

Степанов Алексей Владимирович

ЯЗЫКИ РАЗМЕТКИ. ЧАСТЬ 3: ВЕРСТКА ТАБЛИЦ

В этой статье мы продолжаем описание локальной разметки, разбирая верстку таблиц в LaTeX и HTML.

РАЗМЕТКА ТАБЛИЦ В TEX

LaTeX, также как и Plain TeX, предоставляет два вида таблиц: с предопределенной и с вычисляемой шириной столбцов. Таблицы с предопределенной шириной столбцов могут автоматически переноситься на следующую страницу, прерываться обычным текстом, и возобновляться с той же шириной столбцов. Кроме того, верстка таких таблиц происходит очень быстро (если для вас имеет значение, сколько десятых долей секунды занимает у TeX форматирование страницы). Однако с помощью синтаксиса таблиц с предопределенной шириной столбцов неудобно делать линованные таблицы, да и вообще чаще всего хочется предоставить компьютеру самому выбирать ширину столбцов. Единственным примитивом TeX, предназначенным для изготовления таблиц, является макрокоманда `\halign` (horizontal align – горизонтальное выравнивание). С помощью этой команды можно сделать абсолютно любую таблицу. Все остальные возможности (La)TeX по верстке таблиц являются специализацией этой макрокоманды для различных часто встречающихся случаев. Мы пойдем от простого к сложному: сна-

чала разберем таблицы с предопределенной шириной столбцов, затем стандартные средства LaTeX для верстки таблиц с вычисляемой шириной столбцов, а затем коснемся команды `\halign`. Так как в любой книге про (La)TeX есть глава, посвященная верстке таблиц, мы приведем только основные конструкции и идеи, а также затронем некоторые тонкости, встретившиеся мне на практике.

ИМИТАЦИЯ ПОЗИЦИЙ ТАБУЛЯТОРА

Эта возможность в LaTeX сделана существенно удобнее, чем в Plain TeX. Кроме того, в данном случае LaTeX не надстраивается над, а переопределяет конструкции Plain TeX, поэтому мы разберем только синтаксис LaTeX. Итак, для верстки простой таблицы в LaTeX можно воспользоваться окружением `tabbing`. Внутри этого окружения:

- макрокоманда `\=` отмечает новую позицию табулятора;
- макрокоманда `\>` просит TeX перейти в начало следующего столбца (пробелы после `\>` игнорируются, так что текст, следующий за `\>`, будет напечатан в точности с начала следующего столбца даже если он налезает на предыдущий);
- макрокоманда `\\` отмечает переход на новую строку;
- макрокоманда с суровым названием `\kill` отмечает конец строки, которая не будет выведена на печать; такие строки

предназначены для установки позиций табулятора.

Для того, чтобы понять, как эти команды работают, приведем пример из книги Львовского¹. Если написать листинг 1, то на печати получим рис. 1.

В первой строке мы установили две позиции табуляции, во второй – после двух команд `\>` мы установили третью позицию табулятора. Так как в четвертой строке команда `\=` дана после первой `\>`, то она переставляет вторую позицию табулятора.

Листинг 1

```
\begin{tabbing}
парочка \= позиций \= табуляции \kill
\> плюс \> еще одна здесь:\> \\
теперь \> их \> уже \> три.\\
Вторую \> мы\quad \= сменим \> и посмотрим:\\
где \> эти \> позиции \> теперь.\\
\end{tabbing}
```

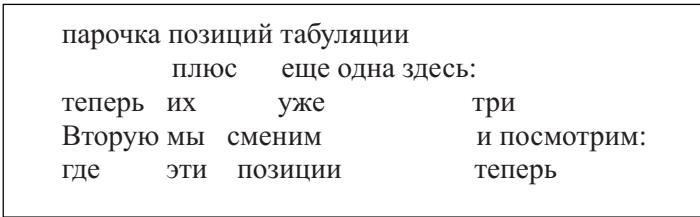


Рис. 1

Листинг 2

```
\newdimen\OneThird \newdimen\ABitSmaller
\OneThird=\textwidth \divide\OneThird by 3 % треть ширины текста
\ABitSmaller=\OneThird \advance\ABitSmaller-.5cm
% 0.5 сантиметра - это расстояние между столбцами
\begin{tabbing}
\hspace{\OneThird}\=\hspace{\OneThird}\=\kill
первый столбец\> второй столбец\> третий столбец\\[10pt]
\vtop{\hsize=\ABitSmaller А теперь, в этом вертикальном боксе можно писать
сколь угодно длинный текст, который будет выровнен по ширине в пределах hsize}\>
\vtop{\hsize=\ABitSmaller \raggedright
А в этой ячейке текст будет выровнен по левому краю за счет команды raggedright
(неровный правый край).} \>
\vtop{\hsize=\ABitSmaller \centering
Наконец, третий столбец демонстрирует возможность выравнивания по центру.}\\
\end{tabbing}
```

Приведем еще несколько возможностей окружения `tabular`, не разбирая их подробно:

- команда `\l` имеет необязательный аргумент (который, как обычно, нужно писать в квадратных скобках) – вертикальный отступ, добавляемый после текущей строки, а также вариант со звездочкой, который запрещает разрыв страницы в этом месте;

- команда `\pushtabs` запоминает текущую установку позиций табулятора, а команда `\poptabs` «вспоминает» эту установку;

- команды `\'` и `\'` задают в ячейке (соответственно, строке) таблицы режим выравнивания по правому краю.

В заключение этого раздела хочется подчеркнуть, что без дополнительных усилий текст в колонках никогда не будет не только выравниваться по ширине, но даже и переноситься на другую строку. Если текст будет слишком длинным, то ячейки будут просто налезать друг на друга. Для того чтобы обеспечить режим выравнивания, нужно написать что-нибудь вроде листинга 2, на печати получим рис. 2.

¹ Львовский С.М. Набор и верстка в пакете TeX.

первый столбец	второй столбец	третий столбец
А теперь в этом вертикальном боксе можно писать сколь угодно длинный текст, который будет выровнен по ширине в пределах <code>hsize</code>	А в этой ячейке текст будет выровнен по левому краю за счет команды <code>raggedright</code> (неровный правый край).	Наконец третий столбец демонстрирует возможность выравнивания по центру.

Рис. 2

Команды `\vtop` и `\hsize` будут обсуждаться позже. Значение команд `\raggedright` и `\centering` понятно из контекста.

СТАНДАРТНЫЕ ТАБЛИЦЫ LATEX

Чаще всего для верстки таблиц в LaTeX используется окружение `tabular`. Это окружение имеет параметр, называемый преамбулой таблицы. *Преамбула* – это строка, указывающая режим выравнивания ячеек и наличие вертикальных разделителей между ними. В преамбуле на месте каждого столбца должно стоять одно из следующих выражений:

- `l` – выравнивание по левому краю;
- `c` – выравнивание по центру;
- `r` – выравнивание по правому краю;
- `p{dim}` – выравнивание по ширине, где `dim` – это ширина столбца таблицы в любых определенных в TeX единицах измерения.

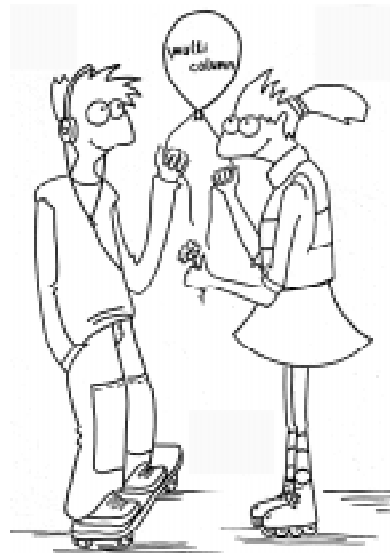
Между этими выражениями, слева или справа от них, может стоять вертикальная линия (символ «|»), обозначающая, что между этими столбцами (соответственно, с левого или с правого края таблицы) должна идти вертикальная линия. На месте символа «|» может также находиться так называемое *at-выражение*, то есть запись вида `@{...}`, где многоточие обозначает произвольную последовательность токенов. Эта последовательность будет вставлена в каждую строку таблицы между соответствующими столбцами.

Внутри окружения `tabular` ячейки таблицы отделяются друг от друга символом `&`, а строки, так же, как и в окружении

`tabbing`, макрокомандой `\\`. Как и раньше, эта команда имеет необязательный аргумент – дополнительный вертикальный промежуток между строками.

Горизонтальные линии по всей ширине таблицы производятся командой `\hline`, данной между строками. Команда `\cline{m-n}` производит горизонтальную линию, простирающуюся от `m`-ого до `n`-ого столбца включительно. Для того чтобы вставить между строками таблицы произвольный вертикальный материал, необходимо воспользоваться нераскрываемой макрокомандой `\nolign{...}`, аргументом которой является вставляемый материал.

Для объединения ячеек в строке таблицы LaTeX предлагает макрокоманду `\multicolumn` взамен примитива¹ `\multispan`. Обе эти команды принимают в качестве



Для объединения ... LaTeX предлагает макрокоманду `\multicolumn`...

¹ *Примитив* – это нераскрываемая макропоследовательность, без всяких негативных эмоций.

первого аргумента количество объединяемых столбцов и, естественно, отменяют форматирование, заданное в преамбуле. Однако у команды `\multispan` больше никаких аргументов и нет, далее до символа `&` или `\\` следует содержимое объединенной ячейки. Выравнивание материала в этом случае может быть задано командами низкоуровневого интерфейса типа `\hfill`. В отличие от этого, `\multicolumn` имеет второй аргумент – способ выравнивания и окружение ячейки, то есть вертикальные линии (смысла в `at`-выражении в данном случае я не вижу, ведь последовательность токенов будет вставлена только в одну ячейку). Содержимое объединенной ячейки должно быть третьим параметром макрокоманды `\multicolumn`. Необходимо отметить, что левый столбец содержит обе окружающие его вертикальные линии или `at`-выражения (если они есть), а остальные столбцы – только правую линию или `at`-выражение. Таким образом, если в преамбуле написано `{|l|l|l|}`, то есть все ячейки выровнены влево и разделены вертикальными линиями, а вы хотите объединить вторую и третью ячейки, оставив вертикальные линии, то в соответствующей строке надо написать

текст первой ячейки & `\multicolumn`
`2{1|}{текст второй ячейки}\\`

Вертикальную черту перед буквой `l` ставить не надо, потому что в данном случае второй параметр команды `\multicolumn` заменяет последовательность символов `l|l|` из преамбулы, а вертикальная черта, разделяющая первый и второй столбцы, относится к первой ячейке.

Для вертикального объединения ячеек никаких специальных средств в LaTeX не предусмотрено. Это всегда можно сделать, например, вставив таблицу в ячейку таблицы. В связи с этим надо отметить, что таблица, созданная окружением `tabular`, является обычным боксом, который может быть вставлен куда угодно, как будто он является одной большой буквой. По умолчанию на базовой линии строки будет находиться середина этого бокса. Умолчание можно изменить с помощью необязательного параметра окружения `tabular`, который принимает значение `t` (от слова `top`) для выравнивания по верхнему краю или `b` (от слова `bottom`) для выравнивания по нижнему краю.

Написав страницу текста про окружение `tabular`, я немного устал. Когда я

Листинг 3

```
\newdimen\colwidth \colwidth=1cm
\begin{tabular}{|l|l|l|p{\colwidth}|p{\colwidth}|p{\colwidth}|}
    p{\colwidth}|p{\colwidth}|p{\colwidth}|p{\colwidth}|c|c|}
\noalign{\centering{\large\bfseries 6399}\par \vspace{5pt}}
\noalign{\hrule height .8pt}
N. & фамилия & Имя & \multicolumn{3}{c|}{Формулировки} &
\multicolumn{4}{c|}{Задачи} & Сумма & Оценка \\
\cline{4-10}
&& \multicolumn{1}{c|}1 & & \multicolumn{1}{c|}2
& \multicolumn{1}{c|}3 & & \multicolumn{1}{c|}1
& \multicolumn{1}{c|}2 & & \multicolumn{1}{c|}3
& \multicolumn{1}{c|}4 & & \cr
\noalign{\hrule height .8pt}
1. & Петров & & Иван & & & & & & & \cr
\hline
2. & Сидорова & & Мария & & & & & & & \cr
\hline
3. & \dots & & \dots & & & & & & & \cr
\noalign{\hrule height .8pt}
\end{tabular}
```

6399											
№	Фамилия	Имя	Формулировки			Задачи				Сумма	Оценка
			1	2	3	1	2	3	4		
1.	Петров	Иван									
2.	Сидорова	Мария									
3.									

Рис. 3

пользовался этим окружением, а не объяснял, как с ним работать, мне казалось, что все намного проще. Теперь, чтобы и вам показалось, что все просто, посмотрите на пару примеров. Первый из них – абсолютно реальный. Такую табличку (листинг 3, рис. 3) я распечатываю к каждому экзамену по алгебре второго семестра (номер группы и фамилии вымышлены).

В преамбуле `p{\colwidth}` поставлено для того, чтобы фиксировать размер столбца (этот размер равен 1 см, для того чтобы удобно было вписывать доли балла, которые студент может заработать за задачу или формулировку). Выражение `\multicolumn{1}` – это способ отказаться в данной ячейке от выравнивания, указанного в преамбуле. Пересечение вертикальных двойных вертикальных линеек с горизонтальными выглядит не очень красиво. Избежать этого можно с помощью пакета `hhline` (то есть написать в преамбуле документа `\usepackage{hhline}`, после чего пользоваться дополнительными

возможностями, предоставляемыми этим пакетом), однако здесь мы не будем обсуждать подобные тонкости¹.

Второй пример взят из книги Львовского. Вместо окружения `tabular` в этом примере использовано окружение `array`, которое полностью эквивалентно `tabular`, кроме того, что в ячейках таблицы `array` стоят математические формулы. Более подробно набор формул будет обсуждаться в следующей статье. Итак, мы верстаем деление многочленов в столбик (см. листинг 4, рис. 4).

$$\begin{array}{r|l}
 x^2 + 2x - 12 & x + 5 \\
 x^2 + 5x & x - 3 \\
 \hline
 - 3x - 12 & \\
 - 3x - 15 & \\
 \hline
 3 &
 \end{array}$$

Рис. 4

Листинг 4

```

$$
\arraycolsep=0.05em
\begin{array}{r|l}
\rrr@{\,}r|r
x^2\{+2x\}-12\&\,x+5\
\cline{5-5}
x^2\{+5x\}\&\,x-3\
\cline{1-2}
\&\{-3x\}-12\
\&\{-3x\}-15\
\cline{2-3}
&3\
\end{array}
$$

```



...фиксировать размер столбца... вписывать доли балла, которые студент может заработать...

¹ Вообще, для изучения тонкостей, кроме документации MikTeX (которая устроена не плохо, а очень плохо!), я использую книгу «Путеводитель по пакету LaTeX и его расширению LaTeX2e» авторов М. Гуссен, Ф. Миттельбах, А. Самарин.

Размер `\arraycolsep` определяет расстояние между столбцами таблицы. В этом примере мы его сильно уменьшили, чтобы расстояние между слагаемыми было разумным. Макрос `\`, означает тонкий пробел. Обратите внимание, что в четвертом столбце таблицы ни в одной строке нет текста. Этот столбец введен только для того, чтобы там печаталась вертикальная линия. Если убрать из преамбулы букву `r` между ат-выражением и вертикальной чертой, и, соответственно, убрать лишние знаки `&` из первой и второй строки, то вертикальная линия, будучи теперь собственностью третьего столбца, напечаталась бы во всех строках таблицы. Знаки `$$` означают переход в «выделенный математический режим», который будет обсуждаться в следующей статье. Наконец, фигурные скобки перед знаками «+» и «-» стоят для того, чтобы TeX воспринимал эти знаки как знаки бинарных операций и проставил бы соответствующие этой ситуации пробелы вокруг них. Такие тонкости в цикле ознакомительных статей естественно обсуждаться не будут, человек, не связанный с полиграфией, скорее всего не заметит никакой разницы. Я оставил фигурные скобки из оригинала только для того, чтобы показать, что такие тонкости вообще существуют, и нам комфортно читать качественные математические тексты потому, что есть профессионалы, обращающие внимание на эти тонкости.

Кроме уже упомянутого размера `\arraycolsep` я знаю еще два параметра, влияющих на внешний вид таблиц, созданных окружением `tabular` или `array`. Один из них называется `\arraystretch` и отвечает за расстояние между строками таблицы. По умолчанию он равен 1. Для того чтобы увеличить его, скажем, в полтора раза, достаточно написать `\def\arraystretch{1.5}` или `\renewcommand\arraystretch{1.5}` на диалекте LaTeX. Обратите внимание, что `\arraystretch` является коэффициентом, а не размером, и поэтому если записать `\arraystretch=1.5`, то на печати получим просто $1=1.5$ вместо желаемого увеличения межстрочного расстояния. И еще одно замечание, касающееся `\arraystretch`. Этот

параметр задает *минимальное* расстояние между базовыми линиями строк. Если же реальная высота строки больше, чем `\baselineskip`, умноженный на `\arraystretch`, то никакой дополнительный промежуток вставляться не будет.

Второй параметр называется `\arrayrulewidth`. Он является размером и определяет толщину линеек. По умолчанию он равен `0.4pt`. Однако вы уже видели в первом примере этого раздела, что для увеличения толщины конкретных линеек я использовал конструкцию, использующую примитив `\noalign`. Дело в том, что `\arrayrulewidth` действует на *все* линейки в таблице, в то время как менять обычно хочется толщину только некоторых из них. Возможность менять толщину некоторых вертикальных линеек предоставляет пакет `array`. Он же предоставляет массу других дополнительных возможностей для форматирования таблиц, которые описаны, например в книге Львовского.

Однако если мне нужны какие-то нестандартные таблицы, то я предпочитаю пользоваться примитивом `\halign`. Кроме того, что он абсолютно универсален, он позволяет верстать таблицы предопределенной ширины с автоматически вычисляемой шириной столбцов – возможность, которой хочется пользоваться постоянно, но которую я не нашел в LaTeX (может быть, правда, плохо искал).

ВЕРСТКА ТАБЛИЦ С ПОМОЩЬЮ \HALIGN

Тело таблицы, набранной с помощью `\halign`, почти не отличается от тела таблицы внутри окружения `tabular`, только строки отделяются примитивом `\cr` вместо макрокоманды `\\`. Они отличаются преамбулой и возможностью в `\halign` задать ширину таблицы и (растяжимые) промежутки между столбцами. Преамбула `\halign`-таблицы – это строка-образец, в которой реальный текст ячейки заменен на символ макропараметра `#`. Итак, синтаксис `\halign`-таблицы таков.

```
\halign to <промежуток> {<строка-образец>\cr
<строка>\cr...<строка>\cr}
```

При этом за расстояние между столбцами таблицы отвечает промежуток `\tabskip`, который можно менять как снаружи таблицы, так и в строке-образце. Этот промежуток является примитивом TeX, поэтому его нельзя менять командами LaTeX типа `\setlength`, а нужно писать просто `\tabskip=<размер> plus <размер> minus <размер>`.

Такие команды являются локальными, то есть их действие теряется при выходе из группы, ограниченной окружением, или фигурными скобками, или каким-нибудь другим образом. Для того чтобы изменить такое поведение, перед названием промежутка (в данном случае `\tabskip`) можно написать макрокоманду `\global`. Обратите внимание на то, что ячейка таблицы по определению является группой.

Расплатой за универсальность является необходимость определять выравнивание в ячейке с помощью низкоуровневых команд типа `\hfil` и `\hfill`. Напомню, что, являясь бесконечно растяжимыми промежутками, они действуют в ячейке как пружины: например, находясь слева от текста, они прижимают текст вправо.

В качестве примера сверстаем (слегка усложненную) таблицу из раздела «Имита-



Тело таблицы, набранной с помощью `\halign`, почти не отступает от тела таблицы внутри окружения `tabular`...

ция позиций табулятора» (см. листинг 5). На печати получим рис. 5.

При этом вертикальные линии поставлены, для того чтобы были четко видны края столбцов и «табличный клей» `\tabskip` между ними. Так как в этом примере в каждом столбце все ячейки имеют одинаковую ширину, то пользоваться командами `\hfill` для выравнивания не пришлось.

Листинг 5

```
\newdimen\OneThird \newdimen\ABitSmaller
\OneThird=\textwidth \divide\OneThird by 3 % треть ширины текста
\ABitSmaller=\OneThird \advance\ABitSmaller-.5cm
% 0.5 сантиметра - это расстояние между столбцами
\parindent=0pt
\halign to \textwidth {% строка-образец
\tabskip=0pt plus 1fill\vrule\top{\hsize=\ABitSmaller #}\hfil\vrule &
\tabskip=0pt plus 1fill\vrule
\top{\hsize=\ABitSmaller \raggedright #}\vrule&
\tabskip=0pt\vrule
\top{\hsize=\ABitSmaller \centering {#}}\vrule \cr
% тело таблицы
первый столбец & второй столбец & третий столбец\cr
\hline
```

А теперь, в этом вертикальном боксе можно писать сколь угодно длинный текст, который будет выровнен по ширине в пределах `hsize.&`
А в этой ячейке текст будет выровнен по левому краю за счет команды `raggedright` (неровный правый край).&
Наконец, третий столбец демонстрирует возможность выравнивания по центру.\cr}

первый столбец	второй столбец	третий столбец
А теперь в этом вертикальном боксе можно писать сколь угодно длинный текст, который будет выровнен по ширине в пределах hsize	А в этой ячейке текст будет выровнен по левому краю за счет команды raggedright (неровный правый край).	Наконец третий столбец демонстрирует возможность выравнивания по центру.

Рис. 5

Между строк таблицы можно вставить любой материал командой `\noalign`. Команда `\hrule`, как вы видели, тоже работает, потому что она является просто сокращением для `\noalign{\hrule}`. Ячейки в строке можно объединять с помощью `\multispan`, а вот команда `\multicolumn` работать не будет (при этом сообщение об ошибке совершенно не адекватное!). Для отмены шаблона в данной ячейке можно написать команду `\omit` сразу после знака `&`, открывающего ячейку (пробелы между ними возможны).

Еще одна возможность `\halign`, полезная особенно при написании макропакетов¹, – это возможность написать шаблон таблицы с неопределенным количеством столбцов. Например, макрокоманда `\matrix` в Plain TeX определена следующим образом (излишние тонкости пропущены):

```
\def\matrix#1{\,\vcenter {\halign
  {\hfil $$$\hfil &&\quad
   \hfil $$$\hfil \cr #1\cr}}\,}
```

Здесь общий шаблон для всех столбцов, кроме первого, задается после символов `&&`. Несмотря на то, что шаблон для первого столбца совпадает с шаблонами остальных, начать строку-образец с символов `&&` нельзя, шаблон для первого столбца должен обязательно присутствовать. Внутри макроопределения символ `#n` обозначает макропараметр с номером `n`, поэтому, когда нужно употребить символ `#` в другом смысле, его приходится удваивать, то есть

писать `##`. Знаки доллара означают начало и конец математического режима. Прimitives `\cr` раскрывается в `\cr`, игнорируя при необходимости макрос `\cr`, присутствующий перед ним. Это делается для того, чтобы пользователь мог набрать `\matrix{1&2\cr 3&4}` или `\matrix{1&2\cr 3&4\cr}`, и ничего не испортилось бы.

Тонкости работы с `\halign`-таблицами можно прочитать в «Библии TeX-а»². Более поздние книги старательно замалчивают этот вопрос.

ТАБЛИЦЫ В HTML

Как вы уже наверняка заметили, в языках разметки запись идет в одну строчку. Грубо говоря, символы перевода каретки приравниваются к пробелу. Поэтому невозможно разбить поле на клеточки и потом в каждую клеточку что-то вставлять. Нужно эти клеточки описывать последовательно. Поэтому ничего удивительного, что в HTML таблицы имеют следующий синтаксис (см. листинг 6). Если взять все параметры по умолчанию и написать

```
<TABLE>
  <TR>
    <TD> 1,1 </TD>
    <TD> 1,2 </TD>
  </TR>
  <TR>
    <TD> 2,1 </TD>
    <TD> 2,2 </TD>
  </TR>
</TABLE>
```

¹ Макропакет – это пакет, то есть файл, с дополнительными макроопределениями. От стилевого пакета отличается только целью: в стилевом пакете большей частью меняются параметры по умолчанию для существующих макрокоманд.

² Донльд Кнут. Все про TeX. (Оригинальное название The TeXBook).

Листинг 6

```
<TABLE атрибуты таблицы>
<ColGroup атрибуты группы столбцов>
  <TR атрибуты строки>
    <TD атрибуты ячейки> (содержимое ячейки) </TD>
    (повтор - по количеству ячеек в данной строке)
  </TR>
  (повтор по количеству строк)
</TABLE>
```

то в результате получится:

1,1	1,2
2,1	2,2

Таким образом, все, что можно изучать про построение таблиц в HTML – это ат-

рибуты тэгов `<TABLE>`, `<TR>`, `<TD>` и `<ColGroup>`. Большая часть из них приведена в таблице 1. Все доступные атрибуты можно посмотреть на сайте компании w3c, разрабатывающей стандарты интернет-сообщества, или в любой книге по языку HTML. Знак «-» в клетке таблицы означает, что

Таблица 1. Атрибуты

	align (valign)	border	cellspacing cellpadding	frame	rules	height width	colspan rowspan
<TABLE>	Выравнивание всей таблицы относительно объемлющего элемента	Толщина рамок	Расстояние от содержимого до рамки, соответственно, расстояние между рамками	Какие рамки рисовать вокруг всей таблицы	Рамки ячеек	Высота, ширина	-
<TR>	Выравнивание (по вертикали) внутри ячеек данной строки	-	-	-	-		-
<TD>	Выравнивание (по вертикали) внутри данной ячейки	-	-	-	-		Количество столбцов (строк), на которые распространяется данная ячейка
<ColGroup> span = количество столбцов в группе	Выравнивание внутри всех ячеек данной группы столбцов	-	-	-	-		-
значение	left right center (для valign: top bottom middle)	Целое число	в пикселях или в процентах	above below hsides lhs rhs vsides void	all cols rows none groups	в пикселях или в процентах	число

данный атрибут отсутствует у данного тэга, если же клетка оставлена пустой, то предполагается, что вы сумеете догадаться, что там должно быть написано.

Например, при наличии заголовка `<TABLE border = 1>` будет напечатана таблица, каждый элемент будет обведён тонкой линией, и вся таблица тоже будет обведена.

В заключение приведем простой пример. Допустим, вы хотите набрать следующее рас-

писание (выравнивание в одном столбце по левому краю, в другом по центру).

пн	9 – 18
вт	10.30 – 18.30
ср	10 – 19

Для этого надо всего лишь написать листинг 7. Правда просто?

Продолжение следует.

Листинг 7

```
<P align=center>  
  <TABLE border=1, cellspacing=0, cellpadding=3, bordercolor=black>  
  <COLGROUP SPAN=1 align = left>  
  <COLGROUP SPAN=1 align = center>  
    <TR>  
      <TD> пн </TD>  
      <TD> 9 &ndash; 18 </TD>  
    </TR>  
    <TR>  
      <TD> вт </TD>  
      <TD> 10.30 &ndash; 18.30 </TD>  
    </TR>  
    <TR>  
      <TD> ср </TD>  
      <TD> 10 &ndash; 19 </TD>  
    </TR>  
  </TABLE>  
</P>
```



Наши авторы, 2008.
Our authors, 2008.

*Степанов Алексей Владимирович,
доцент кафедры Высшей
математики № 2 СПбГЭТУ
«ЛЭТИ».*