Методические указания для выполнения

лабораторной работы № 4 по курсу «Web-технологии»

**Клиентские языки сценариев. Javascript**

**Цель работы**: получение опыта написания и применения функций на языке javascript.

**Правила оценивания работ**

Выполнение базового задания оценивается максимально в 6 баллов. Для получения за работу 10 баллов требуется выполнить не менее пяти пунктов из раздела «Расширенное задание».

Из максимально возможной оценки баллы будут вычитаться при следующих ситуациях:

1. **Срок.** Отчет выложен в лк после срока сдачи – минус 1 балл каждые 2 недели от даты, записанной в ЛК. Дата защиты не влияет;
2. **Защита.** «Пробелы» в знаниях во время устной защиты работы: оценка «хорошо» - минус 1 балл, оценка «удовлетворительно» - минус 2 балла;
3. **Отчет.** В отчете присутствуют незначительные нарушения требований - минус 1 балл, отсутствует несколько пунктов отчета - минус 2 балла или больше;
4. **Работа.** В выполненном задании отсутствуют второстепенные элементы - минус 1 балл, отсутствует несколько пунктов задания - минус 2 балла или больше.

Защита работы возможна только при наличии в ЛК отчета.

**Базовое задание**

Подготовить несколько сценариев по заданиям ниже. Все сценарии встроить в сайт из лабораторной работы № 2. Запрещается создавать отдельную страницу для выполнения данной работы. Варианты в таблице 1 определяют способы реализации сценариев (тип события, способ включения сценария, метод выбора элемента на странице, оформление функции). Вид метода (getElementById, getElementsByName, getElementsByTagName ), заданный вариантом, ОБЯЗАТЕЛЬНО должен применяться для большей части всех случаев обращения к элементам на странице.

1. Вызов всплывающего окна с ФИО и группой автора сайта
2. Добавить сценарий по варианту из Таблицы 1 (столбец «сценарий»)
3. Добавить пустую строку в таблицу, подготовленную в работе 1
4. Изменить цвет элемента
5. Изменить размер элемента

**Расширенное задание**

1.. При добавлении строки в таблицу для заполнения ячеек брать данные, введенные пользователем в элементы интерфейса (input)

2.. Удалять только добавленные через сценарий строки таблицы

3.. Применить лямбда (стрелочную) функцию

4.. Использовать события окна браузера (загрузка, перемещение, скроллинг)

5.. Сделать функцию с несколькими параметрами и показать ее применение

6.. Использовать addEventListener

7.. Добавить и удалить элемент списка

**Содержание отчета**

1. титульный лист;
2. цель работы;
3. вариант задания;
4. таблица со столбцами: название функции, ее обработчик, описание действия функции
5. тексты функций на языке javascript
6. скриншоты страниц сайта с демонстрацией работы всех функций
7. HTML код
8. выводы по работе.

**Элементы языка клиентских сценариев javascript**

**Вставка сценариев**

Существует три способа включения сценария в текст html документа:

1. Сценарий записан внутри парного тега <script>. Можно располагать где угодно, но желательно размещать в разделе <head>.

<script type= "text/javascript">

prompt ('Это диалоговое окно prompt. ', '') </script>

1. Сценарий находится в отдельном текстовом файле, который подключается через тег <script>.

<script type = “text/javascript” src = “filename.js”> </script>

1. Сценарий записан как значение атрибута-события внутри тега элемента страницы:

<img src = “im.png” onclick = “alert (‘Hi’)”>

Место нахождения сценария влияет на порядок его выполнения.

**Функции javascript**

Функции представляют собой объединения нескольких строк javascript-кода, и могут быть вызваны из любой части веб-страницы. Функции задаются в веб-странице в специальных тегах <script>.

Описание функции состоит из заголовка и тела функции. Заголовок включает служебное слово function и список параметров в круглых скобках. Список параметров содержит имена переменных, которые дают входные значения. Если входные данные не нужны, скобки остаются пустыми. Тело функции размещается между фигурными скобками. Если необходимо вернуть результат работы функции, используется оператор return (return переменная).

Существует два варианта оформления заголовка функции:

1. Традиционное описание:

function имя\_функции (список параметров)

{ тело функции }

function jsfunction ()

{confirm ('Это диалоговое окно confirm.'); }

1. Анонимная функция – у функции нет имени, но она привязана к переменной.

Переменная = function ()

{ тело функции}

Вызов анонимной функции выполняется через имя переменной.

summa = function(a,b)

{return a+b;}

**События и их обработчики**

События – это программные оповещения, генерируемые браузером в ответ на действия пользователя. Например, при нажатии пользователем мыши возникает событие click, при нажатии клавиши клавиатуры – событие keypress.

Для обработки событий используются обработчики событий. Название обработчика состоит из префикса on и названия события. Например, обработчик нажатия кнопки мыши – onclick, обработчик нажатия клавиши клавиатуры – onkeypress. Чтобы назначить функцию f() обработчиком события click, возникающего при нажатии на некоторую кнопку button, в HTML-коде нужно написать следующим образом:

<button onclick=”f();”>Нажми меня!</button>

Если щелкнуть мышью по данной кнопке, то будет вызвана функция f().

**Комментарии к листингу 1**

В скрипте описана функция sum, которая вызывается по нажатию кнопки с надписью «я решу задачу!» (<button onclick = "summ()">). После ввода чисел в окне prompt, переменная strmm содержит строку с числами, разделенными пробелом. Применение метода строк split() после разделения по пробелам позволяет получить массив из строк (mm = strmm.split (" ")). Метод документа getElementsByTagName() возвращает **массив** (el) всех элементов страницы с заданным тегом (с тегом span в данном случае). Для обращения к одному элементу нужно использовать индекс массива (el[0]).

С помощью свойства элемента innerHTML можно изменять внешний вид и содержание данного тега. В строке

el[0].innerHTML= "<b style='font-size:14;color:green;'>"+mm +"</b>"

устанавливается новое оформление и выводится на экран массив.

При выполнении действий с массивом необходимо преобразовать строки в числа. Поэтому мы используем функцию Number().

С помощью строки

<div style="width: 100%; height: 3px; background-color: #97070B;"></div>

на экран выводится линия.

**Листинг 1. Документ html со сценарием, решающим задачу с массивом.**

<html> <head>

<title>Пример работы с массивом</title>

<script>

function summ()

{ strmm = prompt("Введите элементы массива через пробел.");

mm = strmm.split (" ")

el= document.getElementsByTagName("span")

el[0].innerHTML= "<b style='font-size:14;color:green;'>"+mm +"</b>"

var ss=0;

for(var i=0; i<mm.length; i++)

ss += Number(mm[i]);

el[1].innerHTML= "<b style='font-size:14;color:green;'>"+ss +"</b>" }

</script> </head>

<body>

Задача: найти сумму элементов массива.<br>

<div style="width: 100%; height: 3px; background-color: #97070B;"></div><br>

введен массив: <span ></span><br>

сумма элементов: <span ></span><br><br><br>

<button onclick = "summ()"> я решу задачу!</button>

</body></html>

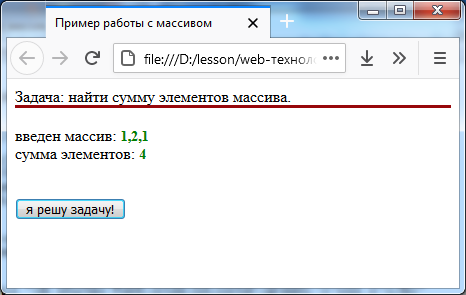


Рис. 1 Пример работы сценария javascript. Листинг 1

**Пример работы со скриптами и функциями**

В примере демонстрируется:

- работа с диалоговыми окнами alert(), confirm(), prompt();

- разные способы включения скрипта на javascript в документ html;

- разные способы оформления функции

**Листинг 2**. Файл jspodkluchenie.htm

<html><head> <title> Script+function </title>

<script type= "text/javascript">

prompt ('Это диалоговое окно prompt. У меня есть поле ввода и две кнопки.\n Из скрипта в теге script', '') </script>

<script type= "text/javascript" src = "myfilejs.js"> </script>

<script type= "text/javascript">

function mult (a,b) {return a\*b}

summa = function(a,b)

{return a+b;} </script>

</head>

<body> <h1> Способы подключения скрипта </h1>

<button onclick = "alert ('Это диалоговое окно alert из скрипта после атрибута-события. \nУ меня одна кнопка OK.')">

скрипт после атрибута-события</button> <br><br>

<button onclick = "jsfunction()"> вызов функции из файла</button>

<h1> Способы написания функции </h1>

<button onclick = "alert('Результат умножения: ' + String (mult (5,10))) "> вызов функции с параметрами</button> <br><br>

<button onclick = "alert('Результат сложения: '+ String(summa(2,4)))"> анонимная функция, вызываемая через переменную</button>

</body> </html>

**Листинг 3**. Файл myfilejs.js

function jsfunction ()

{confirm ('Это диалоговое окно confirm. У меня две кнопки.\n Вызов функции из файла'); }

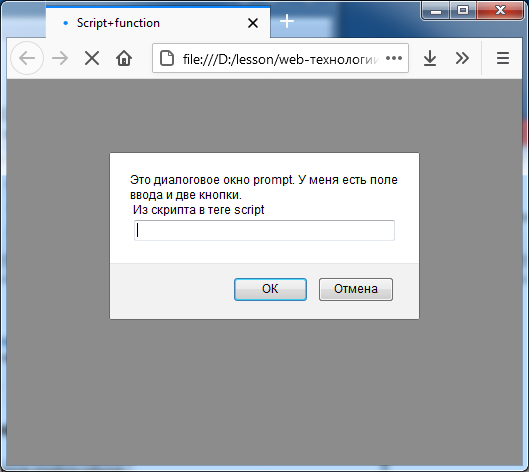


Рис. 2 Выполнение скрипта из Листинга 2.

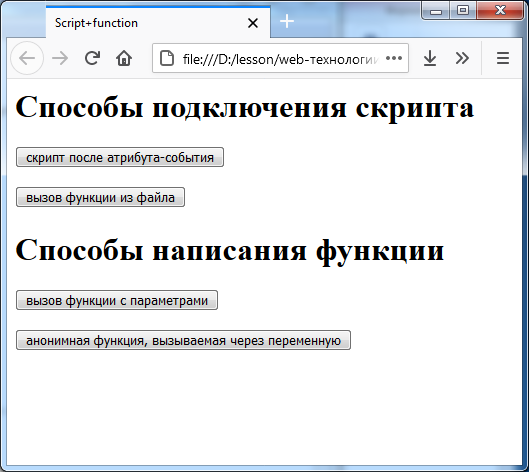


Рис. 3 Отображение документа html из Листинга 2

**Примеры скриптов**

Листинг 4. Галерея. Фрагмент кода документа html

<input type="button" onclick="left\_arrow()" value="Назад"/>

<img src = "DSC05853.JPG" width = "60" height = "40" id = "img" > </img>

<input type="button" onclick="right\_arrow()" value="Вперед"/>

Листинг 5. Галерея. Фрагмент кода скрипта

var mas = ["DSC05853.JPG","DSC05854.JPG","DSC05855.JPG","DSC05856.JPG"]

// массив картинок

var to = -1; // Счетчик, указывающий на текущую картинку

function right\_arrow() // Открытие следующей картинки(движение вправо)

{ var obj = document.getElementById("img");

if (to < mas.length-1) to++ else to = 0;

obj.src = mas[to]; }

function left\_arrow() // Открытие предыдущей картинки(движение влево)

{ var obj = document.getElementById("img");

if (to > 0) to--; else to = mas.length-1;

obj.src = mas[to]; }

Листинг 6. Часы

<html><head>

<script type= "text/javascript">

var tmrid;

function bigben ()

{ var datecur, hour, minute; var time ="";

datecur = new Date();

hour = datecur.getHours();

minute= datecur.getMinutes();

sec = datecur.getSeconds();

time = ((hour <10) ? "0" : "" ) + hour+"\_";

time += ((minute <10) ? "0" : ":\_" ) + minute;

time += ((sec <10) ? "0" : ":\_" ) + sec;

tmr = document.getElementById ('tmr');

tmr.innerText = time;

}

</script> </head>

<body> <h1> Time now : <span id = "tmr"> </span> </h1>

<button onclick = "bigben()"> я покажу время</button> </body> </html>

Таблица 1. **Варианты заданий**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Событие** | **Способ включения сценария** | **Поиск элемента** | **Оформление функции** | **Сценарий** |
| 1 | click | тег <script> | getElementById | с именем | 1 |
| 2 | mouseover | тег <script> | getElementsByTagName | с именем | 2 |
| 3 | mousemove | тег <script> | getElementsByName | с именем | 3 |
| 4 | keydown | тег <script> | getElementById | с именем | 4 |
| 5 | click | тег <script> | getElementsByTagName | с именем | 5 |
| 6 | mouseover | тег <script> | getElementsByName | с именем | 6 |
| 7 | mousemove | тег <script> | getElementById | с именем | 7 |
| 8 | keydown | тег <script> | getElementsByTagName | анонимная | 8 |
| 9 | click | тег <script> | getElementsByName | анонимная | 1 |
| 10 | mouseover | тег <script> | getElementById | анонимная | 2 |
| 11 | mousemove | тег <script> | getElementsByTagName | анонимная | 3 |
| 12 | keydown | тег <script> | getElementsByName | анонимная | 4 |
| 13 | click | тег <script> | getElementById | анонимная | 5 |
| 14 | mouseover | тег <script> | getElementsByTagName | анонимная | 6 |
| 15 | mousemove | тег <script> | getElementsByName | анонимная | 7 |
| 16 | keydown | файл | getElementById | с именем | 8 |
| 17 | click | файл | getElementsByTagName | с именем | 1 |
| 18 | mouseover | файл | getElementsByName | с именем | 2 |
| 19 | mousemove | файл | getElementById | с именем | 3 |
| 20 | keydown | файл | getElementsByTagName | с именем | 4 |
| 21 | click | файл | getElementsByName | с именем | 5 |
| 22 | mouseover | файл | getElementById | с именем | 6 |
| 23 | mousemove | файл | getElementsByTagName | анонимная | 7 |
| 24 | keydown | файл | getElementsByName | анонимная | 8 |
| 25 | click | файл | getElementById | анонимная | 1 |
| 26 | mouseover | файл | getElementsByTagName | анонимная | 2 |
| 27 | mousemove | файл | getElementsByName | анонимная | 3 |
| 28 | keydown | файл | getElementById | анонимная | 4 |
| 29 | click | файл | getElementsByTagName | анонимная | 5 |
| 30 | mouseover | файл | getElementsByName | анонимная | 6 |

Задания для сценария

1. Часы, которые показывают время в трех разных часовых поясах
2. Часы, которые выводят время в формате: сейчас 19 часов 20 минут
3. Часы, в которых в четные минуты цифры окрашены в красный цвет, а в нечетные в синий.
4. Часы, которые показывают время сразу после загрузки страницы
5. Часы, которые показывают постоянно час и минуты. Секунды показываются после нажатия на кнопку.
6. Галерея по таймеру.
7. Галерея с кнопками перехода между картинками и подписями к каждой картинке.
8. Движение текста.

Распределение вариантов для лабораторной работы № 4 по курсу «Web-технологии»

осень 2024

|  |  |  |  |
| --- | --- | --- | --- |
| вариант | Гр z1431 | Гр z1432к |  |
| 1 | Быстров | Андреенко |  |
| 2 | Васильев | Аракчеев |  |
| 3 | Дворников | Бикмурзин |  |
| 4 | Зайцев | Вовк |  |
| 5 | Калинин | Иванов |  |
| 6 | Козлова | Ильюшенко |  |
| 7 | Колесников | Исаков |  |
| 8 | Колобов | Кандаков |  |
| 9 | Краснюков | Куралесов |  |
| 10 | Кутузов | Новиков |  |
| 11 | Леонов | Торопова |  |
| 12 | Лобов | Шульгин |  |
| 13 | Собакин |  |  |
| 14 | Федоров |  |  |
| 15 | Щанкин (перезачет) |  |  |
| 16 |  |  |  |
| 17 |  |  |  |
| 18 |  |  |  |
| 19 |  |  |  |
| 20 |  |  |  |
| 21 |  |  |  |
| 22 |  |  |  |
| 23 |  |  |  |
| 24 |  |  |  |
| 25 |  |  |  |
| 26 |  |  |  |