

Романовский Иосиф Владимирович

АЛЬБОМ «САНКТ-ПЕТЕРБУРГ» В ФОРМАТЕ HTML: ОПЫТ РАЗРАБОТКИ

Последние три года автор занимается созданием альбома «Санкт-Петербург. Любимые места». В альбоме уже больше 3500 снимков (точнее, графических файлов). Там не только снимки, и название уже не вполне отвечает содержанию.¹

Идея альбома родилась вполне естественно. Автор купил цифровой фотоаппарат, стал фотографировать город и показывать снимки друзьям и родным. Но находить заинтересованного зрителя становилось все труднее – никто не выдерживал таких объемов. Трудно было и повторять многократно одно и то же, а снимок часто выигрывает от дополнительной информации.

Вот я и решил собирать эти снимки во что-то вроде интернетовского сайта, только в Интернет не выкладывать: пользователю будет трудно работать с большим числом больших файлов, а маленькие иллюстрации не так эффектны. Скоро альбом вырос до такого размера, что его стало интересно смотреть и показывать – появились доброжелательные критики, которые дали много ценных советов: как выбирать базовые цвета и шрифты, как унифицировать страницы альбома, используя каскадную таблицу стилей (css), как организовывать меню, и многое другое. Они же исправляли мои опечатки, хотя поток новых опечаток был мощнее суммарного потока исправлений.

Обо всем сразу не расскажешь. Мы решили с редакцией сделать серию статей, и я выбрал для первого своего рассказа об альбоме одну тему, которая мне кажется наиболее программистской, – систему меню. Второй статьей, которая идет в этом же номере, будет статья Владимира Мойсеюка о представлении в альбоме известной панорамы А. Тозелли (Вид Санкт-Петербурга с башни Кунсткамеры, 1820 г.).

Итак, меню. Они выполнены в альбоме четырьмя различными способами:

1. Меню главной страницы.
2. Выбор фигуры на картинке.
3. Списковые меню.
4. Меню с картинками.

Рассмотрим их по отдельности.



*...находить заинтересованного зрителя
становилось все труднее...*

¹ Авторы часто упрямятся. Помните, у Пушкина: «Пересмотрел все это строго:/Противоречий очень много,/Но их убраться не хочу...». И я не хочу.

Листинг 1.

```
<table bgcolor="#c0d0ff" width=180 frame="box">
<tr><td><a href="Classic.html">
  Классические виды</a></td></tr>
<tr><td><a href="Churches.html">Храмы</a></td></tr>
<tr><td><a href="Famous.html">
  Некоторые знаменитые здания</a></td></tr>
<tr><td><a href="Bridges.html">Мосты</a></td></tr>
<tr><td><a href="About.html">About</a></td></tr>
</table>
```

1. МЕНЮ ГЛАВНОЙ СТРАНИЦЫ.

Это самое простое меню: почти вся страница занята красивой картинкой: вид на Стрелку и Адмиралтейство с Мытнинской набережной в белую ночь¹. Уменьшать картинку жалко. На меню остается узкая полоска с табличкой, в клетках которой написаны отдельные темы.

Примерно так (листинг 1).

Тег **<table>** определяет таблицу (а параметр **frame** определяет тип рамки ее клеток). Отдельные строки определяются тегом **<tr>**, а отдельные столбцы – тегом **<td>**. Внутри клеточки помещается название темы и навигационный тег **<a>**, задающий параметром **href** адрес вызываемой страницы, в данном случае имя вызываемого файла.

Первое меню – самое простое. Очень близок к нему выбор нужной возможности

по карте или другому изображению. Его мы сейчас и рассмотрим.

2. ВЫБОР ФИГУРЫ НА КАРТИНКЕ.

Характерный пример такого меню – ансамбль Дворцовой площади. Отсканируем план Дворцовой площади из книги [1] и средствами графического редактора доведем эту схему до аккуратного файла. В этом плане вычислим в достаточно грубом виде геометрические контуры, окаймляющие интересные нас объекты: Зимний дворец, Главный штаб, Александровскую колонну, Малый Эрмитаж, Штаб Конногвардейского корпуса, Адмиралтейство (тот, кто знает Петербург, легко опознает все эти объекты). Теперь нужно с помощью тега **<map>** описать сам план и выделенные на нем области (рисунок 1).

Получаем такой текст (с некоторыми упрощениями) (листинг 2).

Каждой области сопоставляется параметром **href** страница меню, а параметром **alt** – текст подсказки, появляющейся при попадании курсора в эту область. Параметр **name** определяет имя карты, оно появится при задании карты стандартным графическим тегом **** в качестве параметра **usemap** (на этом использование этого имени заканчивается, так что сделанный нами выбор даже слишком сложен) (листинг 3).

Этим создание меню такого типа и завершается.

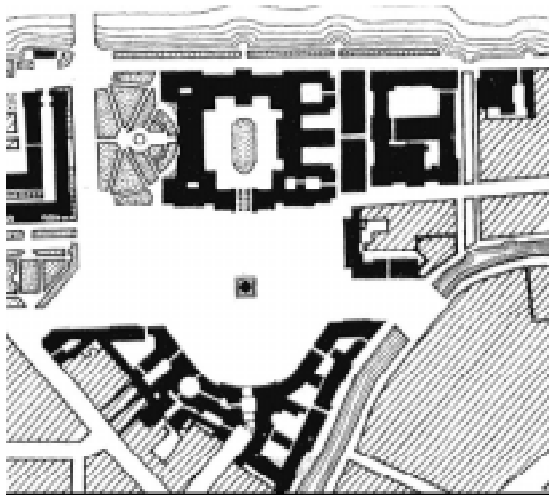


Рисунок 1.

¹ Сейчас этот благородный вид «улучшен» новоделами — «фитнес-фрегатом» у Мытнинской набережной и плавучим фонтаном у самой Стрелки.

Листинг 2.

```
<map name="dvsq">
  <area shape="rect" coords="130,52,280,180"
  href="ds_WP.html" alt="Зимний дворец">
  <area shape="rect" coords="195,230,215,250"
  href="ds_AC.html" alt="Александровская колонна">
  <area shape="rect" coords="281,54,309,162"
  href="ds_mill.html" alt="Малый Эрмитаж">
  <area shape="poly"
  coords="0,143,20,143,16,72,58,70,60,188,0,188"
  href="ds_Admir.html" alt="Адмиралтейство">
  <area shape="poly"
  coords="288,172,300,172,300,220,350,220,350,235,288,235"
  href="ds_guard.html" alt="Штаб Гвардейского корпуса">
  <area shape="poly" coords="30,290,35,280,138,278,180,325,
  234,325,274,270,320,270,292,310,240,420,120,360,50,310"
  href="ds_HQ.html" alt="Главный штаб">
  <area shape="rect" coords="0,0,468,420"
  href="ds_default.html">
```

3. СПИСКОВОЕ МЕНЮ

Рассмотрим маленький фрагмент меню «других мостов»: на странице «Мосты» самые важные водные магистрали имеют отдельные пункты, а один пункт выделен для всего остального. Таких мостов много, при-

ведем только часть меню (параметр **class** в некоторых стилях показывает использование каскадной таблицы стилей, об этом лучше говорить в другой раз) (листинг 4).

Это меню представляет собой, как и первое, таблицу, строки которой относятся к одному из двух типов – разделитель или

Листинг 3.

```

```

Листинг 4.

```
<table class=tabmenu width=960 height=540>
<SCRIPT language=JavaScript>
<!--
  MakeSeparator('Кронверкский пролив')
  MakeItem(660,'br_oth_Ioann','1. Иоанновский мост',
  'Самый старый мост Санкт-Петербурга, . . .');
  MakeItem(700,'br_oth_Kronwerk_2','2. Кронверкский мост',
  'Сам по себе . . .');
  MakeSeparator('Зимняя канавка')
  MakeItem(660,'br_oth_Hermitage','1. Эрмитажный мост',
  '<b>Эрмитажный мост</b> &mdash; самый старый . . .');
  MakeSeparator('Лебяжья канавка')
  MakeItem(700,'br_oth_Lebjazh1','1. Верхне-Лебяжий мост',
  'Коротенькая <b>Лебяжья канавка</b> прорыта . . .');
  MakeSeparator('Справочник')
  MakeSimpleItem('br_plot.html',
  'Длина и ширина многих мостов');
//-->
</SCRIPT>
</table></center>
```

Листинг 5.

[illegible]

объект для вызова. Эти строки формируются специальными процедурами, написанными на языке JavaScript: разделители – процедурой **MakeSeparator**, единственный параметр которой – это текст разделителя, а объекты – одной из процедур **MakeItem** и **MakeSimpleItem**. Их параметры мы опишем позже.

Процедура **MakeSeparator** очень проста, она формирует в таблице одну строку, в которой в нужном формате записан текст разделителя, для этого (внимание!) она формирует HTML-текст, описывающий эту строку, и передает этот текст интерпретатору браузера (листинг 5).

Видите: обычное описание строки таблицы, причем состоящей из одной ячейки, а в этой ячейке немного увеличивается шрифт и просто вставляется параметр **Text**. Получившаяся так строка передается объекту **document**, который применяет к ней метод **write** (это и есть передача строки интерпретатору браузера).

Процедура **MakeSimpleItem** используется в тех случаях, когда в ответ на запрос пользователя нужно вызвать существующую страницу. Ее имя задается параметром **URLName**, а параметр **WinTitle** определяет запись в строке меню (листинг 6).

Процедура **MakeItem** хитрее – она создает и строку в меню, такую же, как предыдущая процедура, и страницу, которую может вызвать пользователь. Строку в меню она создает примерно так же, используя такой же параметр **WinTitle**, а страницу она делает не сама, а вызывает для этого вспомогательную процедуру **LoadPic** (листинг 7).

Формирование строки достаточно понятно, а вот формирование параметров тега **<a>** явно требует комментариев. В качестве параметра **href** этого тега формируется строка, содержащая вызов упомянутой процедуры **LoadPic** и передачу ей необходимых параметров: горизонтального размера картинки **size**, имени файла с картинкой

Листинг 6.

```
function MakeSimpleItem(UrlName,WinTitle) {
    HtmlStr = '<tr><td class=tabmenu width=600>';
    HtmlStr += '<a href="' + UrlName + '">';
    HtmlStr += WinTitle + '</a></td></tr>\n';
    document.write(HtmlStr);
}
```

Листинг 7.

```
function MakeItem(size,URLName,WinTitle,Legend) {
    HtmlStr = '<tr><td class=tabmenu width=600>';
    HtmlStr += '<a href='\"javascript:LoadPic('\";
    HtmlStr += size+'\",'\"+URLName+'\"', '\"';
    HtmlStr += WinTitle+'\"', '\"' + Legend + '\"')\"';
    HtmlStr += '>' + WinTitle + '</a></td></tr>\\n';
    document.write(HtmlStr);
}
```

Листинг 8.

```

var NW = 0;
function LoadPic(size,URLName,WinTitle,Legend) {
HtmlStr = '<HTML>';
HtmlStr += '<head><title>'+ WinTitle + '</title> \n';
HtmlStr += '<META HTTP-EQUIV="Content-Type" ';
HtmlStr += ' CONTENT="text/html"></head> \n';
HtmlStr += '<LINK REL="Stylesheet" HREF="_jvr.css" ';
HtmlStr += ' TYPE="text/css"><body class=page> \n';
HtmlStr += '<center><h1 class=us>' + WinTitle + '</h1>\n';
HtmlStr += '<table class=norm width=720 ';
HtmlStr += ' cellpadding="0" cellspacing="0"> \n';
HtmlStr += '<tr><td class=c> \n';
HtmlStr += '<img id="ImN" name="ImN" ';
HtmlStr += 'src=JPG/'+URLName+'.jpg width=';
HtmlStr += size + ' align=middle></td></tr>';
HtmlStr += '<tr><td width=720 class=norm>';
HtmlStr += Legend+'</td></tr></table>';
HtmlStr += '</center></body></HTML> \n';
if (NW && !NW.closed) NW.close();
NW = window.open( "about:blank","Picture",
  "menubar=no,top=0,left=240,scrollbars=yes,
  status=yes,resizable=yes,location=no,
  width=760,height=640");
NW.document.write(HtmlStr);
}

```

URLName, того же **WinTitle**, используемого теперь в качестве заголовка, и пояснительного текста **Legend**. Отметим, что иногда сопровождающий текст оказывался слишком длинным, и приходилось либо сокращать его, либо отказываться от динамического формирования страницы.

Осталось описать процедуру формирования страницы (листинг 8).

Эта процедура готовит всю страницу полностью, формируя весь нужный файл как строку **HtmlStr**. Эта строка передается во вновь создаваемое окно **NW**, которое перед этим открывается методом окна **window.open**. Почти все его параметры я списал с чьего-то образца (впрочем, они достаточно понятны). Самому мне пришлось добавить только параметры **top=0**, **left=240**, устанавливающие положение верхнего левого угла окна.

4. МЕНЮ С КАРТИНКАМИ

Такой формат оказался очень удобным для верхних уровней меню. По совету кол-

лег, размер картинок был стандартизован; почти всегда это квадрат со стороной в 200 пикселей. Такие картинки хорошо компонуются в ряды, обычно по четыре в ряд, под ними можно что-нибудь написать, а вызвать нужный раздел можно кликнув либо подпись, либо картинку. Удобно такое меню записывать следующим образом (листинг 9).

Здесь мы видим две группы операторов. В каждой из них четыре оператора



Такие картинки хорошо компонуются в ряды, обычно по четыре в ряд...

Листинг 9.

```

<table class=menu>
<SCRIPT language=JavaScript>
<!--
s1 = MenuIcon1('dec_emb_a_ic','Набережные','dec_emb');
s2 = MenuIcon1('dec_fig_ic','Скульптуры','dec_fig');
s3 = MenuIcon1('dec_font_ic','-Фонтаны','dec_font');
s4 = MenuIcon1('dec_fence_ic',
'Решетки и ограды','dec_bars');
MenuIconRow(s1,s2,s3,s4);
s1 = MenuIcon1('dec_lamp_ic','Фонари','dec_lamp');
s2 = MenuIcon1('dec_mask_ic','Маскароны','dec_mask');
s3 = MenuIcon1('dec_board_ic',
'Мемориальные доски','dec_board');
s4 = MenuIcon1('dec_other_ic','Остальное','dec_oth');
MenuIconRow(s1,s2,s3,s4);
//-->
</SCRIPT>
</table></center>

```

MenuIcon1 создают по одной клетке строки нашей таблицы, и один оператор **MenuIconRow**, который создает из этих клеток строку таблицы. Вторая процедура записывается совсем просто: конкатенация строк **s1+s2+s3+s4** окаймляется тегом **<tr>** и его закрывающей парой (листинг 10).

Итак, осталось рассмотреть функцию **MenuIcon1** (это действительно функция, ее результатом является строка, описывающая одну клетку таблицы). Имеем листинг 11.

Но здесь все просто: полностью описана клетка таблицы, а в ней два объекта – картинка и подпись, у каждого из которых есть свой тег **<a>**. Обратите внимание на то, что имя файла с картинкой составляется из трех фрагментов, при обращении к функции каждый раз нужно указывать только специфическую переменную часть. В других страницах меню я пользовался этим генерированием еще больше, и это было во всех отношениях удобно.

Листинг 10.

```

function MenuIconRow(s1,s2,s3,s4) {
  HtmlStr = '<tr>' + s1 + s2 + s3 + s4 + '</tr>\n';
  document.write(HtmlStr);
}

```

Листинг 11.

```

function MenuIcon1(Icon,Text,URLName) {
  Str = <td class = menu><a href=' + URLName + '.html>';
  Str += '</a>\n';
  Str += '<br><a href=' + URLName + '.html>';
  Str += Text + '</a></td>\n';
  return Str;
}

```



Наши авторы, 2006.
Our authors, 2006.

*Романовский Иосиф Владимирович,
доктор физико-математических
наук, профессор СПбГУ.*