

АЛЬБОМ «САНКТ-ПЕТЕРБУРГ» В ФОРМАТЕ HTML: 2. ТОЛКОВЫЙ СЛОВАРЬ

В архитектурных и искусствоведческих описаниях появляется много терминов, которые могут вызвать затруднение. Поэтому, в ходе разработки альбома «Санкт-Петербург» я решил дополнить его небольшим словариком, в котором термины разъяснялись бы и сопровождались хорошо подобранной картинкой. Сами объяснения терминов я позволил себе черпать из разных источников (см. Литературу).

Здесь описывается, как программировался этот словарь¹, а сам словарь находится на прилагаемом к журналу диске.

1. СТРУКТУРА СЛОВАРЯ

Страница словаря, как это часто бывает в интернетовских сайтах, разделена T-образно на верхнюю часть, левую и правую



... в ходе разработки альбома ... я решил дополнить его небольшим словариком...

нижние. В верхней части находится общий заголовок и меню грубого поиска для выбора нужной начальной буквы интересующего нас слова (рисунок 1).

Такое задание окна выполняется в очень простом главном файле, который в альбоме носит название **ab_voc.html** (листинг 1).

Тег **<frameset>** определяет разбиение основного поля на подполя: по горизонтали, если задан параметр **rows** или по вертикали, если задан параметр **cols**. Значение каждого из этих параметров определяет параметры разбиения. Можно делить пространство пропорционально, а можно фиксировать размеры отдельных частей. При этом звездочка означает «все остальное».

Каждая часть имеет свое собственное описание. Тег **<frameset>** означает, что эта часть также будет разделена; так мы поступаем с нижней частью, получившей имя **D**. Тег **<frame>** означает, что часть окончательная. У нее, кроме имени, использован параметр **src**, определяющий имя файла,

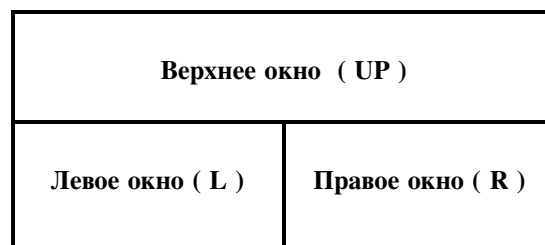


Рисунок 1.

¹ В разработке этого словаря мне очень помогли советы студентки матмеха Е. Клещинской.

значения. Текст появляется автоматически при начальном запуске словаря.

3. ЛЕВАЯ ЧАСТЬ

Файл `ac_voc_L.html` самый сложный – он содержит все словарные статьи и ссылки на иллюстративный материал. Файл состоит из двух частей: описаний используемых процедур и вызовов этих процедур. Начнем с вызовов и приведем несколько типичных примеров (листинг 3).

Мы видим здесь вызовы сравнительно небольшого количества (шести) процедур, которые и опишем по отдельности.

Процедура `MakeLetter` очень простая, она вписывает в выводимый файл строку, содержащую букву `MakeLetter` – параметр вызова, и ставит перед ним HTML-метку, на которую сошлется вызов данной буквы при грубом поиске (листинг 4) (я написал здесь этот текст короче, чем он используется в альбоме).

Процедура `MakeItem0 (Name)` еще проще, она заводит в словарном списке слово, которое пока не снабжено словарной статьей – это очень удобно при разработке, наличие слова в списке «мобилизует» автора. Параметром процедуры является объясняемое слово (листинг 5).

Листинг 3.

```

MakeLetter('A')
MakeItem1('Акант', '160', 'acanthus',
  'Акантовый узор базы колонны д. 17 на Конногвард. бульв.',
  'Травянистое растение, . . . ордера.')
MakeItem0('Алебастр')
MakeLetter('К')
MakeItem3('Кавальер', '160', 'cavalier', '160', 'cav_chess',
  'Кавальер Анны Иоанновны в Петропавловской крепости '+
  'и шахматный кавальер.',
  'Фортификационное . . . и сооружение.')
MakeItem1('Камея', '160', 'cameo',
  'Камея из собрания Эрмитажа (колл. герцога Орлеанского).',
  'Резной камень (гемма) . . . или нескольких.')
MakeItem2('Кариатида', '320', 'cariatide',
  'Кариатиды дома на наб.<br>к. Грибоедова.',
  'Статуя одетой, . . . атланты. - <i>Брок.-Ефрон</i>')
MakeItem0('Корильон')
MakeItem1('Курдонер', '160', 'courtdhonneur',
  'Курдонер Капеллы.',
  'Парадный двор . . . с воротами. - <i>ЗСП-19</i>')
MakeLetter('М')
MakeItem1('Маскарон', '160', 'maskaron',
  'Маскарон дома на 8 линии В. О.',
  'Выпуклый лепной . . . и изящества. - <i>Брок.-Ефрон</i>')
MakeItem2a('Меандр', '320', 'meandre',
  'Меандр <i>алагрек</i> с решетки Литейного моста.',
  'Один из . . . как алагрек. - <i>Брок.-Ефрон</i>')

```

Листинг 4.

```

function MakeLetter(Letter) {
  HtmlStr = '<hr width=500 color=#e0e0b7>';
  HtmlStr += '<a name='+Letter+'></a>';
  HtmlStr += '<p class=lett>'+Letter+'</p>';
  document.write(HtmlStr);
}

```

Остальные слова однотипны, разберем одно из них, а для остальных укажем их назначение.

Процедура `MakeItem1 (Name, size, Img, Cap, Leg)` создает гипертекстовый вызов процедуры `LoadItem1 (Name, size, Img, Cap, Leg)`, где `Name` – объясняемое слово, `size` – ширина картинка, `Img` – главная часть имени файла с картинкой, `Cap` – подпись под картинкой, `Leg` – словарная статья (листинг 6).

Процедура `MakeItem3 (Name, size1, Img1, size2, Img2, Cap, Leg)` используется для статей с двумя картинками (смысл параметров понятен).

Процедуры `MakeItem2` и `MakeItem2a` отличаются от `MakeItem1` только тем, что вызывают другие процедуры, иначе формирующие текст статьи и расположение подписи под картинкой.

Рассмотрим теперь особенности процедуры. `LoadItem1 (Name, size, Img, Cap, Leg)` (листинг 7).



...заводит в словарном списке слово, которое пока не снабжено словарной статьёй...

Поскольку пришлось писать пять вариантов процедуры `LoadItem`, мы сделали специальные процедуры, формирующие начало вызова и запись результата (листинг 8).

Здесь интересно формирование во второй процедуре адреса, по которому пишется готовая словарная статья: `parent.R.document`. Нужно обратить внимание также на вызов функции `open` (эту возможность я долго

Листинг 5.

```
function MakeItem0 (Name) {
    HtmlStr = Name + '<br>';
    document.write (HtmlStr);
}
```

Листинг 6.

```
function MakeItem1 (Name, size, Img, Cap, Leg) {
    HtmlStr = '<a href="javascript:LoadItem1(\'';
    HtmlStr += Name + '\', '+size+', \'' + Img + '\', \'';
    HtmlStr += Cap + '\', \'' + Leg + '\')">';
    HtmlStr += Name + '</a><br>';
    document.write (HtmlStr);
}
```

Листинг 7.

```
function LoadItem1 (Name, size, Img, Cap, Leg) {
    HtmlStr = FormHead (Name);
    HtmlStr += '<tr><td class=norm width=720>';
    HtmlStr += '<img id="ImN" name="ImN" src=IDic/';
    HtmlStr += Img + '.jpg align=left valign=top';
    HtmlStr += hspace=30 width='+size+'>'+Leg;
    HtmlStr += + '<br><font color=#20A000>';
    HtmlStr += Cap + '</font></td></tr>';
    HtmlStr += '</table></body></HTML>';
    WriteTarget (HtmlStr);
}
```

Листинг 8.

```

function FormHead(Name) {
  TStr = '<HTML><head> <META ' ;
  TStr += ' HTTP-EQUIV="Content-Type charset=windows-1251" ' ;
  TStr += ' CONTENT="text/html"></head>' ;
  TStr += '<LINK REL="Stylesheet" HREF="_jvr.css" ' ;
  TStr += ' TYPE="text/css"><body class=page> ' ;
  TStr += '<h1 class=us>' + Name + '</h1>' ;
  TStr += '<table class=norm width=720 cellspacing="0" ' ;
  TStr += ' cellpadding=0>' ;
  return TStr ;
}
function WriteTarget(HtmlStr) {
  with (parent.R.document) {
    close() ; open("", "replace") ; write(HtmlStr) ;
  }
}

```

искал). Параметр «**replace**» обеспечивает сокращение информации в стеке истории вызовов статей словаря, и возвращение из словаря происходит проще. К сожалению, мне не удалось добиться того же с грубым поиском – переходы по кнопке Назад просматривают в обратном порядке все переходы по буквам. Пусть тот, кто сможет, сделает это лучше.

4. ИЗГОТОВЛЕНИЕ ФАЙЛА СО СЛОВАРЕМ

Выписывать все, что требуется в разобранных выше форматах, оказалось доста-

точно неприятно, и мы сделали дополнительный формат разработки, который, с одной стороны, очень похож на результат, а с другой стороны технически проще. Начнем с того, что выпишем ту часть исходного файла, которая соответствует части приведенного выше фрагмента файла **ac_voc_R.html** (листинг 9).

Как видите, все поля помечены тегами, по виду напоминающими теги HTML, но на кириллической шрифтовой основе. Б – выбрано для Буквы, П – для подписи, Д – для Детального объяснения и т. д.

Для обработки этого исходного текста была написана программа на языке AWK.

Листинг 9.

```

<Б>А</Б>
<И1>Акант</И1>
<Ш>160</Ш>
<Ф>acanthus</Ф>
<П>Акантовый узор базы колонны дома 17 на . . . </П>
<Д>Травянистое растение, . . . ордера.</Д>
<И0>Алебастр</И0>
<Б>К</Б>
<И3>Кавальер</И3>
<Ш>160</Ш>
<Ф>cavalier</Ф>
<Ш>160</Ш>
<Ц>cav_chess</Ц>
<П>Кавальер Анны Иоанновны в . . . и шахматный кавальер.</П>
<Д>Фортификационное . . .
и сооружение.</Д>
<БК>

```

Листинг 10.

```
END{
  print "//->"
  print "</SCRIPT>"
  print "</table>"
  print "</td></tr></table></body>"
  print "</html>"
}
```

Сама эта программа `src2voc.awk` очень скучна, мы ее опишем совсем коротко. Она состоит из четырех частей: вспомогательной функции `OutItem()`, блока `BEGIN{...}`, печатающего шапку файла, блока `END{...}`, печатающего завершение файла, и главного блока `{...}`, последовательно накапливающего параметры вызовов процедуры и аккуратно печатающего обращения к процедурам и параметры. Все печати производятся в системный поток вывода `sysout`. В качестве примера приведем блок `END{...}` (листинг 10).

Для вызова этой процедуры был подготовлен `batch`-файл `src2voc.bat`, учитывающий, конечно, особенности расположения программ в моем компьютере (`gawk-w32` – это интерпретатор программ на AWK, предназначенный для работы в Windows32) (листинг 11).

Сейчас в словаре 150 статей, из которых почти 120 самых простых, а 8 неопианных. Он легко пополняется, когда в этом возникает потребность.

Листинг 11.

```
echo off
c:\util\gawk-w32 -f src2voc.awk ab_voc.src > ..\ab_voc_1.html
exit
```

Литература

(БСЭ) Большая Советская Энциклопедия, 3-е издание.

(Брок-Ефрон) Энциклопедический словарь; изд. Брокгауза и Ефрона. СПб., 1892–1907.

(ЗСП-19) Зодчие Санкт-Петербурга XIX–начало XX века, сост. В.Г. Исаченко, СПб.: Лениздат, 1998.

(А.Г.Булах) Книги А.Г. Булаха и его соавторов в серии «Каменное убранство Петербурга», СПб.: изд. «Сударыня», 1987–2004.

(Б.М.Кириков) Б.М. Кириков. Архитектурные памятники Санкт-Петербурга. Стили и мастера. СПб.: изд. «Белое и Черное», 2003.

(МАЭ) Н.И. Баторевич, Т.Д. Кожицева. Малая архитектурная энциклопедия. СПб.: изд. «Дмитрий Буланин», 2005.

(МинСССР) Минералы и горные породы СССР. Отв. ред. А.И. Гинзбург. М.: изд. «Мысль», 1970.

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



Наши авторы, 2006.
Our authors, 2006.