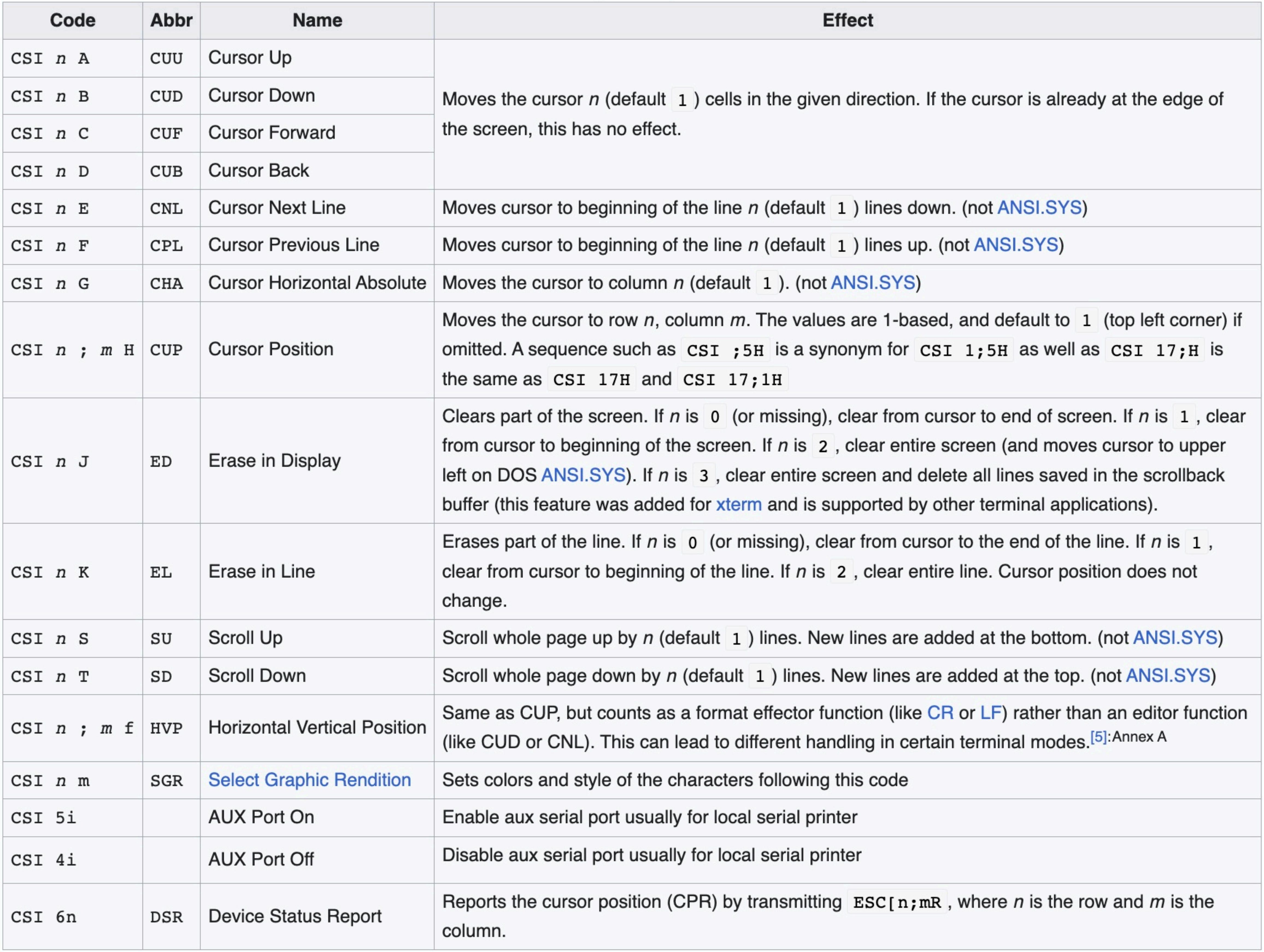
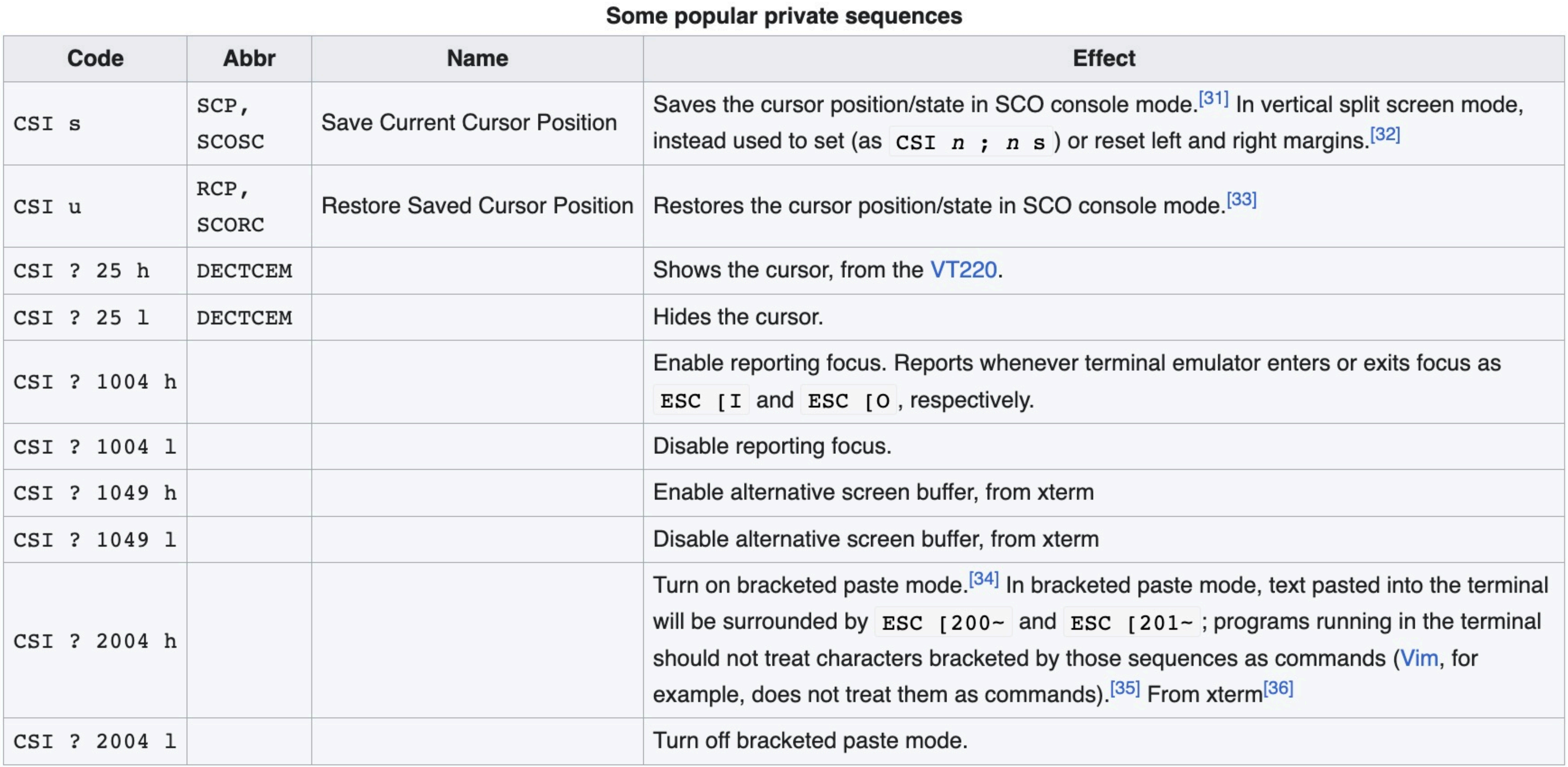
Some ANSI control sequences (not an exhaustive list)

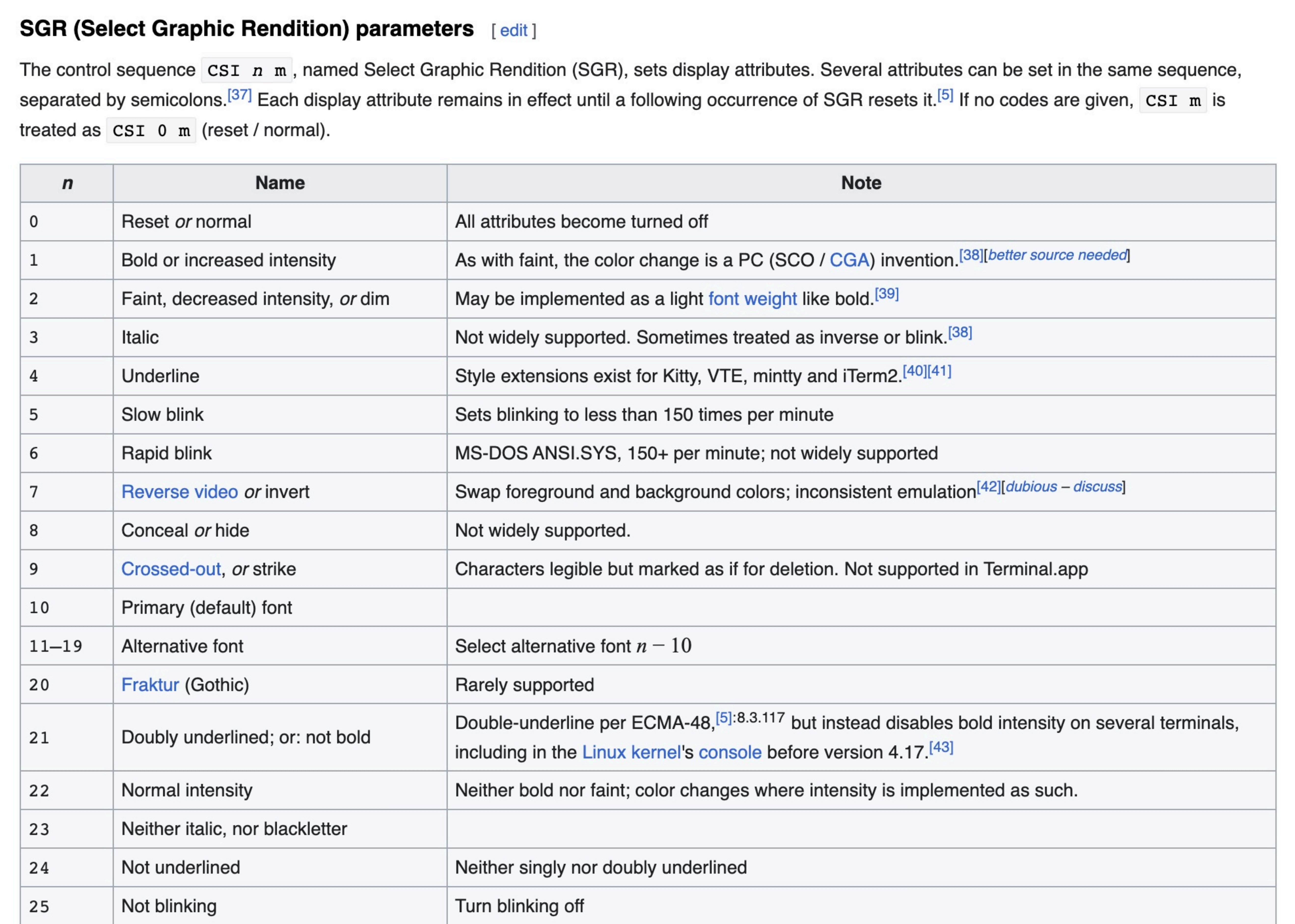


Code	Abbr	Name	Effect
`CSI n A`	`CUU`	Cursor Up	Moves the cursor *n* (default `1`) cells in the given direction. If the cursor is already at the edge of the screen, this has no effect.
`CSI n B`	`CUD`	Cursor Down	
`CSI n C`	`CUF`	Cursor Forward	
`CSI n D`	`CUB`	Cursor Back	
`CSI n E`	`CNL`	Cursor Next Line	Moves cursor to beginning of the line *n* (default `1`) lines down. (not ANSI.SYS)
`CSI n F`	`CPL`	Cursor Previous Line	Moves cursor to beginning of the line *n* (default `1`) lines up. (not ANSI.SYS)
`CSI n G`	`CHA`	Cursor Horizontal Absolute	Moves the cursor to column *n* (default `1`). (not ANSI.SYS)
`CSI n ; m H`	`CUP`	Cursor Position	Moves the cursor to row *n*, column *m*. The values are 1-based, and default to `1` (top left corner) if omitted. A sequence such as `CSI ;5H` is a synonym for `CSI 1;5H` as well as `CSI 17;H` is the same as `CSI 17H` and `CSI 17;1H`
`CSI n J`	`ED`	Erase in Display	Clears part of the screen. If *n* is `0` (or missing), clear from cursor to end of screen. If *n* is `1`, clear from cursor to beginning of the screen. If *n* is `2`, clear entire screen (and moves cursor to upper left on DOS ANSI.SYS). If *n* is `3`, clear entire screen and delete all lines saved in the scrollback buffer (this feature was added for xterm and is supported by other terminal applications).
`CSI n K`	`EL`	Erase in Line	Erases part of the line. If *n* is `0` (or missing), clear from cursor to the end of the line. If *n* is `1`, clear from cursor to beginning of the line. If *n* is `2`, clear entire line. Cursor position does not change.
`CSI n S`	`SU`	Scroll Up	Scroll whole page up by *n* (default `1`) lines. New lines are added at the bottom. (not ANSI.SYS)
`CSI n T`	`SD`	Scroll Down	Scroll whole page down by *n* (default `1`) lines. New lines are added at the top. (not ANSI.SYS)
`CSI n ; m f`	`HVP`	Horizontal Vertical Position	Same as CUP, but counts as a format effector function (like CR or LF) rather than an editor function (like CUD or CNL). This can lead to different handling in certain terminal modes.[5]:Annex A
`CSI n m`	`SGR`	Select Graphic Rendition	Sets colors and style of the characters following this code
`CSI 5i`		AUX Port On	Enable aux serial port usually for local serial printer
`CSI 4i`		AUX Port Off	Disable aux serial port usually for local serial printer
`CSI 6n`	`DSR`	Device Status Report	Reports the cursor position (CPR) by transmitting `ESC[n;mR`, where *n* is the row and *m* is the column.



Some popular private sequences

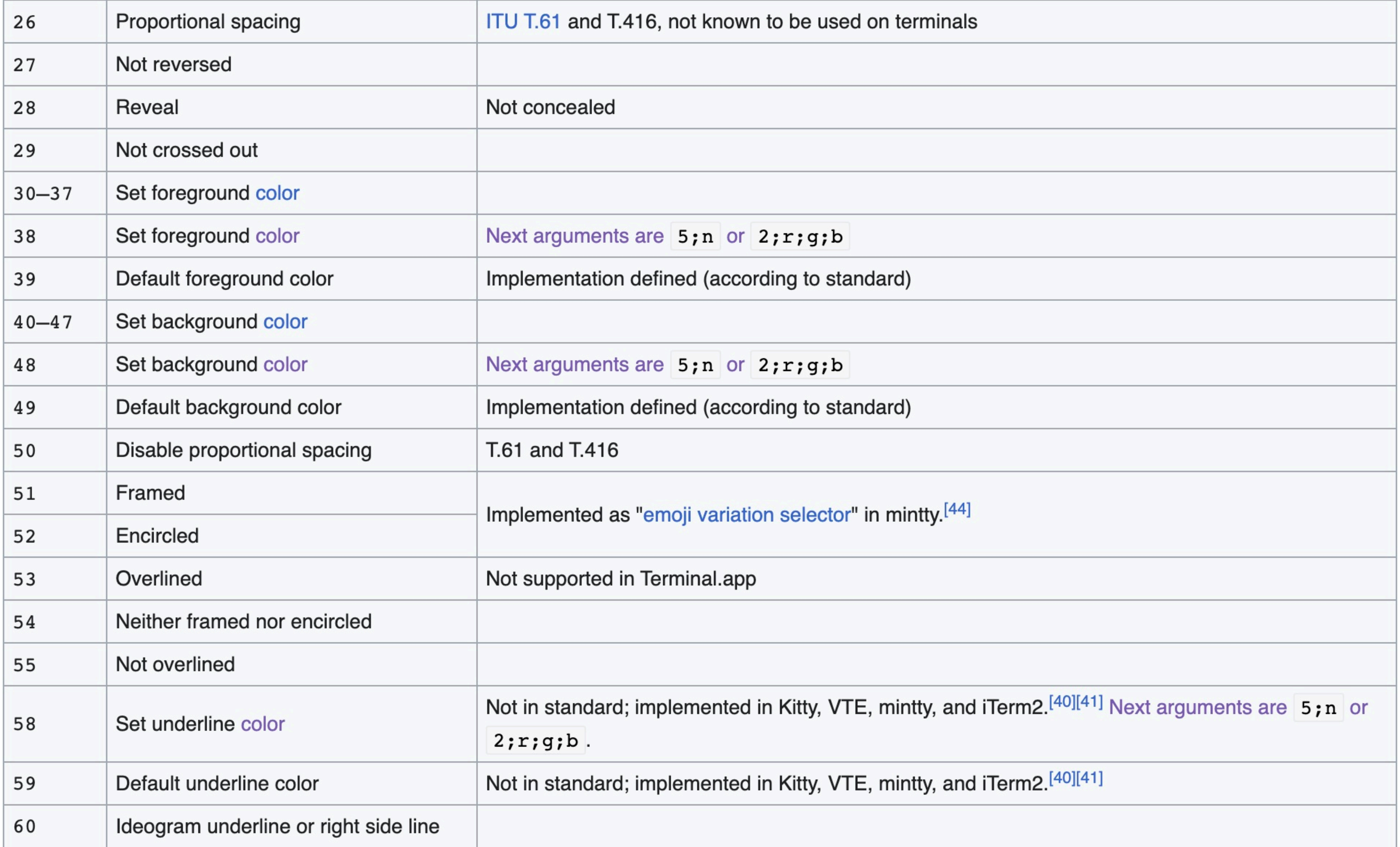
Code	Abbr	Name	Effect
`CSI s`	`SCP`, `SCOSC`	Save Current Cursor Position	Saves the cursor position/state in SCO console mode.[31] In vertical split screen mode, instead used to set (as `CSI n ; n s`) or reset left and right margins.[32]
`CSI u`	`RCP`, `SCORC`	Restore Saved Cursor Position	Restores the cursor position/state in SCO console mode.[33]
`CSI ? 25 h`	`DECTCEM`		Shows the cursor, from the VT220.
`CSI ? 25 l`	`DECTCEM`		Hides the cursor.
`CSI ? 1004 h`			Enable reporting focus. Reports whenever terminal emulator enters or exits focus as `ESC [I` and `ESC [O`, respectively.
`CSI ? 1004 l`			Disable reporting focus.
`CSI ? 1049 h`			Enable alternative screen buffer, from xterm
`CSI ? 1049 l`			Disable alternative screen buffer, from xterm
`CSI ? 2004 h`			Turn on bracketed paste mode.[34] In bracketed paste mode, text pasted into the terminal will be surrounded by `ESC [200~` and `ESC [201~`; programs running in the terminal should not treat characters bracketed by those sequences as commands (Vim, for example, does not treat them as commands).[35] From xterm[36]
`CSI ? 2004 l`			Turn off bracketed paste mode.



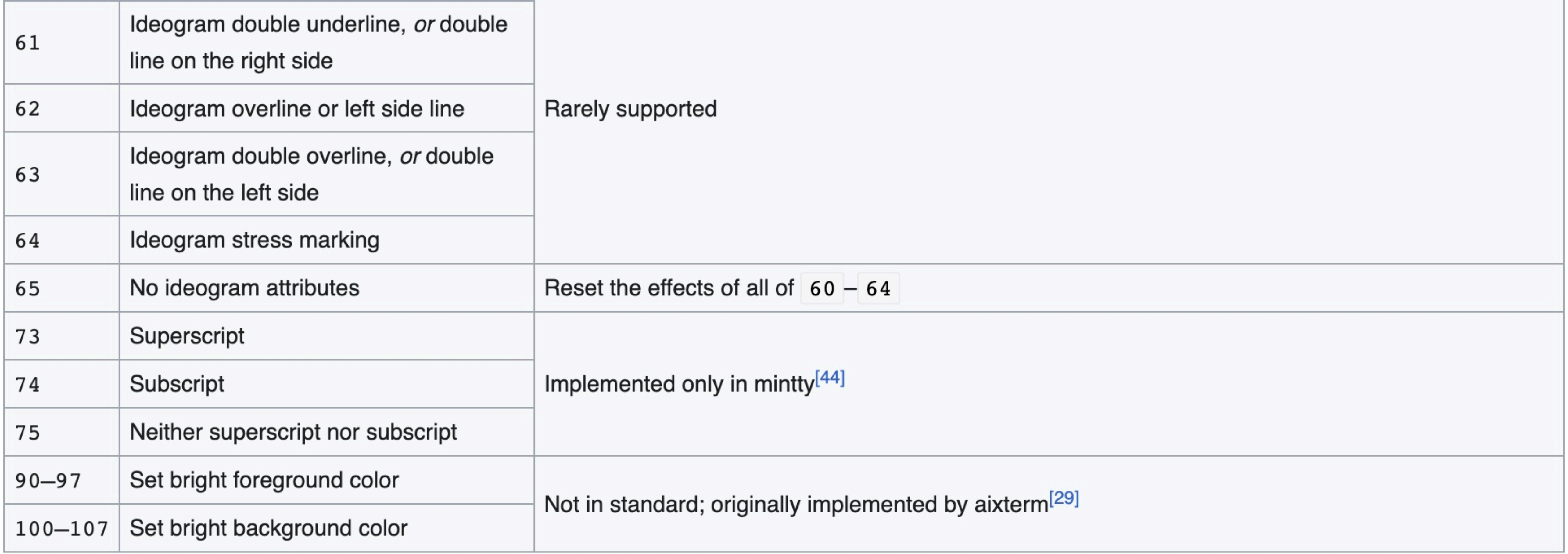
SGR (Select Graphic Rendition) parameters [edit]

The control sequence `CSI n m`, named Select Graphic Rendition (SGR), sets display attributes. Several attributes can be set in the same sequence, separated by semicolons.[37] Each display attribute remains in effect until a following occurrence of SGR resets it.[5] If no codes are given, `CSI m` is treated as `CSI 0 m` (reset / normal).

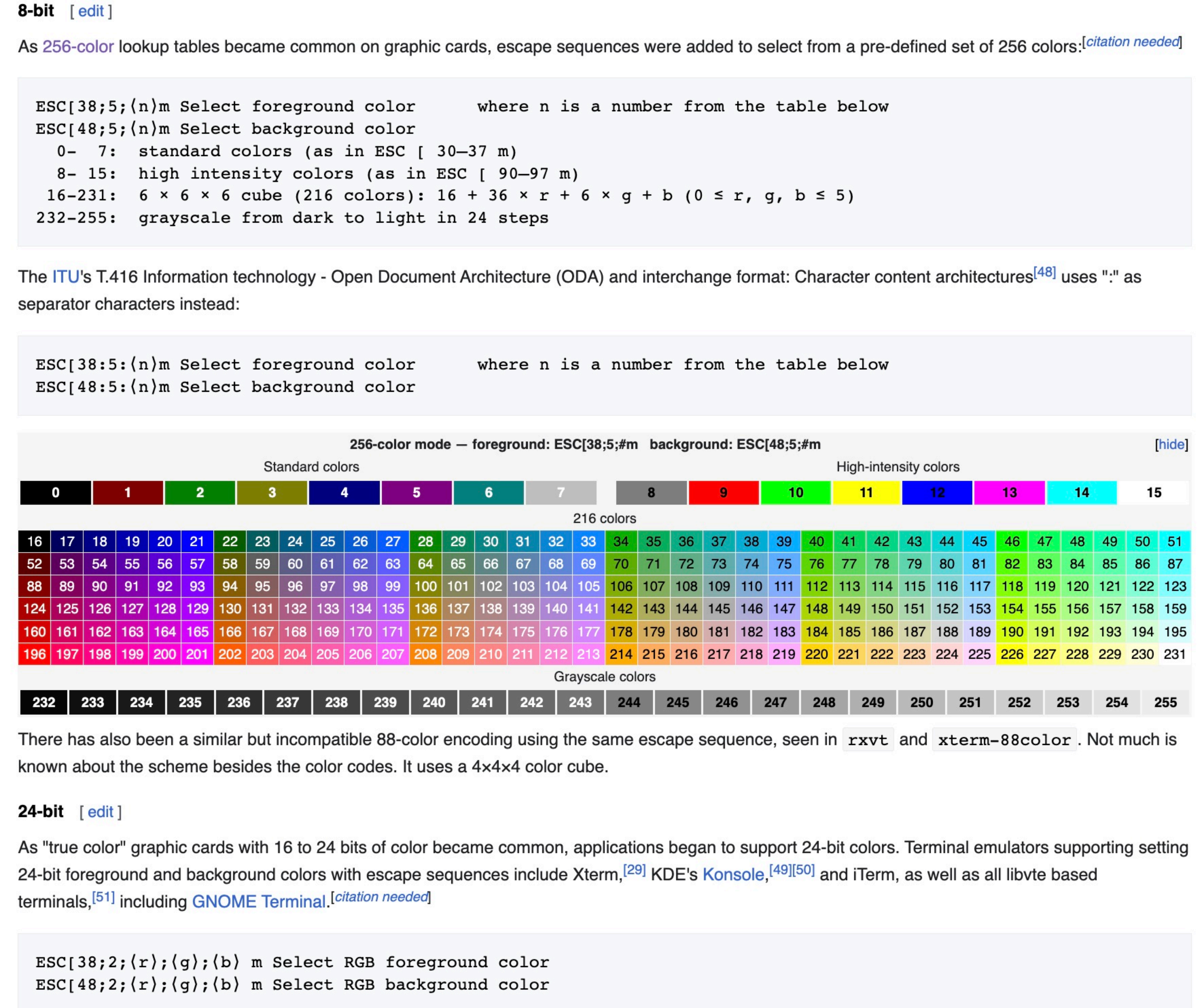
n	Name	Note
0	Reset *or* normal	All attributes become turned off
1	Bold or increased intensity	As with faint, the color change is a PC (SCO / CGA) invention.[38][*better source needed*]
2	Faint, decreased intensity, *or* dim	May be implemented as a light font weight like bold.[39]
3	Italic	Not widely supported. Sometimes treated as inverse or blink.[38]
4	Underline	Style extensions exist for Kitty, VTE, mintty and iTerm2.[40][41]
5	Slow blink	Sets blinking to less than 150 times per minute
6	Rapid blink	MS-DOS ANSI.SYS, 150+ per minute; not widely supported
7	Reverse video *or* invert	Swap foreground and background colors; inconsistent emulation[42][*dubious – discuss*]
8	Conceal *or* hide	Not widely supported.
9	Crossed-out, *or* strike	Characters legible but marked as if for deletion. Not supported in Terminal.app
10	Primary (default) font	
11–19	Alternative font	Select alternative font $n - 10$
20	Fraktur (Gothic)	Rarely supported
21	Doubly underlined; or: not bold	Double-underline per ECMA-48,[5]:8.3.117 but instead disables bold intensity on several terminals, including in the Linux kernel's console before version 4.17.[43]
22	Normal intensity	Neither bold nor faint; color changes where intensity is implemented as such.
23	Neither italic, nor blackletter	
24	Not underlined	Neither singly nor doubly underlined
25	Not blinking	Turn blinking off



26	Proportional spacing	ITU T.61 and T.416, not known to be used on terminals
27	Not reversed	
28	Reveal	Not concealed
29	Not crossed out	
30–37	Set foreground color	
38	Set foreground color	Next arguments are `5;n` or `2;r;g;b`
39	Default foreground color	Implementation defined (according to standard)
40–47	Set background color	
48	Set background color	Next arguments are `5;n` or `2;r;g;b`
49	Default background color	Implementation defined (according to standard)
50	Disable proportional spacing	T.61 and T.416
51	Framed	Implemented as "emoji variation selector" in mintty.[44]
52	Encircled	
53	Overlined	Not supported in Terminal.app
54	Neither framed nor encircled	
55	Not overlined	
58	Set underline color	Not in standard; implemented in Kitty, VTE, mintty, and iTerm2.[40][41] Next arguments are `5;n` or `2;r;g;b`.
59	Default underline color	Not in standard; implemented in Kitty, VTE, mintty, and iTerm2.[40][41]
60	Ideogram underline or right side line	



61	Ideogram double underline, *or* double line on the right side	Rarely supported
62	Ideogram overline or left side line	
63	Ideogram double overline, *or* double line on the left side	
64	Ideogram stress marking	
65	No ideogram attributes	Reset the effects of all of 60 – 64
73	Superscript	Implemented only in mintty[44]
74	Subscript	
75	Neither superscript nor subscript	
90–97	Set bright foreground color	Not in standard; originally implemented by aixterm[29]
100–107	Set bright background color	



8-bit [edit]

As 256-color lookup tables became common on graphic cards, escape sequences were added to select from a pre-defined set of 256 colors:[citation needed]

```
ESC[38;5;⟨n⟩m Select foreground color      where n is a number from the table below
ESC[48;5;⟨n⟩m Select background color
  0-  7:  standard colors (as in ESC [ 30–37 m)
  8- 15:  high intensity colors (as in ESC [ 90–97 m)
 16-231:  6 × 6 × 6 cube (216 colors): 16 + 36 × r + 6 × g + b (0 ≤ r, g, b ≤ 5)
232-255:  grayscale from dark to light in 24 steps
```

The ITU's T.416 Information technology - Open Document Architecture (ODA) and interchange format: Character content architectures[48] uses ":" as separator characters instead:

```
ESC[38:5:⟨n⟩m Select foreground color      where n is a number from the table below
ESC[48:5:⟨n⟩m Select background color
```

256-color mode — foreground: ESC[38;5;#m   background: ESC[48;5;#m [hide]

Standard colors								High-intensity colors							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

216 colors

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87
88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123
124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195
196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231

Grayscale colors

232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

There has also been a similar but incompatible 88-color encoding using the same escape sequence, seen in `rxvt` and `xterm-88color`. Not much is known about the scheme besides the color codes. It uses a 4×4×4 color cube.

24-bit [edit]

As "true color" graphic cards with 16 to 24 bits of color became common, applications began to support 24-bit colors. Terminal emulators supporting setting 24-bit foreground and background colors with escape sequences include Xterm,[29] KDE's Konsole,[49][50] and iTerm, as well as all libvte based terminals,[51] including GNOME Terminal.[citation needed]

```
ESC[38;2;⟨r⟩;⟨g⟩;⟨b⟩ m Select RGB foreground color
ESC[48;2;⟨r⟩;⟨g⟩;⟨b⟩ m Select RGB background color
```