

The `cellprops` package

CSS-like cell and table properties*

Julien “_FrnchFrfgg_” RIVAUD[†]

Released 2016/04/05

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

1.1 Example

The code:

```
\[
\cellprops{
  td { padding: 1ex; }
  td:nth-child(4) { color: red; }
  tr:nth-child(2) { color: red; }
  tr:nth-child(1) td:nth-child(4) { color: inherit; }
  td:nth-child(even) { background-color: black!5!white; }
  tr:nth-child(2n+1) { background-color: yellow!30!white; }
  tr:nth-child(odd) td:nth-child(even) {
    background-color: black!20!yellow!20!white;
  }
}
\begin{array}{Tlccc}
\hline
This is text & A_2 & A_3 & A_4 \\
\hline
B1 & This is maths & B_3 & \\
C1 & C_2 & X & Y \\
D1 & D_2 & DX & v \\
E & F & \int_a^b f(t) dt & v \\
\hline
\end{array}\]
```

*This file describes v1.0, last revised 2016/04/05.

[†]E-mail: frnchfrgg@free.fr

produces:

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
E	F	$\int_a^b f(t)dt$	v

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>]
<pseudo-class>: ':nth-child(' <nth> ')'
<nth>:        <number> | 'odd' | 'even' | <number> 'n+' <number>
<properties>: [ <property> ';' ]*
<property>:   'padding: ' <dimension> [ ' ' <dimension> ]* |
               'padding-top: ' <dimension> |
               'padding-left: ' <dimension> |
               'padding-bottom: ' <dimension> |
               'padding-right: ' <dimension> |
               'min-height: ' <dimension> |
               'min-depth: ' <dimension> |
               'min-width: ' <dimension> |
               'text-align: ' ( 'left' | 'right' | 'center' ) |
               'math-mode: ' ( 'text' | 'math' | 'auto' ) |
               'color: ' ( <xcolor-expression> | 'inherit' ) |
               'background-color: ' ( <xcolor-expression> | 'transparent' )

```

Most of these properties are straight-forward. Some explanations:
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`.
- 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.
- Currently, only `2n`, `2n+1`, `3n`, `3n+1` and `3n+2` are supported, without any space, as arguments of `nth-child()` other than a single number, `odd` or `even` without any space. Supporting others is trivial, but increase the running time. In a near-future release, I intend to detect which multiples are actually used and automatically support those.

1.3 TODO

Add `border` properties. Add automatic support of `:nth-child(an+b)` for $a \geq 4$.

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 } }
10
11

```

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.

```

12 \cs_set:Nn \__cellprops_generic_setter:nnn {
13   \exp_not:N \tl_set:Nn
14   \exp_not:c { l__cellprops_property_value_#2_tl }
15   {#1 {#3}}
16 }
17
18 \cs_set_nopar:Nn \__cellprops_get_property:n {
19   \tl_use:c { l__cellprops_property_value_#1_tl }
20 }
21
22 \cs_set_protected_nopar:Nn \__cellprops_get_property:nN {
23   \tl_set_eq:Nc #2 { l__cellprops_property_value_#1_tl }
24 }

```

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

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

```

25 \cs_set_protected:Nn \__cellprops_define_properties:nn {
26   \clist_map_inline:nn {#2} {
27     \cs_set:cpn { __cellprops_property_type_##1:nn } {#1}
28   }
29 }

```

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

```

30 \cs_set_protected:Nn \__cellprops_use_setter:nn {
31   \use:x {
32     \use:c { __cellprops_property_type_#1:nn } {#1} {#2}
33   }
34 }

```

(End definition for `__cellprops_use_setter:nn`.)

```

35 \cs_set_protected:Nn \__cellprops_parse_properties:nn {
36   \tl_if_exist:cF { l__cellprops_property_group_#1_tl } {
37     \tl_clear:c { l__cellprops_property_group_#1_tl }
38   }

```

```

39 \seq_set_split:Nnn \l_tmpa_seq {;} {#2}
40 \seq_map_inline:Nn \l_tmpa_seq {
41   \tl_if_empty:nF {##1} {
42     \exp_args:NNV \seq_set_split:Nnn \l_tmpb_seq \c_colon_str {##1}
43     \int_compare:nNnTF {\seq_count:N \l_tmpb_seq} = \c_two {
44       \seq_get_left:NN \l_tmpb_seq \l_tmpa_tl
45       \exp_args:NNV \str_set:Nn \l_tmpa_str \l_tmpa_tl
46       \seq_get_right:NN \l_tmpb_seq \l_tmpa_tl
47       \cs_if_exist:cTF { __cellprops_property_type_\l_tmpa_str :nn } {
48         \tl_put_right:cx { l__cellprops_property_group_#1_tl } {
49           \exp_args:Nc \exp_args:NVV
50             { __cellprops_property_type_\l_tmpa_str :nn }
51           \l_tmpa_str \l_tmpa_tl
52         }
53       }{
54         % TODO: ERROR-no property with that name
55       }
56     }{
57       % TODO: ERROR-too many : or none at all
58     }
59   }
60 }
61 }
62
63 \cs_set:Nn \__cellprops_fourdim_setter:nn {
64   \__cellprops_fourdim_setter_aux:w
65   {#1}#2~{\q_no_value}~{\q_no_value}~{\q_no_value}~\q_stop
66 }
67 \cs_set:Npn \__cellprops_fourdim_setter_aux:w #1#2~#3~#4~#5~#6\q_stop {
68   \__cellprops_generic_setter:nnn \exp_not:n {#1-top} {#2}
69   \quark_if_no_value:nTF {#3} {
70     \__cellprops_generic_setter:nnn \exp_not:n {#1-right} {#2}
71     \__cellprops_generic_setter:nnn \exp_not:n {#1-left} {#2}
72   }{
73     \__cellprops_generic_setter:nnn \exp_not:n {#1-right} {#3}
74     \quark_if_no_value:nTF {#5} {
75       \__cellprops_generic_setter:nnn \exp_not:n {#1-left} {#3}
76     }{
77       \__cellprops_generic_setter:nnn \exp_not:n {#1-left} {#5}
78     }
79   }
80   \quark_if_no_value:nTF {#4} {
81     \__cellprops_generic_setter:nnn \exp_not:n {#1-bottom} {#2}
82   }{
83     \__cellprops_generic_setter:nnn \exp_not:n {#1-bottom} {#4}
84   }
85 }
86
87 \tl_const:Nn \c__cellprops_inherit_color_tl { \q_nil }
88
89 \cs_new_nopar:Nn \__cellprops_color_setter:nn {
90   \str_if_eq:nnTF {#2} {inherit} {
91     \__cellprops_generic_setter:nnn \exp_not:n {#1} {\c__cellprops_inherit_color_tl}
92   }{

```

```

93     \str_case_x:nnF { \str_range:nnn {#2} {1} {4} } {
94         {rgb()} {
95             \__cellprops_generic_setter:nnn \use:n {#1} {
96                 \exp_not:n {\color[RGB]} {\str_range:nnn {#2} {5} {-2}}
97             }}
98         {hsl()} {
99             \__cellprops_generic_setter:nnn \use:n {#1} {
100                 \exp_not:n {\color[Hsb]} {\str_range:nnn {#2} {5} {-2}}
101             }}
102     }{
103         \__cellprops_generic_setter:nnn \exp_not:n {#1} {
104             \color{#2}
105         }
106     }
107 }
108 }
109 \cs_new_nopar:Nn \__cellprops_bgcolor_setter:nn {
110     \str_if_eq:nnTF {#2} {transparent} {
111         \__cellprops_color_setter:nn {#1} {inherit}
112     }{
113         \__cellprops_color_setter:nn {#1} {#2}
114     }
115 }
116
117
118 \__cellprops_define_properties:nn {
119     \__cellprops_generic_setter:nnn \exp_not:n
120 }{
121     padding-top,
122     padding-right,
123     padding-bottom,
124     padding-left,
125     min-height,
126     min-depth,
127     min-width,
128 }
129
130 \__cellprops_define_properties:nn {
131     \__cellprops_fourdim_setter:nn
132 }{
133     padding
134 }
135
136 \__cellprops_define_properties:nn {
137     \__cellprops_generic_setter:nnn \tl_to_str:n
138 }{
139     text-align,
140     math-mode,
141 }
142
143 \__cellprops_define_properties:nn {
144     \__cellprops_color_setter:nn
145 }{
146     color,

```

```

147 }
148
149 \__cellprops_define_properties:nn {
150     \__cellprops_bgcolor_setter:nn
151 }{
152     background-color,
153 }
154
155
156 % TODO: properties for array
157 \NewDocumentCommand \cellprops { m } {
158     \__cellprops_parse_css:n {#1}
159 }
160
161 \cs_new_protected:Nn \__cellprops_parse_css:n {
162     \__cellprops_parse_css:w #1 \q_mark {\q_nil} \q_stop
163 }
164 \NewDocumentCommand \__cellprops_parse_css:w { lmu{\q_stop} } {
165     \quark_if_nil:nF {#2} {
166         \clist_map_inline:nn {#1} {
167             \__cellprops_parse_properties:nn {##1} {#2}
168         }
169         \__cellprops_parse_css:w #3 \q_stop
170     }
171 }
172
173 \cs_set_protected:Nn \__cellprops_recall_properties:n {
174     \tl_if_exist:cT { l__cellprops_property_group_#1_tl } {
175         \tl_use:c { l__cellprops_property_group_#1_tl }
176     }
177     \clist_map_inline:nn { \@currenvir } {
178         \tl_if_exist:cT { l__cellprops_property_group_##1~#1_tl } {
179             \tl_use:c { l__cellprops_property_group_##1~#1_tl }
180         }
181     }
182 }
183
184 \dim_new:N \l__cellprops_colsep_dim
185 \ExplSyntaxOff
186 \cellprops{
187     td {
188         padding: Opt \csname l__cellprops_colsep_dim\endcsname;
189         min-height: Opt;
190         min-depth: Opt;
191         min-width: Opt;
192         text-align: left;
193         math-mode: auto;
194         color: inherit;
195         background-color: transparent;
196     }
197     tr {
198         color: inherit;
199         background-color: transparent;
200     }

```

```

201     table {
202         padding: Opt; % No change at load time
203         color: inherit;
204         background-color: transparent;
205     }
206 }
207 \ExplSyntaxOn
208
209 \int_new:N \g__cellprops_row_int
210 \int_new:N \g__cellprops_col_int
211 \bool_new:N \g__cellprops_inrow_bool
212 \bool_gset_false:N \g__cellprops_inrow_bool
213
214 \box_new:N \l__cellprops_cell_box
215 \skip_new:N \l__cellprops_left_skip
216 \skip_new:N \l__cellprops_right_skip
217 \dim_new:N \g__cellprops_ht_dim
218 \dim_new:N \g__cellprops_dp_dim
219
220 \tl_new:N \l__cellprops_restore_tl
221
222 \dim_new:N \l__cellprops_tablepadding_top_dim
223 \dim_new:N \l__cellprops_tablepadding_bottom_dim
224 \tl_new:N \l__cellprops_color_tl
225 \tl_new:N \l__cellprops_bgcolor_tl
226
227 % To count rows and columns
228 \cs_set_eq:NN \__cellprops_orig_array: \@array
229 \cs_set_protected_nopar:Npn \@array[#1]#2 {
230     \tl_set:Nx \l__cellprops_restore_tl {
231         \bool_if:NTF \g__cellprops_inrow_bool {
232             \exp_not:n {\bool_gset_true:N \g__cellprops_inrow_bool}
233         }{
234             \exp_not:n {\bool_gset_false:N \g__cellprops_inrow_bool}
235         }
236         \exp_not:n { \int_gset:Nn \g__cellprops_row_int }
237         { \int_use:N \g__cellprops_row_int }
238         \exp_not:n { \int_gset:Nn \g__cellprops_col_int }
239         { \int_use:N \g__cellprops_col_int }
240         \exp_not:n { \dim_gset:Nn \g__cellprops_ht_dim }
241         { \dim_use:N \g__cellprops_ht_dim }
242         \exp_not:n { \dim_gset:Nn \g__cellprops_dp_dim }
243         { \dim_use:N \g__cellprops_dp_dim }
244     }
245     \int_gzero:N \g__cellprops_row_int
246     \bool_gset_false:N \g__cellprops_inrow_bool
247     \cs_set_eq:NN \__cellprops_orig_tab@initread: \tab@initread
248     \cs_set_protected_nopar:Npn \tab@initread {
249         \cs_set_eq:NN \tab@initread \__cellprops_orig_tab@initread:
250         \tab@initread
251         \tl_put_left:Nn \tab@tabtext {\int_gincr:N \g__cellprops_col_int}
252     }
253 }
254 % \end{macrocode}
255 % Zero \cs{col@sep} but remember its value for the default padding.

```



```

255 % \begin{macrocode}
256   \dim_set_eq:NN \l__cellprops_colsep_dim \col@sep
257   \dim_zero:N \col@sep
258 % \end{macrocode}
259 % Also ignore \cs{*extrasep} dimensions that are not part of cellprop
260 % interface and should be replaced by CSS equivalents.
261 % \begin{macrocode}
262   \dim_zero:N \tab@extrasep
263   \group_begin:
264     \__cellprops_recall_properties:n {table}
265     \dim_gset:Nn \g_tmpa_dim { \__cellprops_get_property:n {padding-top} }
266     \dim_gset:Nn \g_tmpb_dim { \__cellprops_get_property:n {padding-bottom} }
267     \__cellprops_update_colors:
268     \tl_gset_eq:NN \g_tmpa_tl \l__cellprops_color_tl
269     \tl_gset_eq:NN \g_tmpb_tl \l__cellprops_bgcolor_tl
270   \group_end:
271   \dim_set_eq:NN \l__cellprops_tablepadding_top_dim \g_tmpa_dim
272   \dim_set_eq:NN \l__cellprops_tablepadding_bottom_dim \g_tmpb_dim
273   \tl_set_eq:NN \l__cellprops_color_tl \g_tmpa_tl
274   \tl_set_eq:NN \l__cellprops_bgcolor_tl \g_tmpb_tl
275   \__cellprops_recall_properties:n {tr}
276   \__cellprops_orig_array:[{#1}]{#2}
277   \hbox{s[\l__cellprops_tablepadding_top_dim]}
278 }
279
280 % \end{macrocode}
281 % The color inheritance is handled with \cs{l__cellprops_inherit_color_tl},
282 % \cs{l__cellprops_color_tl} and \cs{l__cellprops_bgcolor_tl}. The role of
283 % \cs{__cellprops_update_color:Nn} is to set the inherit fallback to the already
284 % existing value of \verb|#1| then set \verb|#1| to the CSS value, which can be
285 % the inherit variable.
286 % \begin{macrocode}
287
288 \cs_new_protected_nopar:Nn \__cellprops_update_color:Nn {
289   \__cellprops_get_property:n {#2} \l_tmpc_tl
290   \exp_args:NV \tl_if_eq:NNF \l_tmpc_tl \c__cellprops_inherit_color_tl {
291     \tl_set_eq:NN #1 \l_tmpc_tl
292   }
293 }
294
295 \cs_set_protected_nopar:Nn \__cellprops_update_colors: {
296   \__cellprops_update_color:Nn \l__cellprops_color_tl {color}
297   \__cellprops_update_color:Nn \l__cellprops_bgcolor_tl {background-color}
298 }
299
300 \cs_set_eq:NN \__cellprops_orig_endarray: \endarray
301 \cs_set_nopar:Npn \endarray {
302   \crrr
303   \hbox{s[\l__cellprops_tablepadding_bottom_dim]}
304   \__cellprops_orig_endarray:
305   \tl_use:N \l__cellprops_restore_tl
306 }
307
308 \cs_set_eq:NN \__cellprops_orig_tab@startrow: \tab@startrow

```

```

309 \cs_set_nopar:Npn \tab@startrow {
310   \__cellprops_orig_tab@startrow:
311   \bool_if:NF \g__cellprops_inrow_bool {
312     \bool_gset_true:N \g__cellprops_inrow_bool
313     \int_gincr:N \g__cellprops_row_int
314     \int_gset_eq:NN \g__cellprops_col_int \c_one
315     \dim_gzero:N \g__cellprops_ht_dim
316     \dim_gzero:N \g__cellprops_dp_dim
317   }
318 }
319
320 \cs_set_eq:NN \__cellprops_orig_tab@arraycr: \tab@arraycr
321 \cs_set_protected_nopar:Npn \tab@arraycr {
322   \bool_if:NT \g__cellprops_inrow_bool {
323     \bool_gset_false:N \g__cellprops_inrow_bool
324     \__cellprops_fix_valign_end:
325   }
326   \__cellprops_orig_tab@arraycr:
327 }
328
329
330
331 \cs_set_eq:NN \__cellprops_orig_multicolumn:w \multicolumn
332 \cs_set:Npn \multicolumn#1#2#3 {
333   \__cellprops_orig_multicolumn:w {#1}{#2}{
334     #3
335     \int_gadd:Nn \g__cellprops_col_int {#1}
336   }
337 }
338
339
340 \cs_new_protected_nopar:Nn \__cellprops_fix_valign_end: {
341   \exp_args:NV \tl_if_eq:nnTF \tab@hlstate {b} {
342     \dim_gadd:Nn \tab@endheight {
343       \g__cellprops_dp_dim - \box_dp:N \@arstrutbox
344     }
345   }{
346     \int_compare:nNnT \g__cellprops_row_int = \c_one {
347       \dim_gadd:Nn \tab@endheight {
348         \g__cellprops_ht_dim - \box_ht:N \@arstrutbox
349       }
350     }
351   }
352 }
353
354 \colpush{tabular}
355
356 \coldef n{\tabcoltype{
357   \__cellprops_begincell:n{
358 }{
359   \__cellprops_endcell:
360 }}
361 \coldef l{\tabcoltype{
362   \__cellprops_begincell:n

```

```

363         {\_cellprops\_use\_setter:nn {text-align} {left}}
364     }{
365         \_cellprops\_endcell:
366     }}
367 \coldef c{\tabcoltype{
368     \_cellprops\_begin\_cell:n
369         {\_cellprops\_use\_setter:nn {text-align} {center}}
370 }{
371     \_cellprops\_endcell:
372 }}
373 \coldef r{\tabcoltype{
374     \_cellprops\_begin\_cell:n
375         {\_cellprops\_use\_setter:nn {text-align} {right}}
376 }{
377     \_cellprops\_endcell:
378 }}
379 \coldef M#1{\_cellprops\_MTcol:nn {math}{#1}}
380 \coldef T#1{\_cellprops\_MTcol:nn {text}{#1}}
381 \cs\_new\_protected\_nopar:Nn \_cellprops\_MTcol:nn {
382     % TODO: error if align not l, c, or r
383     \exp\_args:Nx \tabcoltype {
384         \exp\_not:N \_cellprops\_begin\_cell:n {
385             \exp\_not:n {\_cellprops\_use\_setter:nn {math-mode} {#1}} }
386             \exp\_not:n {\_cellprops\_use\_setter:nn {text-align}} {
387                 \str\_case:nn {#2} {
388                     {l} {left}
389                     {c} {center}
390                     {r} {right}
391                 }
392             }
393         }
394     }{
395         \_cellprops\_endcell:
396     }
397 }
398
399 \coldef p#1{\tabcoltype{
400     \_cellprops\_begin\_par\_cell:nn \vtop {#1}
401 }{
402     \_cellprops\_end\_par\_cell:n {}
403 }}
404 \coldef m#1{\tabcoltype{
405     \_cellprops\_begin\_par\_cell:nn {\c\_math\_toggle\_token\vcenter} {#1}
406 }{
407     \_cellprops\_end\_par\_cell:n{\c\_math\_toggle\_token}
408 }}
409 \coldef b#1{\tabcoltype{
410     \_cellprops\_begin\_par\_cell:nn \vbox {#1}
411 }{
412     \_cellprops\_end\_par\_cell:n {}
413 }}
414
415
416 \colpop

```

```

417
418 % \end{macrocode}
419 % Handle various \verb|:nth-child()| forms.
420 % \begin{macrocode}
421
422 \cs_new_protected_nopar:Nn \__cellprops_nthchild_anb:Nnn {
423   \tl_set:Nx \l_tmpa_tl { +\int_eval:n{\int_mod:nn{#2}{#3}} }
424   \exp_args:NV \tl_if_eq:nnT \l_tmpa_tl { +0 } {
425     \tl_clear:N \l_tmpa_tl
426   }
427   \seq_put_right:Nx #1 { #3n\l_tmpa_tl }
428 }
429
430 \cs_new_protected_nopar:Nn \__cellprops_seq_nthchild:Nn {
431   \seq_clear:N #1
432   \seq_put_right:Nx #1 {
433     \int_compare:nNnTF{\int_mod:nn{#2}{2}}={0}{even}{odd}
434   }
435   \__cellprops_nthchild_anb:Nnn #1 {#2} {2}
436   \__cellprops_nthchild_anb:Nnn #1 {#2} {3}
437   \seq_put_right:Nx #1 { \int_eval:n{#2} }
438 }
439
440 \cs_new_protected_nopar:Nn \__cellprops_begincell:n {
441   \__cellprops_begin_raw_cell:n {
442     \__cellprops_seq_nthchild:Nn \l_tmpa_seq { \g__cellprops_row_int }
443     \__cellprops_seq_nthchild:Nn \l_tmpb_seq { \g__cellprops_col_int }
444     \seq_map_inline:Nn \l_tmpa_seq {
445       \__cellprops_recall_properties:n {tr:nth-child(##1)}
446     }
447     \__cellprops_update_colors:
448     \__cellprops_recall_properties:n {td}
449     \__cellprops_recall_properties:n {tr~td}
450     \seq_map_inline:Nn \l_tmpb_seq {
451       \__cellprops_recall_properties:n {td:nth-child(##1)}
452     }
453     \seq_map_inline:Nn \l_tmpa_seq {
454       \__cellprops_recall_properties:n {tr:nth-child(##1)~td}
455     }
456     \seq_map_inline:Nn \l_tmpa_seq {
457       \seq_map_inline:Nn \l_tmpb_seq {
458         \__cellprops_recall_properties:n {tr:nth-child(##1)~
459           td:nth-child(###1)}
460       }
461     }
462     \__cellprops_update_colors:
463     #1
464     \hbox_set:Nw \l__cellprops_cell_box
465     \str_case_x:nnF {\__cellprops_get_property:n {math-mode}} {
466       { text } { \tab@btext }
467       { math } { \tab@bmaths }
468     }{% any other treated as |auto|
469     \tab@bgroup
470   }

```

```

471     }
472 }
473
474 \cs_new_protected_nopar:Nn \__cellprops_endcell: {
475   \str_case:x:nnF {\__cellprops_get_property:n {math-mode}} {
476     { text } { \tab@etext }
477     { math } { \tab@emaths }
478   }{% any other treated as |auto|
479     \tab@egroup
480   }
481   \hbox_set_end:
482   \__cellprops_end_raw_cell:
483 }
484
485 \cs_new_protected_nopar:Nn \__cellprops_begin_par_cell:nn {
486   \savenotes
487   \__cellprops_begin_raw_cell:n{
488     \hbox_set:Nw \l__cellprops_cell_box
489     #1
490     \bgroup
491     \hsize#2\relax
492     \@arrayparboxrestore
493     \global\@minipagetrue
494     \everypar{
495       \global\@minipagefalse
496       \everypar{}}
497   }
498   \__cellprops_recall_properties:n {td~p}
499 }
500 }
501 \cs_new_protected_nopar:Nn \__cellprops_end_par_cell:n {
502   \ifhmode\@maybe@unskip\par\fi
503   \unskip
504   \egroup
505   #1
506   \hbox_set_end:
507   \__cellprops_end_raw_cell:
508   \spewnotes\hfil
509 }
510
511 \cs_new_protected_nopar:Nn \__cellprops_begin_raw_cell:n {
512   \group_begin:
513   % Additional init code
514   #1
515   % Install the cell color
516   \__cellprops_update_colors:
517   \tl_use:N \l__cellprops_color_tl
518 }
519
520 \cs_new_protected_nopar:Nn \__cellprops_end_raw_cell: {
521   % Here \l__cellprops_cell_box must contain the contents of the cell
522   %
523   % Handle vpadding and min-height/depth
524   \box_set_ht:Nn \l__cellprops_cell_box {

```

```

525         \dim_max:nn
526         {\box_ht:N \l__cellprops_cell_box}
527         {\__cellprops_get_property:n {min-height}}
528         + (\__cellprops_get_property:n {padding-top})
529     }
530     \box_set_dp:Nn \l__cellprops_cell_box {
531         \dim_max:nn
532         {\box_dp:N \l__cellprops_cell_box}
533         {\__cellprops_get_property:n {min-depth}}
534         + (\__cellprops_get_property:n {padding-bottom})
535     }
536     % To fix vertical alignment later
537     \dim_gset:Nn \g__cellprops_ht_dim {
538         \dim_max:nn
539         {\g__cellprops_ht_dim}
540         {\box_ht:N \l__cellprops_cell_box}
541     }
542     \dim_gset:Nn \g__cellprops_dp_dim {
543         \dim_max:nn
544         {\g__cellprops_dp_dim}
545         {\box_dp:N \l__cellprops_cell_box}
546     }
547     % Handle hpadding and halign
548     \skip_set:Nn \l_tmpa_skip {
549         \dim_max:nn
550         {Opt}
551         { (\__cellprops_get_property:n {min-width})
552           - \box_wd:N \l__cellprops_cell_box }
553     }
554     \skip_add:Nn \l_tmpa_skip {
555         Opt plus 1fil
556     }
557     \skip_set:Nn \l__cellprops_left_skip
558         {\__cellprops_get_property:n {padding-left}}
559     \skip_set:Nn \l__cellprops_right_skip
560         {\__cellprops_get_property:n {padding-right}}
561     \str_case_x:nnF {\__cellprops_get_property:n {text-align}} {
562         { right } {
563             \skip_add:Nn \l__cellprops_left_skip { \l_tmpa_skip }
564         }
565         { center } {
566             \skip_add:Nn \l__cellprops_left_skip { \l_tmpa_skip / 2 }
567             \skip_add:Nn \l__cellprops_right_skip { \l_tmpa_skip / 2 }
568         }
569     }{% any other treated as |left|
570         \skip_add:Nn \l__cellprops_right_skip { \l_tmpa_skip }
571     }
572     \kern\c_zero_dim
573     \tl_if_empty:NF \l__cellprops_bgcolor_tl {
574         \group_begin:
575         % Paint a background with leaders
576         \tl_use:N \l__cellprops_bgcolor_tl % install the color
577         \skip_set:Nn \l_tmpa_skip {
578             \l__cellprops_left_skip

```

```

579         + \box_wd:N \l__cellprops_cell_box
580         + \l__cellprops_right_skip
581     }
582     \leaders
583         \vrule
584         \skip_horizontal:N \l_tmpa_skip
585         \skip_horizontal:n {-\l_tmpa_skip}
586     \group_end:
587 }
588 \skip_horizontal:N \l__cellprops_left_skip
589 \box_use_clear:N \l__cellprops_cell_box
590 \skip_horizontal:N \l__cellprops_right_skip
591 \kern\c_zero_dim
592 \group_end:
593 }
594 \end{package}

```