The University Of Sheffield.

The University of Sheffield CiCS Research IT

PRACTISE SESSION 3 SOLUTIONS

Define a function, shut_down(), that takes string 's' as an argument. If the shut_down() function receives the string "yes", then it should return "Shutting down". If the string 's' is equal to "no", then the function should return "Shutdown aborted". Finally, if shut_down() gets anything other than yes/no, the function should return "Unknown parameter".

```
def shut_down(s):
   if s == 'yes':
      return 'Shutting down'
   elif s == 'no':
      return 'Shutdown aborted'
   else:
      return 'Sorry'

answer = input('Shut down the system? ')
msg = shut_down(answer)
print(msg)
```

Define a function called distance_from_zero(), with one argument. If the type of the argument is either int or float, the function should return the absolute value of the input variable. Otherwise, the function returns "Error". Call the function with -5, -5.6 and "-5.6".

a) Write distance_from_zero() (use if and return statements).
b) Use the input() function.
c) Call the function distance_from_zero()
d) Display the result.

How can you check if an input is a number or a string? Google it.

Step 1

```
def distance_from_zero(num):
  # Phythonic way, we use this
  return abs(num)

def distance_from_zero_2(num):
  # Other way
  if num >= 0:
    return num
  elif num < 0:
    return -1 * num

# Test number: -4.3
res = distance_from_zero(-4.3)
print(res)
```

Step II

```
def is_number(s):
    try:
        float(s)
        return True
    except ValueError:
        return False

def distance_from_zero(num):
  if is_number(num) == True:
    return abs(float(num))
  else:
    return 'Nope'

number = input('Number? ')
res = distance_from_zero(number)
print(res)
```

The University Of Sheffield.

The University of Sheffield CiCS Research IT

PRACTISE SESSION 4 SOLUTIONS

1) Generate a random number between 1 and 9. Ask the user to guess the number, then tell them whether they guessed too low, too high, or exactly right. The user has only 3 attempts. For this task use a for loop, loop breaking, input, if condition, random module and function. This is a complex task, design your code before you start to write it.

```
import random

def guess_game(guess, rnd):
  if guess == rnd:
    return True
  else:
    return False


a = random.randint(1, 9)

for i in range(3):
  guess = input('? ')
  if guess_game(int(guess), a) == True:
    print('You win!')
    break

if i == 2:
  print('You lost!')
```

The University Of Sheffield.

The University of Sheffield CiCS Research IT

MODULES

- A module allows you to logically organise your Python code.

- Grouping related code into a module makes the code easier to understand and use.

- Simply, a module is a file consisting of Python code.

Step 1 - write your function(s)

```
def area(r):
    return 3.14 * (r ** 2)

def circumference(r):
    return 2 * 3.14 * r
```

Save your code (circle.py) and start a new session.

Step 1 - import your custom functions.

```
# Now you can use the functions from circle.py
import circle

r=5

print(circle.area(r))
print(circle.circumference(r))
```

Important! This file and circle.py have to be in the same folder.

Tip 1 - Make your code simpler

```
# Now you can use the functions from circle.py
import circle as c

r=5

print(c.area(r))
print(c.circumference(r))
```

Important! This file and circle.py have to be in the same folder.

Tip 11 - Import only one function

```
# Now you can use the functions from circle.py
from circle import area

r=5

print(area(r))
```

Important! This file and circle.py have to be in the same folder.

The University Of Sheffield.

The University of Sheffield CiCS Research IT

Write a Quadratic Equation Solver module.

Input: a = 1, b = 9, c = 6

Solution: -8.27 and -0.72

For more information visit:

https://www.mathsisfun.com/quadratic-equation-solver.html

The University Of Sheffield.

The University of Sheffield CiCS Research IT

Module file: quadratic.py

```
def solver(a,b,c):
  sol1 = ((-1*b) - (b**2 - 4*a*c)**(1/2)) / (2*a)
  sol2 = ((-1*b) + (b**2 - 4*a*c)**(1/2)) / (2*a)
  return sol1, sol2
```

Main file.

```
import quadratic

print(quadratic.solver(1, 9, 6))
```

The University Of Sheffield.

The University of Sheffield CiCS Research IT

THIRD PARTY MODULES

You have learned how to write a new module. However, you should always check the existing third party modules. Your task may be already implemented in someone else's module.

Anaconda Python contains numerous third-party modules. See the complete list here: https://docs.continuum.io/anaconda/pkg-docs

A short list of popular packages:

- Pandas: easy-to-use data structures and data analysis tools
- NumPy: fundamental package for scientific computing
- SciPy: scientific computing tools
- math (inbuilt) : basic math library

Solve the quadratic equation again.

Step 1: Understand the problem.

Step 2: Check existing modules.

Type into Google: python quadratic solver

Did you find anything?

https://docs.scipy.org/doc/numpy-1.10.0/reference/generated/numpy.roots.html

Solve the quadratic equation again.

Step 3: Check the documentation

```
import numpy as np

print(np.roots([1, 9, 6]))
```

The University Of Sheffield.

The University of Sheffield CiCS Research IT

Numpy is able to solve your problem in two lines.

```
import numpy # Existing library
import quadratic # your solution


print(quadratic.solver(1, 9, 6))
print(numpy.roots([1, 9, 6]))
```

You had to write your custom model before.

Remember?

```
def solver(a,b,c):
 sol1 = ((-1*b) - (b**2 - 4*a*c)**(1/2)) / (2*a)
 sol2 = ((-1*b) + (b**2 - 4*a*c)**(1/2)) / (2*a)
 return sol1, sol2
```

Numpy provides a 2 lines solution vs your solution which takes 6 lines. You wasted 4 lines.

Do not reinvent the wheel !

Your input is:

[4,6,2,3,4,6,8,9,3,4,5,6,7]

Calculate the average, variance and standard deviation. Use Numpy.

```
import numpy as np

a = [4,6,2,3,4,6,8,9,3,4,5,6,7]


print(np.mean(a))
print(np.var(a))
print(np.std(a))
```

How many lines did you save here?

The University Of Sheffield.

The University of Sheffield CiCS Research IT

PRACTICE SESSION 5

The University Of Sheffield.

The University of Sheffield CiCS Research IT

FILES

READING A FILE LINE BY LINE

Step 1 - open the file

```
file = open("test.txt","r")
```

“r" means you cannot write / overwrite the file.

READING A FILE LINE BY LINE

Step 2 - iterate over the file

```
# Create an empty list for the content
file_list = []

# Read line by line.
for line in file:
    file_list.append(line)
```

Create a new list item by item according to the file.

READING A FILE LINE BY LINE

Step 3 - close the file

```
# close the file after reading the lines.
file.close()

# Print the list.
print(file_list)
```

The University Of Sheffield.

The University of Sheffield CiCS Research IT

READING A FILE LINE BY LINE

Tip 1: Remove new line

```
string = ‘1\n’
line = line.rstrip(“\n")
print(line)
```

READING A FILE LINE BY LINE

Tip 2: Multiple columns or reading a table

```
string = ‘1 3’
print(string)

splinted_string = string.split(‘ ‘)
print(splinted_string[0])
print(splinted_string[1])
```

READING A FILE LINE BY LINE

```
file = open("test2.txt","r")

result1=[]
result2=[]
for line in file:
    line = line.rstrip(‘\n')
    line = line.split(' ')
    result1.append(line[0])
    result2.append(line[1])
file.close()

print(result1)
print(result2)
```

The University Of Sheffield.

The University of Sheffield CiCS Research IT

```
1 53453
2 43554
3 4321
4 43r5
5 4354
6 56
7 45q32
8 4536
9
x 54365
```

Read the following file: wrong.txt

The file contains several rows and two columns of numbers. For some reason, the format is damaged: missing numbers and random non-numeric characters. Write a code which is able to read the data without the corrupted lines.

Hint: use isnumeric(), len()

```
file = open(“wrong.txt","r")

result1=[]
result2=[]
for line in file:
  line = line.rstrip('\n')
  line = line.split(' ')
  if len(line) == 2:
    if line[0].isnumeric() is True:
      if line[1].isnumeric():
        result1.append(line[0])
        result2.append(line[1])
file.close()

print(result1)
print(result2)
```

The University Of Sheffield.

The University of Sheffield CiCS Research IT

PRACTICE SESSION 6