constants

April 9, 2022

This file describes the constants used in the project.

[1]:
```
#Define the required constants
class Constants:
    '''
    An instance object of the Constant class can:
    1. allow access to the defined constants and display them

    SUGGESTIONS FOR IMPROVEMENTS
    1.
    '''



    def __init__(self):
        self.constants = {
                #'symbol': ['value', unit', 'digits', '10 ^ power', 'number of digits', 'description'],
                'e' : [1.602176634 * (10**(-19)), 'C', 1.602176634, -19, 10, \
                       'The charge of an electron, or the elementary charge'],
                'm_e': [9.109383701 * (10**(-31)), 'kg', 9.109383701, -31, 10, \
                        'The mass of an electron'],
                'amu': [1.6605390666 * (10**(-27)), 'kg', 1.6605390666, -27, 11, \
                        'The unified atomic mass unit or Dalton'],
                'N_A': [6.02214076 * (10**(23)), 'mol', 6.02214076, 23, 9, \
                        'Avogadro\'s number'],
                'epsilon_0': [8.854187812 * (10**(-12)), 'F/m', 8.854187812, -12, 10, \
                              'Permittivity of vacuum'],
                'mu_0': [1.2566370621 * (10**(-6)), 'H/m', 1.2566370621, -6, 11, \
                         'Permeability of vaccum'],
                'K': [1.380649 * (10**(-23)), 'J/K', 1.380649, -23, 7, 'The Boltzmann constant'],
                'R': [8.31446261815324, 'J/Kmol', 8.31446261815324, 0, 15, 'Universal gas constant'],
```

```
                'm_H': [1.008, 'amu', 1.008, 0, 4, 'The relative atomic mass of␣
↪Hydrogen atom in amu or g/mol']
                #Define other constants like mass and charge of other ions here
                    }
    def show_constant(self, symbol):
        '''
        Print a constant if it is defined in the dictionary constants.

        Arguments:
        symbol: string representing the defined symbol of the constant

        Returns:
        nothing
        '''



        if symbol in constants.keys():
            (print(f'{symbol} = {constants[symbol][1]} x␣
↪10^({constants[symbol][2]}) {constants[symbol][0]} \
                    \n {constants[symbol][3]} significant figures \
                    \n {constants[symbol][4]}'))
        else:
            print('Constant not in record or symbol mismatch.')
```

[1]:
```
len('8.31446261815324')
```

[1]:
```
16
```

[ ]:
```

```