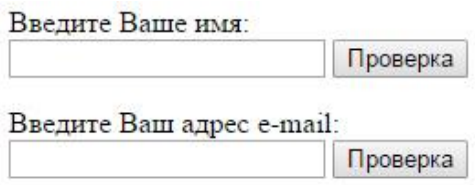
# Part 7: Forms

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 введеная пользователем в форму информация корректна. Легко проверить ее перед пересылкой в Интернет можно с помощью языка JavaScript. Сначала я бы хотел продемонстрировать, как можно выполнить проверку формы. А затем мы рассмотрим, какие есть возможности для пересылки информации по Интернет.

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.




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 ввведете '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 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.

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("OK!");
}
// -->
</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"
onClick="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 - ok.
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. Я полагаю, что это будет сделать легко, опираясь лишь на представленную здесь информацию.
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). С ним Вы уже

знакомились в шестой части Введения.
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. Решение задачи продемонстрировано в следующем примере, взятом из моей книги о JavaScript: :


Исходный код этого скрипта:

```
<html>
<head>
<script language="JavaScript">
<!-- hide

// ****************************************************
// Script from Stefan Koch - Voodoo's Intro to JavaScript
//     http://rummelplatz.uni-mannheim.de/~skoch/js/
//       JS-book: http://www.dpunkt.de/javascript
//    You can use this code if you leave this message
// ****************************************************

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. И никакого программирования на JavaScript здесь вовсе не нужно! Конечно за исключением

того случая, если Вам понадобится перед пересылкой проверить данные, занесенные в форму - и здесь уже Вам действительно понадобится JavaScript. Я должен лишь добавить, что команда mailto работает не повсюду - например, поддержка для ее отсутствует в Microsoft Internet Explorer 3.0.

```
<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. Это значительно упрощает чтение такой почты.

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 дружественной. Так, Вы можете выбрать, который элемент будет выделен в первую очередь. Либо Вы можете 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. Сделать это Вы можете с помощью следующего фрагмента скрипта:

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

bla bla bla Focus

Эта запись могла бы выделить первый элемент для ввода текста в скрипте, который я уже показывал. 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 Ваш 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:


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