

DAILY ASSESSMENT FORMAT

Date:	17/06/2020	Name:	SAMRUDDI SHETTY
Course:	Python	USN:	4AL16IS047
Topic:	Day 4	Semester & Section:	8 th sem
Github Repository:			

AFTERNOON SESSION DETAILS

Introduction

In this section of the tutorial, we are going to build a real-time most popular python application known as website blocker. This application can be used to block the websites so that the user can not open them during the specific period.

Objective

The objective of Python website blocker is to block some certain websites which can distract the user during the specified amount of time. In this, we will block the access to the list of some particular websites during the working hours so that the user can only access those websites during the free time only. The working time in this python application is considered from 9 AM to 5 PM. The time period except that time will be considered as free time.

Process

To block the access to a specific website on the computer, we need to configure the **hosts** file.

The hosts file

The hosts file is a local file which was used to map hostnames to IP addresses in the old days. Although the DNS service is used nowadays to map the hostnames to the IP addresses, the host file is still very complex and can be used to configure the mapping of the IP addresses locally.

Location of hosts file

The location of the hosts file varies from operating system to operating system.

Windows: C:\Windows\System32\drivers\etc

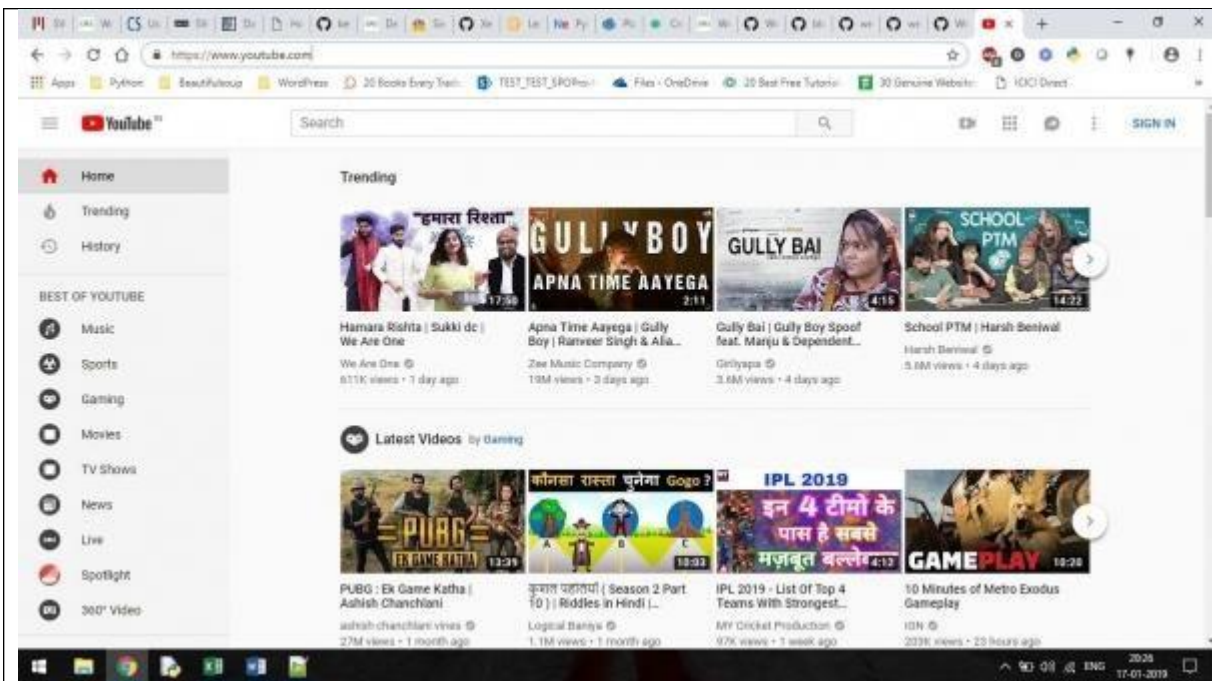
mac and Linux: /etc/hosts

Configuration

Since the hosts file contains the mapping between the host names and IP addresses, let's look at the content of our hosts file first stored as **/etc/hosts** as we use CentOS 7 (Linux). As we can see in the above image, the mappings are saved for the localhost with the IP address 127.0.0.1. Similarly, we can redirect any hostname back to the localhost IP (127.0.0.1). It will block the access to that hostname and redirect that to localhost on each request. For testimony, let's edit the content of hosts file and add some of the following lines to it. To make changes to the **/etc/hosts** file, we will need to change its permissions.

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com          # source server
#       38.25.63.10       x.acme.com             # x client host
#
# localhost name resolution is handled within DNS itself.
#   127.0.0.1       localhost
#   ::1             localhost
```

As I have not mentioned, any website name in my host file. If I try to open “youtube.com”, I can do it without any problem. Below is the screenshot (just to make sure after running my scripts, this website shouldn't open if I want to block it.)



Below is our website blocker program –

#Import libraries

import time

from datetime import datetime as dt

#Windows host file path

hostsPath=r"C:\Windows\System32\drivers\etc\hosts"

redirect="127.0.0.1"

#Add the website you want to block, in this list

websites=["www.youtube.com", "youtube.com", "www.facebook.com", "facebook.com"]

while True:

 #Duration during which, website blocker will work

 if dt(dt.now().year,dt.now().month,dt.now().day,9) < dt.now() <
dt(dt.now().year,dt.now().month,dt.now().day,18):

 print ("Sorry Not Allowed...")

 with open(hostsPath,'r+') as file:

```
content = file.read()

for site in websites:

    if site in content:

        pass

    else:

        file.write(redirect+" "+site+"\n")

else:

    with open(hostsPath,'r+') as file:

        content = file.readlines()

        file.seek(0)

        for line in content:

            if not any(site in line for site in websites):

                file.write(line)

        file.truncate()

    print ("Allowed access!")

    time.sleep(5)
```

```
2 import time
3 from datetime import datetime as dt
4 #Windows host file path
5 hostsPath=r"C
    :\Windows\System32\drivers\etc\hosts"
6 redirect="127.0.0.1"
7 #Add the website you want to block, in
    this list
8 websites=["www.youtube.com", "youtube.com"
    , "www.facebook.com", "facebook.com"]
9 while True:
10     #Duration during which, website
        blocker will work
11     if dt(dt.now().year,dt.now().month,dt
        .now().day,9) < dt.now() < dt(dt
        .now().year,dt.now().month,dt.now
        ().day,18):
12         print ("Sorry Not Allowed...")
13         with open(hostsPath,'r+') as file
            :
14             content=file.read()
15             for site in websites:
16                 if site in content:
17                     pass
18                 else:
19                     file.write(redirect+"
                        "+site+"\n")
20     else:
21         with open(hostsPath,'r+') as file
            :
22             content=file.readlines()
23             file.seek(0)
24             for line in content:
25                 if not any(site in line
                    for site in websites
                    ):
26                     file.write(line)
27             file.truncate()
28             print ("Allowed access!")
29             time.sleep(5)
```

Result



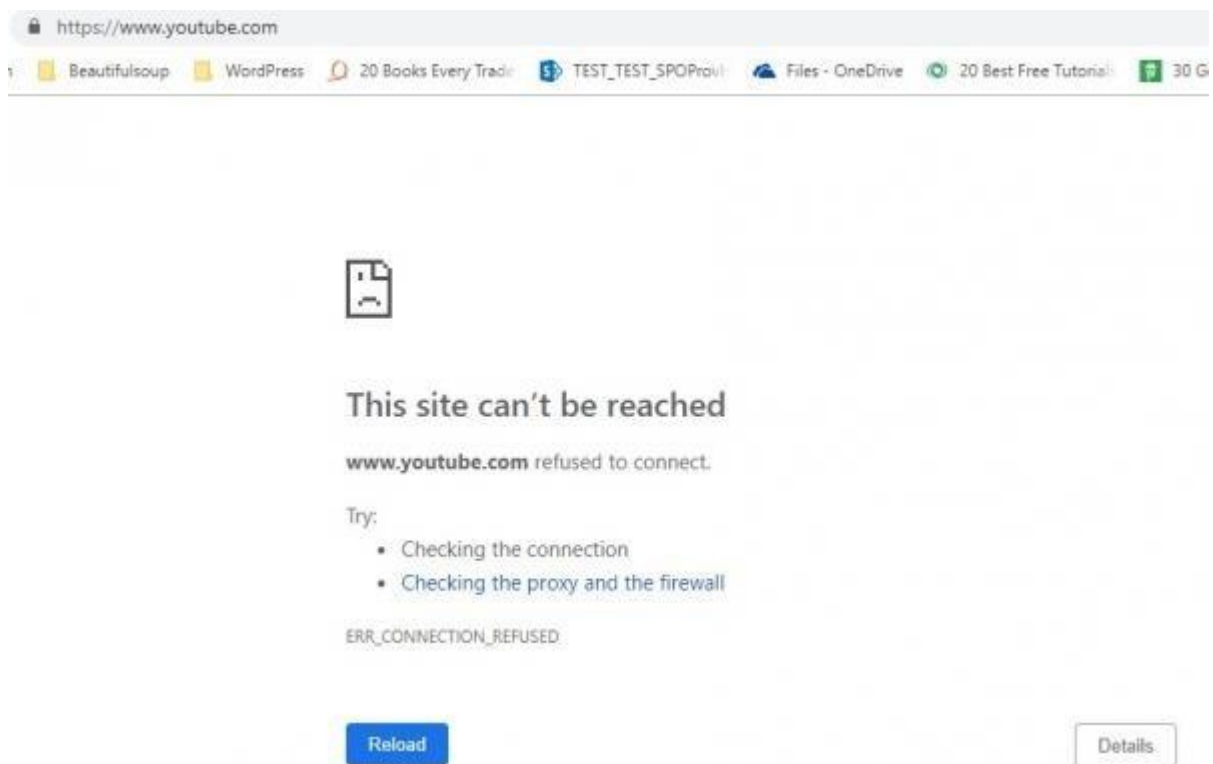
```
$python3 main.py
```

Sorry Not Allowed...

Output

Sorry Not Allowed...
Sorry Not Allowed...
Sorry Not Allowed...
Sorry Not Allowed...
Sorry Not Allowed...
Sorry Not Allowed...
Sorry Not Allowed...
....

Now if I try to open – youtube.com or facebook.com, we'll get –

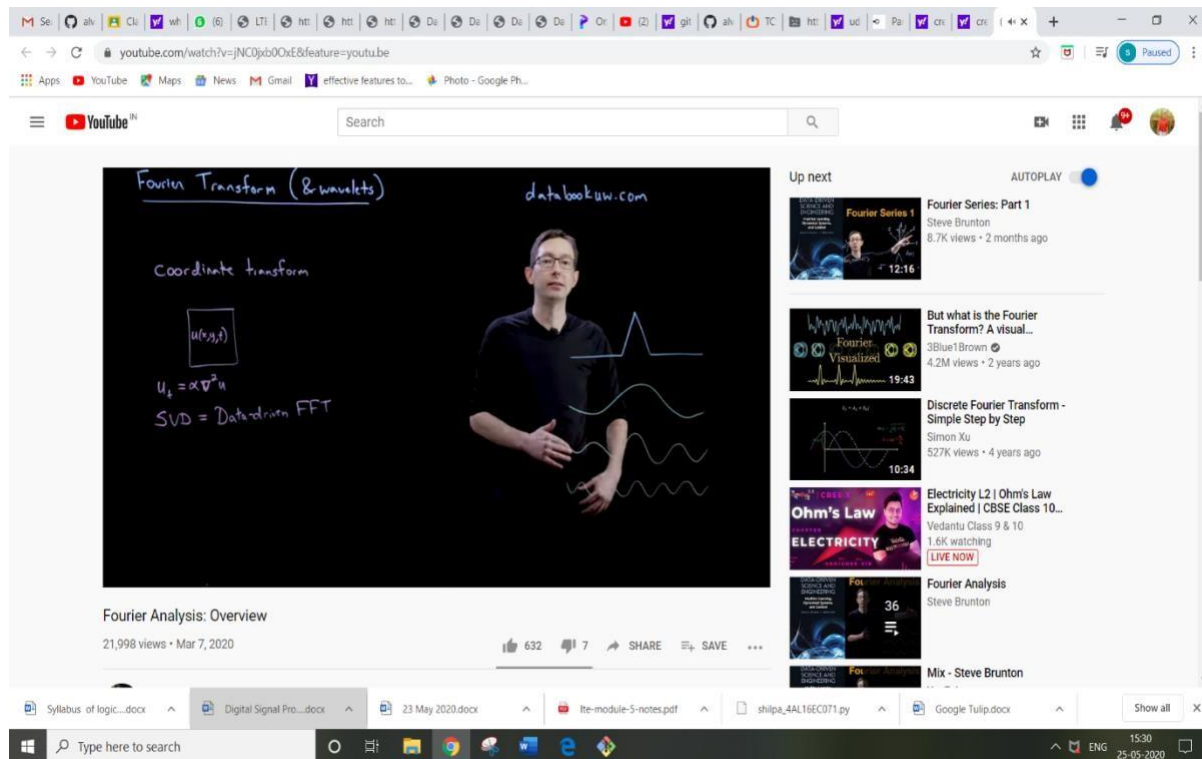


We can customize above code as per our requirement like duration, websites, custom messages etc.

Date:	25/05/2020	Name:	Ramanath Naik
Course:	DSP	USN:	4AL16EC054
Topic:	DAY 1	Semester & Section:	8th sem , 'B' section
Github Repository:			

FORENOON SESSION DETAILS

Image of session



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

Fourier transform and wavelets

FT is a coordinate transformation used in the representation of data in images. Spoke about history of FT. we can take arbitrary function and we can approximate this by some sines of increasingly frequency. Discussed about Laplacian operator, Hilbert spaces. Fast forward two hundred years, and the fast Fourier transform has become the cornerstone of computational mathematics, enabling real-

time image and audio compression, global communication networks, modern devices and hardware, numerical physics and engineering at scale, and advanced data analysis. Simply put, the fast Fourier transform has had a more significant and profound role in shaping the modern world than any other algorithm to date.

FFT is used to compute Fourier series on a computer. Modern digital communication is depending on FFT. Fourier series convert any component in time to frequency. They have discussed it using noise signal. Discussed about basics of DFT. How to expand summation and how to deal with complex coefficients was discussed. How to deal with complex number and how to plot it on a plane was briefed. Solved some examples.

Fourier series- A Fourier series is an expansion of a [periodic function](#) $f(x)$ in terms of an infinite sum of [sines](#) and [cosines](#). Fourier series make use of the [orthogonality](#) relationships of the [sine](#) and [cosine](#) functions. The computation and study of Fourier series is known as [harmonic analysis](#) and is extremely useful as a way to break up an *arbitrary* periodic function into a set of simple terms that can be plugged in, solved individually, and then recombined to obtain the solution to the original problem or an approximation to it to whatever accuracy is desired or practical. How to solve Fourier series was discussed with problem. How to solve complex valued function.

Inner products of functions and vectors

In this section, we will make use of inner products and norms of functions. In particular, we will use the common Hermitian inner product for functions $f(x)$ and $g(x)$ defined for x on a domain $x \in [a, b]$:

$$\langle f(x), g(x) \rangle = \int_a^b f(x)^* g(x) dx \quad (2.1)$$

where $*$ denotes the complex conjugate. The inner product of functions may seem strange or unmotivated at first, but this definition becomes clear when we consider the inner product of vectors of data. If we discretize the functions $f(x)$ and $g(x)$ into vectors of data, as in we would like the vector inner product to converge to the function inner product as the sampling resolution is increased. The inner product of the data vectors $f = [f_1 \ f_2 \ \dots \ f_n]^T$ and $g = [g_1 \ g_2 \ \dots \ g_n]^T$ is

defined by:

$$\langle f, g \rangle = g^* f = \sum_{k=1}^n f_k^* g_k = \sum_{k=1}^n f(x_k) g(x_k) \quad (2.2)$$

The magnitude of this inner product will grow as more data points are added; i.e., as n increases. Thus, we may normalize by $\Delta x = (b-a)/(n-1)$:

$$\int_a^b f(x)g(x)dx \approx \sum_{k=1}^n f(x_k)g(x_k)\Delta x, \text{-----} \textcircled{7} (2.3)$$

which is the Riemann approximation to the continuous function inner product. It is now clear that as we take the limit of $n \rightarrow \infty$ (i.e., infinite data resolution, with $\Delta x \rightarrow 0$), the vector inner product converges to the inner product of functions in (2.1).

This inner product also induces a norm on functions, given by

$$\|f\|_2 = (\langle f, f \rangle)^{1/2} = \sqrt{\langle f, f \rangle} = \left(\int_a^b f(x) \cdot f(x) dx \right)^{1/2} \text{-----} \textcircled{7} (2.4)$$

The set of all functions with bounded norm define the set of square integrable functions, denoted by $L^2([a,b])$; this is also known as the set of Lebesgue integrable functions. The interval $[a,b]$ may also be chosen to be infinite (e.g., $(-\infty, \infty)$), semi-infinite (e.g., $[a, \infty)$), or periodic (e.g., $[-\pi, \pi]$). A fun example of a function in $L^2([1, \infty))$ is $f(x) = 1/x$. The square of f has finite integral from 1 to ∞ , although the integral of the function itself diverges. The shape obtained by rotating this function about the x -axis is known as Gabriel's horn, as the volume is finite (related to the integral of f^2), while the surface area is infinite (related to the integral of f). As in finite-dimensional vector spaces, the inner product may be used to project a function into an new coordinate system defined by a basis of orthogonal functions. A Fourier series representation of a function f is precisely a projection of this function onto the orthogonal set of sine and cosine functions with integer period on the domain $[a,b]$. This is the subject of the following sections.

Fourier series

$$f(x) = a_0/2 + \sum_{k=1}^{\infty} (a_k \cos(kx) + b_k \sin(kx))$$

Thus, the functions $\psi_k = e^{ikx}$ for $k \in \mathbb{Z}$ (i.e., for integer k) provide a basis for periodic, complex-valued functions on an interval $[0, 2\pi]$. It is simple to see that these functions are orthogonal:

$$\langle \psi_j, \psi_k \rangle = \int_{-\pi}^{\pi} e^{ijx} e^{-ikx} dx = \int_{-\pi}^{\pi} e^{i(j-k)x} dx = \left[\frac{e^{i(j-k)x}}{i(j-k)} \right]_{-\pi}^{\pi} = \begin{cases} 0 & \text{if } j \neq k \\ 2\pi & \text{if } j = k. \end{cases}$$

$$\begin{aligned} \vec{f} &= \langle \vec{f}, \vec{x} \rangle \frac{\vec{x}}{\|\vec{x}\|^2} + \langle \vec{f}, \vec{y} \rangle \frac{\vec{y}}{\|\vec{y}\|^2} \\ &= \langle \vec{f}, \vec{u} \rangle \frac{\vec{u}}{\|\vec{u}\|^2} + \langle \vec{f}, \vec{v} \rangle \frac{\vec{v}}{\|\vec{v}\|^2}. \end{aligned}$$

- Fourier series approximation to a hat function.

```

% Define domain
dx = 0.001; L = pi;
x = (-1+dx:dx:1)*L;
n = length(x); nquart = floor(n/4);
% Define hat function
f = 0*x; f(nquart:2*nquart) = 4*(1:nquart+1)/n;
f(2*nquart+1:3*nquart) = 1-4*(0:nquart-1)/n;
plot(x,f,'-k','LineWidth',1.5), hold on
% Compute Fourier series
CC = jet(20);
A0 = sum(f.*ones(size(x)))*dx;
fFS = A0/2;
for k=1:20
A(k) = sum(f.*cos(pi*k*x/L))*dx; % Inner product
B(k) = sum(f.*sin(pi*k*x/L))*dx;
fFS = fFS + A(k)*cos(k*pi*x/L) + B(k)*sin(k*pi*x/L);
plot(x,fFS,'-', 'Color',CC(k,:), 'LineWidth',1.2)
end

```

- The truncated Fourier series is plagued by ringing oscillations, known as Gibbs phenomena, around the sharp corners of the step function. This example highlights the challenge of applying the Fourier series to discontinuous functions:

```

dx = 0.01; L = 10;
x = 0:dx:L;
n = length(x); nquart = floor(n/4);
f = zeros(size(x)); f(nquart:3*nquart) = 1;
A0 = sum(f.*ones(size(x)))*dx*2/L;
fFS = A0/2;
for k=1:100
Ak = sum(f.*cos(2*pi*k*x/L))*dx*2/L;
Bk = sum(f.*sin(2*pi*k*x/L))*dx*2/L;
fFS = fFS + Ak*cos(2*k*pi*x/L) + Bk*sin(2*k*pi*x/L);
end

```

```
plot(x,f,'k','LineWidth',2), hold on  
plot(x,fFS,'r-','LineWidth',1.2)
```