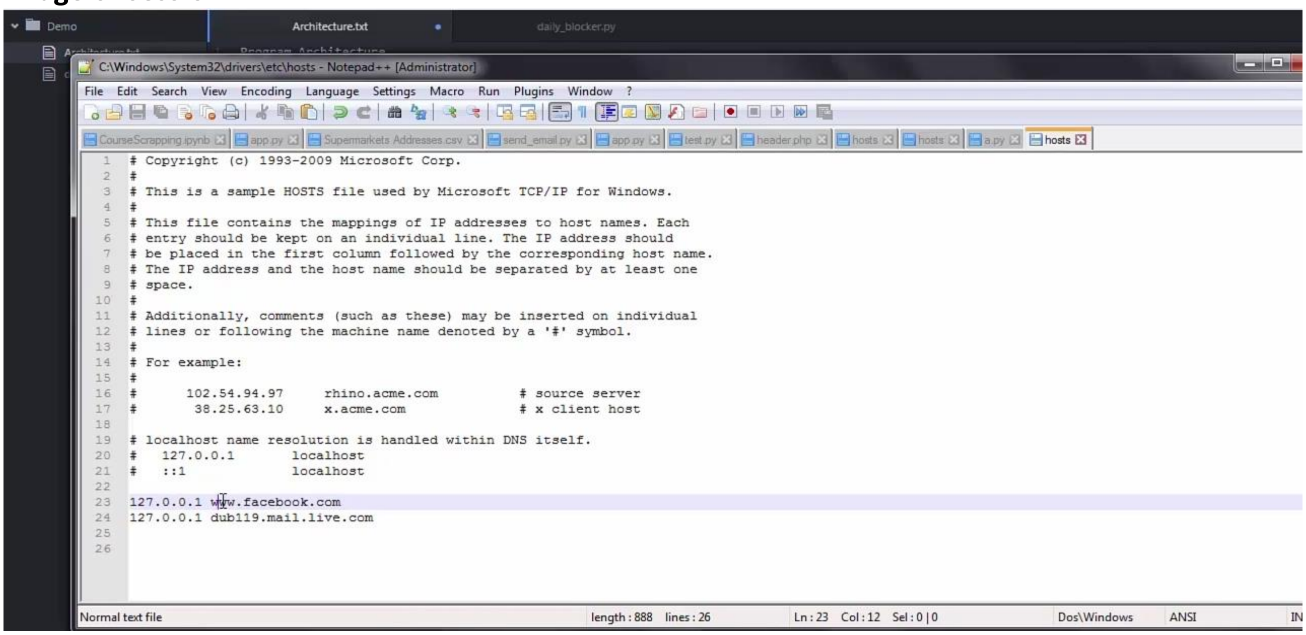
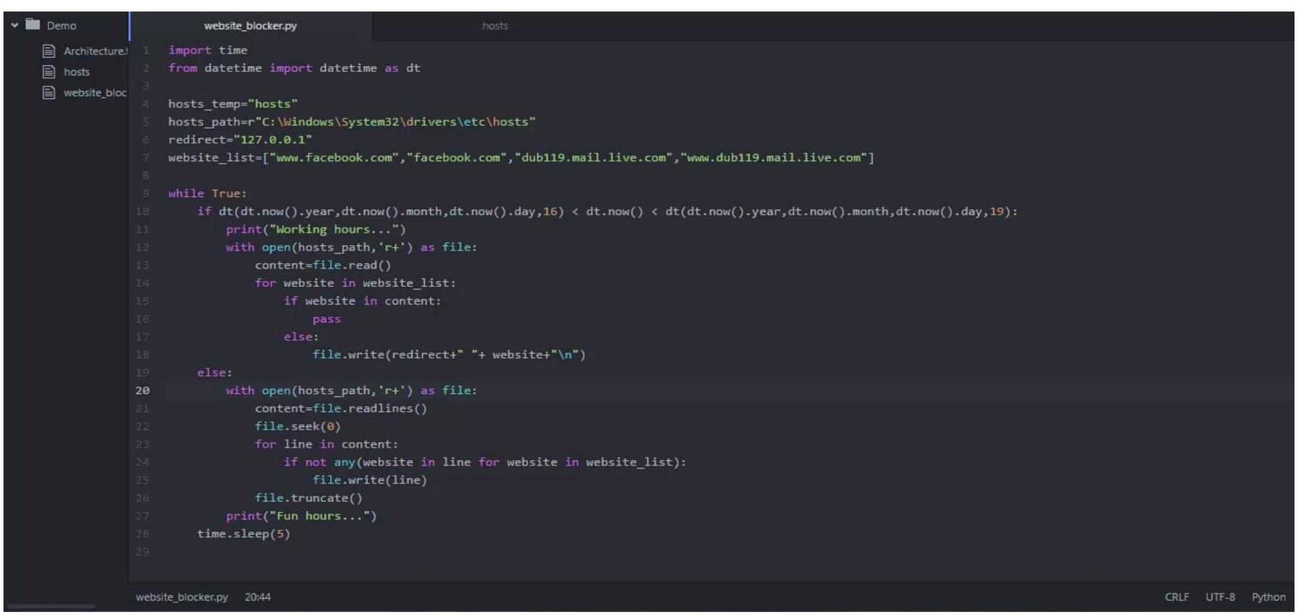
DAILY ASSESSMENT FORMAT

Date:	25th May 2020	**Name:**	Soundarya NA
Course:	UDEMY	**USN:**	4AL16EC077
Topic:	PYTHON: Application 4: Build a personal Website with Python and Flask	**Semester & Section:**	8th - B

AFTERNOON SESSION DETAILS

Image of session





Report:

Code1:

```
from flask import Flask


app = Flask(__name__)


@app.route("/")
def home():
    return "Hello, World!"


@app.route("/salvador")
def salvador():
    return "Hello, Salvador"


if __name__ == "__main__":
    app.run(debug=True)
```

We will write code that will take care of the server-side processing. Our code will receive requests. It will figure out what those requests are dealing with and what they are asking. It will also figure out what response to send to the user. It makes the process of designing a web application simpler. Flask lets us focus on what the users are requesting and what sort of response to give back.

HTTP is the protocol for websites. The internet uses it to interact and communicate with computers and servers. Let me give you an example of how you use it every day.

When you type the name of a website in the address bar of your browser and you hit enter. What happens is that an HTTP request has been sent to a server.

For example, when I go to my address bar and type google.com, then hit enter, an HTTP request is sent to a Google Server. The Google Server receives the request and needs to figure how to interpret that request. The Google Server sends back an HTTP response that contains the information that my web browser receives. Then it displays what you asked for on a page in the browser.

Code2:

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>About Flask</title>
  </head>
  <body>
    {% extends "template.html" %}
    {% block content %}

    <h1> About Flask </h1>
    <p> Flask is a micro web framework written in Python.</p>
    <p> Applications that use the Flask framework include Pinterest,
      LinkedIn, and the community web page for Flask itself.</p>

    {% endblock %}
  </body>
</html>
```

Code3:

Linking our CSS with our HTML file

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Flask Parent Template</title>

    <link rel="stylesheet" href="{{ url_for('static',    filename='css/template.css') }}">

</head>
```

```
 <body>
   <header>
     <div class="container">
       <h1 class="logo">First Web App</h1>
       <strong><nav>
         <ul class="menu">
           <li><a href="{{ url_for('home') }}">Home</a></li>
           <li><a href="{{ url_for('about') }}">About</a></li>
         </ul>
       </nav></strong>
     </div>
   </header>

{% block content %}
{% endblock %}

 </body>
</html>
```

Why use virtualenv?

You may use Python for others projects besides web-development. Your projects might have different versions of Python installed, different dependencies and packages. We use virtualenv to create an isolated environment for your Python project. This means that each project can have its own dependencies regardless of what dependencies every other project has.

Deploy Your Web Application to the Cloud:

To deploy our web application to the cloud, we will use Google App Engine (Standard Environment). This is an example of a Platform as a Service (PaaS). PaaS refers to the delivery of operating systems and associated services over the internet without downloads or installation. The approach lets customers create and deploy applications without having to invest in the underlying infrastructure (More info on PaaS check out TechTarget).

Activating the virtual environment:

Now go to your terminal or command prompt. Go to the directory that contains the file called activate. The file called activate is found inside a folder called Scripts for Windows and bin for OS X and Linux.

The Application:

Now check the URL of your application. The application will be store in the following way:

"your project id".appspot.com

Conclusion:

- Use the framework called Flask to use Python as a Server Side Language.
- Learned how to use HTML, CSS, and Flask to make a website.
- Learned how to create Virtual Environments using virtualenv.
- Use Google App Engine Standard Environment to deploy an application to the cloud.

Date:	25^{th} May 2020	**Name:**	Soundarya NA
Course:	DSP	**USN:**	4AL16EC077
Topic:	Digital Signal Processing	**Semester & Section:**	8^{th} - B

Report:

Introduction to fourier series and fourier transform:

Fourier series:

Fourier series expansion or harmonic analysis extracts appropriately weighted harmonic components from a general periodic waveform. Any function f(x), which is periodic between $-\pi$ and $+\pi$(or L to + L) can be expanded in this interval by a Fourier series. The Fourier series expansion of the function f(x) is defined by

f(x) = ∑∞ n=0 c(n) einπx⁄L

Fourier transform:

The concept of Fourier series is quite useful for introducing the concept of harmonic analysis, and the concept of the discrete Fourier transform to be discussed later. Next, the standard concepts of the Fourier transform are introduced.

c(k) = 1 c ∫ +∞ −∞ f(x)eikxdk

The Fourier series synthesis equation creates a continuous periodic signal with a fundamental frequency, f, by adding scaled cosine and sine waves with frequencies: f, 2f, 3f, 4f, etc. The amplitudes of the cosine waves are held in the variables: a1, a2, a3, a3, etc., while the amplitudes of the sine waves are held in: b1, b2, b3, b4, and so on. In other words, the "a" and "b" coefficients are the real and imaginary parts of the frequency spectrum, respectively. In addition, the coefficient a0 is used to hold the DC value of the time domain waveform. This can be viewed as the amplitude of a cosine wave with zero frequency (a constant value). Sometimes is grouped with the other "a" coefficients, but it is often handled separately because it requires special calculations. There is no b0 coefficient since a sine wave of zero frequency has a constant value of zero, and would be quite useless.

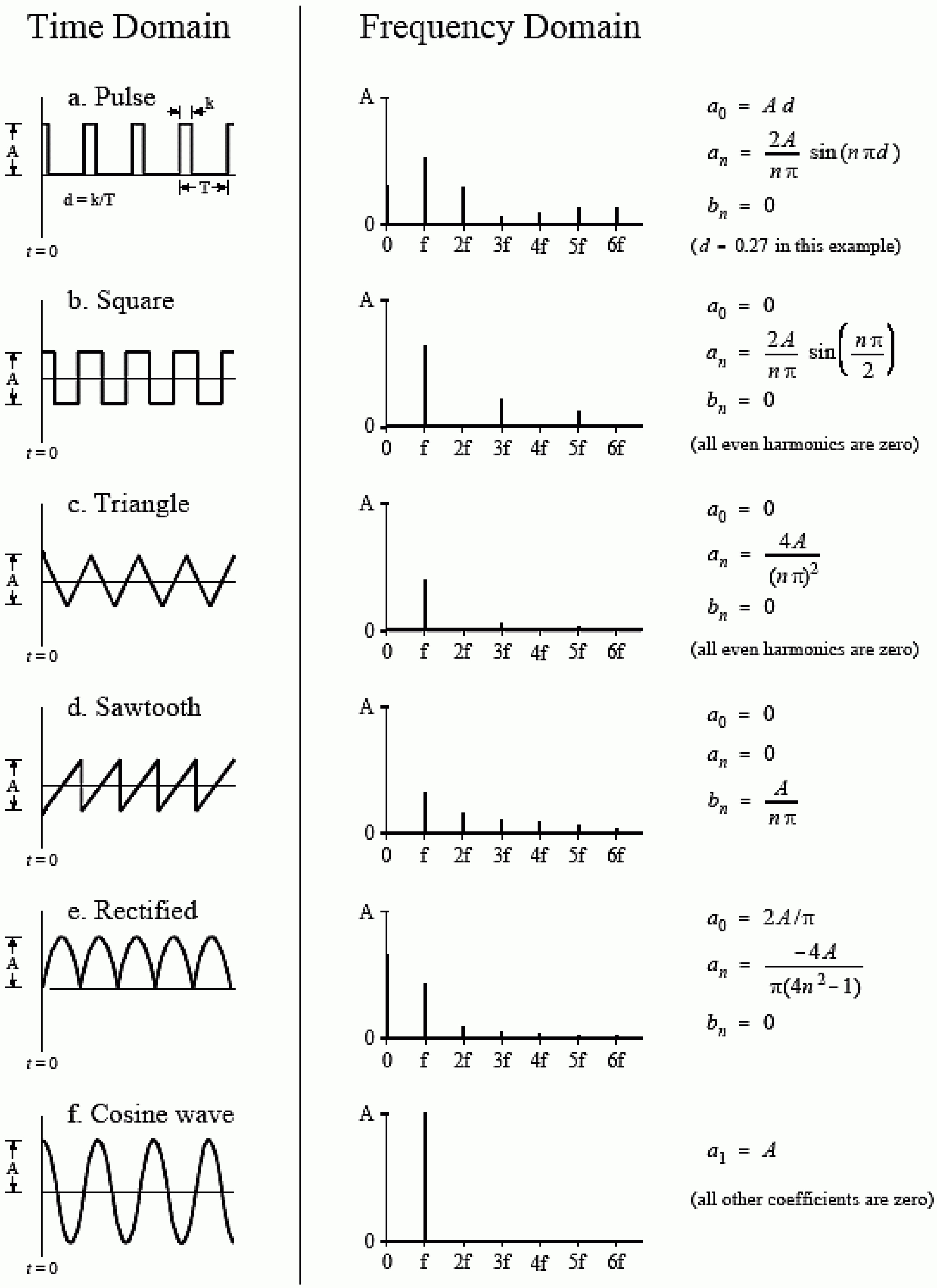


FIGURE 13-10
Examples of the Fourier series. Six common time domain waveforms are shown, along with the equations to calculate their "a" and "b" coefficients.

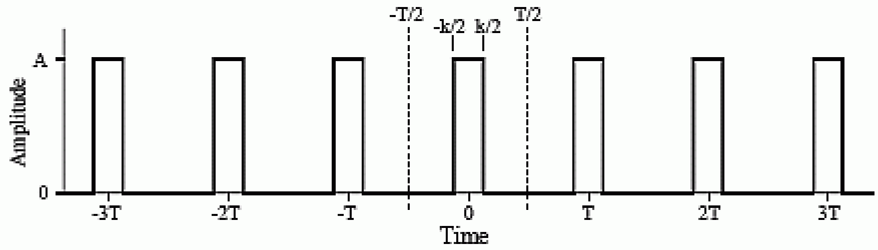


FIGURE 13-11
Example of calculating a Fourier series. This is a pulse train with a duty cycle of $d = k/T$. The Fourier series coefficients are calculated by correlating the waveform with cosine and sine waves over any full period. In this example, the period from $-T/2$ to $T/2$ is used.

Hilbert Transform:

The Hilbert transform of u can be thought of as the convolution of u(t) with the function h(t) = 1/(πt), known as the Cauchy kernel. Because h(t) is not integrable, the integral defining the convolution does not always converge. Instead, the Hilbert transform is defined using the Cauchy principal value (denoted here by p.v.). Explicitly, the Hilbert transform of a function (or signal) u(t) is given by

$$H(u)(t) = \frac{1}{\pi}\ \mathrm{p.v.}\int_{-\infty}^{+\infty} \frac{u(\tau)}{t-\tau}\,d\tau,$$

Fourier series using Python:

```
import numpy as np
import matplotlib.pyplot as plt
resolution = 0.0001
 x = np.arange(-np.pi,np.pi,resolution)# -pi to pi with the interval of 0.0001
 square = np.array(x)
square[range(x.size)] = 0
square[range(int(x.size/2))] = 1
square[range(int(x.size/2), int(x.size))]= 0
 np.trapz(square,x) # integration of f(x)
a0 = (np.trapz(square,x))/ np.pi # dividing by pi which is present out side the integration
 n=1
harm = np.sin(n*x)
mult1 = square*harm
 fund = np.trapz(mult1,x)
 np.trapz(mult1,x)
```

```
b1 = (np.trapz(mult1,x))/np.pi
n=3
harm = np.sin(n*x)
mult2 = square*harm
third = np.trapz(mult2,x)
np.trapz(mult2,x)
b3 = (np.trapz(mult2,x))/np.pi
20*np.log10(abs(third/fund))
plt.subplot(311)
plt.plot(x,square)
plt.xlabel('(x)')
plt.ylabel('f(x)')
plt.title('SIGNAL', fontsize=18)
plt.subplot(312)
plt.plot(x,mult1)
plt.plot(x,square)
plt.xlabel('(x)')
plt.ylabel('sin(1*x)*f(x)')
plt.subplot(313)
plt.plot(x,mult2)
plt.plot(x,square)
plt.xlabel('(x)')
plt.ylabel('sin(3*x)*f(x)')
plt.show()
```