

The `cellprops` package

CSS-like cell and table properties*

Julien “_FrnchFrgg_” RIVAUD†

Released 2018/07/16

1 `cellprops` documentation

This package reworks the internals of `tabular`, `array`, and similar constructs, and adds a `\cellprops` command accepting CSS-like selectors and properties. It implements the `border-collapse: separate` CSS model.

It depends on `mdwtab`, `xcolor` and of course `expl3` and `xparse`.

`cellprops` default settings mimic the LaTeX layout, that is left and right padding equal to `\tabcolsep` or `\arraycolsep`, zero top and bottom padding, but minimum height and depth corresponding to the table strut box.

I recommend to add globally:

```
cellprops{ td { padding: 1ex; min-height: Opt; min-depth: Opt; } }
```

so that you get better-looking tables by default.

1.1 Examples

To produce:

| | | | |
|--------------|--------------------|-------------------|-------|
| This is text | A_2 | A_3 | A_4 |
| B1 | <i>Thisismaths</i> | B_3 | |
| C1 | C_2 | X | Y |
| D1 | D_2 | DX | v |
| E | F | $\int_a^b f(t)dt$ | v |

you can use:

```
\[
\cellprops{
  td {
    padding: 1ex;
    min-height: Opt;
  }
}
```

*This file describes v1.4, last revised 2018/07/16.

†E-mail: frnchfrgg@free.fr

```

        min-depth: 0pt;
        border-style: none solid solid none;
        text-align: center;
    }
    table {
        background-color: black!5!white;
    }
    tr:nth-child(even) {
        background-color: black!15!white;
    }
    td:nth-child(even) {
        background-color: yellow!20!white
    }
    tr:nth-child(even) td:nth-child(even) {
        background-color: yellow!50!white;
    }
    tr:first-child td {
        border-top-style: solid;
    }
    td:first-child {
        border-left-style: solid;
        math-mode: text;
        text-align: left;
    }
}
\begin{array}{nnnn}
This is text & A_2 & A_3 & A_4 \\
B_1 & This is maths & B_3 & \\
C_1 & C_2 & X & Y \\
D_1 & D_2 & DX & v \\
E & F & \int_a^b f(t) dt & v
\end{array}
\]

```

You can also use the `longtable` environment:

| | | | |
|-------|-------|--------|-------|
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |

| | | | |
|-------|-------|--------|-------|
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |
| aaaaa | baaaa | caaaaa | dbbbb |

This table has been produced by:

```
cellprops{
  td { border: thin solid black; }
  tr:nth-child(4n) td:first-child,
  tr:nth-child(4n+1) td:nth-child(2),
  tr:nth-child(4n+2) td:nth-child(3),
  tr:nth-child(4n+3) td:nth-child(4) {
    border: thick solid red;
  }
}

begin{longtable}{nnnn}
  aaaaa & baaaa & caaaaa & dbbbb \\
  ...
  aaaaa & baaaa & caaaaa & dbbbb \\
end{longtable}
```

1.2 Usage guide

```
<usage>:      '\cellprops{' [ <selectors> '{' <properties> '}' ]* '}'
<selectors>:  <selector> [, <selectors> ]
<selector>:   [<environment> ' ' ] <element1>
<element1>:   'table' | 'tr' [<pseudo-class>] [ ' ' <element2> ] | <element2>
<element2>:   'td' [<pseudo-class>] [ ' ' <parbox> ] | <parbox>
<parbox>:     'p'
<pseudo-class>: ':nth-child(' <nth> ')'
<nth>:        <number> | 'odd' | 'even' | <number>'n+' <number>
<properties>: [ <property> ';' ]*
```

```

<property>: 'padding: ' ( <dimension> ) {1,4} |
'padding-top: ' <dimension> |
'padding-right: ' <dimension> |
'padding-bottom: ' <dimension> |
'padding-left: ' <dimension> |
'min-height: ' <dimension> |
'min-depth: ' <dimension> |
'min-width: ' <dimension> |
'text-align: ' ( 'left' | 'right' | 'center' ) |
'math-mode: ' ( 'text' | 'math' | 'auto' ) |
'color: ' <color> |
'background-color: ' ( <color> | 'transparent' ) |
'border: ' [ <bd-width> ] [ <bd-style> ] [ <color> ] |
'border-top: ' [ <bd-width> ] [ <bd-style> ] [ <color> ] |
'border-right: ' [ <bd-width> ] [ <bd-style> ] [ <color> ] |
'border-bottom: ' [ <bd-width> ] [ <bd-style> ] [ <color> ] |
'border-left: ' [ <bd-width> ] [ <bd-style> ] [ <color> ] |
'border-width: ' ( <bd-width> ) {1,4} |
'border-top-width: ' <bd-width> ) |
'border-right-width: ' <bd-width> ) |
'border-bottom-width: ' <bd-width> ) |
'border-left-width: ' <bd-width> ) |
'border-style: ' ( <bd-style> ) {1,4} |
'border-top-style: ' <bd-style> ) |
'border-right-style: ' <bd-style> ) |
'border-bottom-style: ' <bd-style> ) |
'border-left-style: ' <bd-style> ) |
'border-color: ' ( <color> ) {1,4} |
'border-top-color: ' <color> ) |
'border-right-color: ' <color> ) |
'border-bottom-color: ' <color> ) |
'border-left-color: ' <color> )

<color>: 'inherit' | <xcolor-expression> |
'rgb(' <red-0-255> ', ' <green-0-255> ', ' <blue-0-255> ') ' |
'hsl(' <hue-0-360> ', ' <sat-0-1> ', ' <lum-0-1> ') '

```

Most of these properties are straight-forward. You should check a CSS documentation to get more information. A very good source is the Mozilla Developer Network.

Here are the supported column types:

- **n**: The most basic cell type, hbox, honoring all properties.
- **l**, **c**, and **r**: Same as **n** but with forced `text-align`.
- **M**<coltype>: Same as <coltype> but enforces `math-mode: math`. The net effect is that **Mc** will create a centered column whose contents are in non-display math mode.
- **T**<coltype>: Same as <coltype> but enforces `math-mode: text`.
- **p**{<width>}, **m**<width> and **b**<width>: parbox cell with the corresponding vertical alignment (`\vtop`, `\vcenter` or `\vbox`).

- `*{<count>}{<coltypes>}`: same as in `array` or `mdwtab`.
- `>{<prefix>}` and `<{<suffix>}`: same as in `array` or `mdwtab`.
- You can try to use constructs of `array` or `mdwtab`, but they might alter the function of `cellprops`. Most should be fine though.

The intended usage is to use n-type columns and set the properties with CSS, but L^AT_EX-like columns in the preamble are often less verbose.

Details for some properties:

- `math-mode`: `auto` means that the cell will be in math mode in environments `array`, `matrix`, `...`, and in text mode in environments like `tabular`, `...`
- `background-color` is only painted on the cell, and `transparent` actually means `inherit` except that if all values encountered are `inherit/transparent` no background is painted at all. That means that (currently) you cannot paint a row in some color and rely on transparency to have it bleed through a cell background.
- There are no columns in the CSS object model so you have to use `td:nth-child()` to select a column. Currently, cells spanning several columns actually increase the child count by the number of column they span, so that `nth-child` can still be used to select columns. This is not consistent with the CSS specification.
- Any `:nth-child(A n +B)` or `:nth-child(A n)` or `:nth-child(B)` is supported, with arbitrary A and B . The performance will slowly degrade the more different A are active (but if in a T_EX group, they will become inactive again when leaving the group). A *big difference* with the CSS specification is that currently n is not enforced non-negative. In particular `:nth-child(2 n +8)` *will* match for the second element. This also prevents tricks like `:nth-child(- n +3)`. I am investigating how to handle those without slowing down the general case.

1.3 Compatibility

This package has been tested compatible with `diagbox`, `spreadtab`, `colcell`. Compatibility with `longtable` has been specifically taken care of, provided `cellprops` is loaded afterwards. Table packages that only introduce new column types should be loaded after `mdwtab`, so either you load `mdwtab` manually and load your package in between `mdwtab` and `cellprops`, or you load your package after `cellprops` (provided it doesn't overwrite the machinery).

1.4 TODO

Add a test suite with compatibility tests. Improve the documentation, and test more L^AT_EX table constructs and preamble column types. Enforce $n \geq 0$ in `:nth-child` selectors to match the CSS specification.

2 cellprops implementation

```

1 <*package>
2 <@@=cellprops>
3 \ProvidesExplPackage

```

```

4   {\ExplFileName}{\ExplFileDate}{\ExplFileVersion}{\ExplFileDescription}
5
6   \RequirePackage{xparse}
7   \RequirePackage{xcolor}

```

2.1 Loading and fixing mdwtab

There is a bug in the command `\colpop` of `mdwtab`: instead of just popping one name in the stack of column sets currently used, it empties it completely because one `\expandafter` is missing. This is proof that not many package authors really use this API as recommended by Mark WOODING... We thus load `mdwtab` and fix `\colpop`.

```

8   \RequirePackage{mdwtab}
9   \cs_set_nopar:Npn \tab@pop #1 { \tl_set:Nx #1 { \tl_tail:N #1 } }

```

2.2 Parsing CSS properties

Properties are parsed once at setting time, by expandable parsers that leave definitions in the input stream. All these resulting definitions are saved in a token list that will be expanded when we need the values. The goal is to have multiple token lists for multiple contexts, yet not to do the full parsing dance once per cell.

We first define a generic setter which just uses `\l__cellprops_property_value_<name>_tl` to store the value of the property. We define getters, one that leaves the value in the stream, and one saving the value in a token list.

```

\l__cellprops_property_value_<name>_tl
  \__cellprops_generic_setter:nnn
\__cellprops_get_property:n
\__cellprops_get_property:nN
10 \cs_new:Nn \__cellprops_generic_setter:nnn {
11   \exp_not:N \tl_set:Nn
12   \exp_not:c { l__cellprops_property_value_#2_tl }
13   {#1 {#3}}
14 }
15
16 \cs_set_nopar:Nn \__cellprops_get_property:n {
17   \tl_use:c { l__cellprops_property_value_#1_tl }
18 }
19
20 \cs_new_protected_nopar:Nn \__cellprops_get_property:nN {
21   \tl_if_exist:cTF { l__cellprops_property_value_#1_tl } {
22     \tl_set_eq:Nc #2 { l__cellprops_property_value_#1_tl }
23   }{
24     \tl_clear:N #2
25   }
26 }

```

(End definition for `\l__cellprops_property_value_<name>_tl` and others.)

`__cellprops_property_type_<name>:nn` The control sequence `__cellprops_property_type_<name>:nn` holds the setter for the property *<name>*. It can be set by the following helper:

```

27 \cs_new_protected:Nn \__cellprops_define_properties:nn {
28   \clist_map_inline:nn {#2} {
29     \cs_set:cpn { __cellprops_property_type_##1:nn } {#1}
30   }
31 }

```

(End definition for `__cellprops_property_type_<name>:nn` and `__cellprops_define_properties:nn`.)

`__cellprops_use_setter:nn` Sometimes we need to use a setter right away rather than save its action somewhere. The following helper does that with an x-expansion.

```

32 \cs_new:Nn \__cellprops_delegate_setter:nn {
33   \use:c {__cellprops_property_type_#1:nn} {#1} {#2}
34 }
35 \cs_new_protected:Nn \__cellprops_use_setter:nn {
36   \use:x {
37     \__cellprops_delegate_setter:nn {#1} {#2}
38   }
39 }

```

(End definition for `__cellprops_use_setter:nn`.)

```

40 \cs_new_protected:Nn \__cellprops_parse_properties:Nn {
41   \tl_clear:N #1
42   \seq_set_split:Nnn \l_tmpa_seq {;} {#2}
43   \seq_map_inline:Nn \l_tmpa_seq {
44     \tl_if_empty:nF {##1} {
45       \exp_args:NNV \seq_set_split:Nnn \l_tmpb_seq \c_colon_str {##1}
46       \int_compare:nNnTF {\seq_count:N \l_tmpb_seq} = \c_two {
47         \seq_get_left:NN \l_tmpb_seq \l_tmpa_tl
48         \exp_args:NNV \str_set:Nn \l_tmpa_str \l_tmpa_tl
49         \seq_get_right:NN \l_tmpb_seq \l_tmpa_tl
50         \cs_if_exist:cTF { __cellprops_property_type_\l_tmpa_str :nn } {
51           \tl_put_right:Nx #1 {
52             \exp_args:NVV \__cellprops_delegate_setter:nn
53               \l_tmpa_str \l_tmpa_tl
54           }
55         }{
56           % TODO: ERROR-no property with that name
57         }
58       }{
59         % TODO: ERROR-too many : or none at all
60       }
61     }
62   }
63 }
64
65 \cs_new:Nn \__cellprops_fourval_setter:nnnnnn {
66   \__cellprops_fourval_setter_aux:w
67     {#1}{#2}{#3}{#4}#6~{\q_no_value}~{\q_no_value}~{\q_no_value}~\q_stop
68 }
69 \cs_new:Npn \__cellprops_fourval_setter_aux:w #1#2#3#4#5~#6~#7~#8~#9\q_stop {
70   \__cellprops_delegate_setter:nn {#1} {#5}
71   \quark_if_no_value:nTF {#6} {
72     \__cellprops_delegate_setter:nn {#2} {#5}
73     \__cellprops_delegate_setter:nn {#4} {#5}
74   }{
75     \__cellprops_delegate_setter:nn {#2} {#6}
76     \quark_if_no_value:nTF {#8} {
77       \__cellprops_delegate_setter:nn {#4} {#6}
78     }{
79       \__cellprops_delegate_setter:nn {#4} {#8}
80     }
81   }

```

```

82 \quark_if_no_value:nTF {#7} {
83   \__cellprops_delegate_setter:nn {#3} {#5}
84 }{
85   \__cellprops_delegate_setter:nn {#3} {#7}
86 }
87 }
88
89 \cs_new_protected:Nn \__cellprops_define_fourval_properties:nnnnnn {
90   \__cellprops_define_properties:nn {#1} { #3, #4, #5, #6 }
91   \__cellprops_define_properties:nn {
92     \__cellprops_fourval_setter:nnnnnn {#3}{#4}{#5}{#6}
93   }{
94     #2
95   }
96 }
97
98 \tl_const:Nn \c__cellprops_inherit_color_tl { \q_nil }
99
100 \cs_new_nopar:Nn \__cellprops_color_setter:nn {
101   \str_if_eq:nnTF {#2} {inherit} {
102     \__cellprops_generic_setter:nnn \exp_not:n {#1} {\c__cellprops_inherit_color_tl}
103   }{
104     \str_case:x:nnF { \str_range:nnn {#2} {1} {4} } {
105       {rgb()} {
106         \__cellprops_generic_setter:nnn \use:n {#1} {
107           \exp_not:n {\color[RGB]} {\str_range:nnn {#2} {5} {-2}}
108         }
109       }{
110         {hsl()} {
111           \__cellprops_generic_setter:nnn \use:n {#1} {
112             \exp_not:n {\color[Hsb]} {\str_range:nnn {#2} {5} {-2}}
113           }
114         }{
115           \__cellprops_generic_setter:nnn \exp_not:n {#1} {
116             \color{#2}
117           }
118         }
119       }
120     }
121     \cs_new_nopar:Nn \__cellprops_bgcolor_setter:nn {
122       \str_if_eq:nnTF {#2} {transparent} {
123         \__cellprops_color_setter:nn {#1} {inherit}
124       }{
125         \__cellprops_color_setter:nn {#1} {#2}
126       }
127     }
128     \cs_new_nopar:Nn \__cellprops_linewidth_setter:nn {
129       \str_case:nnF {#2} {
130         {thin} { \__cellprops_generic_setter:nnn \exp_not:n {#1} { \fboxrule } }
131         {medium} { \__cellprops_generic_setter:nnn \exp_not:n {#1} { 2\fboxrule } }
132         {thick} { \__cellprops_generic_setter:nnn \exp_not:n {#1} { 3\fboxrule } }
133       }{
134         \__cellprops_generic_setter:nnn \exp_not:n {#1} {#2}
135       }

```



```

136 }
137
138 \cs_new_nopar:Nn \__cellprops_border_setter:nn {
139   \__cellprops_border_setter_aux:nw
140   {#1}#2~{\q_no_value}~{\q_no_value}~\q_stop
141 }
142 \cs_new:Npn \__cellprops_border_setter_aux:nw #1#2~#3~#4~#5\q_stop {
143   \quark_if_no_value:nTF {#4} {
144     \__cellprops_border_setter_isstyle:nTF {#2} {
145       \__cellprops_delegate_setter:nn {#1-width} {thin}
146       \__cellprops_delegate_setter:nn {#1-style} {#2}
147       \quark_if_no_value:nTF {#3} {
148         \__cellprops_delegate_setter:nn {#1-color} {inherit}
149       }{
150         \__cellprops_delegate_setter:nn {#1-color} {#3}
151       }
152     }{
153       \quark_if_no_value:nTF {#3} {
154         %% TODO: Error, one no-style value, ambiguous
155       }{
156         \__cellprops_border_setter_isstyle:nTF {#3} {
157           \__cellprops_delegate_setter:nn {#1-width} {#2}
158           \__cellprops_delegate_setter:nn {#1-style} {#3}
159           \__cellprops_delegate_setter:nn {#1-color} {inherit}
160         }{
161           \__cellprops_delegate_setter:nn {#1-width} {#2}
162           \__cellprops_delegate_setter:nn {#1-style} {none}
163           \__cellprops_delegate_setter:nn {#1-color} {#3}
164         }
165       }
166     }
167   }{
168     \__cellprops_delegate_setter:nn {#1-width} {#2}
169     \__cellprops_delegate_setter:nn {#1-style} {#3}
170     \__cellprops_delegate_setter:nn {#1-color} {#4}
171   }
172 }
173 \cs_new:Npn \__cellprops_border_setter_isstyle:nTF #1 {
174   \str_case:nnTF {#1} {
175     {none}{} {hidden}{} {dotted}{} {dashed}{} {solid}{}
176     {double}{} {groove}{} {ridge}{} {inset}{} {outset}{}
177   }
178 }
179
180 \__cellprops_define_properties:nn {
181   \__cellprops_generic_setter:nnn \exp_not:n
182 }{
183   min-height,
184   min-depth,
185   min-width,
186 }
187
188 \__cellprops_define_fourval_properties:nnnnn
189 { \__cellprops_generic_setter:nnn \exp_not:n }

```

```

190     {padding}
191     {padding-top}{padding-right}{padding-bottom}{padding-left}
192
193 \__cellprops_define_properties:nn {
194     \__cellprops_generic_setter:nnn \tl_to_str:n
195 }{
196     text-align,
197     math-mode,
198 }
199
200 \__cellprops_define_properties:nn {
201     \__cellprops_color_setter:nn
202 }{
203     color,
204 }
205
206 \__cellprops_define_properties:nn {
207     \__cellprops_bgcolor_setter:nn
208 }{
209     background-color,
210 }
211
212 \__cellprops_define_fourval_properties:nnnnnn
213     { \__cellprops_linewidth_setter:nn }
214     {border-width}
215     {border-top-width}{border-right-width}
216     {border-bottom-width}{border-left-width}
217
218 \__cellprops_define_fourval_properties:nnnnnn
219     { \__cellprops_generic_setter:nnn \tl_to_str:n }
220     {border-style}
221     {border-top-style}{border-right-style}
222     {border-bottom-style}{border-left-style}
223
224 \__cellprops_define_fourval_properties:nnnnnn
225     { \__cellprops_color_setter:nn }
226     {border-color}
227     {border-top-color}{border-right-color}
228     {border-bottom-color}{border-left-color}
229
230 \__cellprops_define_properties:nn {
231     \__cellprops_border_setter:nn
232 }{
233     border, border-top, border-right, border-bottom, border-left
234 }
235
236 \NewDocumentCommand \cellprops { m } {
237     \__cellprops_parse_css:n {#1}
238 }
239
240 \cs_new_protected:Nn \__cellprops_parse_css:n {
241     \__cellprops_parse_css:w #1 \q_mark {\q_nil} \q_stop
242 }
243

```

```

244 \tl_new:N \l__cellprops_parse_tmp_tl
245 \NewDocumentCommand \__cellprops_parse_css:w { lmu{\q_stop} } {
246   \quark_if_nil:nF {#2} {
247     \__cellprops_parse_properties:Nn \l__cellprops_parse_tmp_tl {#2}
248     \clist_map_inline:nn {#1} {
249       \__cellprops_parse_css_addprops:nV {##1} \l__cellprops_parse_tmp_tl
250     }
251     \__cellprops_parse_css:w #3 \q_stop
252   }
253 }
254
255
256 \seq_new:N \l__cellprops_parse_selector_seq
257 \tl_new:N \l__cellprops_parse_desc_tl
258
259 \str_const:Nn \c__cellprops_parse_nthchild_str { :nth-child( }
260 \prop_new:N \c__cellprops_parse_replace_prop
261 \prop_put:Nnn \c__cellprops_parse_replace_prop { :first-child } { :nth-child(1) }
262
263 \cs_new_protected:Nn \__cellprops_parse_selector:Nn {
264   \str_set:Nx \l_tmpa_str {#2}

```

Replace some aliases with their meaning:

```

265   \prop_map_inline:Nn \c__cellprops_parse_replace_prop {
266     \use:x {
267       \exp_not:n { \tl_replace_all:Nnn \l_tmpa_str }
268       { \tl_to_str:n { ##1 } } { \tl_to_str:n { ##2 } }
269     }
270   }

```

Replace all spaces by `\q_stop` to defeat the space removal feature of `\seq_set_split:Nnn`.

```

271   \tl_replace_all:Nnn \l_tmpa_str {~} {\q_stop}
272   \exp_args:NNVV
273   \seq_set_split:Nnn \l_tmpa_seq \c__cellprops_parse_nthchild_str \l_tmpa_str
274   \seq_pop_left:NN \l_tmpa_seq \l_tmpa_tl

```

And replace them back.

```

275   \tl_replace_all:Nnn \l_tmpa_tl {\q_stop} {~}
276   \seq_clear:N \l__cellprops_parse_selector_seq
277   \seq_put_right:NV \l__cellprops_parse_selector_seq \l_tmpa_tl
278   \seq_map_inline:Nn \l_tmpa_seq {
279     \tl_set:Nn \l_tmpa_tl { ##1 }

```

And replace them back again.

```

280     \tl_replace_all:Nnn \l_tmpa_tl {\q_stop} {~}

```

Now replace the first closing parenthesis by `\q_stop\prg_do_nothing:` to use `\q_stop` as a delimiter for `\seq_set_split:Nnn` and thus split at most once. Note that here the space trimming feature is desired for the left part, but not just at the right of the parenthesis, so `\prg_do_nothing:` will act as a guard and will be removed afterwards.

```

281     \tl_replace_once:Nnn \l_tmpa_tl { ) } { \q_stop\prg_do_nothing: }
282     \seq_set_split:NnV \l_tmpa_seq { \q_stop } \l_tmpa_tl
283     \seq_pop_right:NN \l_tmpa_seq \l__cellprops_parse_desc_tl

```

```

284     \tl_replace_once:Nnn \l__cellprops_parse_desc_tl { \prg_do_nothing: } {}
285     \seq_get_left:NNT \l_tmpa_seq \l_tmpa_tl {
286         \exp_args:NNV \__cellprops_parse_nth:Nn \l_tmpa_tl \l_tmpa_tl
287         \tl_put_left:Nn \l__cellprops_parse_desc_tl { ) }
288         \tl_put_left:NV \l__cellprops_parse_desc_tl \l_tmpa_tl
289     }
290     \seq_put_right:NV \l__cellprops_parse_selector_seq \l__cellprops_parse_desc_tl
291 }
292 \tl_set:Nx #1 {
293     \exp_args:NNV \seq_use:Nn
294         \l__cellprops_parse_selector_seq \c__cellprops_parse_nthchild_str
295 }
296 }
297
298 \str_const:Nn \c__cellprops_parse_n_str {n}
299 \seq_new:N \l__cellprops_used_nth_factors_seq
300 \cs_new_protected:Nn \__cellprops_parse_nth:Nn {
301     \str_case:nnF {#2} {
302         {even} { \str_set:Nn \l_tmpa_str {2n} }
303         {odd} { \str_set:Nn \l_tmpa_str {2n+1} }
304     }{
305         \str_set:Nn \l_tmpa_str {#2}
306     }
307     \exp_args:NNVV
308     \seq_set_split:Nnn \l_tmpa_seq \c__cellprops_parse_n_str \l_tmpa_str
309     \seq_pop_right:NN \l_tmpa_seq \l_tmpa_tl
310     \int_set:Nn \l_tmpb_int { 0\l_tmpa_tl }
311     \seq_get_left:NNTF \l_tmpa_seq \l_tmpa_tl {
312         \int_set:Nn \l_tmpa_int { 0\l_tmpa_tl }
313     }{
314         \int_zero:N \l_tmpa_int
315     }
316     \int_compare:nNnTF \l_tmpa_int = { 0 } {
317         \tl_set:Nx #1 { \int_use:N \l_tmpb_int }
318     }{
319         \int_set:Nn \l_tmpb_int { \int_mod:nn {\l_tmpb_int} {\l_tmpa_int} }
320         \tl_set:Nx #1 {
321             \int_use:N \l_tmpa_int \exp_not:V \c__cellprops_parse_n_str
322             + \int_use:N \l_tmpb_int }
323         \seq_put_right:Nx
324             \l__cellprops_used_nth_factors_seq { \int_use:N \l_tmpa_int }
325     }
326 }
327
328 \cs_new_protected:Npn \__cellprops_parse_css_addprops:nV #1 #2 {
329     \__cellprops_parse_selector:Nn \l_tmpa_tl {#1}
330     \tl_set:Nx \l_tmpa_tl { l__cellprops_property_group_\l_tmpa_tl _tl }
331     \tl_if_exist:cF { \l_tmpa_tl } { \tl_clear:c { \l_tmpa_tl } }
332     \tl_put_right:cV { \l_tmpa_tl } #2
333 }
334
335 \cs_set_protected:Nn \__cellprops_recall_properties:n {
336     \tl_if_exist:cT { l__cellprops_property_group_#1_tl } {
337         \tl_use:c { l__cellprops_property_group_#1_tl }

```

```

338 }
339 \clist_map_inline:nn { \@currentvir } {
340   \tl_if_exist:cT { l__cellprops_property_group_##1~#1_tl } {
341     \tl_use:c { l__cellprops_property_group_##1~#1_tl }
342   }
343 }
344 }
345
346 \dim_new:N \l__cellprops_colsep_dim
347 \dim_new:N \l__cellprops_strut_ht_dim
348 \dim_new:N \l__cellprops_strut_dp_dim
349
350 \ExplSyntaxOff
351 \cellprops{
352   td {
353     padding: Opt \csname l__cellprops_colsep_dim\endcsname;
354     min-height: \csname l__cellprops_strut_ht_dim\endcsname;
355     min-depth: \csname l__cellprops_strut_dp_dim\endcsname;
356     min-width: Opt;
357     text-align: left;
358     math-mode: auto;
359     color: inherit;
360     background-color: transparent;
361     border: thin none inherit;
362   }
363   tr {
364     color: inherit;
365     background-color: transparent;
366   }
367   table {
368     padding: Opt; % No change at load time
369     color: inherit;
370     background-color: transparent;
371   }
372 }
373 \ExplSyntaxOn
374
375 \int_new:N \g__cellprops_row_int
376 \int_new:N \g__cellprops_col_int
377 \bool_new:N \g__cellprops_inrow_bool
378 \bool_gset_false:N \g__cellprops_inrow_bool
379
380 \box_new:N \l__cellprops_cell_box
381 \skip_new:N \l__cellprops_left_skip
382 \skip_new:N \l__cellprops_right_skip
383 \dim_new:N \g__cellprops_ht_dim
384 \dim_new:N \g__cellprops_dp_dim
385 \tl_new:N \g__cellprops_borders_tl
386
387 \tl_new:N \l__cellprops_restore_tl
388
389 \dim_new:N \l__cellprops_tablepadding_top_dim
390 \dim_new:N \l__cellprops_tablepadding_bottom_dim
391 \tl_new:N \l__cellprops_color_tl

```

```

392 \tl_new:N \l__cellprops_bgcolor_tl
393
394 % To count rows and columns
395 \cs_new_protected:Nn \__cellprops_array_init: {
396   \tl_set:Nx \l__cellprops_restore_tl {
397     \bool_if:NTF \g__cellprops_inrow_bool {
398       \exp_not:n {\bool_gset_true:N \g__cellprops_inrow_bool}
399     }{
400       \exp_not:n {\bool_gset_false:N \g__cellprops_inrow_bool}
401     }
402     \exp_not:n { \int_gset:Nn \g__cellprops_row_int }
403     { \int_use:N \g__cellprops_row_int }
404     \exp_not:n { \int_gset:Nn \g__cellprops_col_int }
405     { \int_use:N \g__cellprops_col_int }
406     \exp_not:n { \dim_gset:Nn \g__cellprops_ht_dim }
407     { \dim_use:N \g__cellprops_ht_dim }
408     \exp_not:n { \dim_gset:Nn \g__cellprops_dp_dim }
409     { \dim_use:N \g__cellprops_dp_dim }
410     \exp_not:n { \tl_gset:Nn \g__cellprops_borders_tl }
411     { \exp_not:V \g__cellprops_borders_tl }
412   }
413   \int_gzero:N \g__cellprops_row_int
414   \bool_gset_false:N \g__cellprops_inrow_bool
415   \tl_gclear:N \g__cellprops_borders_tl
416   \cs_set_eq:NN \__cellprops_orig_tab@readpreamble:n \tab@readpreamble
417   \cs_set_eq:NN \tab@readpreamble \__cellprops_readpreamble:n

```

Zero \col@sep but remember its value for the default padding.

```

418   \dim_set_eq:NN \l__cellprops_colsep_dim \col@sep
419   \dim_zero:N \col@sep

```

Also ignore *extrasep dimensions that are not part of cellprop interface and should be replaced by CSS equivalents.

```

420   \dim_zero:N \tab@extrasep
421   \group_begin:
422     \__cellprops_recall_properties:n {table}
423     \dim_gset:Nn \g_tmpa_dim { \__cellprops_get_property:n {padding-top} }
424     \dim_gset:Nn \g_tmpb_dim { \__cellprops_get_property:n {padding-bottom} }
425     \__cellprops_update_colors:
426     \tl_gset_eq:NN \g_tmpa_tl \l__cellprops_color_tl
427     \tl_gset_eq:NN \g_tmpb_tl \l__cellprops_bgcolor_tl
428   \group_end:
429   \dim_set_eq:NN \l__cellprops_tablepadding_top_dim \g_tmpa_dim
430   \dim_set_eq:NN \l__cellprops_tablepadding_bottom_dim \g_tmpb_dim
431   \tl_set_eq:NN \l__cellprops_color_tl \g_tmpa_tl
432   \tl_set_eq:NN \l__cellprops_bgcolor_tl \g_tmpb_tl
433   \__cellprops_recall_properties:n {tr}
434   \dim_set:Nn \l__cellprops_strut_ht_dim { \box_ht:N \@arstrutbox }
435   \dim_set:Nn \l__cellprops_strut_dp_dim { \box_dp:N \@arstrutbox }
436   \box_clear:N \@arstrutbox
437 }
438
439 \cs_set_nopar:Nn \__cellprops_array_startcontent: {
440   \hlx{s[\l__cellprops_tablepadding_top_dim]}

```

```

441 }
442
443 \cs_set_protected_nopar:Nn \__cellprops_readpreamble:n {
444   \cs_set_eq:NN \tab@readpreamble \__cellprops_orig_tab@readpreamble:n
445   \tl_put_left:Nn \tab@multicol {\__cellprops_startrow:}
446   \tl_put_left:Nn \tab@tabtext {\int_gincr:N \g__cellprops_col_int}
447   \tab@readpreamble{#1}
448   \exp_args:Nx \tab@preamble
449   { \exp_not:N\__cellprops_startrow: \the\tab@preamble \exp_not:N\__cellprops_endrow: }
450 }
451
452
453 \cs_set_eq:NN \__cellprops_orig_array:w \@array
454 \cs_set_protected_nopar:Npn \@array[#1]#2 {
455   \__cellprops_array_init:
456   \__cellprops_orig_array:w [#1]{#2}
457   \__cellprops_array_startcontent:
458 }
459
460 \cs_set_eq:NN \__cellprops_orig_LTmkpream:n \@mkpream
461 \cs_set_protected_nopar:Npn \@mkpream#1 {
462   \group_end:
463   \__cellprops_array_init:
464   \group_begin:
465   \__cellprops_orig_LTmkpream:n {#1}
466 }
467
468 \cs_set_eq:NN \__cellprops_orig_LTarray:w \LT@array
469 \cs_set_protected_nopar:Npn \LT@array [#1]#2 {
470   \__cellprops_orig_LTarray:w [#1]{#2}
471   \__cellprops_array_startcontent:
472 }

```

The color inheritance is handled with `\l__cellprops_inherit_color_tl`, `\l__cellprops_color_tl` and `\l__cellprops_bgcolor_tl`. The role of `__cellprops_update_color:Nn` is to set the inherit fallback to the already existing value of #1 then set #1 to the CSS value, which can be the inherit variable.

```

473 \cs_new_protected_nopar:Nn \__cellprops_update_color:Nn {
474   \__cellprops_get_property:nN {#2} \l_tmpa_tl
475   \exp_args:NV \tl_if_eq:NNF \l_tmpa_tl \c__cellprops_inherit_color_tl {
476     \tl_set_eq:NN #1 \l_tmpa_tl
477   }
478 }
479
480 \cs_new_protected_nopar:Nn \__cellprops_update_colors: {
481   \__cellprops_update_color:Nn \l__cellprops_color_tl {color}
482   \__cellprops_update_color:Nn \l__cellprops_bgcolor_tl {background-color}
483 }
484
485 \cs_new_nopar:Nn \__cellprops_end_array:n {
486   \tl_if_empty:NF \g__cellprops_borders_tl { \ }
487   \crrc
488   \hlx{s[\l__cellprops_tablepadding_bottom_dim]}
489   #1

```

```

490     \tl_use:N \l__cellprops_restore_tl
491 }
492
493 \cs_set_eq:NN \__cellprops_orig_endarray: \endarray
494 \cs_set_nopar:Npn \endarray {
495     \__cellprops_end_array:n { \__cellprops_orig_endarray: }
496 }
497 \cs_set_eq:NN \endtabular \endarray
498 \cs_set_eq:cN {endtabular*} \endarray
499
500 \cs_set_eq:NN \__cellprops_orig_endLT: \endlongtable
501 \cs_set_nopar:Npn \endlongtable {
502     \__cellprops_end_array:n { \__cellprops_orig_endLT: }
503 }
504
505 \cs_new_protected_nopar:Nn \__cellprops_startrow: {
506     \bool_if:NF \g__cellprops_inrow_bool {
507         \bool_gset_true:N \g__cellprops_inrow_bool
508         \int_gincr:N \g__cellprops_row_int
509         \int_gset_eq:NN \g__cellprops_col_int \c_one
510         \dim_gzero:N \g__cellprops_ht_dim
511         \dim_gzero:N \g__cellprops_dp_dim
512     }
513 }
514
515 \cs_new_protected_nopar:Nn \__cellprops_endrow: {
516     \bool_if:NT \g__cellprops_inrow_bool {
517         \bool_gset_false:N \g__cellprops_inrow_bool
518     }
519 }
520
521 \cs_new_protected_nopar:Nn \__cellprops_cr:n {
522     \__cellprops_endrow:
523     \tl_if_empty:NF \g__cellprops_borders_tl {
524         \cr
525         \noalign{\nobreak}
526         \tl_use:N \g__cellprops_borders_tl
527         \tl_gclear:N \g__cellprops_borders_tl
528     }
529     \cr
530     \__cellprops_fix_valign_end:n {#1}
531     \use_none:n
532 }
533
534 \cs_set_protected_nopar:Npn \tab@tabcr #1#2 { \__cellprops_cr:n {#2} }
535 \cs_set_protected_nopar:Npn \@xargarraycr #1 { \__cellprops_cr:n {#1} }
536 \cs_set_protected_nopar:Npn \@yargarraycr #1 { \__cellprops_cr:n {#1} }
537 \tl_if_exist:NT \LT@echunk {
538     \tl_put_left:Nn \LT@echunk {
539         \tl_if_empty:NF \g__cellprops_borders_tl { \\ }
540     }
541 }
542
543 \cs_set_eq:NN \__cellprops_orig_multicolumn:w \multicolumn

```



```

544 \cs_set:Npn \multicolumn#1#2#3 {
545   \__cellprops_orig_multicolumn:w {#1}{#2}{
546     #3
547     \int_gadd:Nn \g__cellprops_col_int {#1}
548   }
549 }
550
551 \cs_new_nopar:Nn \__cellprops_fix_valign_end:n {
552   \noalign{
553     \dim_set:Nn \l_tmpa_dim {#1}
554     \skip_vertical:n {\l_tmpa_dim}
555     \exp_args:NV \tl_if_eq:nnTF \tab@hlstate {b} {
556       \dim_gadd:Nn \tab@endheight { \g__cellprops_dp_dim + \l_tmpa_dim }
557     }{
558       \int_compare:nNnT \g__cellprops_row_int = \c_one {
559         \dim_gadd:Nn \tab@endheight { \g__cellprops_ht_dim }
560       }
561     }
562   }
563 }

```

Reset `\firstline` and `\lastline` to `\hline` because the version from array which might be loaded already will mess up the spacing and is unneeded anyway.

```

564 \cs_set_eq:NN \firstline \hline
565 \cs_set_eq:NN \lastline \hline
566
567 \colpush{tabular}
568
569 \coldef n{\tabcoltype{
570   \__cellprops_begincell:n{
571 }{
572   \__cellprops_endcell:
573 }}
574 \coldef l{\tabcoltype{
575   \__cellprops_begincell:n
576   {\__cellprops_use_setter:nm {text-align} {left}}
577 }{
578   \__cellprops_endcell:
579 }}
580 \coldef c{\tabcoltype{
581   \__cellprops_begincell:n
582   {\__cellprops_use_setter:nm {text-align} {center}}
583 }{
584   \__cellprops_endcell:
585 }}
586 \coldef r{\tabcoltype{
587   \__cellprops_begincell:n
588   {\__cellprops_use_setter:nm {text-align} {right}}
589 }{
590   \__cellprops_endcell:
591 }}
592 \coldef M#1{\__cellprops_MTcol:nm {math}{#1}}
593 \coldef T#1{\__cellprops_MTcol:nm {text}{#1}}
594 \cs_new_protected_nopar:Nn \__cellprops_MTcol:nm {

```

```

595 % TODO: error if align not l, c, or r
596 \exp_args:Nx \tabcoltype {
597   \exp_not:N \__cellprops_begincell:n {
598     \exp_not:n {\__cellprops_use_setter:nn {math-mode} {#1} }
599     \exp_not:n {\__cellprops_use_setter:nn {text-align} {
600       \str_case:nn {#2} {
601         {l} {left}
602         {c} {center}
603         {r} {right}
604       }
605     }
606   }
607 }{
608   \__cellprops_endcell:
609 }
610 }
611
612 \coldef p#1{\tabcoltype{
613   \__cellprops_begin_par_cell:nn \vtop {#1}
614 }{
615   \__cellprops_end_par_cell:n { }
616 }}
617 \coldef m#1{\tabcoltype{
618   \__cellprops_begin_par_cell:nn {\c_math_toggle_token\vcenter} {#1}
619 }{
620   \__cellprops_end_par_cell:n{\c_math_toggle_token}
621 }}
622 \coldef b#1{\tabcoltype{
623   \__cellprops_begin_par_cell:nn \vbox {#1}
624 }{
625   \__cellprops_end_par_cell:n { }
626 }}
627
628
629 \colpop
630   Handle various :nth-child() forms.
631 \cs_new_protected_nopar:Nn \__cellprops_seq_nthchild:Nn {
632   \seq_clear:N #1
633   \seq_map_inline:Nn \l__cellprops_used_nth_factors_seq {
634     \seq_put_right:Nx #1 {
635       ##1 n + \int_eval:n{\int_mod:nn{#2}{##1}}
636     }
637   }
638   \seq_put_right:Nx #1 { \int_eval:n{#2} }
639 }
640 \cs_new_protected_nopar:Nn \__cellprops_begincell:n {
641   \__cellprops_begin_raw_cell:n {
642     #1
643     \hbox_set:Nw \l__cellprops_cell_box
644     \str_case_x:nnF {\__cellprops_get_property:n {math-mode}} {
645       { text } { \tab@btext }
646       { math } { \tab@bmaths }

```

```

647     }{% any other treated as |auto|
648     \tab@bgroup
649   }
650 }
651 }
652
653 \cs_new_protected_nopar:Nn \__cellprops_endcell: {
654   \str_case_x:nnF {\__cellprops_get_property:n {math-mode}} {
655     { text } { \tab@etext }
656     { math } { \tab@emaths }
657   }{% any other treated as |auto|
658   \tab@egroup
659 }
660 \hbox_set_end:
661 \__cellprops_end_raw_cell:
662 }
663
664 \cs_new_protected_nopar:Nn \__cellprops_begin_par_cell:nn {
665   \savenotes
666   \__cellprops_begin_raw_cell:n{
667     \hbox_set:Nw \l__cellprops_cell_box
668     #1
669     \bgroup
670     \hsize#2\relax
671     \@arrayparboxrestore
672     \global\@minipagetrue
673     \everypar{
674       \global\@minipagefalse
675       \everypar{}
676     }
677     \__cellprops_seq_nthchild:Nn \l_tmpa_seq { \g__cellprops_row_int }
678     \__cellprops_seq_nthchild:Nn \l_tmpb_seq { \g__cellprops_col_int }
679     \__cellprops_recall_properties:n {td~p}
680     \seq_map_inline:Nn \l_tmpa_seq {
681       \__cellprops_recall_properties:n {tr:nth-child(##1)~p}
682     }
683     \seq_map_inline:Nn \l_tmpb_seq {
684       \__cellprops_recall_properties:n {td:nth-child(##1)~p}
685     }
686     \__cellprops_recall_properties:n {tr~td~p}
687     \seq_map_inline:Nn \l_tmpa_seq {
688       \__cellprops_recall_properties:n {tr:nth-child(##1)~td~p}
689     }
690     \seq_map_inline:Nn \l_tmpa_seq {
691       \seq_map_inline:Nn \l_tmpb_seq {
692         \__cellprops_recall_properties:n {tr:nth-child(##1)~
693           td:nth-child(###1)~p}
694       }
695     }
696   }
697 }
698 \cs_new_protected_nopar:Nn \__cellprops_end_par_cell:n {
699   \ifhmode\@maybe@unskip\par\fi
700   \unskip

```

```

701 \egroup
702 #1
703 \hbox_set_end:
704 \__cellprops_end_raw_cell:
705 \spewnotes\hfil
706 }
707
708 \cs_new_protected_nopar:Nn \__cellprops_begin_raw_cell:n {
709   \group_begin:
710   \__cellprops_seq_nthchild:Nn \l_tmpa_seq { \g__cellprops_row_int }
711   \__cellprops_seq_nthchild:Nn \l_tmpb_seq { \g__cellprops_col_int }
712   \seq_map_inline:Nn \l_tmpa_seq {
713     \__cellprops_recall_properties:n {tr:nth-child(##1)}
714   }
715   \__cellprops_update_colors:
716   \__cellprops_recall_properties:n {td}
717   \__cellprops_recall_properties:n {tr~td}
718   \seq_map_inline:Nn \l_tmpb_seq {
719     \__cellprops_recall_properties:n {td:nth-child(##1)}
720   }
721   \seq_map_inline:Nn \l_tmpa_seq {
722     \__cellprops_recall_properties:n {tr:nth-child(##1)~td}
723   }
724   \seq_map_inline:Nn \l_tmpa_seq {
725     \seq_map_inline:Nn \l_tmpb_seq {
726       \__cellprops_recall_properties:n {tr:nth-child(##1)~
727         td:nth-child(###1)}
728     }
729   }
730   \__cellprops_update_colors:
731   % Additional init code
732   #1
733   % Install the cell color
734   \__cellprops_update_colors:
735   \tl_use:N \l__cellprops_color_tl
736 }
737
738 \cs_new_protected_nopar:Nn \__cellprops_make_solid_hborder:nnn {
739   \group_begin:
740     \hbox_set_to_wd:Nnn \l_tmpa_box {1pt} {
741       \hss
742       \hbox:n {
743         #3 % install color
744         \vrule height~\dim_eval:n{#1+#2}
745           ~depth~\dim_eval:n{#2}
746           ~width~3pt
747       }
748       \hss
749     }
750   \box_set_ht:Nn \l_tmpa_box { \c_zero_dim }
751   \box_set_dp:Nn \l_tmpa_box { \c_zero_dim }
752   \kern 1pt
753   \box_use:N \l_tmpa_box
754   \xleaders

```

```

755         \box_use:N \l_tmpa_box
756         \skip_horizontal:n {-4pt~plus~1fil}
757     \box_use:N \l_tmpa_box
758     \kern 1pt
759     \skip_horizontal:n {0pt~plus~-1fil}
760 \group_end:
761 }
762 \cs_new_protected_nopar:Nn \__cellprops_make_solid_vborder:nnn {
763     \group_begin:
764     \hbox_set_to_wd:Nnn \l_tmpa_box {0pt} {
765         \hbox:n {
766             #3 % install color
767             \vrule height~\dim_eval:n{#2}~width~\dim_eval:n{#1}
768         }
769         \hss
770     }
771     \box_set_ht:Nn \l_tmpa_box { \c_zero_dim }
772     \box_set_dp:Nn \l_tmpa_box { \c_zero_dim }
773     \box_use:N \l_tmpa_box
774 \group_end:
775 }
776 \clist_map_inline:nn {
777     dotted, dashed, solid, double,
778     groove, ridge, inset, outset
779 }{
780     \cs_set_eq:cN {__cellprops_make_#1_hborder:nnn} \__cellprops_make_solid_hborder:nnn
781     \cs_set_eq:cN {__cellprops_make_#1_vborder:nnn} \__cellprops_make_solid_vborder:nnn
782 }
783
784 \dim_new:N \l__cellprops_border_width_dim
785 \str_new:N \l__cellprops_border_style_str
786 \tl_new:N \l__cellprops_border_color_tl
787 \cs_new_protected_nopar:Nn \__cellprops_get_border_info:n {
788     \dim_set:Nn \l__cellprops_border_width_dim {\__cellprops_get_property:n {border-#1-width}}
789     \__cellprops_get_property:n{border-#1-style} \l_tmpa_tl
790     \exp_args:NNV \str_set:Nn \l__cellprops_border_style_str \l_tmpa_tl
791     \tl_clear:N \l__cellprops_border_color_tl
792     \cs_if_exist:cTF {__cellprops_make_#1__cellprops_border_style_str_hborder:nnn} {
793         \__cellprops_update_color:Nn \l__cellprops_border_color_tl {border-#1-color}
794     }{
795         \dim_zero:N \l__cellprops_border_width_dim
796     }
797 }
798
799 \cs_new_protected_nopar:Npn \__cellprops_make_hborder:nnnn #1 {
800     \use:c { __cellprops_make_#1_hborder:nnn }
801 }
802 \cs_new_protected_nopar:Npn \__cellprops_make_vborder:nnnn #1 {
803     \use:c { __cellprops_make_#1_vborder:nnn }
804 }
805
806 \cs_new_protected_nopar:Nn \__cellprops_end_raw_cell: {
807     % Here \l__cellprops_cell_box must contain the contents of the cell
808     %

```

```

809 % Prepare the borders token list
810 \int_compare:nNnT \g__cellprops_col_int = 1 {
811   \tl_gclear:N \g__cellprops_borders_tl
812 }
813 \tl_gput_right:Nx \g__cellprops_borders_tl {
814   \tl_if_empty:NF \g__cellprops_borders_tl { \exp_not:n {&} }
815   \exp_not:n { \omit \kern \c_zero_dim }
816 }
817 % Handle padding-top, min-height and border-top
818 \__cellprops_get_border_info:n {top}
819 \box_set_ht:Nn \l__cellprops_cell_box {
820   \dim_max:nn
821     {\box_ht:N \l__cellprops_cell_box}
822     {\__cellprops_get_property:n {min-height}}
823   + (\__cellprops_get_property:n {padding-top})
824   + \l__cellprops_border_width_dim
825 }
826 \dim_compare:nNnT \l__cellprops_border_width_dim > \c_zero_dim {
827   \tl_gput_right:Nx \g__cellprops_borders_tl {
828     \exp_not:N \__cellprops_make_hborder:nnnn
829     { \exp_not:V \l__cellprops_border_style_str }
830     { \dim_use:N \l__cellprops_border_width_dim }
831     {
832       \exp_not:n { \g__cellprops_dp_dim + \g__cellprops_ht_dim - }
833       \dim_use:N \l__cellprops_border_width_dim
834     }
835     { \exp_not:V \l__cellprops_border_color_tl }
836   }
837 }
838 % Handle padding-bottom, min-depth and border-bottom
839 \__cellprops_get_border_info:n {bottom}
840 \box_set_dp:Nn \l__cellprops_cell_box {
841   \dim_max:nn
842     {\box_dp:N \l__cellprops_cell_box}
843     {\__cellprops_get_property:n {min-depth}}
844   + (\__cellprops_get_property:n {padding-bottom})
845   + \l__cellprops_border_width_dim
846 }
847 \dim_compare:nNnT \l__cellprops_border_width_dim > \c_zero_dim {
848   \tl_gput_right:Nx \g__cellprops_borders_tl {
849     \exp_not:N \__cellprops_make_hborder:nnnn
850     { \exp_not:V \l__cellprops_border_style_str }
851     { \dim_use:N \l__cellprops_border_width_dim }
852     { \exp_not:n { Opt } }
853     { \exp_not:V \l__cellprops_border_color_tl }
854   }
855 }
856 % To fix vertical alignment later
857 \dim_gset:Nn \g__cellprops_ht_dim {
858   \dim_max:nn
859     {\g__cellprops_ht_dim}
860     {\box_ht:N \l__cellprops_cell_box}
861 }
862 \dim_gset:Nn \g__cellprops_dp_dim {

```

```

863     \dim_max:nn
864     {\g__cellprops_dp_dim}
865     {\box_dp:N \l__cellprops_cell_box}
866 }
867 % Handle hpadding and halign
868 \skip_set:Nn \l_tmpa_skip {
869     \dim_max:nn
870     {Opt}
871     { (\__cellprops_get_property:n {min-width})
872       - \box_wd:N \l__cellprops_cell_box }
873 }
874 \skip_add:Nn \l_tmpa_skip {
875     Opt plus 1fil
876 }
877 % padding-left and border-left
878 \__cellprops_get_border_info:n {left}
879 \skip_set:Nn \l__cellprops_left_skip
880     {\__cellprops_get_property:n {padding-left} + \l__cellprops_border_width_dim}
881 \dim_compare:nNnT \l__cellprops_border_width_dim > \c_zero_dim {
882     \tl_gput_right:Nx \g__cellprops_borders_tl {
883         \exp_not:N \__cellprops_make_vborder:nnnn
884         { \exp_not:V \l__cellprops_border_style_str }
885         { \dim_use:N \l__cellprops_border_width_dim }
886         { \exp_not:n { \g__cellprops_dp_dim + \g__cellprops_ht_dim } }
887         { \exp_not:V \l__cellprops_border_color_tl }
888     }
889 }
890 \tl_gput_right:Nx \g__cellprops_borders_tl {
891     \exp_not:n {
892         \skip_horizontal:n {Opt~plus~1fil}
893         \kern \c_zero_dim
894     }
895 }
896 \__cellprops_get_border_info:n {right}
897 \skip_set:Nn \l__cellprops_right_skip
898     {\__cellprops_get_property:n {padding-right} + \l__cellprops_border_width_dim}
899 \dim_compare:nNnT \l__cellprops_border_width_dim > \c_zero_dim {
900     \tl_gput_right:Nx \g__cellprops_borders_tl {
901         \exp_not:N \skip_horizontal:n
902         { - \dim_use:N \l__cellprops_border_width_dim }
903         \exp_not:N \__cellprops_make_vborder:nnnn
904         { \exp_not:V \l__cellprops_border_style_str }
905         { \dim_use:N \l__cellprops_border_width_dim }
906         { \exp_not:n { \g__cellprops_dp_dim + \g__cellprops_ht_dim } }
907         { \exp_not:V \l__cellprops_border_color_tl }
908         \exp_not:N \skip_horizontal:n
909         { \dim_use:N \l__cellprops_border_width_dim }
910         \exp_not:n { \kern \c_zero_dim }
911     }
912 }
913 \str_case:x:nnF {\__cellprops_get_property:n {text-align}} {
914     { right } {
915         \skip_add:Nn \l__cellprops_left_skip { \l_tmpa_skip }
916     }

```

```

917     { center } {
918         \skip_add:Nn \l__cellprops_left_skip { \l_tmpa_skip / 2 }
919         \skip_add:Nn \l__cellprops_right_skip { \l_tmpa_skip / 2 }
920     }
921 }{% any other treated as |left|
922     \skip_add:Nn \l__cellprops_right_skip { \l_tmpa_skip }
923 }
924 \kern\c_zero_dim
925 \tl_if_empty:NF \l__cellprops_bgcolor_tl {
926     \group_begin:
927     % Paint a background with leaders
928     \tl_use:N \l__cellprops_bgcolor_tl % install the color
929     \skip_set:Nn \l_tmpa_skip {
930         \l__cellprops_left_skip
931         + \box_wd:N \l__cellprops_cell_box
932         + \l__cellprops_right_skip
933     }
934     \leaders
935         \vrule
936         \skip_horizontal:N \l_tmpa_skip
937         \skip_horizontal:n {-\l_tmpa_skip}
938     \group_end:
939 }
940 \skip_horizontal:N \l__cellprops_left_skip
941 \box_use_clear:N \l__cellprops_cell_box
942 \skip_horizontal:N \l__cellprops_right_skip
943 \kern\c_zero_dim
944 \group_end:
945 }
946 \end{package}

```