DAY 6 REPORT

Date:	25/05/20	Name:	ANKITHA C C
Course:	DIGITAL SIGNAL PROCESSING	USN:	4AL16EC004
Topic:	1.Introduction to Fourier Series & Fourier Transform. 2. Fourier Series – Part 1 and part-2. 3. Inner Product in Hilbert Transform. 4. Complex Fourier Series. 5. Fourier Series using Matlab, python & Gibbs Phenomena Using Matlab	Semester & Section:	8th & 'A' SECTION
Github Repository:	ankitha-course		

FORENOON SESSION DETAILS
Image of session

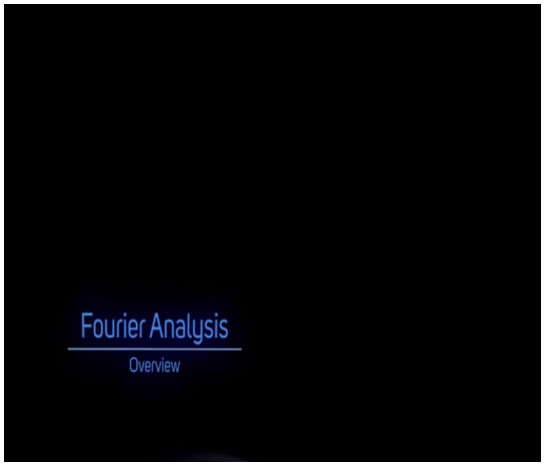


Fig: introduction to fourier system

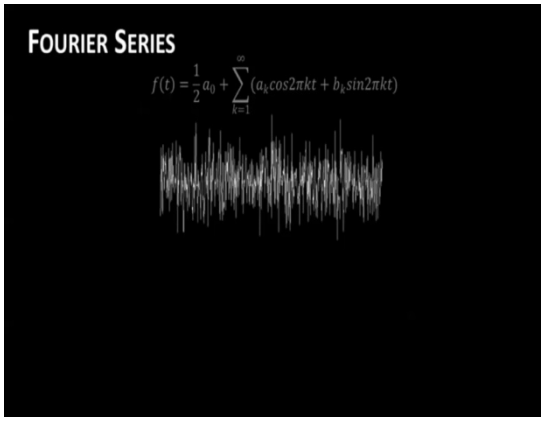


Fig: fourier series formula

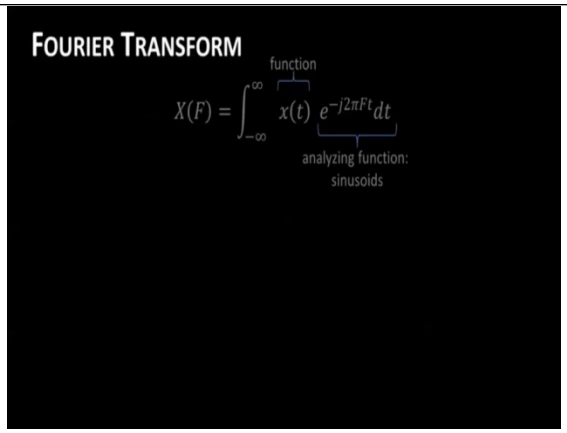
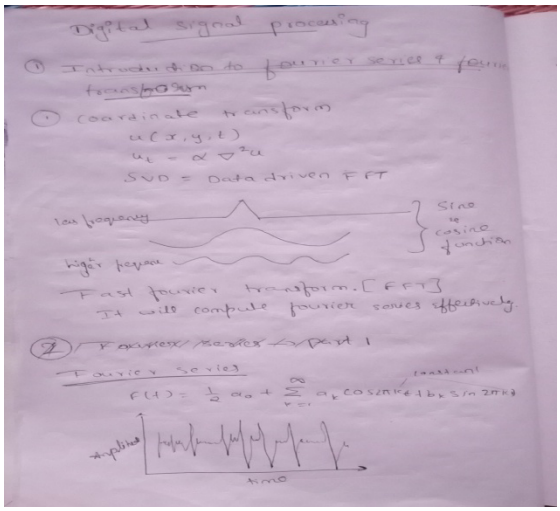
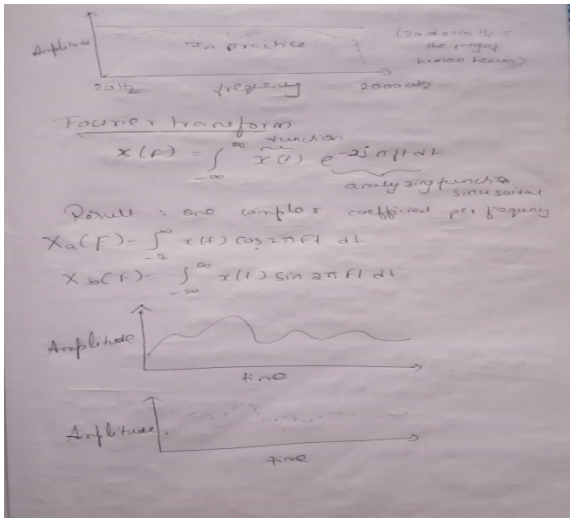


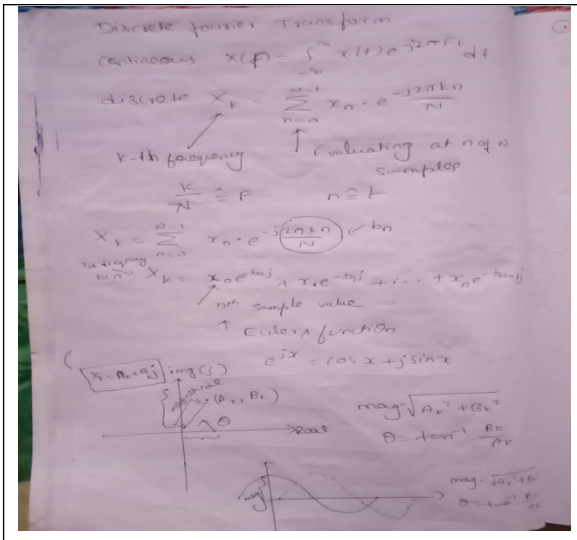
Fig: fourier transform formula

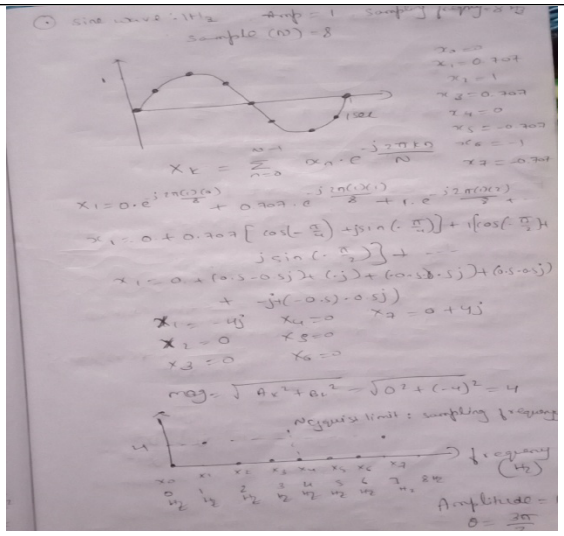
Report – Report can be typed or hand written for up to two pages.

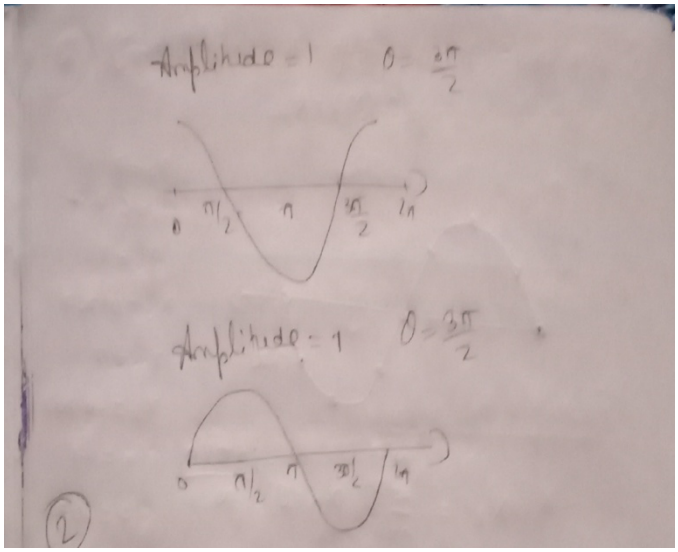
1. Introduction to Fourier Series & Fourier Transform.



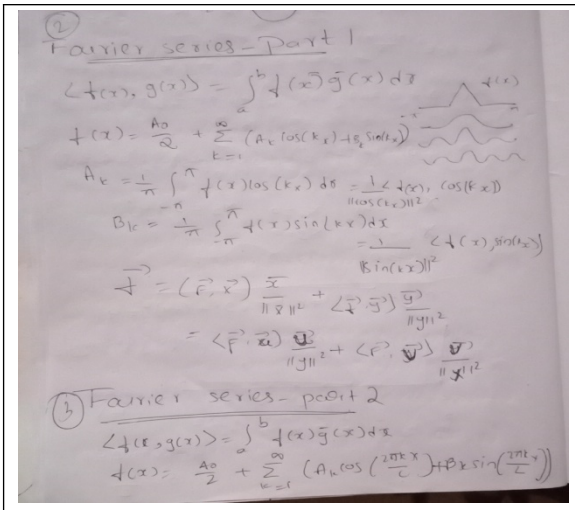


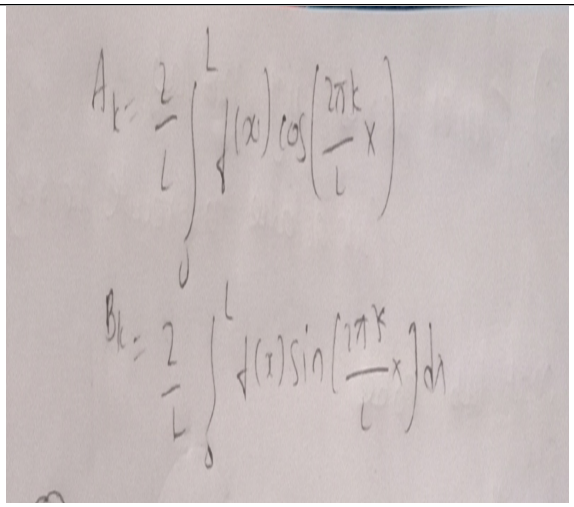


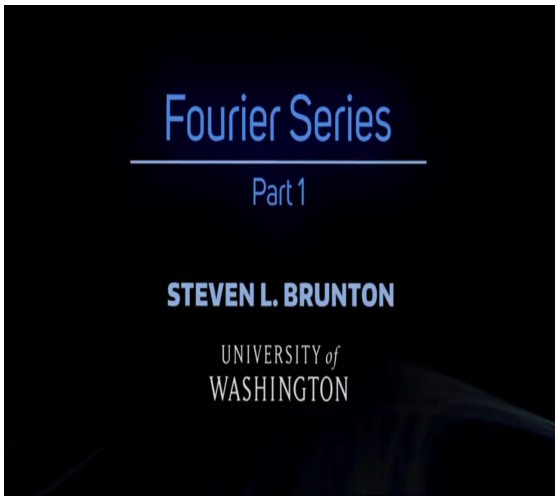


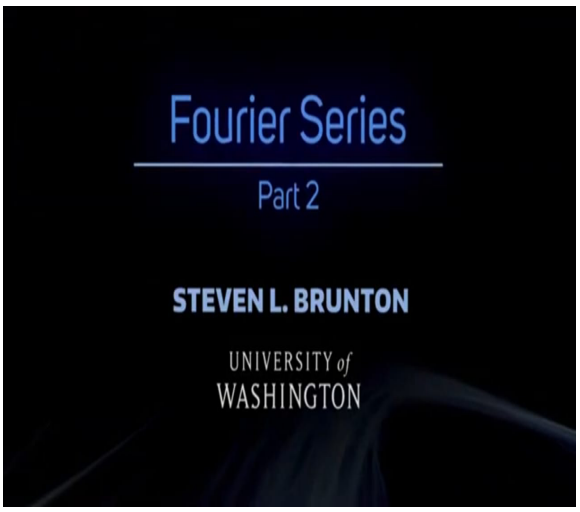


2. Fourier Series – Part 1 and part-2.

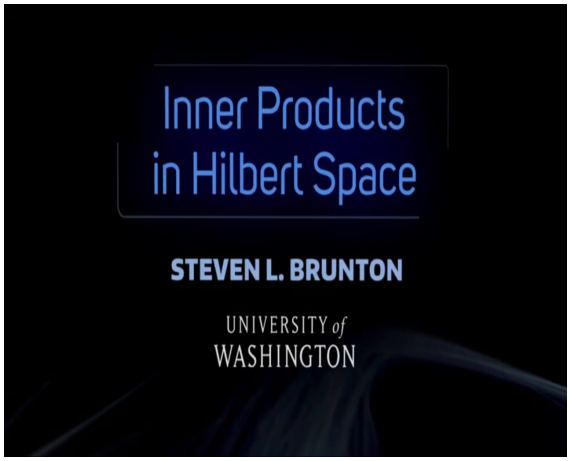


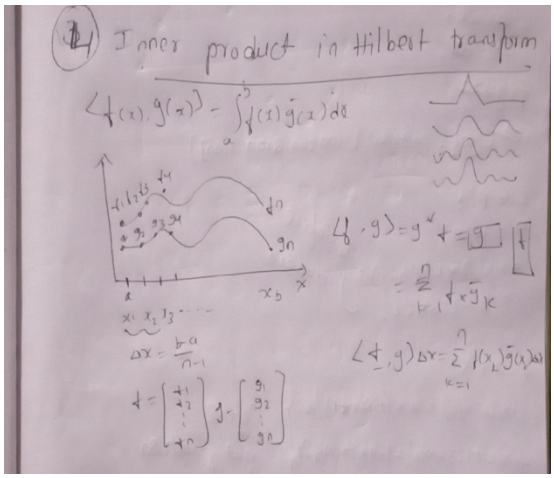




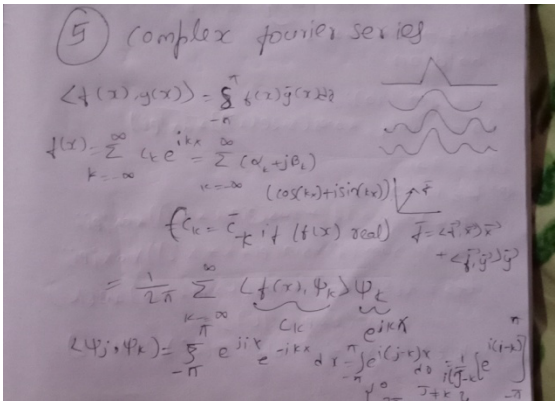


3. Inner Product in Hilbert Transform.



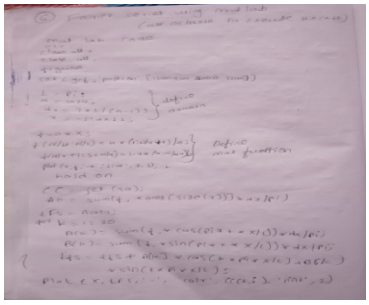


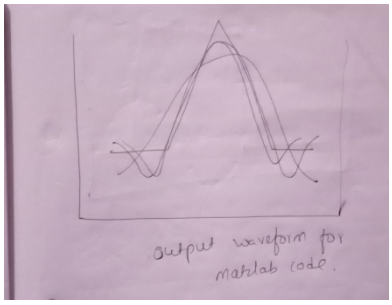
4. Complex Fourier Series.

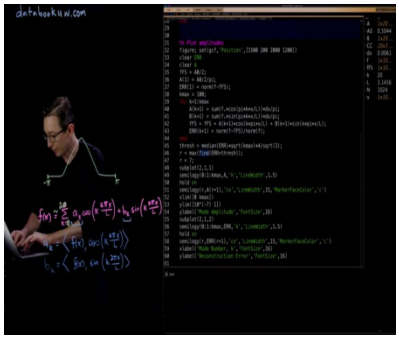


5. Fourier Series using

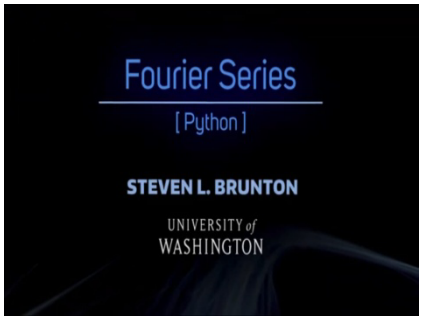
a. Matlab

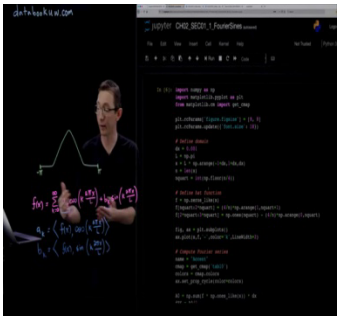


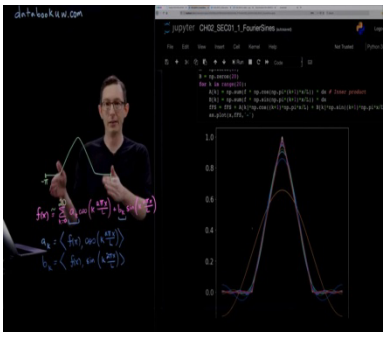


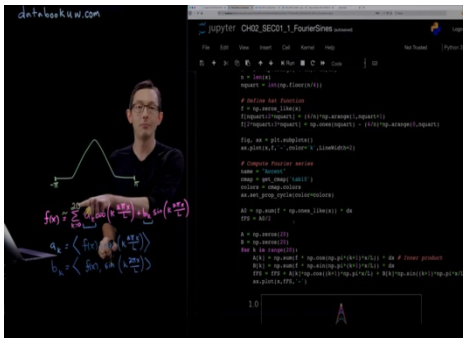


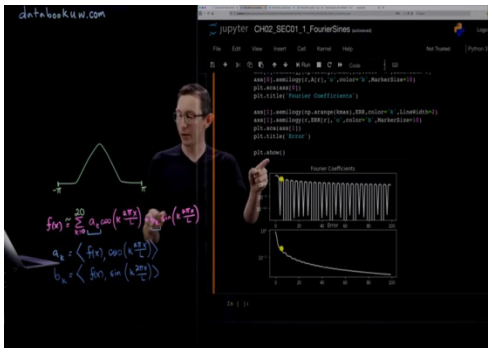
b. python



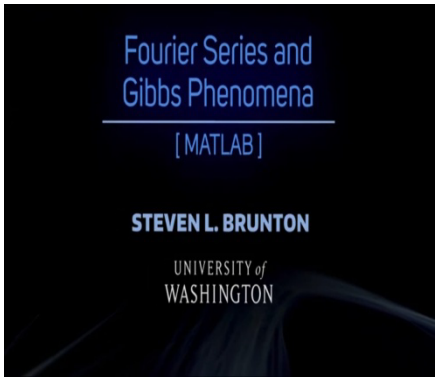


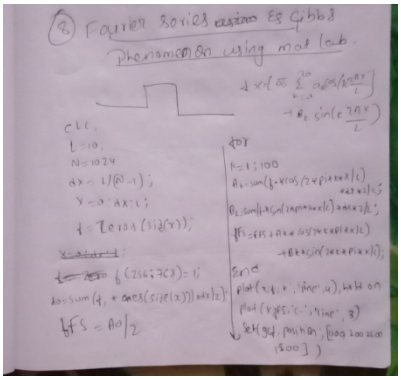


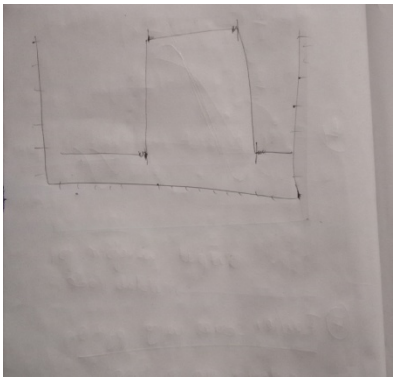


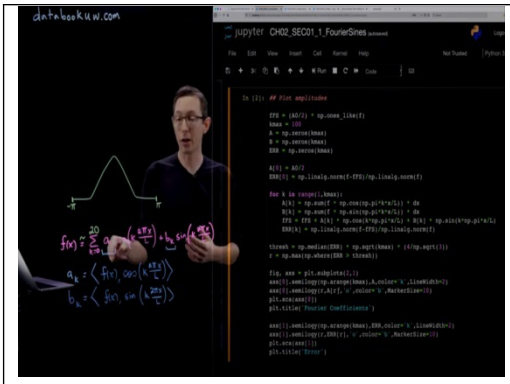


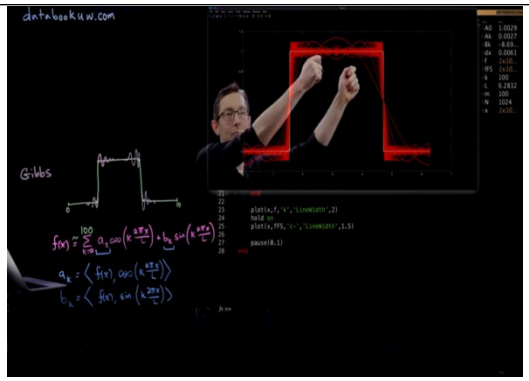
c. Gibbs Phenomena Using Matlab





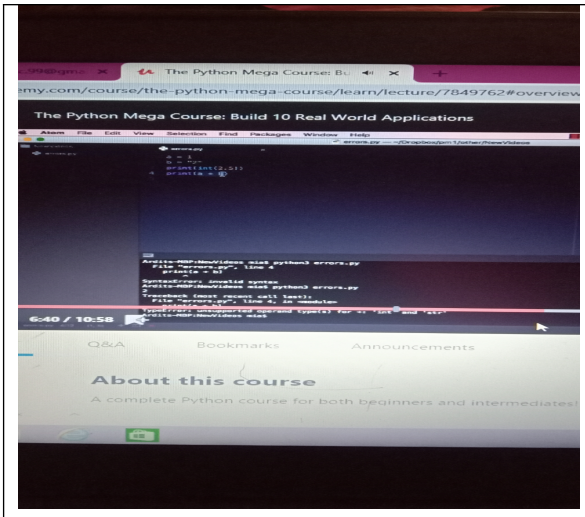


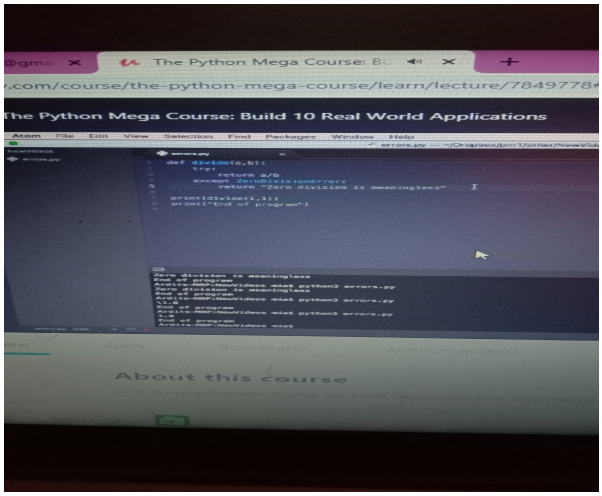




Date:	25/5/20	Name:	Ankitha c c
Course:	Python	USN:	4al16ec004
Topic:	1. Fixing programming error 2. Build website blocker	Semester & Section:	8th a

AFTERNOON SESSION DETAILS
Image of session





Report – Report can be typed or hand written for up to two pages.

1. Fixing programming error

A. Syntax error

Syntax errors – usually the easiest to spot, syntax errors occur when you make a typo. Not ending an if statement with the colon is an example of an syntax error, as is misspelling a Python keyword (e.g. using whille instead of while). Syntax error usually appear at compile time and are reported by the interpreter. Here is an example of a syntax error:

Eg:

```
x = int(input('Enter a number: '))
while x%2 == 0:
    print('You have entered an even number.')
else:
    print ('You have entered an odd number.')
```

B. Runtime error

As you do more and more programming, you will naturally encounter a lot of errors (or bugs). Causing, understanding, and fixing errors is an important part of programming. Python will do its best to run anything that you tell it to run, but if it can't understand what you're asking, then it won't run the program. All the same, Python will try to tell you a little bit of information about what went wrong, in order to help you try to fix it.

Here are two Python errors.

Example: A Syntax Error

```
print "Gee golly"
```

In this first example, we forget to use the parenthesis that are required by print(...). Python does not understand what you are trying to do.

Here is a second example of a bug in Python.

Example: A Run-Time Error

```
print(greeting)
```

2. Building website blocker

A. The any() function

```
>>> lines  = ["trees are good", "pool is fresh", "face is round"]
>>> website_list = ["face", "clock", "trend"]
>>> for line in lines:
...     any(website in line for website in website_list)
...
False
False
True
```

We start iterating over the items of website_list using a for loop. In the first iteration we would have:

any(website in "trees are good" for website in website_list)

Inside the parenthesis of any() there's another loop that iterates over website_list:

("face" in "trees are good")

("clock" in "trees are good")

("trend" in "trees are good")

If any of the above is True you get the expression evaluated to True. In this case none of them is True, so you get False.

If you want to return True (if all of them are True), use all() instead of any().

So, the part any(website in line for website in website_list) will either be equal to True or False.

B. Scheduling a Python Program on a Server

Keeping your computer on 24-7 is not practical, so if you want to execute a Python script at a particular time every day,

you probably need a computer that is on all the time.

PythonAnywhere gives you access to such a 24-7 computer. You can upload a Python script and schedule it to run at a certain time every day. This availability can be useful, for example, when you want to extract some values (e.g., weather data) from a website and generate a text file with the value or other reports every day.

To schedule a Python script for execution on PythonAnywhere, follow these simple steps:

Sign up for a free account at https://www.pythonanywhere.com.

Go to your Dashboard, Files, Upload a File, and upload the Python file you want to schedule for execution.

Go to Tasks and set the time of the day you want your script to be executed and type in the name of the Python file you uploaded (e.g., myscript.py). Note that the time you enter should be in UTC.

Your Python file will now be executed every day at your specified time. If you don't have a Python script and you're still confused about the benefit of this, here is a very simple Python script that you can use to try the above steps:

If you don't have a Python script and you're still confused about the benefits of this PythonAnywhere feature, here is a very simple Python script you can use to schedule for execution:

```
from datetime import datetime

with open(datetime.now().strftime("%Y-%m-%d-%H-%M-%S"), "w") as myfile:

myfile.write("Hi there!")
```

The above code creates a text file and writes the string "Hi there!" in that text file. The name of the text file will be the current date and time. For example one file name example would be 2018-02-16-18-20-33.txt.

That name is generated by datetime.now() indicating the date and time the script was executed. Every time the script is executed, the script generates a new text file with a different name. You will have a new text file created every day.