

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федерально автономное образовательное учреждение высшего образования
«Севастопольский государственный университет»
кафедра Информационных систем

Куркчи Ариф Эрнестович

Институт информационных технологий и управления в технических системах
курс 3 группа ИС/б-31-о
09.03.02 Информационные системы и технологии (уровень бакалавриата)

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Веб-технологии»
на тему «Исследование объектной модели документа (DOM)
и системы событий Java Script.»

Отметка о зачете _____ (дата)

Руководитель практикума

ст. преподаватель
(должность)

(подпись)

А. Л. Овчинников
(инициалы, фамилия)

1. ЦЕЛЬ РАБОТЫ

Изучить динамическую объектную модель документа, предоставляемую стандартом DOM и систему событий языка JavaScript, возможность хранения данных на стороне клиента. Приобрести практические навыки работы с событиями JavaScript, деревом документа, Local Storage и Cookies.

2. ПОСТАНОВКА ЗАДАЧИ

Реализовать интерактивное графическое меню сайта, в соответствии со следующими требованиями:

- при наведении мыши на соответствующий пункт меню изображение, соответствующее этому пункту меняется на другое (в том случае если этот пункт не активен, т.е. не загружена соответствующая ему страница);

- сделать пункт меню “Мои интересы” в виде выпадающего меню, каждый элемент которого ведет на соответствующую ему подсекцию данной страницы. Выпадающее меню реализовать с помощью тегов `` и ``. Данный список должен раскрываться в зависимости от варианта задания:

1	Выпадающее меню раскрывается по наведению
---	---

Реализовать отображение в области меню сайта текущих даты и времени (обновление времени 1 раз в секунду). Формат даты определяется как остаток от деления последней цифры зачетной книжки на 4:

3	ЧЧ Месяц ГГГГ
---	---------------

(«День недели» – полная запись дня недели на русском языке, «Месяц» - полная запись названия месяца на русском языке)

3.3. На странице «Контакт» добавить поле «Дата рождения», для которого реализовать всплывающий снизу элемент «календарь».

При этом элемент должен давать возможность менять дату, месяц и код. Верстка элемента должна быть выполнена с помощью тегов `<div>` и `<select>` (для выбора месяца и года). Варианты задания элемента «календарь» приведены ниже:

3	English	день.месяц.год
---	---------	----------------

3.4. Реализовать динамическую проверку корректности заполнения пользователем формы на странице «Контакт» таким образом, чтобы при потере фокуса заполняемого поля осуществлялась проверка корректности его заполнения. В случае если поле заполнено корректно, оно должно быть подсвечено зеленым цветом, иначе оно должно быть подсвечено красным, а после данного поля должна появиться надпись, поясняющая характер ошибки. После исправления пользователем ошибки, надпись должна исчезнуть. Если все поля формы заполнены корректно должна стать активной кнопка «Отправить».

3.5. Реализовать открытие в динамически формируемом новом окне (блоке DIV) соответствующих больших фото при щелчке мыши по маленьким фото на странице «Фотоальбом».

3.6. Добавить страницу «История просмотра». На данной странице реализовать отображение двух таблиц:

- «История текущего сеанса» - в данной таблице отображается количество посещений каждой страницы за время текущего сеанса. Реализовать хранение этих данных в Local Storage.

- «История за все время» - в данной таблице отображается количество посещений каждой страницы за все время. Реализовать хранение этих данных в хранилище Cookies. Для выполнения этого задания необходимо создать два JavaScript метода: `getCookie(name)` и `setCookie(name, value, expiration_days)`.

Для реализации страницы «История просмотра» необходимо добавить на каждую страницу вызов JavaScript функции, которая будет сохранять информацию о просмотре страницы в Local Storage и Cookies.

3. ИСХОДНЫЙ КОД

calendar.js

```

1 function calendar_parse(value, not_now) {
2     if (value) {
3         value = value.split('.');
4         return new Date(value[2], value[1] - 1, value[0]);
5     }
6     return not_now ? undefined : new Date();
7 }
8
9 function calendar_month_length(date) {
10     var tmp = new Date(date.getFullYear(), date.getMonth() + 1, -1);
11     return tmp.getDate() + 1;
12 }
13
14 function calendar_week_offset(date) {
15     var tmp = new Date(date.getFullYear(), date.getMonth(), 1);
16     return tmp.getDay() == 0 ? 7 : tmp.getDay() - 1;
17 }
18
19 function calendar_day_click() {
20     var cal = document.getElementById('calendar'),
21         field = cal.parentElement.getElementsByTagName('input')[0],
22         tds = cal.getElementsByTagName('td');
23
24     for (var i = 0; i < tds.length; i++) {
25         tds[i].classList.remove('selected');
26     }
27     this.classList.add('selected');
28     field.value = leftPad(this.innerHTML, 2, '0') + '.' + leftPad(document.getElementById('calendar-month').value, 2, '0') +
29     '.' + document.getElementById('calendar-year').value;
30     fireEvent(field, 'change');
31     cal.remove();
32 }
33
34 function calendar_select_change() {
35     var cal = document.getElementById('calendar'),
36         field = cal.parentElement.getElementsByTagName('input')[0],
37         tbody = document.getElementById('calendar-body'),
38         month = document.getElementById('calendar-month'),
39         year = document.getElementById('calendar-year');
40     tbody.innerHTML = '';
41     calendar_generate_body(tbody, new Date(year.value, month.value - 1, 1), calendar_parse(field.value));
42 }
43
44 function calendar_generate_months(date) {
45     var month = document.createElement('select'),
46         months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October',
47     'November', 'December'];
48     for (var i = 0; i < months.length; i++) {
49         var option = document.createElement('option');
50         option.value = i + 1;
51         option.innerHTML = months[i];
52         if (i == date.getMonth()) {
53             option.setAttribute('selected', 'selected');
54         }
55         month.appendChild(option);
56     }
57     month.id = 'calendar-month';
58     month.addEventListener('change', calendar_select_change);
59     return month;
60 }
61
62 function calendar_generate_years(date) {
63     var year = document.createElement('select'),
64         now = new Date();
65     for (var i = now.getFullYear() - 100; i <= now.getFullYear(); i++) {
66         var option = document.createElement('option');
67         option.innerHTML = i;
68         if (i == date.getFullYear()) {
69             option.setAttribute('selected', 'selected');
70         }
71         year.appendChild(option);
72     }
73     year.id = 'calendar-year';
74     year.addEventListener('change', calendar_select_change);
75     return year;
76 }
77
78 function calendar_generate_week() {
79     var tr = document.createElement('tr');
80     var week_days = ['Mo', 'Tu', 'We', 'Th', 'Fr', 'Sa', 'Su'];

```

```

79     for (var i = 0; i < week_days.length; i++) {
80         var th = document.createElement('th');
81         th.innerHTML = week_days[i];
82         tr.appendChild(th);
83     }
84     return tr;
85 }
86
87 function calendar_generate_body(tbody, date, selected) {
88     var month_length = calendar_month_length(date),
89         week_offset = calendar_week_offset(date),
90         tr = document.createElement('tr');
91     for (var day = 1, pos = 1; day <= month_length || pos % 7 != 1; pos++) {
92
93         var td = document.createElement('td');
94         td.className = 'disabled';
95         if (week_offset > 0) {
96             week_offset--;
97         } else if (day <= month_length) {
98             td.innerHTML = day + '';
99             td.className = selected &&
100                 selected.getFullYear() == date.getFullYear() &&
101                 selected.getMonth() == date.getMonth() &&
102                 selected.getDate() == day ? 'selected' : '';
103             td.addEventListener('click', calendar_day_click);
104             day++;
105         }
106         tr.appendChild(td);
107         if (pos % 7 == 0) {
108             tbody.appendChild(tr);
109             tr = document.createElement('tr');
110         }
111     }
112 }
113
114 function create_calendar(value) {
115     var el = document.createElement('div'),
116         header = document.createElement('div'),
117         body = document.createElement('div'),
118         table = document.createElement('table'),
119         thead = document.createElement('thead'),
120         tbody = document.createElement('tbody'),
121         date = calendar_parse(value);
122
123     header.appendChild(calendar_generate_months(date));
124     header.appendChild(calendar_generate_years(date));
125     header.className = 'calendar-header';
126
127     thead.appendChild(calendar_generate_week());
128
129     calendar_generate_body(tbody, date, calendar_parse(value, true));
130     tbody.id = 'calendar-body';
131
132     table.appendChild(thead);
133     table.appendChild(tbody);
134
135     body.appendChild(table);
136     body.className = 'calendar-body';
137
138     el.appendChild(header);
139     el.appendChild(body);
140     el.id = 'calendar';
141
142     el.addEventListener('mouseleave', function() {
143         el.remove();
144     });
145
146     return el;
147 }
148
149 function open_calendar() {
150     var calendar = document.getElementById('calendar');
151     if (calendar) {
152         calendar.remove();
153     }
154     fireEvent(this, 'change');
155     this.parentNode.appendChild(create_calendar(this.value));
156 }
157
158 var once_calendar = true;
159 document.body.addEventListener('load', function () {
160     if (!once_calendar) {
161         return;
162     }
163     once_calendar = false;
164
165     var calendars = document.getElementsByClassName('js-calendar');
166     for (var i=0; i<calendars.length; i++) {
167         calendars[i].addEventListener('click', open_calendar);
168     }
169 }, true);

```

contact.js

```

1 function validate() {
2     var valid_global = validate_form(document.form_contact),
3         valid_specific = validate_contacts(document.form_contact);
4     document.form_contact.submit.disabled = valid_global || valid_specific;
5 }
6

```

```

7 function validate_contacts(form) {
8   var validates = [
9     validate_field(form.name, /^[^а-яё]+\s+[^а-яё]+\s*[^а-яё]*$/i, 'Введите в формате: Фамилия Имя Отчество'),
10    validate_field(form.tel, /^[+]?(:380(\d){9}|7\s*(\d{3})?\s*(?:-\d){7})$/i, 'Допустимы действительные номера +7 и
+380'),
11    validate_field(form.email, /^[^+@.\s]+$/i, 'Введите действительную почту')
12  ];
13  for (var i = 0; i < validates.length; i++) {
14    if (validates[i]) {
15      return true;
16    }
17  }
18  return false;
19 }
20
21 function initValidation() {
22   var form = document.form__contact,
23       inputs = form.getElementsByTagName('input'),
24       selects = form.getElementsByTagName('select'),
25       i;
26   for (i = 0; i < inputs.length; i++) {
27     inputs[i].addEventListener('change', validate);
28     if (inputs[i].type != 'checkbox' && inputs[i].type != 'radio') {
29       inputs[i].addEventListener('keyup', validate);
30     }
31   }
32   for (i = 0; i < selects.length; i++) {
33     selects[i].addEventListener('change', validate);
34   }
35 }
36
37 var once_contact = true;
38 document.body.addEventListener('load', function () {
39   if (!once_contact) {
40     return;
41   }
42   once_contact = false;
43
44   initValidation();
45 }, true);

```

main.js

```

1 function validate_form(form) {
2   var hasError = false;
3   for (var i = 0, len = form.elements.length; i < len; i++) {
4     var element = form.elements[i],
5         p = parent(element, 'form-group');
6     if (p) {
7       var messages = p.getElementsByClassName('validation-message');
8       for (var j = 0; j < messages.length; j++) {
9         messages[j].remove();
10      }
11      p.classList.remove('has-success', 'has-error');
12      if (element.required && element.value.trim() == '') {
13        p.classList.add('has-error');
14        hasError = true;
15      } else {
16        p.classList.add('has-success');
17      }
18    }
19  }
20  return hasError;
21 }
22
23 function parent(el, cls) {
24   while ((el = el.parentElement) && !el.classList.contains(cls));
25   return el;
26 }
27
28 function validate_field(field, regex, message) {
29   var hasError = false;
30   if (field) {
31     var value = field.value,
32         p = parent(field, 'form-group');
33     if (!regex.test(value)) {
34       p.classList.remove('has-success');
35       p.classList.add('has-error');
36       hasError = true;
37       if (message) {
38         var msg = document.createElement('div');
39         msg.classList.add('validation-message');
40         msg.innerHTML = message;
41         p.appendChild(msg);
42         field.setCustomValidity(message);
43       } else {
44         field.setCustomValidity('Поле заполнено некорректно');
45       }
46     } else {
47       p.classList.add('has-success');
48       p.classList.remove('has-error');
49       field.setCustomValidity('');
50     }
51   }
52   return hasError;
53 }
54
55 function navbar_hover() {
56   var navbar = document.getElementById('navbar'),
57       lis = navbar.getElementsByTagName('li');

```

```

58     for (var i = 0; i < lis.length; i++) {
59         lis[i].addEventListener('mouseover', function () {
60             this.classList.add('hover');
61         }, true);
62         lis[i].addEventListener('mouseout', function () {
63             this.classList.remove('hover');
64         }, true);
65     }
66 }
67
68 function initDropdown(cont) {
69     var drops = cont.getElementsByClassName('dropdown');
70     for (var i = 0; i < drops.length; i++) {
71         var drop = drops.item(i);
72         drop.addEventListener('mouseenter', function (e) {
73             this.classList.add('open');
74         });
75         drop.addEventListener('mouseleave', function (e) {
76             this.classList.remove('open');
77         });
78     }
79 }
80 // Storage functions
81 var storage = typeof(Storage) !== undefined ? localStorage : undefined;
82 function localStore(action, key, val) {
83     var ret_val = undefined;
84     if (storage !== undefined) {
85         var data;
86         try {
87             data = JSON.parse(storage[key]);
88         } catch (e) {
89             data = {};
90         }
91         ret_val = data;
92         if (action === 'set' && val !== undefined) {
93             data = val;
94         }
95         storage[key] = JSON.stringify(data);
96     }
97     return ret_val;
98 }
99 function localSet(key, val) {
100     return localStore('set', key, val);
101 }
102 function localGet(key, def) {
103     return localStore('get', key, def);
104 }
105 function setCookie(name, value, days) {
106     var expires = "";
107     if (days) {
108         var date = new Date();
109         date.setTime(date.getTime() + (days * 24 * 60 * 60 * 1000));
110         expires = "; expires=" + date.toUTCString();
111     }
112     document.cookie = name + "=" + JSON.stringify(value);
113 }
114 function getCookie(name) {
115     var matches = document.cookie.match(new RegExp(
116         "(?^|; )" + name.replace(/[\.\$\*\{\}\(\)\[\]\\/\+\^]/g, "\\$1") + "=([^\;]*)"
117     ));
118     return matches ? JSON.parse(decodeURIComponent(matches[1])) : undefined;
119 }
120
121 function visit() {
122     var local = localGet('visits', {}),
123         cookie = getCookie('visits'),
124         loc = window.location.pathname;
125     if (cookie === undefined) {
126         cookie = {};
127     }
128     if (!local[loc]) {
129         local[loc] = 0;
130     }
131     if (!cookie[loc]) {
132         cookie[loc] = 0;
133     }
134     local[loc]++;
135     cookie[loc]++;
136     localSet('visits', local);
137     setCookie('visits', local);
138 }
139
140 var clock = document.getElementById('clock'),
141     months = [
142         'Января', 'Февраля', 'Марта', 'Апреля', 'Мая', 'Июня', 'Июля', 'Августа', 'Сентября', 'Октября', 'Ноября', 'Декабря'
143     ];
144
145 function leftPad(str, len, ch) {
146     str = str + '';
147     len = len - str.length;
148     if (len <= 0) return str;
149     if (!ch && ch !== 0) ch = ' ';
150     ch = ch + '';
151     var pad = '';
152     while (true) {
153         if (len & 1) pad += ch;
154         len >>= 1;
155         if (len) ch += ch;
156         else break;

```

```

157     }
158     return pad + str;
159 }
160
161 function fireEvent(element, event_type) {
162     var event;
163     if (document.createEvent) {
164         event = document.createEvent('HTMLEvents');
165         event.initEvent(event_type, true, true);
166     } else {
167         event = document.createEventObject();
168         event.eventType = event_type;
169     }
170
171     event.eventName = event_type;
172
173     if (document.createEvent) {
174         element.dispatchEvent(event);
175     } else {
176         element.fireEvent('on' + event.eventType, event);
177     }
178 }
179
180 function initClock() {
181     var current = new Date();
182
183     clock.innerHTML = leftPad(current.getDay(), 2, '0') + ' ' +
184         months[current.getMonth() - 1] + ' ' +
185         current.getFullYear() + ' ' +
186         leftPad(current.getHours(), 2, 0) + ':' +
187         leftPad(current.getMinutes(), 2, '0') + ':' +
188         leftPad(current.getSeconds(), 2, '0');
189
190     setTimeout(initClock, 100);
191 }
192
193 var once = true;
194 document.body.addEventListener('load', function () {
195     if (!once) {
196         return;
197     }
198     once = false;
199     navbar_hover();
200     initDropdown(this);
201     initClock();
202
203     visit();
204 }, true);

```

history.js

```

1 function fillHistory(cont, history) {
2     var html = '';
3     for(var page in history) {
4         html += '<tr><td><a href="'+page+'"'>'+page+'</a></td><td>'+history[page]+'</td></tr>';
5     }
6     cont.innerHTML = html;
7 }
8
9 var onceHistory = true;
10 document.body.addEventListener('load', function() {
11     if(!onceHistory) {
12         return;
13     }
14     onceHistory = false;
15     var localCont = document.getElementById('history-local'),
16         cookieCont = document.getElementById('history-cookies');
17     fillHistory(localCont, localGet('visits', {}));
18     fillHistory(cookieCont, getCookie('visits'));
19 }, true);

```

ВЫВОДЫ

В ходе лабораторной работы были повторены основы работы с динамической объектной моделью документа, представляемой стандартов DOM и систему событий языка JavaScript. Написаны функции взаимодействия с деревом документа, событиями и хранилищами Local Storage и Cookies.