Лабораторная работа №4. События JavaScript. Встроенные объекты.

Теоретическая часть

События JavaScript

Событие - это сигнал от браузера о том, что что-то произошло, можно выделить несколько их категорий:

1. События, связанные с документом;
2. События, связанные с элементами документа;
3. События, связанные с окнами.

Для того чтобы скрипт реагировал на событие - нужно назначить обработчик события. Обычно обработчики называют "on+имя события", например: `onclick`.

Назначение обработчиков событий для элементов

Существует несколько способов назначать обработчик на конкретное событие элемента. Один из этих способов – обработчик события записывается прямо в открывающем теге элемента. Например, для обработки события `click` на кнопке `input`, можно назначить обработчик `onclick` вот так:

```
<input type="button" value="Нажми" onclick="alert('Нажал!');" />
```

В этом случае JavaScript код пишется в кавычках в одну строку.
Такой способ установки обработчиков очень удобен - он нагляден и прост, поэтому часто используется в решении простых задач.

Событие Load и его обработчик onLoad

Событие `Load` возникает для элементов `body` и `frameset` когда закончена загрузка документа. Например, в данном примере, после загрузки страницы вызывается метод `document.write()`, который затирает то, что было загружено, рисунок 4.1.
Пример страницы:

```
<html>
 <head>
  <meta charset="UTF-8">
  <title></title>
 </head>
 <style type="text/css"></style>
 <body onload="document.write('загрузка страницы завершена...и она уже
затёрта');">
 <script type="text/javascript"></script>
 <H1>Заголовок</H1>
 </body>
</html>
/* загрузка страницы завершена... и она уже затёрта */
```

Рис. 4.1 Событие Load

Событие Click и его обработчик onClick

Событие `Click` – одинарный щелчок (нажата и отпущена кнопка мыши) возникает фактически для всех элементов страницы, рисунок 4.2.

```
<!DOCTYPE html>
<html>
 <head><meta charset="UTF-8"><title></title></head>
 <body>
  <input type="button" value="Кликни" onclick="alert('Хватит переводить ресурс мышиного манипулятора.');">
 </body>
    </html>
```

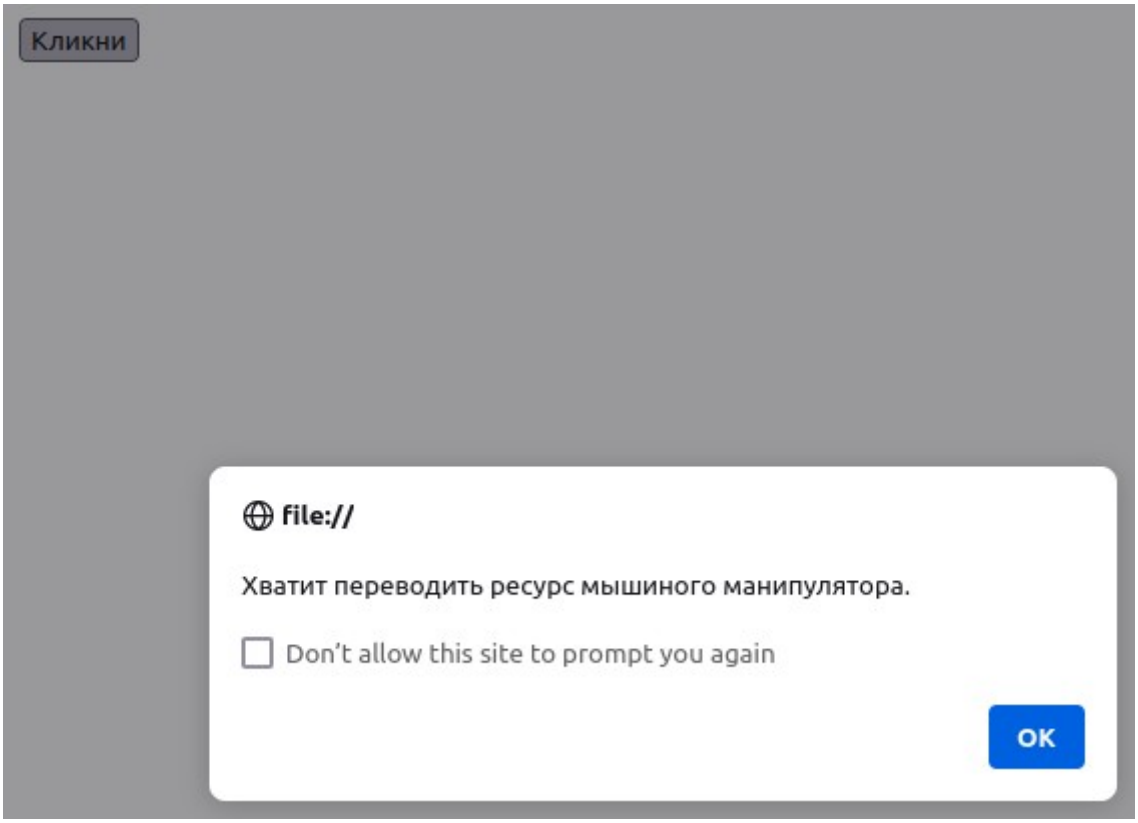

Рис. 4.2 Событие Click

Назначение обработчиков событий

Способ установки обработчика событий непосредственно как атрибута тега имеет свои плюсы и минусы. Как только обработчик начинает занимать больше одной строки - читабельность резко падает. В этом случае для обработки события нужно использовать функцию. При этом в обработчике события указывают только имя функции, также вы можете описать так вашу функцию, что она будет принимать какие угодно вам параметры. Например, в следующем примере при загрузке страницы выдаётся сообщение и на странице есть кнопка и абзац щёлканье по которым увеличивает значение счётчика. После загрузки страницы выдаётся сообщение «загрузка страницы завершена…», рисунки 4.3 и 4.4. Также следует отметить, что функция описана с применением функционального литерала.

```
<html>
 <head>
  <meta charset="UTF-8">
  <title></title>
 </head>
 <style type="text/css"></style>
```

```
 <body onload="alert('загрузка страницы завершена...');">
  <p id="p_out">0</p>
  <p onclick="f();">это параграф, по которому можно щёлкать.</p>
  <input type="button" value="Кликни" onclick="f();">
  <script type="text/javascript">
   var f = function() {
    var count = parseInt(document.getElementById("p_out").innerHTML);
    console.log(count);//отладочный вывод
    document.getElementById("p_out").innerHTML = ++count;
   };
  </script>
 </body>
</html>
```

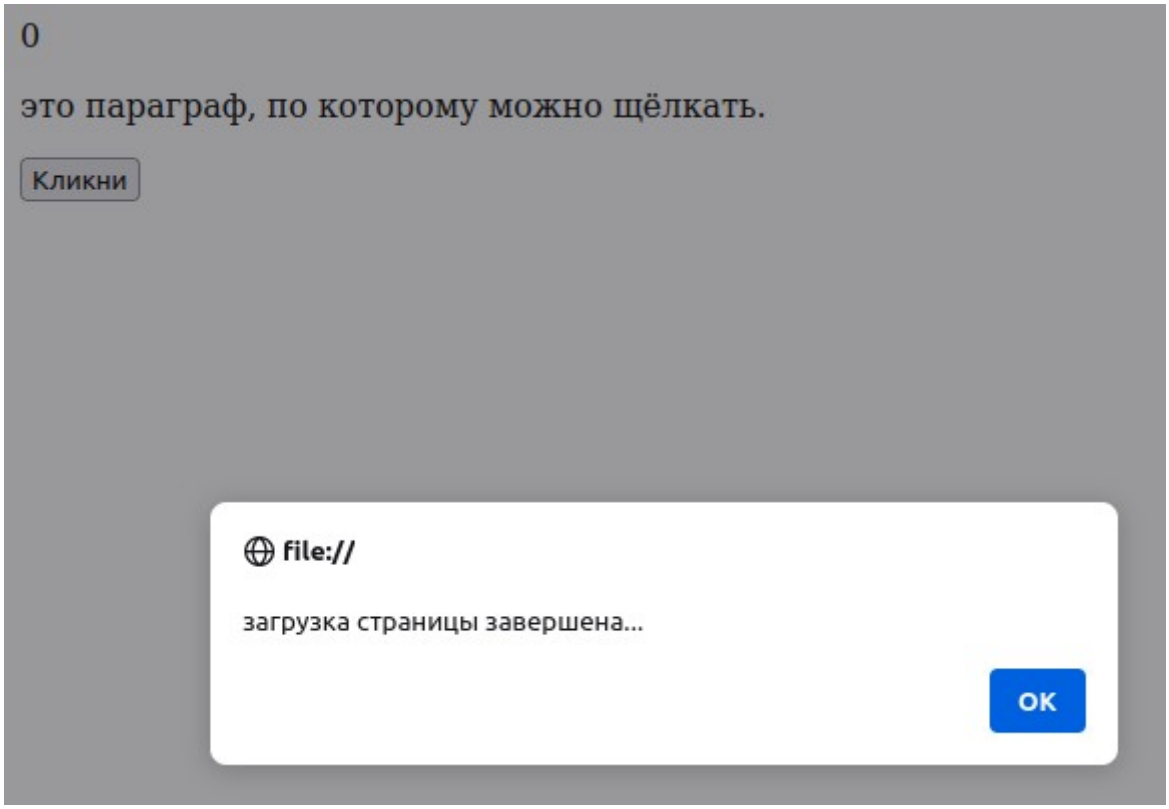

Рис. 4.3 Установки обработчика событий непосредственно как атрибута тега

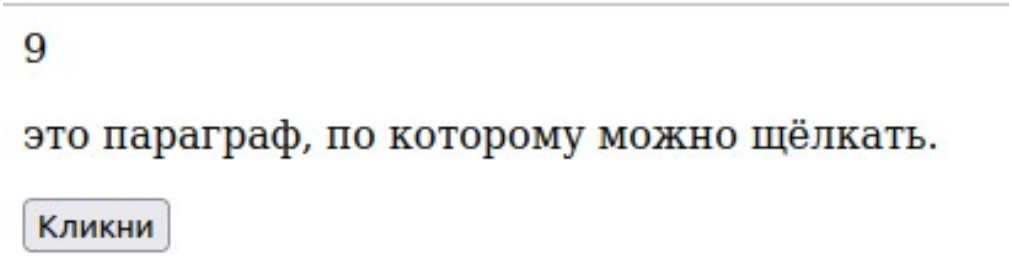

Рис. 4.4 Установки обработчика событий непосредственно как атрибута тега

Встроенные объекты

Основной единицей в языке JavaScript является объект, который объединяет в себе данные (свойства) и средства обработки этих данных (методы). Все объекты, которые используются в языке JavaScript, делятся на две большие группы:

1. Встроенные объекты базового языка
2. Объекты документа.

Объект Math

В языке JavaScript существует специальный класс `Math`, в котором собраны основные математические функции и константы. Все свойства и методы данного класса являются статическими, поэтому объект класса `Math` создавать не нужно. Ниже приведён список некоторых из его методов, таблица 4.1.

Табл. 4.1 Методы класса `Math`

Метод	**Описание**
`abs(число)`	Возвращает модуль (абсолютную величину) числа.
`sqrt(число)`	Возвращает квадратный корень из числа.
`pow(основание, степень)`	Возвращает результат возведения основания в указанную степень. Например, Math.pow(5, 3) вернёт 5 3 =125.
`random()`	Возвращает псевдослучайное число в диапазоне от 0 до 1.
`ceil(число)`	Производит округление в большую сторону, то есть возвращает наименьшее целое число, большее либо равное аргументу.
`floor(число)`	Производит округление в меньшую сторону, то есть возвращает наибольшее целое число, меньшее либо равное аргументу.
`round(число)`	Округляет указанный аргумент до целочисленного значения.
`cos(число)`	Возвращает косинус числа.
`sin(число)`	Возвращает синус числа.
`exp(число)`	Возводит число e (основание натурального логарифма) в указанную степень.
`log(число)`	Возвращает натуральный логарифм числа.

Чтобы задействовать метод (обычно это называют вызовом метода), в операторе JavaScript нужно сделать на него ссылку. Для этого используется синтаксис с использованием точки. Очень часто используется также свойство этого объекта, которое имеет значение числа π, `Math.PI`.

```
document.write(Math.PI);
document.write(" "+Math.floor(12.9));
/*3.141592653589793 12 */
```

Класс Date

Объект класса `Date` предназначен для манипуляций с датами и временем. Его примитивным значением является число, равное количеству миллисекунд относительно базового времени, равного полуночи 1 января 1970 г. День состоит из 86400000 миллисекунд.

Создание объекта класса Date

Создание объекта класса с текущей датой и временем:

```
var a = new Date();
```

В переменной `a` текущая дата и время.
Создание объекта класса с датой и временем, заданными аргументами конструктора:

```
new Date(год,месяц, день [,часы [,минуты [,секунды [,мс]]]])
```

Здесь:

- год — числовое выражение, задающее полный номер года (например, 1988, а не 88);
- месяц — числовое выражение, задающее номер месяца (0 = январь, 1 = февраль, ...,11 = декабрь);
- день — числовое выражение, задающее номер дня в месяце от 1 до 31;
- часы — необязательное числовое выражение, задающее номер часа от 0 до 23;
- минуты — необязательное числовое выражение, задающее номер минуты от 0 до 59;
- секунды — необязательное числовое выражение, задающее номер секунды от 0 до 59;
- мс — необязательное числовое выражение, задающее номер миллисекунды от 0 до 999.

Например,

```
var c = new Date(1961, 3, 12, 09, 07);
```

В переменной «с» дата – 12 апреля 1961 года и время 09 часов 07 мин. Нумерация месяцев начинается с 0.

Методы класса Date

Метод это действие, которое выполняется для объекта или с объектом. По своей сути это команда, но её действия связаны с определенным объектом. У каждого объекта может быть много методов.
Далее приводятся некоторые методы класса `Date`:

- `getTime()` - Возвращает количество миллисекунд, прошедших с полуночи 01.01.1970.
- `getFullYear()` - Возвращает год (четыре цифры).
- `getMonth()` - Возвращает месяц по местному времени (от 0 до 11; 0 = январь, 1 = февраль и т.д.).
- `getDate()` - Возвращает день месяца по местному времени.
- `getDay()` - Возвращает день недели (от 0 до 6; 0 = воскресенье, 1 = понедельник и т.д.) для указанной даты по местному времени.
- `getHours()` - Возвращает часы по местному времени (от 0 до 23).
- `getMinutes()` - Возвращает минуты по местному времени (от 0 до 59).
- `getSeconds()` - Возвращает секунды по местному времени (от 0 до 59).

Есть и друге методы, также для методов `get` есть парные методы `set`.

Обращение к элементам страницы

Для размещения значений, возвращаемых описанными методами на Web–странице, необходимо уметь обратиться к элементам этой страницы (документа). Один из простых способ заключается в том, чтобы обратится к элементу по `id`. Для этого нужно у каждого требуемого элемент документа присвоить атрибут `id`.

```
<p id='first'>
```

Обратите внимание, что все ключевые слова чувствительны к регистру. В названии метода используются три буквы в верхнем регистре. Метод `getElementById()` относится к объекту `document`. С его помощью мы можем найти любой поименованный элемент документа. Для разделения элементов иерархической ссылки используется точка.

Обращение к свойствам объекта

Каждый объект обладает некими характеристиками, которые называются свойствами. Например: пусть объект текстовое поле описывается так:

```
<input type='text' id='entry' value='User Name?'>
```

Для получения доступа к свойству объекта используется тот же тип синтаксиса с точками. Ссылка на данное свойство состоит из ссылки на данный объект плюс ещё одно расширение, указывающее на нужное свойство. Например, чтобы обратиться к свойству `value` элемента текстовое поле, который имеет `id 'entry'` можно записать выражение вида:

```
document.getElementById('entry').value
```

Или можно сопоставить наш объект с переменной ob и присвоить свойству value этого объекта новое значение 'Введите имя'.

```
var ob=document.getElementById('entry');
ob.value='Введите имя';
```

Примеры

Пример №1

Далее приводится пример странички, на которой сверху вниз размещены: абзац, поле для ввода, ещё абзац и кнопка. При этом нажатие на кнопку приводит к тому, что текст из второго абзаца и текстового поля с добавлением количества сделанных нажатий на кнопку размещается в верхнем абзаце. Дополнительно к названию кнопки при каждом её нажатии приписывается номер очередного нажатия на эту кнопку, рисунок 4.5.

```
<!DOCTYPE html>
<html>
 <head><meta charset="UTF-8"><title>Тест</title></head>
 <body>
  <p id="p_2">Суда мы запишем значения абзаца и текстового поля при нажатии на
кнопку.</p>
  <input type="text" id="input_1" value="Текстовое поле: текст.">
  <p id="p_1">Абзац: текст - копия.</p>
  <input type="button" value="Кнопка" id="input_2" onclick="func_2()">
 </body>
 <script type="text/javascript">
  var counter = 0;
  function func_2() {
   var p_1 = document.getElementById("p_1");
   var p_2 = document.getElementById("p_2");
   var input_1 = document.getElementById("input_1");
   var input_2 = document.getElementById("input_2");
   input_2.value += ++counter;//смена названия кнопки
   p_2.innerHTML = p_1.innerHTML + " " + input_1.value + " " + counter;
   /*Внимание:          так для абзаца            и так для поля*/
  }
 </script>
</html>
```

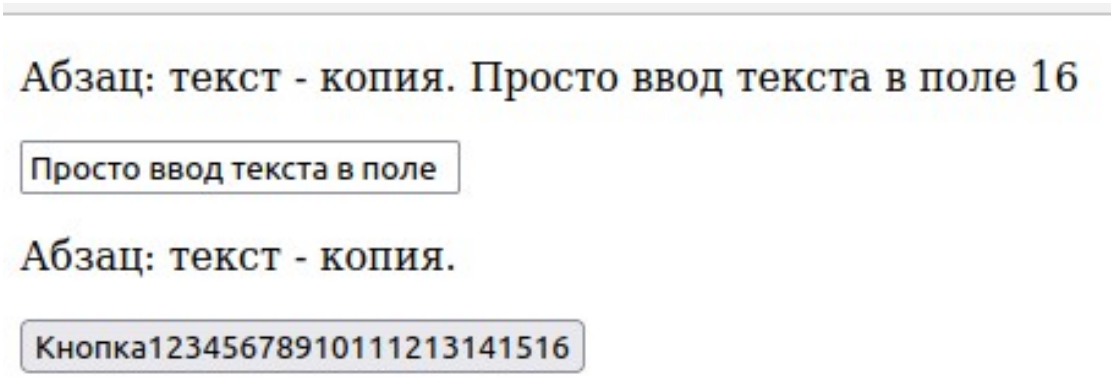

Рис. 4.5 Демонстрация работы пример №1

Пример №2

В следующем примере показано как определить номер дня с начала текущего года, рисунки 4.6 и 4.7:

```
<!DOCTYPE html>
<html>
 <head><meta charset="UTF-8"><title>Тест</title></head>
 <body>
  <p id="p_1">Нажми на кнопку, чтобы определить номер дня с начала текущего
года.</p>
  <input type="button" id="input_1" onclick="func();">
 </body>
 <script type="text/javascript">
  function func() {
   /*количество милисекунд прошедших с 01.01.1970 на текущий момент*/
   var d1 = new Date();
   console.log(d1);//отладочный вывод в консоль
   console.log(d1.getFullYear());//отладочный вывод в консоль
   var d0 = new Date(d1.getFullYear(), 0, 1, 0, 0);
   var days = Math.ceil((d1.getTime() - d0.getTime()) / 86400000);
   document.getElementById("p_1").innerHTML = days;
  }
 </script>
</html>
```

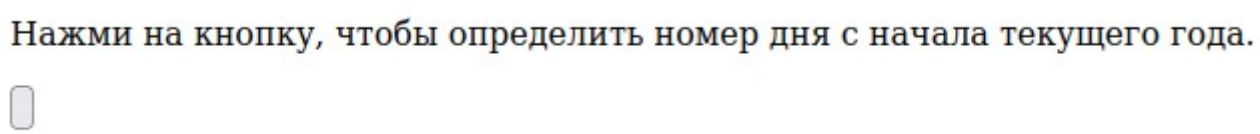

Рис. 4.6 Демонстрация работы пример №2

Рис. 4.7 Демонстрация работы пример №2

Пример №3

В следующем примере приводится несколько стилизованных элементов размещаемых на html — странице. Сам пример позволяет находить объём цилиндра, рисунки 4.8 и 4.9:

```
<!DOCTYPE html>
<html>
 <head><meta charset="UTF-8"><title>Тест</title></head>
 <style type="text/css">
  h1 {
    color: darkred;
    text-align: center;
  }
  body {
    background: lightyellow;
    font-weight: bold;
  }
  input {
    font-weight: bold;
  }
 </style>
 <body>
  <h1>Определение объёма цилиндра.</h1>
  <p id="p_1">Задайте радиус основания и высоту цилиндра.</p>
  <hr>
  r = <input id="input_1" type="text" value="0.0">
  h = <input id="input_2" type="text" value="0.0">
  <input type="button" id="input_1" onclick="func();">
  <hr>
 </body>
 <script type="text/javascript">
  function func() {
   var r = document.getElementById("input_1").value;
   var h = document.getElementById("input_2").value;
   if(parseFloat(r) && parseFloat(h)) {
    document.getElementById("p_1").innerHTML = Math.PI * Math.pow(r, 2) * h;
   } else {
    alert("Введи то, что может быть хоть как-то преобразовано в число!");
   }
  }
 </script>
</html>
```

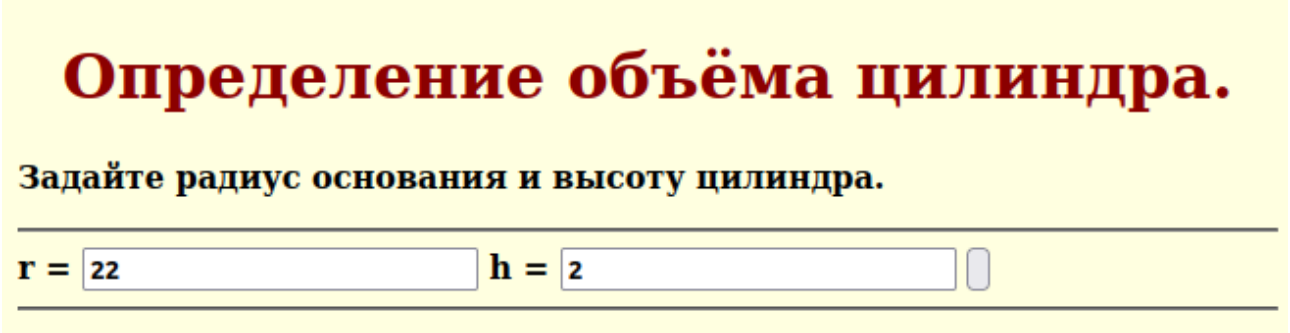

Рис. 4.8 Демонстрация работы пример №3

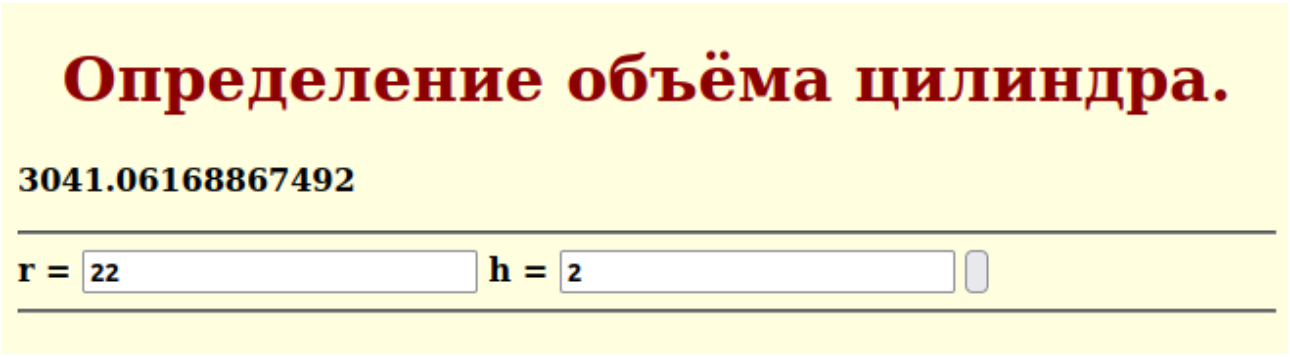

Рис. 4.9 Демонстрация работы пример №3

Практическая часть

1. Нужно разработать HTML - страничку, содержащую: текстовое поле для ввода данных, необходимых для решения поставленной задачи из лабораторной №3; интерактивный элемент, нажатие на который приводит к решению поставленной задачи; элемент для отображения результата решения поставленной задачи.
2. Нужно использовать как минимум две функции: первая - обработчик события нажатия на интерактивном элементе; вторая - функция «исполнитель», которая собственно и решает поставленную задачу.
3. В качестве интерактивного элемента использовать: 1 — кнопку, 2 — абзац с текстом.
4. Обеспечить использование синтаксиса функциональных литералов: 1 — нет, 2 — да.
5. Обеспечить использование синтаксиса ассоциативных массивов: 1 — нет, 2 — да.
6. Размещать данные, необходимые для решения поставленной задачи[3]: 1 — в объектах, 2 — в массивах.
7. При загрузке на странице должен выводиться текущий день недели, число, месяц и год. Для месяцев и дней недели следует организовать массивы. При этом вывод должен быть организован с применением таблицы, в которой: 1 — три строки, где в первой строке размещается день недели, во второй число и месяц, а в третьей год; 2 — две строки, где в первой строке размещается год число и месяц, а во второй день недели.
8. Далее на разрабатываемой HTML - странице выводить количество дней оставшихся до сессии:[4] 1 — в текстовом поле; 2 — в абзаце; 3 — в кнопке; 4 — в не редактируемом текстовом поле.[5]
9. Далее на HTML - странице нужно разместить кнопку, при щелчке на которую следует запрашивать памятную для вас дату и ниже (или сбоку) выводить: 1 — в текстовом поле количество дней, которые минуло после неё (дата должна быть такой, что количество дней превосходит 365); 2 — выводить в абзаце; 3 — в кнопке; 4 — в не редактируемом текстовом поле.
10. Найти и вывести на HTML - странице сумму заданного количества членов числовой последовательности, начиная с указанного (по номеру), задав эти параметры следующим далее образом (см. ниже следующие пп.). Указанную сумму предлагается вычислять для одной из следующих целочисленных числовых последовательностей:
    1 — $1+(-2)^n$
    2 — $5+3\cdot n$
    3 — $(n-2)^2$
    4 — $(-2)^{3n}$
    5 — $(-1)^n \cdot (n-1)^3$
    6 — $(-1)^{n-1} \cdot (n+4)^2$
    7 — n^2+n^3
    8 — $3-16\cdot n+n^3$
11. Задав номер первого её члена и количество слагаемых последующих членов в: 1 — отдельных текстовых полях; 2 — в одном текстовом поле, 3 — поочерёдным запросом из двух диалоговых окон; 4 — одновременным запросом из одного диалогового окна.[6]

---

3 Например, дано число и нужно переставить его цифры в обратном порядке. Тогда, если данные требуется хранить в массиве, то создаётся массив на соответствующее число элементов (равное числу цифр в числе) и каждая цифра помещается в отдельный элемент. Если же данные требуется разместить в объекте, то в объекте создаются новые свойства (в самом простом случае это — "0", "1","2"...), соответствующие номерам позиций цифр в числе, и в них помещаются сами цифры числа. Далее решается поставленная задача.

4 Считать, что сессия начинается 10 июня 2024 года.

5 Для этого достаточно задать пустой либо же заполненный атрибут текстового поля: <input disabled="">

6 Пример кода:

```
var str = "Как у тебя дела сегодня?";
```

Нужно уточнить, что, если данные поступают через вызов диалогового окна (или двух следующих поочерёдно окон), то после их считывания они отображаются в соответствующих не редактируемых текстовых полях на html-странице. Вызов диалога должен осуществляется по щелчку на соответствующую кнопку.

12. В код кнопки для вычисления суммы определённого числа членов последовательности ввести проверку на корректность введённых данных. При этом для проверки корректности ввода данных запрещено использовать функцию: 1 — isNaN(); 2 — parseFloat(); 3 — parseInt(); 4 — Number(). Но при этом требует обеспечить корректность ввода данных.

---

var res = str.split(" ");//пробел

В результате переменная res будет содержать массив: Array ["Как", "у", "тебя", "дела", "сегодня?"]

Задания на лабораторную работу

Табл. №4.2

№	(П.1)	(П.3)	(П.4)	(П.5)	(П.6)	(П.7)	(П.8)	(П.9)	(П.10)	(П.11)	(П.12)
1	4	1	2	2	1	2	2	2	7	4	3
2	2	2	2	1	1	2	1	3	1	2	2
3	23	1	2	1	2	1	2	2	3	3	2
4	26	2	1	1	2	1	1	2	5	4	3
5	14	1	1	1	1	1	2	2	5	3	4
6	7	2	1	1	2	1	3	3	3	2	1
7	12	1	1	2	2	2	2	2	4	4	4
8	27	2	1	1	2	1	3	2	2	2	3
9	31	1	1	2	2	1	2	3	6	4	3
10	27	1	2	2	2	2	1	1	4	1	2
11	2	2	1	1	2	1	2	4	3	1	3
12	15	1	1	1	2	2	4	2	3	2	2
13	24	2	1	1	2	2	3	3	2	2	3
14	26	1	1	2	2	1	1	3	6	2	2
15	3	1	1	1	2	1	2	1	3	2	2
16	17	1	2	1	2	2	2	3	3	3	1
17	20	1	1	1	1	2	1	2	5	4	1
18	26	2	2	2	1	1	1	2	1	4	3
19	18	1	1	2	2	1	1	2	2	4	3
20	26	2	2	2	1	2	3	3	3	2	1
21	5	1	1	1	1	2	3	3	8	4	2
22	13	2	2	2	2	1	2	3	4	4	2
23	24	2	1	1	2	1	4	2	6	2	1
24	16	1	2	2	2	2	2	2	3	1	2
25	27	1	2	1	1	1	2	3	4	1	4
26	22	1	2	1	1	2	3	3	7	2	4
27	14	1	1	2	2	2	4	4	1	2	2
28	11	2	1	1	1	2	1	3	1	4	1
29	10	1	2	1	1	1	2	3	3	3	1
30	9	1	2	1	1	2	2	3	2	2	3
31	5	1	1	1	2	2	1	2	6	2	4
32	31	1	1	1	1	2	1	4	2	2	2
33	17	2	1	1	1	2	3	3	5	3	2
34	22	1	1	1	2	1	3	1	3	4	2
35	21	2	1	2	2	1	1	1	7	1	3
36	27	2	2	1	1	1	2	1	2	2	3
37	19	2	2	2	2	2	3	3	7	2	3
38	15	2	2	1	2	2	1	2	4	2	2
39	5	1	2	1	1	1	2	3	4	2	3
40	11	1	1	2	2	2	3	2	5	2	1
41	25	1	2	2	2	1	2	2	7	3	2
42	24	2	1	1	2	1	2	2	5	1	2
43	17	1	2	2	2	1	3	1	3	1	4
44	9	2	1	2	1	2	3	1	7	2	4
45	31	2	1	1	1	2	2	2	6	4	4
46	4	1	1	2	1	2	2	1	1	1	4
47	14	1	1	2	2	1	2	1	6	3	3
48	2	2	2	1	2	1	3	2	3	4	1

49	25	1	2	1	2	1	2	2	5	3	2
50	8	2	1	2	1	1	3	2	6	3	2
51	7	1	2	1	1	1	1	3	6	1	2
52	4	1	1	2	1	2	3	2	3	2	2
53	20	1	1	2	1	2	1	4	8	1	3
54	4	2	2	2	1	2	3	1	5	3	3
55	19	1	2	1	1	2	3	2	3	2	2
56	24	2	2	1	1	2	3	4	1	1	3
57	27	2	2	1	2	2	2	1	3	2	1
58	25	1	2	2	2	1	4	2	2	3	2
59	25	1	1	2	2	2	1	2	4	1	4
60	29	1	1	2	1	1	3	3	2	2	2
61	24	2	2	2	1	2	2	2	8	3	1
62	6	2	2	1	2	2	3	3	6	3	3
63	20	1	2	1	2	2	3	4	7	3	3
64	32	1	1	1	2	1	1	2	4	1	2
65	22	2	1	1	1	2	1	4	3	1	1
66	23	2	2	1	1	2	3	2	3	3	4
67	29	2	1	1	1	1	2	4	7	4	3
68	12	2	1	2	1	1	2	4	3	4	4
69	27	1	2	1	1	1	2	2	2	2	2
70	3	1	2	1	1	1	2	3	2	2	3
71	10	2	2	1	2	2	3	2	7	2	2
72	2	2	2	2	1	1	1	3	6	3	1
73	26	1	1	2	2	1	2	3	5	2	2
74	11	2	1	1	2	1	3	1	1	1	3
75	25	2	1	1	1	2	3	2	6	3	4
76	13	2	1	2	2	1	1	3	6	1	2
77	12	2	2	1	2	2	2	3	7	3	3
78	30	2	1	1	1	2	3	3	6	3	4
79	6	2	2	1	1	1	3	4	5	4	3
80	21	1	1	1	2	1	3	2	4	1	1
81	25	1	1	1	1	1	3	2	7	3	3
82	29	2	2	1	2	2	1	2	7	3	1
83	25	1	2	2	1	2	2	1	5	1	2
84	8	2	2	1	1	1	3	2	5	3	2
85	6	1	1	2	1	1	1	3	6	3	2
86	8	2	1	1	2	1	1	1	3	2	2
87	19	2	2	2	2	1	1	2	7	4	3
88	8	2	1	2	2	2	3	3	4	1	1
89	13	2	2	2	1	1	3	1	3	4	2
90	21	1	2	2	1	1	2	2	2	3	2
91	29	2	1	1	2	1	4	2	2	3	3
92	2	1	1	2	1	2	4	3	3	4	2
93	18	1	1	2	2	1	3	2	6	1	3
94	5	1	1	1	1	2	4	3	2	1	2
95	15	1	1	2	2	2	2	3	4	1	3
96	24	2	2	2	2	1	1	1	4	3	1
97	31	1	2	2	2	2	2	3	1	4	4
98	19	2	1	2	1	1	2	3	4	1	1
99	7	2	1	1	2	1	1	4	3	3	3
100	11	2	2	1	2	2	2	3	1	3	3