

Методы и средства решения стандартных задач профессиональной деятельности

ЛАБОРАТОРНАЯ РАБОТА №4

Взаимодействие с пользователями через веб формы.

Теория

Основным элементом языка JavaScript является функция.

Описание функции имеет вид

```
function F (V)
```

```
{S},
```

где F - идентификатор функции, задающий имя, по которому можно обращаться к функции; V - список параметров функции, разделяемых запятыми; S - тело функции, в нем задаются действия, которые нужно выполнить, чтобы получить результат.

Необязательный оператор return определяет возвращаемое функцией значение. Обычно все определения и функции задаются в разделе <head> документа. Это обеспечивает интерпретацию и сохранение в памяти всех функций при загрузке документа в браузер.

Создадим форму, в которой поля Основание и Высота служат для ввода соответствующих значений. Кроме того, в форме создадим кнопку Вычислить. При щелчке мышью по этой кнопке мы хотим получить значение площади треугольника. Действие пользователя (например, щелчок кнопкой мыши) вызывает событие. События в основном связаны с действиями, производимыми пользователем с элементами форм HTML. Обычно перехват и обработка события задается в параметрах элементов форм. Имя параметра обработки события начинается с приставки on, за которой следует имя самого события. Например, параметр обработки события click будет выглядеть как onclick.

Листинг 1. Реакция на событие Click.

```
<HTML>
<HEAD>
<title>Обработка значений из формы</title>

<script language="JavaScript">

function care (a, h)
{
    let s=(a*h)/2;
    document.write ("Площадь прямоугольного
    треугольника равна ",s); return s
}

</script>

</HEAD>
<BODY>
<P>Пример сценария со значениями из формы</P>
<FORM name="form1">
Основание:
<input type="text" size=5 name="st1"><hr>
Высота:
<input type="text" size=5 name="st2"><hr>
<input type="button" value=Вычислить
onClick="care(document.form1.st1.value,
document.form1.st2.value)"> /*По клику мыши на кнопке в
функцию care передаются два параметра - содержимое полей ввода*/
</FORM>
</BODY>
</HTML>
```

При интерпретации HTML-страницы браузером создаются объекты JavaScript. Взаимосвязь объектов между собой представляет иерархическую структуру. На самом верхнем уровне иерархии находится объект windows, представляющий окно браузера. Объект windows является предком или родителем" всех остальных объектов. Каждая страница кроме объекта windows имеет объект document. Свойства объекта document определяются содержимым самого документа: цвет фона, цвет шрифта и т. д. Для получения значения основания треугольника, введенного в первом поле формы, должна быть выполнена конструкция

```
document.form1.st1.value
```

читаем с конца: используем данные value из поля ввода с именем st1 находящегося на форме с именем form1 объекта document.

Напишем сценарий, определяющий площадь квадрата по заданной стороне. Площадь должна вычисляться в тот момент, когда изменилось значение его стороны. Пусть форма содержит два текстовых поля: одно для длины стороны квадрата, другое для вычисленной площади. Кнопка Обновить очищает поля формы. Площадь квадрата вычисляется при возникновении события **change**, которое происходит в тот момент, когда значение элемента формы с именем num1 изменилось, и элемент потерял фокус.

Листинг 2. Реакция на событие Change

```
<HTML> <HEAD>
<title>Обработка события Change - изменение значения
элемента</title>
</HEAD>

<BODY>
<script>
function srec(obj)
{obj.res.value=obj.num1.value* obj.num1.value}
</script>

<P>Вычисление площади квадрата</P>
<FORM name="form1">
Сторона: <input type="text" size=7
name="num1" onChange="srec(form1)">
<hr>
Площадь: <input type="text" size=7 name="res"><hr>
<input type="reset" value=Обновить>
</FORM>
</BODY>
</HTML>
```

Рассмотрим далее примеры с условиями. Синтаксис выполнения действий при выполнении условий:

```
if (condition) {
  code to run if condition is true
} else {
  run some other code instead
}
```

Данный код читается: "if (если) condition (условие) возвращает true (истина), запусти код А, else (иначе) запусти В"

Запретим посетителю переключаться с поля ввода, если введённое значение не

прошло валидацию, для этого рассмотрим методы `elem.focus()` и `elem.blur()` которые устанавливают/снимают фокус:

```
<style>
  .error {
    background:
    red;
  }
</style>
```

```
Ваш email: <input type="email" id="input">
<input type="text" style="width:280px" placeholder="введите
неверный email и кликните сюда">
```

```
<script>
input.onblur = function() {
if (!this.value.includes('@')) { // не email
// показать ошибку this.classList.add("error");
// ...и вернуть фокус обратно input.focus();
} else { this.classList.remove("error");
}
};
</script>
```

Это работает во всех браузерах, кроме Firefox (bug).

Если мы что-нибудь введём и нажмём Tab или кликнем в другое место, тогда `onblur` вернёт фокус обратно.

Дополнительная информация:

Случайные числа: `Math.random()` – генерирует случайное число от 0 до 1.

Объект `Math` содержит набор методов, который используется для округления чисел:

`round()` — округление по обычным правилам;

`floor()` — округление вниз;

`ceil()` — округление вверх;

`trunc()` — отбрасывание дробной части, не обращая внимания на знак аргумента. до 1 не включительно.

Пример:

```
let X = Math.ceil(Math.random()*99999999);
```

в данном примере объявлена переменная `X` значение которой равно округленному до большего целого значения произведению (Случайно значение от 0 до 1 умноженное на 99999999).

ЗАДАНИЕ:

Разработать страницу анкеты для получения карты постоянного клиента магазина. Для работы со строковыми данными используйте справочники по JavaScript.

Сценарий кода страницы анкеты должен обрабатывать вводимую информацию в текстовых полях формы:

1. В полях фамилия и имя должно срабатывать событие onChange при котором все прописные буквы будут заменяться заглавными (метод toUpperCase() см.справочник)
2. Если после заполнения анкеты нажать кнопку «Отправить», на экране появится диалоговое окно с информацией о номере карты (случайное число), Фамилию и имя из анкеты. Кнопка «Ввести данные заново» устанавливает поля в исходное состояние.
3. Если указать возраст, меньший 18 лет, сценарий выдает сообщение «Карта выдается только после 18 лет».

The image shows a web form for card registration and a confirmation dialog. The form is titled "Заполните анкету" and contains input fields for "Фамилия" (ИВАНОВ), "Имя" (ИВАН), "Телефон" (47738990864), and "Возраст" (25). There are two buttons at the bottom: "Зарегистрировать карту" and "Очистить данные". The confirmation dialog is titled "Подтвердите действие" and displays the entered information: "Фамилия: ИВАНОВ", "Имя: ИВАН", and "Номер карты клиента: 49171431". It has an "OK" button.

Литература:

1. Справочники html, CSS, JavaScript <https://www.basicweb.ru>