E:\di1.py

```
class HTML:

    def __init__(self, Header, tableStyles = {}, trStyles = {}, thStyles = {}):
        self.tableStyles = HTML._styleConverter(tableStyles)
        trStyles = HTML._styleConverter(trStyles)
        thStyles = HTML._styleConverter(thStyles)
        self.rows = []
        self.Header= f'<tr {trStyles} >'
        for th in Header:
            self.Header += f'\n<th {thStyles} >{th}</th>'
        self.Header += '\n</tr>'

    @staticmethod
    def _styleConverter(styleDict : dict):
        if styleDict == {}:
            return ''
        styles = ''
        for [style, value] in styleDict.items():
            styles +=f'{style}: {value};'
        return f'style="{styles}"'

    def addRow(self, row, trStyles = {}, tdStyles = {}):
        trStyles = HTML._styleConverter(trStyles)
        tdStyles = HTML._styleConverter(tdStyles)
        temp_row = f'\n<tr {trStyles} >'
        for td in row:
            temp_row += f'\n<td {tdStyles} >{td}</td>'
        temp_row += '\n</tr>'
        self.rows.append(temp_row)


    def __str__(self):


        return \
f'''
<table {self.tableStyles} >
{self.Header}
{''.join(self.rows)}
</table>
'''



def dictionaryToHTMLTable(dict : dict):
    html = HTML(Header = dict.keys(),
                tableStyles={'margin': '3px'},
                trStyles={'background-color': '#7cc3a97d'},
                thStyles={ 'color': 'white'})
    for i, row in enumerate(zip(*dict.values())):
        print(row)
        if i%2 == 0:
            BGC = 'aliceblue'
        else:
```

```
            BGC = '#c2d4e4'
        html.addRow(row, trStyles={'background-color' : BGC}, tdStyles={'padding': '1rem'})
    return html
```