасовой оплаты: фамилия, оплата за час, дата начала работы, дата окончания. Напишите сценарий, при работе которого определяется:

- количество рабочих дней для каждого сотрудника;

- количество выходных дней между заданными датами;

- размер гонорара, учитывая, что каждый рабочий день длится 8 часов (стоимость одного часа работы указана в договоре и содержится в анкете).

Вариант 2

Напишите сценарий, который для заданной даты определяет, сколько дней прошло после некоторого события.

В анкете заполняется информация о семи сотрудниках: фамилия, дата заключения контракта. Ровно через одиннадцать месяцев после начала работы сотрудникам предоставляется отпуск продолжительностью 24 рабочих дня. Создайте форму для ввода данных. Напишите сценарий, определяющий:

- дату начала и окончания отпуска для каждого из сотрудников;

- фамилии сотрудников, отпуск которым будет предоставлен в заданном месяце (месяц выбирается пользователем);

- фамилии сотрудников, отпуск которых будет завершен в заданном квартале (номер квартала выбирается пользователем);

- фамилии сотрудников, отпуск которым будет предоставлен в следующем календарном году;

Вариант 3

Напишите сценарий, который по заданной дате определяет номер недели в году.

В анкете заполняется информация о десяти сотрудниках: фамилия, дата заключения контракта и срок (в годах), на какой заключен контракт. Создайте форму для ввода данных. Напишите сценарий, определяющий:

сотрудников, контракт с которыми заканчивается в определенное время года (время года выбирается);

сотрудников, контракт с которыми заканчивается в определенный квартал (номер квартала выбирается пользователем);

в зависимости от введенной даты тех сотрудников, с которыми заключен контракт на заданный момент;

сотрудников, контракт с которыми завершен на заданную дату.

Вариант 4

Напишите сценарий, который по дате рождения человека определяет, под каким знаком зодиака родился человек.

В анкете заполняется информация о десяти сотрудниках: фамилия, дата заключения контракта и срок (в годах), на какой заключен контракт. Создайте форму для ввода данных. Напишите сценарий, определяющий:

дату окончания контракта;

месяц, в который закончится контракт;

день недели, в который закончится контракт;

сотрудников, контракт с которыми заканчивается в определенном году (год выбирается пользователем);

Вариант 5

Напишите сценарий, который рассчитывает, сколько рабочих и выходных дней между двумя заданными датами.

В анкете приводятся данные о шести сотрудниках: фамилия и дата приема на работу. Все сотрудники в зависимости от стажа работы разделяются на категории: проработавшие меньше года относятся к категории "молодой", от года до пяти лет — "опытный", более пяти лет — "ветеран". Напишите сценарий определения стажа работы каждого сотрудника и числа сотрудников в каждой категории. Постройте диаграмму, отражающую число сотрудников в категориях.