

The `cellprops` package

CSS-like cell and table properties*

Julien “`_FrnchFrgg_`” RIVAUD†

Released 2019/09/29

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 mimick 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: 0pt; min-depth: 0pt; } }
```

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>This is maths</i>	B_3	
C1	C_2	X	Y
D1	D_2	DX	v
	F	$\int_a^b f(t)dt$	v

you can use:

```
\[
  \cellprops{
    td {
      padding: 1ex;
      min-height: 0pt;
```

*This file describes v1.6, last revised 2019/09/29.

†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 \\
    B1 & This is maths & B_3 & \\
    C1 & C_2 & X & Y \\
    D1 & D_2 & DX & v \\
    E & F & \int_a^b f(t) dt & v \\
\end{array}
\]

```

You can also use the `longtable` environment:

aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb

aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	baaaa	caaaaa	dbbbb
aaaaaa	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 & caaaa & dbbbb \\
...
aaaaa & baaaa & caaaa & 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**.
- **Ml**, **Mc** and **Mr**: Same as column **l**, **c** and **r** 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<align>**: Same as **M<align>** 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(An+B)` or `:nth-child(An)` 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(2n+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`, `collcell`. 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}
8 \RequirePackage{etoolbox}
```

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`.

```

9 \RequirePackage{mdwtab}
10 \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.

```

\l_cellprops_property_value_<name>_tl
  \cellprops_generic_setter:nnn
\__cellprops_get_property:n
\__cellprops_get_property:nN
\__cellprops_property_value_<name>_tl
  \cellprops_generic_setter:nnn {
    \exp_not:N \tl_set:Nn
    \exp_not:c { \l_cellprops_property_value_#2_tl }
    {#1 {#3}}
  }
  \cs_set_nopar:Nn \__cellprops_get_property:n {
    \tl_use:c { \l_cellprops_property_value_#1_tl }
  }
  \cs_new_protected_nopar:Nn \__cellprops_get_property:nN {
    \tl_if_exist:cTF { \l_cellprops_property_value_#1_tl } {
      \tl_set_eq:Nc #2 { \l_cellprops_property_value_#1_tl }
    }{
      \tl_clear:N #2
    }
  }
  
```

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

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

```

28 \cs_new_protected:Nn \__cellprops_define_properties:nn {
29   \clist_map_inline:nn {#2} {
30     \cs_set:cpn { \__cellprops_property_type_##1:nn } {#1}
31   }
32 }
```

(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.

```
33 \cs_new:Nn \__cellprops_delegate_setter:nn {
34     \use:c {__cellprops_property_type_#1:nn} {#1} {#2}
35 }
36 \cs_new_protected:Nn \__cellprops_use_setter:nn {
37     \use:x {
38         \__cellprops_delegate_setter:nn {#1} {#2}
39     }
40 }
```

(End definition for `__cellprops_use_setter:nn`.)

```
41 \cs_new_protected:Nn \__cellprops_parse_properties:Nn {
42     \tl_clear:N #1
43     \seq_set_split:Nnn \l_tmpa_seq {;} {#2}
44     \seq_map_inline:Nn \l_tmpa_seq {
45         \tl_if_empty:nF {##1} {
46             \exp_args:NNV \seq_set_split:Nnn \l_tmpb_seq \cColonStr {##1}
47             \int_compare:nNnTF {\seq_count:N \l_tmpb_seq} = { 2 } {
48                 \seq_get_left:NN \l_tmpb_seq \l_tmpa_tl
49                 \exp_args:NNV \str_set:Nn \l_tmpa_str \l_tmpa_tl
50                 \seq_get_right:NN \l_tmpb_seq \l_tmpa_tl
51                 \cs_if_exist:cTF { __cellprops_property_type_\l_tmpa_str :nn } {
52                     \tl_put_right:Nx #1 {
53                         \exp_args:NNV \__cellprops_delegate_setter:nn
54                             \l_tmpa_str \l_tmpa_tl
55                     }
56                 }{
57                     % TODO: ERROR-no property with that name
58                 }
59             }{
59                 % TODO: ERROR-too many : or none at all
60             }
61         }
62     }
63 }
64 }
65
66 \cs_new:Nn \__cellprops_fourval_setter:nnnnnn {
67     \__cellprops_fourval_setter_aux:w
68     {#1}{#2}{#3}{#4}#6~{\q_no_value}~{~\q_no_value}~{~\q_no_value}~{\q_stop}
69 }
70 \cs_new:Npn \__cellprops_fourval_setter_aux:w #1#2#3#4#5~#6~#7~#8~#9\q_stop {
71     \__cellprops_delegate_setter:nn {#1} {#5}
72     \quark_if_no_value:nTF {#6} {
73         \__cellprops_delegate_setter:nn {#2} {#5}
74         \__cellprops_delegate_setter:nn {#4} {#5}
75     }{
76         \__cellprops_delegate_setter:nn {#2} {#6}
77         \quark_if_no_value:nTF {#8} {
78             \__cellprops_delegate_setter:nn {#4} {#6}
79         }{
80             \__cellprops_delegate_setter:nn {#4} {#8}
81         }
82     }
83 }
```

```

81         }
82     }
83     \quark_if_no_value:nTF {#7} {
84         \__cellprops_delegate_setter:nn {#3} {#5}
85     }{
86         \__cellprops_delegate_setter:nn {#3} {#7}
87     }
88 }
89
90 \cs_new_protected:Nn \__cellprops_define_fourval_properties:nnnnnn {
91     \__cellprops_define_properties:nn {#1} { #3, #4, #5, #6 }
92     \__cellprops_define_properties:nn {
93         \__cellprops_fourval_setter:nnnnnn {#3}{#4}{#5}{#6}
94     }{
95         #2
96     }
97 }
98
99 \tl_const:Nn \c__cellprops_inherit_color_tl { \q_nil }
100
101 \cs_new_nopar:Nn \__cellprops_color_setter:nn {
102     \str_if_eq:nnTF {#2} {inherit} {
103         \__cellprops_generic_setter:nnn \exp_not:n {#1} {\c__cellprops_inherit_color_tl}
104     }{
105         \str_case_e:nnF { \str_range:nnn {#2} {1} {4} } {
106             {rgb()} {
107                 \__cellprops_generic_setter:nnn \use:n {#1} {
108                     \exp_not:n {\color[RGB]} {\str_range:nnn {#2} {5} {-2}}
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
129 \cs_new_nopar:Nn \__cellprops_linewidth_setter:nn {
130     \str_case:nnF {#2} {
131         {thin} { \__cellprops_generic_setter:nnn \exp_not:n {#1} { \fboxrule} }
132         {medium} { \__cellprops_generic_setter:nnn \exp_not:n {#1} { 2\fboxrule} }
133         {thick} { \__cellprops_generic_setter:nnn \exp_not:n {#1} { 3\fboxrule} }
134     }{

```

```

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

```

```

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

```

```

243 }
244
245 \tl_new:N \l__cellprops_parse_tmp_tl
246 \NewDocumentCommand \__cellprops_parse_css:w { lmu{\q_stop} } {
247     \quark_if_nil:nF {#2} {
248         \__cellprops_parse_properties:Nn \l__cellprops_parse_tmp_tl {#2}
249         \clist_map_inline:nn {#1} {
250             \__cellprops_parse_css_addprops:nV {##1} \l__cellprops_parse_tmp_tl
251         }
252         \__cellprops_parse_css:w #3 \q_stop
253     }
254 }
255
256
257 \seq_new:N \l__cellprops_parse_selector_seq
258 \tl_new:N \l__cellprops_parse_desc_tl
259
260 \str_const:Nn \c__cellprops_parse_nthchild_str { :nth-child( }
261 \prop_new:N \c__cellprops_parse_replace_prop
262 \prop_put:Nnn \c__cellprops_parse_replace_prop { :first-child } { :nth-child(1) }
263
264 \cs_new_protected:Nn \__cellprops_parse_selector:Nn {
265     \str_set:Nx \l_tmpa_str {#2}

```

Replace some aliases with their meaning:

```

266     \prop_map_inline:Nn \c__cellprops_parse_replace_prop {
267         \use:x {
268             \exp_not:n { \tl_replace_all:Nnn \l_tmpa_str }
269             { \tl_to_str:n {##1} } { \tl_to_str:n {##2} }
270         }
271     }

```

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

```

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

```

And replace them back.

```

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

```

And replace them back again.

```

281         \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.

```

282         \tl_replace_once:Nnn \l_tmpa_tl { ) } { \q_stop\prg_do_nothing: }

```

```

283     \seq_set_split:NnV \l_tmpa_seq { \q_stop } \l_tmpa_tl
284     \seq_pop_right:NN \l_tmpa_seq \l__cellprops_parse_desc_tl
285     \tl_replace_once:Nnn \l__cellprops_parse_desc_tl { \prg_do_nothing: } {}
286     \seq_get_left:NNT \l_tmpa_seq \l_tmpa_tl {
287         \exp_args:NNV \l__cellprops_parse_nth:Nn \l_tmpa_tl \l_tmpa_tl
288         \tl_put_left:Nn \l__cellprops_parse_desc_tl { ) }
289         \tl_put_left:NV \l__cellprops_parse_desc_tl \l_tmpa_tl
290     }
291     \seq_put_right:NV \l__cellprops_parse_selector_seq \l__cellprops_parse_desc_tl
292 }
293 \tl_set:Nx #1 {
294     \exp_args:NNV \seq_use:Nn
295         \l__cellprops_parse_selector_seq \c__cellprops_parse_nthchild_str
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 \l__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 \l__cellprops_parse_css_addprops:nV #1 #2 {
329     \l__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 \l__cellprops_recall_properties:n {
336 
```

```

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

```

```

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

```

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

```

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

```

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

```

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

```

```

440 \cs_set_nopar:Nn \__cellprops_array_startcontent: {
441     \hbox{s[\l__cellprops_tablepadding_top_dim]}
442 }
443
444 \cs_set_protected_nopar:Nn \__cellprops_readpreamble:n {
445     \cs_set_eq:NN \tab@readpreamble \__cellprops_orig_tab@readpreamble:n
446     \tl_put_left:Nn \tab@multicol {\__cellprops_startrow:}
447     \tl_put_left:Nn \tab@tabtext {\int_gincr:N \g__cellprops_col_int}
448     \tab@readpreamble{#1}
449     \exp_args:Nx \tab@preamble
450         { \exp_not:N \__cellprops_startrow: \the\tab@preamble \exp_not:N \__cellprops_endrow: }
451 }

```

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.

```

452 \cs_new_protected_nopar:Nn \__cellprops_update_color:Nn {
453     \__cellprops_get_property:nN {#2} \l_tmpa_tl
454     \exp_args:NV \tl_if_eq:NNT \l_tmpa_tl \c__cellprops_inherit_color_tl {
455         \tl_set_eq:NN #1 \l_tmpa_tl
456     }
457 }
458
459 \cs_new_protected_nopar:Nn \__cellprops_update_colors: {
460     \__cellprops_update_color:Nn \l__cellprops_color_tl {color}
461     \__cellprops_update_color:Nn \l__cellprops_bgcolor_tl {background-color}
462 }

```

Patch the `\@array`, `\LT@array`, `\@mkpream`, `\endarray` and `\endlongtable` commands, so that we can properly setup our line and column counting system. This is the most brittle part of `cellprops`, and subject to compatibility problems with other packages that patch those (`hyperref` in particular).

```

463 \AtEndPreamble{%
464 \cs_set_eq:NN \__cellprops_orig_array:w \@array
465 \cs_set_protected_nopar:Npn \@array[#1]#2 {
466     \__cellprops_array_init:
467     \__cellprops_orig_array:w [#1]{#2}
468     \__cellprops_array_startcontent:
469 }
470
471 \cs_set_eq:NN \__cellprops_orig_LTmkpream:n \@mkpream
472 \cs_set_protected_nopar:Npn \@mkpream#1 {
473     \group_end:
474     \__cellprops_array_init:
475     \group_begin:
476     \__cellprops_orig_LTmkpream:n {#1}
477 }
478
479 \cs_set_eq:NN \__cellprops_orig_LTarray:w \LT@array
480 \cs_set_protected_nopar:Npn \LT@array [#1]#2 {
481     \__cellprops_orig_LTarray:w [#1]{#2}
482     \__cellprops_array_startcontent:
483 }

```

```

484 \cs_new_nopar:Nn \__cellprops_end_array:n {
485     \tl_if_empty:NF \g__cellprops_borders_tl { \\ }
486     \crr
487     \hlx{s[\l__cellprops_tablepadding_bottom_dim]}
488     #1
489     \tl_use:N \l__cellprops_restore_tl
490 }
491
492 \cs_set_eq:NN \__cellprops_orig_endarray: \endarray
493 \cs_set_nopar:Npn \endarray {
494     \__cellprops_end_array:n { \__cellprops_orig_endarray: }
495 }
496 \cs_set_eq:NN \endtabular \endarray
497 \cs_set_eq:cN {endtabular*} \endarray
498
499 \cs_set_eq:NN \__cellprops_orig_endLT: \endlongtable
500 \cs_set_nopar:Npn \endlongtable {
501     \__cellprops_end_array:n { \__cellprops_orig_endLT: }
502 }
503
504 \cs_new_protected_nopar:Nn \__cellprops_startrow: {
505     \bool_if:NF \g__cellprops_inrow_bool {
506         \bool_gset_true:N \g__cellprops_inrow_bool
507         \int_gincr:N \g__cellprops_row_int
508         \int_gset_eq:NN \g__cellprops_col_int \c_one_int
509         \dim_gzero:N \g__cellprops_ht_dim
510         \dim_gzero:N \g__cellprops_dp_dim
511     }
512 }
513
514 \cs_new_protected_nopar:Nn \__cellprops_endrow: {
515     \bool_if:NT \g__cellprops_inrow_bool {
516         \bool_gset_false:N \g__cellprops_inrow_bool
517     }
518 }
519
520 \cs_new_protected_nopar:Nn \__cellprops_cr:n {
521     \__cellprops_endrow:
522     \tl_if_empty:NF \g__cellprops_borders_tl {
523         \cr
524         \noalign{\nobreak}
525         \tl_use:N \g__cellprops_borders_tl
526         \tl_gclear:N \g__cellprops_borders_tl
527     }
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 \oxargarraycr #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 }
552
553 \cs_new_nopar:Nn \__cellprops_fix_valign_end:n {
554     \noalign{
555         \dim_set:Nn \l_tmpa_dim {#1}
556         \skip_vertical:n {\l_tmpa_dim}
557         \exp_args:NV \tl_if_eq:nnTF \tab@hlstate {b} {
558             \dim_gadd:Nn \tab@endheight { \g__cellprops_dp_dim + \l_tmpa_dim }
559         }{
560             \int_compare:nNnT \g__cellprops_row_int = \c_one_int {
561                 \dim_gadd:Nn \tab@endheight { \g__cellprops_ht_dim }
562             }
563         }
564     }
565 }

```

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

```

566 \cs_set_eq:NN \firsthline \hline
567 \cs_set_eq:NN \lasthline \hline
568
569 \colpush{tabular}
570
571 \coldef n{\tabcoltype{
572     \__cellprops_begincell:n{}}
573 }{
574     \__cellprops_endcell:
575 }
576 \coldef l{\tabcoltype{
577     \__cellprops_begincell:n
578     {\__cellprops_use.setter:nn {text-align} {left}}}
579 }{
580     \__cellprops_endcell:
581 }
582 \coldef c{\tabcoltype{
583     \__cellprops_begincell:n
584     {\__cellprops_use.setter:nn {text-align} {center}}}
585 }{
586     \__cellprops_endcell:
587 }
588 \coldef r{\tabcoltype{

```

```

589      \__cellprops_begincell:n
590          {\__cellprops_use_setter:nn {text-align} {right}}
591      }{
592          \__cellprops_endcell:
593      }{
594      \coldef M#1{\__cellprops_MTcol:nn {math}{#1}}
595      \coldef T#1{\__cellprops_MTcol:nn {text}{#1}}
596      \cs_new_protected_nopar:Nn \__cellprops_MTcol:nn {
597          % TODO: error if align not l, c, or r
598          \exp_args:Nx \tabcoltype {
599              \exp_not:N \__cellprops_begincell:n {
600                  \exp_not:n {\__cellprops_use_setter:nn {math-mode} {#1} }
601                  \exp_not:n {\__cellprops_use_setter:nn {text-align} {
602                      \str_case:nn {#2} {
603                          {l} {left}
604                          {c} {center}
605                          {r} {right}
606                      }
607                  }
608              }
609          }{
610              \__cellprops_endcell:
611          }
612      }
613
614      \coldef p#1{\tabcoltype{
615          \__cellprops_begin_par_cell:nn \vtop {#1}
616      }{
617          \__cellprops_end_par_cell:n {}
618      }{
619      \coldef m#1{\tabcoltype{
620          \__cellprops_begin_par_cell:nn {\c_math_toggle_token\vcenter} {#1}
621      }{
622          \__cellprops_end_par_cell:n{\c_math_toggle_token}
623      }{
624      \coldef b#1{\tabcoltype{
625          \__cellprops_begin_par_cell:nn \vbox {#1}
626      }{
627          \__cellprops_end_par_cell:n {}
628      }
629
630
631 \colpop
Handle various :nth-child() forms.

632 \cs_new_protected_nopar:Nn \__cellprops_seq_nthchild:Nn {
633     \seq_clear:N #1
634     \seq_map_inline:Nn \l__cellprops_used_nth_factors_seq {
635         \seq_put_right:Nx #1 {
636             ##1 n + \int_eval:n{\int_mod:nn{#2}{##1}}
637         }
638     }
639     \seq_put_right:Nx #1 { \int_eval:n{#2} }
640 }

```

```

641
642 \cs_new_protected_nopar:Nn \__cellprops_begincell:n {
643     \__cellprops_begin_raw_cell:n {
644         #1
645         \hbox_set:Nw \l__cellprops_cell_box
646         \str_case_e:nnF {\__cellprops_get_property:n {math-mode}} {
647             { text } { \tab@btext }
648             { math } { \tab@bmaths }
649         }{\% any other treated as |auto|
650             \tab@bgroup
651         }
652     }
653 }
654
655 \cs_new_protected_nopar:Nn \__cellprops_endcell: {
656     \str_case_e:nnF {\__cellprops_get_property:n {math-mode}} {
657         { text } { \tab@etext }
658         { math } { \tab@emath }
659     }{\% any other treated as |auto|
660         \tab@egroup
661     }
662     \hbox_set_end:
663     \__cellprops_end_raw_cell:
664 }
665
666 \cs_new_protected_nopar:Nn \__cellprops_begin_par_cell:nn {
667     \savenotes
668     \__cellprops_begin_raw_cell:n{
669         \hbox_set:Nw \l__cellprops_cell_box
670         #1
671         \bgroup
672         \hsize#2\relax
673         \arrayparboxrestore
674         \global\minipagetrue
675         \everypar{
676             \global\minipagetrue
677             \everypar{ }
678         }
679         \__cellprops_seq_nthchild:Nn \l_tmpa_seq { \g__cellprops_row_int }
680         \__cellprops_seq_nthchild:Nn \l_tmpb_seq { \g__cellprops_col_int }
681         \__cellprops_recall_properties:n {td~p}
682         \seq_map_inline:Nn \l_tmpa_seq {
683             \__cellprops_recall_properties:n {tr:nth-child(##1)~p}
684         }
685         \seq_map_inline:Nn \l_tmpb_seq {
686             \__cellprops_recall_properties:n {td:nth-child(##1)~p}
687         }
688         \__cellprops_recall_properties:n {tr~td~p}
689         \seq_map_inline:Nn \l_tmpa_seq {
690             \__cellprops_recall_properties:n {tr:nth-child(##1)~td~p}
691         }
692         \seq_map_inline:Nn \l_tmpa_seq {
693             \seq_map_inline:Nn \l_tmpb_seq {
694                 \__cellprops_recall_properties:n {tr:nth-child(##1)~
```

```

695             td:nth-child(####1)-p}
696         }
697     }
698 }
699 }
700 \cs_new_protected_nopar:Nn \__cellprops_end_par_cell:n {
701     \ifhmode\@maybe@unskip\par\fi
702     \unskip
703     \egroup
704     #1
705     \hbox_set_end:
706     \__cellprops_end_raw_cell:
707     \spewnotes
708 }
709
710 \cs_new_protected_nopar:Nn \__cellprops_begin_raw_cell:n {
711     \group_begin:
712     \__cellprops_seq_nthchild:Nn \l_tmpa_seq { \g__cellprops_row_int }
713     \__cellprops_seq_nthchild:Nn \l_tmpb_seq { \g__cellprops_col_int }
714     \seq_map_inline:Nn \l_tmpa_seq {
715         \__cellprops_recall_properties:n {tr:nth-child(#1)}
716     }
717     \__cellprops_update_colors:
718     \__cellprops_recall_properties:n {td}
719     \__cellprops_recall_properties:n {tr-td}
720     \seq_map_inline:Nn \l_tmpb_seq {
721         \__cellprops_recall_properties:n {td:nth-child(#1)}
722     }
723     \seq_map_inline:Nn \l_tmpa_seq {
724         \__cellprops_recall_properties:n {tr:nth-child(#1)-td}
725     }
726     \seq_map_inline:Nn \l_tmpa_seq {
727         \seq_map_inline:Nn \l_tmpb_seq {
728             \__cellprops_recall_properties:n {tr:nth-child(#1)-
729                             td:nth-child(####1)}
730         }
731     }
732     \__cellprops_update_colors:
733     % Additional init code
734     #1
735     % Install the cell color
736     \__cellprops_update_colors:
737     \tl_use:N \l__cellprops_color_tl
738 }
739
740 \cs_new_protected_nopar:Nn \__cellprops_make_solid_hborder:nnn {
741     \group_begin:
742         \hbox_set_to_wd:Nnn \l_tmpa_box {1pt} {
743             \hss
744             \hbox:n {
745                 #3 % install color
746                 \vrule height~\dim_eval:n{#1+#2}
747                 ~depth~- \dim_eval:n{#2}
748                 ~width~3pt

```

```

749         }
750         \hss
751     }
752     \box_set_ht:Nn \l_tmpa_box { \c_zero_dim }
753     \box_set_dp:Nn \l_tmpa_box { \c_zero_dim }
754     \kern 1pt
755     \box_use:N \l_tmpa_box
756     \xleaders
757         \box_use:N \l_tmpa_box
758         \skip_horizontal:n {-4pt~plus~-1fil}
759     \box_use:N \l_tmpa_box
760         \kern 1pt
761         \skip_horizontal:n {0pt~plus~-1fil}
762 \group_end:
763 }
764 \cs_new_protected_nopar:Nn \__cellprops_make_solid_vborder:nnn {
765     \group_begin:
766         \hbox_set_to_wd:Nnn \l_tmpa_box {0pt} {
767             \hbox:n {
768                 #3 % install color
769                 \vrule height~\dim_eval:n{#2}~width~\dim_eval:n{#1}
770             }
771             \hss
772         }
773         \box_set_ht:Nn \l_tmpa_box { \c_zero_dim }
774         \box_set_dp:Nn \l_tmpa_box { \c_zero_dim }
775         \box_use:N \l_tmpa_box
776     \group_end:
777 }
778 \clist_map_inline:nn {
779     dotted, dashed, solid, double,
780     groove, ridge, inset, outset
781 }{
782     \cs_set_eq:cN {\__cellprops_make_#1_hborder:nnn} \__cellprops_make_solid_hborder:nnn
783     \cs_set_eq:cN {\__cellprops_make_#1_vborder:nnn} \__cellprops_make_solid_vborder:nnn
784 }
785 \dim_new:N \l__cellprops_border_width_dim
786 \str_new:N \l__cellprops_border_style_str
787 \tl_new:N \l__cellprops_border_color_tl
788 \cs_new_protected_nopar:Nn \__cellprops_get_border_info:n {
789     \dim_set:Nn \l__cellprops_border_width_dim {\__cellprops_get_property:n {border-#1-width}}
790     \__cellprops_get_property:nN {border-#1-style} \l_tmpa_tl
791     \exp_args:NNV \str_set:Nn \l__cellprops_border_style_str \l_tmpa_tl
792     \tl_clear:N \l__cellprops_border_color_tl
793     \cs_if_exist:cTF {\__cellprops_make_\l__cellprops_border_style_str _hborder:nnn} {
794         \__cellprops_update_color:Nn \l__cellprops_border_color_tl {border-#1-color}
795     }{
796         \dim_zero:N \l__cellprops_border_width_dim
797     }
798 }
799 }
800 \cs_new_protected_nopar:Npn \__cellprops_make_hborder:nnnn #1 {

```

```

802     \use:c { __cellprops_make_#1_hborder:nnn }
803 }
804 \cs_new_protected_nopar:Npn \__cellprops_make_vborder:nnnn #1 {
805     \use:c { __cellprops_make_#1_vborder:nnn }
806 }
807
808 \cs_new_protected_nopar:Nn \__cellprops_end_raw_cell: {
809     % Here \l__cellprops_cell_box must contain the contents of the cell
810     %
811     % Prepare the borders token list
812     \int_compare:nNnT \g__cellprops_col_int = 1 {
813         \tl_gclear:N \g__cellprops_borders_tl
814     }
815     \tl_gput_right:Nx \g__cellprops_borders_tl {
816         \tl_if_empty:NF \g__cellprops_borders_tl { \exp_not:n {&} }
817         \exp_not:n { \omit \kern \c_zero_dim }
818     }
819     % Handle padding-top, min-height and border-top
820     \__cellprops_get_border_info:n {top}
821     \box_set_ht:Nn \l__cellprops_cell_box {
822         \dim_max:nn
823             { \box_ht:N \l__cellprops_cell_box }
824             { \__cellprops_get_property:n {min-height} }
825             + ( \__cellprops_get_property:n {padding-top} )
826             + \l__cellprops_border_width_dim
827     }
828     \dim_compare:nNnT \l__cellprops_border_width_dim > \c_zero_dim {
829         \tl_gput_right:Nx \g__cellprops_borders_tl {
830             \exp_not:N \__cellprops_make_hborder:nnnn
831                 { \exp_not:V \l__cellprops_border_style_str }
832                 { \dim_use:N \l__cellprops_border_width_dim }
833                 {
834                     \exp_not:n { \g__cellprops_dp_dim + \g__cellprops_ht_dim - }
835                     \dim_use:N \l__cellprops_border_width_dim
836                 }
837                 { \exp_not:V \l__cellprops_border_color_tl }
838             }
839     }
840     % Handle padding-bottom, min-depth and border-bottom
841     \__cellprops_get_border_info:n {bottom}
842     \box_set_dp:Nn \l__cellprops_cell_box {
843         \dim_max:nn
844             { \box_dp:N \l__cellprops_cell_box }
845             { \__cellprops_get_property:n {min-depth} }
846             + ( \__cellprops_get_property:n {padding-bottom} )
847             + \l__cellprops_border_width_dim
848     }
849     \dim_compare:nNnT \l__cellprops_border_width_dim > \c_zero_dim {
850         \tl_gput_right:Nx \g__cellprops_borders_tl {
851             \exp_not:N \__cellprops_make_hborder:nnnn
852                 { \exp_not:V \l__cellprops_border_style_str }
853                 { \dim_use:N \l__cellprops_border_width_dim }
854                 { \exp_not:n { Opt } }
855                 { \exp_not:V \l__cellprops_border_color_tl }

```

```

856     }
857 }
858 % To fix vertical alignment later
859 \dim_gset:Nn \g__cellprops_ht_dim {
860     \dim_max:nn
861         {\g__cellprops_ht_dim}
862         {\box_ht:N \l__cellprops_cell_box}
863 }
864 \dim_gset:Nn \g__cellprops_dp_dim {
865     \dim_max:nn
866         {\g__cellprops_dp_dim}
867         {\box_dp:N \l__cellprops_cell_box}
868 }
869 % Handle padding-left and border-left
870 \__cellprops_get_border_info:n {left}
871 \skip_set:Nn \l__cellprops_left_skip
872     {\__cellprops_get_property:n {padding-left} + \l__cellprops_border_width_dim}
873 \dim_compare:nNnT \l__cellprops_border_width_dim > \c_zero_dim {
874     \tl_gput_right:Nx \g__cellprops_borders_tl {
875         \exp_not:N \__cellprops_make_vborder:nnnn
876             { \exp_not:V \l__cellprops_border_style_str }
877             { \dim_use:N \l__cellprops_border_width_dim }
878             { \exp_not:n { \g__cellprops_dp_dim + \g__cellprops_ht_dim } }
879             { \exp_not:V \l__cellprops_border_color_tl }
880     }
881 }
882 \tl_gput_right:Nx \g__cellprops_borders_tl {
883     \exp_not:n {
884         \skip_horizontal:n {0pt~plus~-1fil}
885         \kern \c_zero_dim
886     }
887 }
888 \__cellprops_get_border_info:n {right}
889 \skip_set:Nn \l__cellprops_right_skip
890     {\__cellprops_get_property:n {padding-right} + \l__cellprops_border_width_dim}
891 \dim_compare:nNnT \l__cellprops_border_width_dim > \c_zero_dim {
892     \tl_gput_right:Nx \g__cellprops_borders_tl {
893         \exp_not:N \skip_horizontal:n
894             { - \dim_use:N \l__cellprops_border_width_dim }
895         \exp_not:N \__cellprops_make_vborder:nnnn
896             { \exp_not:V \l__cellprops_border_style_str }
897             { \dim_use:N \l__cellprops_border_width_dim }
898             { \exp_not:n { \g__cellprops_dp_dim + \g__cellprops_ht_dim } }
899             { \exp_not:V \l__cellprops_border_color_tl }
900         \exp_not:N \skip_horizontal:n
901             { \dim_use:N \l__cellprops_border_width_dim }
902         \exp_not:n { \kern \c_zero_dim }
903     }
904 }
905 % Handle hpadding and halign
906 \skip_set:Nn \l_tmpa_skip {
907     \dim_max:nn
908     {0pt}
909     { (\__cellprops_get_property:n {min-width})}

```

```

910           - \box_wd:N \l__cellprops_cell_box }
911     }
912   \skip_add:Nn \l_tmpa_skip {
913     1sp plus 1fil
914   }
915   \str_case_e:nnF {\_cellprops_get_property:n {text-align}} {
916     { right } {
917       \skip_add:Nn \l__cellprops_left_skip { \l_tmpa_skip }
918     }
919     { center } {
920       \skip_add:Nn \l__cellprops_left_skip { \l_tmpa_skip / 2 }
921       \skip_add:Nn \l__cellprops_right_skip { \l_tmpa_skip / 2 }
922     }
923   }{%
924     any other treated as |left|
925     \skip_add:Nn \l__cellprops_right_skip { \l_tmpa_skip }
926   }
927   \kern\c_zero_dim
928   \tl_if_empty:NF \l__cellprops bgcolor_tl {
929     \group_begin:
930     % Paint a background with leaders
931     \tl_use:N \l__cellprops bgcolor_tl % install the color
932     \skip_set:Nn \l_tmpa_skip {
933       \l__cellprops_left_skip
934       + \box_wd:N \l__cellprops_cell_box
935       + \l__cellprops_right_skip
936     }
937     \leaders
938       \vrule
939       \skip_horizontal:N \l_tmpa_skip
940       \skip_horizontal:n {-\l_tmpa_skip}
941     \group_end:
942   }
943   \skip_horizontal:N \l__cellprops_left_skip
944   \box_use_drop:N \l__cellprops_cell_box
945   \skip_horizontal:N \l__cellprops_right_skip
946   \kern\c_zero_dim
947   \group_end:
948 }
```