Министерство сельского хозяйства Российской Федерации
Федеральное государственное бюджетное образовательное
учреждениевысшего образования
Казанский государственный энергетический университет
Кафедра информационных технологий и интеллектуальных систем

JavaScript и DOM-манипуляции

Методические указания для выполнения лабораторных работ

Казань, 2024

Лабораторные работы

JavaScript и DOM-манипуляции. Создание простых скриптов на JavaScript

Задание 1. Для расширения функциональных возможностей web-страниц используются скрипты ***JavaScript***, которые включаются в html-документы несколькими способами:

– в теговом контейнере <***body***>...</***body***>:

```
<body>
…
<script > команды скрипта</script>
…
</body>
```

– в контейнере <***head***>...</***head***>, если скрипт представляет собой функцию, вызываемую в ответ на какое-либо событие:

```
<head>
…
<script type=”text/javascript”> команды сценария </script>
</head>
```

– во внешнем файле с расширением ***js:***

```
<head>
…
<script type = "text/javascript" src = "my.js"> </script>
</head>
```

Любые текстовые данные в языке ***JavaScript*** являются строками. В ***JavaScript*** не существует отдельного типа «символ», который есть в ряде других языков, а внутренний формат для строк – всегда UTF-16, вне зависимости от кодировки страницы.

Для создания строк в ***JavaScript*** можно использовать одинарные, двойные или обратные кавычки.

Одинарные и двойные кавычки работают одинаково, а строка в обратных кавычках может содержать произвольное выражение, заключенное в ${…}, и располагаться более чем в одной строке.

Многострочные строки также можно создавать с помощью одинарных и двойных кавычек, используя символ перевода строки ***\n***.

Свойство ***length*** содержит длину строки:

```
alert( `My\n`.length );        // 3
```

здесь ***\n*** – один спецсимвол, поэтому длина строки равна 3.

Получить символ, который занимает позицию ***pos***, можно с помощью квадратных скобок: [***pos***]. Также можно использовать метод ***charAt***.

Поиск подстроки. Метод ***str.indexOf(substr, pos)*** ищет подстроку ***substr*** в строке ***str***, начиная с позиции ***pos***, и возвращает позицию, на которой располагается совпадение, либо –1 при отсутствии совпадений. Чтобы найти все вхождения подстроки, можно использовать метод ***indexOf()*** в цикле.

Для получения подстроки можно использовать методы ***substring***, ***substr*** и ***slice***. Например, ***str.slice(start [, end])*** вернет часть строки от ***start*** до (не включая) ***end***.

Порядок выполнения работы.

Выполнить следующие задания.

1 Создать строку текста из 22 первых букв русского алфавита: var str = 'abcde … '. Используя функцию ***alert*** вывести символы с указанными в столбце «Задание 1» таблицы 2.1 номерами и разделить их номером варианта, например, для варианта 11: a-11-c-11-e11 и т. д.

2 Построить строку из цифр «Номера символов» первого задания. Из полученной строки, состоящей из 12 цифр, выделить четыре трехзначных числа. Используя полученные числа и заданные в колонке «Задание 2» таблицы 2.1 операции, построить оператор присваивания с построенным арифметическим выражением, полученный результат вывести в окно браузера.

3 Построить строку текста из букв, номера которых заданы в «Задание 1». Используя свойство ***innerHTML*** метода ***document.getElementById(id),*** вывести на страницу html построенную строку.

4 С помощью функции ***confirm*** построить запрос, содержащий две кнопки: Да и Нет. В зависимости от выбранной кнопки вычислить заданные в «Задании 3» таблицы 2.1 выражения: для ***Да*** – ***y***, для ***Нет*** – ***f***. Результат вывести на страницу ***html*** используя функцию ***writeln***.

Таблица 2.1 – Задания к лабораторной работе

Номер варианта	Задание 1 (номера символов)	Задание 2 (операции)			Задание 3 (арифметическое выражение)	
					y =	f =
1	01, 03, 05, 06, 08, 09	+	–	*	$(a + b)^2/c$	$d/(f - e/2)$
2	02, 03, 04, 06, 07, 15	+	–	/	(a * b) – c^2	$2*d/(f + e/4)$
3	04, 05, 07, 09, 12, 18	+	–	%	$(a - b/c)^2$	$d*(2*f - e/2)$
4	03, 05, 06, 12, 20, 21	/	*	–	$(a^2 + b)/c$	$d/(f/5 + e$^2)
5	01, 02, 08, 15, 17, 19	/	*	+	$(a - b/c)^2/c$	$2*d/(f/1{,}4 - e/2)$
6	13, 15, 16, 17, 18, 20	/	*	%	$(a + b/c^2)/c$	$d/(f*e/2) + d$
7	05, 07, 09, 13, 16, 18	*	+	–	$(a - b/a)/c^2$	$d/(f*e + 2*d) - d$
8	03, 04, 09, 12, 14, 17	*	+	/	$(a + b/a^2)/2*c$	$d/(f$*e – $2*d) + e$
9	02, 09, 16, 17, 18, 21	*	+	%	$(a - b/a*2)/c^2$	$d/($e$*f/2) + d*e/f$
10	03, 07, 08, 09, 10, 20	–	*	/	$(a + b/c^2)/c - a$	$d/(f*e/2 + d) - 2*f$
11	01, 03, 05, 07, 10, 11	–	*	+	(a – $b/c/2)/c^a$	$d/(f$*e + $2*f) + d$
12	06, 08, 09, 12, 13, 19	–	*	%	$(a + b/c+2)/c–2*b$	$d/(f$*e – $2/f) – d$

Контрольные вопросы

1 Какие функции вывода на страницу ***html*** Вы знаете?
2 Какие математические операции используются в ***JavaScripn***?
3 Как извлечь символ строки по его номеру?
4 Как встроить ***JavaScript***-код в ***html***-документ?
5 Как выделить символ из строки текста?
6 Какие комментарии используются в языке ***JavaScript***?
7 Для каких целей используются методы ***prompt*** и ***Confirm***?
8 Для чего используются методы ***document.write()***, ***alert()***, ***console.log()***?
9 Какие способы включения ***JavaScript***-кода в ***html***-документ Вы знаете?
10 Прокомментируйте технологию использования метода ***document.getElementById(id)*** вывода на страницу ***html***.

Задание 2. Изучение функций обработки событий JavaScript

Событие – это сигнал от браузера о том, что что-то произошло. Среди событий можно выделить: события мыши: ***click***, ***contextmenu***, ***mouseover*** / ***mouseout***, ***mousedown*** / ***mouseup***, ***mousemove***; события на элементах управления: ***submit***, ***focus***; клавиатурные события: ***keydown*** и ***keyup***.

Событию можно назначить обработчика, т. е. функцию, которая сработает, как только событие произошло.

Если при наступлении события требуется произвести много действий, то удобно написать сценарий в виде функции и разместить его в контейнере <***script***> …<***/script***>, предназначенном для сценариев. Например, для вывода модуля заданного числа используется функция ***Math.abs***, а для округления чисел – функции ***Math.round***, ***Math.ceil*** и ***Math.floor***, а также методы ***toFixed*** и ***toPrecision***.

Функции ***Math.min***, ***Math.max***, ***Math.sqrt***, ***Math.pow***, ***Math.random*** используются для определения минимального, максимального значений, вычисления квадратного корня, возведения в степень и генерации псевдослучайных чисел с равномерным законом распределения.

Для работы со строками текста используются следующие методы: ***length***, ***toUpperCase***, ***toLowerCase***, ***substr***, ***substring***, ***slice***, ***indexOf***, ***replace***, ***split*** и функция ***join***. Чтобы обратить внимание пользователя web-сайта на определённый элемент ***html***-документа, его свойства можно менять, например, цвет или размер, при попадании на него курсора мышки, а при снятии курсора восстанавливать прежние значения.

Для взаимодействия с пользователем в ***JavaScript*** определен механизм событий. Например, когда пользователь нажимает кнопку, то возникает событие нажатия кнопки. В коде ***JavaScript*** можно определить возникновение события и обработать его.

Событие в ***JavaScript*** – это определенное действие, которые вызвано либо пользователем, либо браузером, например, клик мыши по кнопке, движение мыши, наведение фокуса на элемент, изменение значения в каком-либо текстовом поле, изменение размеров окна браузера и т. д.

В ***JavaScript*** используются следующие типы событий:

– события мыши (перемещение курсора, нажатие мыши и т. д.);

– события клавиатуры (нажатие или отпускание клавиши клавиатуры);

– события жизненного цикла элементов (например, загрузки web-страницы);

– события элементов форм (нажатие кнопки на форме и т. д.);

– события, возникающие при изменении элементов DOM;

– события, возникающие при касании на сенсорных экранах;

– события, возникающие при возникновении ошибок.

В таблице 3.1 приведены события ***JavaScript***, объекты, которые могут их генерировать, и причины возникновения соответствующих событий.

Таблица 3.1 – События ***JavaScript***

Событие	Объект	Причина возникновения
Abort	Image	Прерывание загрузки изображения
Blur	Button, Checkbox, FileUpload, Frame, Layer, Password, Radio, Reset, Select, Submit, Text, Textarea, Window	Потеря фокуса элемента
Change	FileUpload, Select, Text, Textarea	Смена значения
Click	Area, Button, Checkbox, Document, Link, Radio, Reset, Submit	Клик мыши на элементе
DblClick	Area, Document, Link	Двойной клик на элементе
DragDrop	Window	Перемещение в окно браузера
Focus	Button, Checkbox, FileUpload, Frame, Layer, Password, Radio, Reset, Select, Submit, Text, Textarea, Window	Установка фокуса на элементе
KeyDown	Document, Image, Link, Textarea	Нажатие клавиши на клавиатуре
KeyPress	Document, Image, Link, Textarea	Удержание клавиши на клавиатуре
KeyUp	Document, Image, Link, Textarea	Отпуск клавиши
Load	Document, Image, Layer, Window	Загрузка элемента
MouseDown	Button, Document, Link	Нажатие кнопки мыши
MouseMove	Window	Мышь в движении
MouseOut	Area, Layer, Link	Мышь выходит за границы элемента
MouseOver	Area, Layer, Link	Мышь находится над элементом
MouseUp	Button, Document, Link	Отпускание кнопки мыши
Move	Frame	Перемещение элемента
Reset	Form	Сброс формы
Resize	Frame, Window	Изменение размеров
Select	Text, Textarea	Выделение текста
Submit	Form	Передача данных
Unload	Window	Выгрузка текущей страницы

Для использования событий используются их обработчики, которые определяют, что будет происходить при возникновении определённого события. Обработчики событий в JavaScript имеют вид: ***onНазваниеСобытия***, т. е. вначале записывается приставка «***on***», а за ней – название события. Например, обработчики событий: ***onFocus***, ***onClick***, ***onSubmit*** и др. Область применения событий в ***JavaScript*** огромна.

Рассмотрим простейшую обработку событий. Например, на web-странице записан элемент div:

```
<div id="rect" onclick="alert('Нажато')"
        style="width:50px;height:50px;background-color:blue;"></div>
```

Здесь определен обычный блок ***div*** с атрибутом ***onclick***, который задает обработчик события нажатием на блок ***div***. То есть, чтобы обработать какое-либо событие, нам необходимо определить для него обработчик, который представляет собой соответствующий код на языке ***JavaScript***.

Также можно вынести все действия по обработке события в отдельную функцию:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
  </head>
  <body>
  <div id="rect" onclick="displayMessage()"
        style="width:50px;height:50px;background-color:blue;"></div>
        <script>
              function displayMessage(){

                 alert('Нажато');
              }
        </script>
  </body>
</html>
```

Теперь обработчиком события будет выступать функция ***displayMessage***.

Обработчик событий может быть назначен прямо в HTML-разметке, в атрибуте, который называется ***on*** <событие>. Например, чтобы назначить обработчик события ***click*** на элементе ***input***, можно использовать атрибут ***onclick***:

```
<input value="Нажми меня" onclick="alert('Клик!')" type="button">
```

При клике мышкой на кнопке выполнится код, указанный в атрибуте ***onclick***.

Порядок выполнения работы.

Выполнить задания, приведенные в таблице 3.2.

Таблица 3.2 – Задания к лабораторной работе

Номер варианта	Задание 1		Задание 2
	Используя Math.abs, вычислить модуль числа	Округлить число до *n* знаков в дробной части, используя функцию	Выполнить операции с элементами одномерного массива из *n* вещественных чисел, используя функцию
1	278,785	Math.round, *n* = 1	Math.min, *n* = 8
2	547,473	Math.ceil, *n* = 2	Math.max, *n* = 9
3	182,487	Math.floor, *n* = 3	Math.sqrt, *n* = 5 и просуммировать
4	264,289	toFixed, *n* = 2	Math.pow, *n* = 6 и просуммировать
5	452,297	Math.round, *n* = 2	Math.random, *n* = 9 и просуммировать
6	984,573	Math.ceil, *n* = 4	Math.min, *n* = 7
7	893,284	Math.floor, *n* = 1	Math.max, *n* = 8
8	487,651	toFixed, *n* = 3	Math.sqrt, *n* = 5 и просуммировать
9	774,652	Math.round, *n* = 3	Math.pow, *n* = 9 и просуммировать
10	682,571	Math.ceil, *n* = 1	Math.random, *n* = 7 и просуммировать
11	455,628	Math.floor, *n* = 2	Math.min, *n* = 9
12	671,475	toFixed, *n* = 4	Math.max, *n* = 6

Контрольные вопросы

1 Какие функции и методы округления чисел Вы знаете?

2 Прокомментируйте технологию использования функций ***Math.sqrt, Math.pow, Math.random***.

3 Прокомментируйте технологию использования функций ***isNaN***, ***isFinite***, ***parseInt***, ***parseFloat***.

4 Какая функция используется для округления вещественного числа?

5 Как получить модуль числа?

6 В каких ситуациях возникает необходимость использования регулярных выражений при работе с текстом?

7 Как сформировать последовательность псевдослучайных чисел с равномерным распределением?

8 Как выполняется глобальный поиск и замена символа в строке текста?

9 Для чего используется метод ***indexOf*** при работе со строками текста?

10 Какие методы для работы со строками Вы знаете?