

```

1  '''
2      Costants and Utility Function used by Cell and Matrix
3  '''
4
5  from sys import version_info
6
7  # Flag Constants
8  BTFLAG_UP = 1          # Backtrace Up Flag
9  BTFLAG_LEFT = 2        # Backtrace Left Flag
10 BTFLAG_DIAG = 4        # Backtrace Diagonal Flag
11
12 # unicode characters - need Python 3!
13 if version_info.major == 3:
14     BTMATRIX_SYMBOLS = [
15         " ",          # 0 - No flags were set
16         "\u2191",      # 1 - Up flag set
17         "\u2190",      # 2 - Left flag set
18         "\u2190\u2191", # 3 - Up flag and Left flag set
19         "\u2196",      # 4 - Diagonal flag set
20         "\u2196\u2191", # 5 - Diagonal flag and Up flag set
21         "\u2196\u2190", # 6 - Diagonal flag and Left flag set
22         "\u2196\u2190\u2191", # 7 - All flags are set
23     ]
24 else:
25     BTMATRIX_SYMBOLS = [
26         " ",          # 0 - No flags were set
27         "U",          # 1 - Up flag set
28         "L",          # 2 - Left flag set
29         "U",          # 3 - Up flag and Left flag set
30         "G",          # 4 - Diagonal flag set
31         "G",          # 5 - Diagonal flag and Up flag set
32         "G",          # 6 - Diagonal flag and Left flag set
33         "G",          # 7 - All flags are set
34     ]
35
36 # Spacing constant
37 FORMAT_SPACING = 4
38
39 def CellFormat(value):
40     '''
41     Returns a right justified string based on spacing constant.
42     Returns a str()
43     -Returns the string " ###..." if the value passed to large to display
44     with the current spacing
45     '''
46     value_length = len(str(value))
47     if value_length >= FORMAT_SPACING:
48         return " " + "#" * (FORMAT_SPACING - 1)
49     else:
50         return " " * (FORMAT_SPACING - value_length) + str(value)

```