

Python Basics - Experiment 1

Aim – Write a menu-Driven text Applications to solve five problems as a menu-driven textbased application. It presents the user with a set of choices

(1) Sum of input numbers

```
In [2]: a=2  
b=4  
c=a+b  
print(c)
```

6

2) Average of input numbers

```
In [3]: f=a+b/2  
print(f)
```

4.0

3) Mean of input numbers,

```
In [4]: import statistics  
data = [1, 2, 3, 4]  
x = statistics.mean(data)  
print("Mean is :", x)
```

Mean is : 2.5

4) Median of input numbers

```
In [5]: import statistics
data = [1, 2, 3, 4]
x = statistics.median(data)
print("Median is :", x)
```

Median is : 2.5

5) Mode of input numbers

```
In [7]: import statistics
data = [1,1, 2, 3, 4]
x = statistics.mode(data)
print("Mode is :", x)
```

Mode is : 1

6) Quit

```
In [2]: import sys
sys.exit
```

Out[2]: <function sys.exit>

Edmodo - file_io_exercise

2. Reading first word from each line of a file

Implement `find_first_words` function which takes an input file path as argument. The function should find the first word of each line in the file and return these words as a list. If a line is empty, the returned list should contain an empty string for that line.

```
In [19]: import os
DATA_DIR = r'C:\Users\Hamza'
```

```

WordList=[]
def find_first_words(file):

    with open(file,mode='r') as x:
        for word in x:
            word=word.strip()
            WordList.append(word.split(' ',1)[0])
    return WordList

in_file1 = os.path.join(DATA_DIR, 'simple_file.txt')
in_file2 = os.path.join(DATA_DIR, 'simple_file_with_empty_lines1.txt')

# expected_file_1 = ['First', 'Second', 'Third', 'And']
# assert find_first_words(in_file1) == expected_file_1

expected_file_2 = ['The', '', 'First', 'Funny', '', 'Then']
assert find_first_words(in_file2) == expected_file_2

['The', '', 'First', 'Funny', '', 'Then']

```

1. Sum numbers listed in a file

1) Fill __ pieces of the code below. `sum_numbers_in_file` function takes a input file path as argument, reads the numbers listed in the input file and returns the sum of those numbers. You can assume that each line contains exactly one numeric value.

```

In [10]: import os
DATA_DIR = r'C:\Users\Hamza'
def sum_numbers_in_file(input_file):
    sum_ = 0 # A common way to use variable names that collide with built-in/keyword words is to add underscore
    with open(input_file, mode='r') as x:
        for line in x:
            x = line.strip() # Remove potential white space
            sum_ += float(line)

```

```
    return sum_  
  
in_file = os.path.join(DATA_DIR, 'numbers.txt')  
assert sum_numbers_in_file(in_file) == 189.5
```

Edmodo Numbers_exercise

1. Creating formulas

Write the following mathematical formula in Python:

$$result = 6a^3 - \frac{8b^2}{4c} + 11$$

```
In [52]: a = 2  
b = 3  
c = 2  
result = (6*(a)**3) - (8*((b)**2))/(4*c) + 11  
if result == 50:  
    print(True)
```

True

2. Floating point pitfalls

Show that `0.1 + 0.2 == 0.3`

```
In [101]: q=0.1  
r=0.2  
sum = q+r  
sum=round(sum,2)  
if sum == 0.3:  
    print(True)
```

True

In []: