NPTEL (Python)

Variables: → x="Sonu" → string.

concacteration ways. print("Hello", t_variable), Print(a, end=" ")
or print("Hello" + t_variable)

★ # ',' adds a space too.
↓
For commenting.

Input: →

n = input() → Simple input. (String)

n = input("Enter") Takes input with a text first. on same line

n = int(input())
↳ integer input.

and → str * 5 = strstrstrstrstr

S = int(c) # S will become integer.
string

a,b = map(int, input().split())

Inputting an array

array = input().split()

or

a = [int(i) for i in input().split()]

Scanned with CamScanner

IF-ELSE

```
if (condition):
        statements          # use indentation otherwise
                            # there will be indentation
elif(    ):                 # error
else:
```

```
print (a if a>b else b, end="")
```

For loop:-

```
for i in range(100):
    print("Sorry")
```

While loop:-

```
while (cond^n):
    print("    ")
```

```
for in range (0,20, 2):
    print(i)
```

>> all even nos from 0 to 19

Lists in Python

- List is a flexible data structure. These are ordered, changeable and allow duplicate values.

```
shopping = [ ]          # empty list
shopping = ["Bread", "Coffee", "Sugar"]
```

- Two ways to print :-

```
print(shopping)     or.     for item in shopping:
                                print(item)
```

Variable

- Adding element at last.

```
shopping.append("curd")
```

Scanned with CamScanner

- we can pick elements of list like.

  Shopping [0] = 'Bread'

- Inserting at specified position

  Shopping. insert (1, "Oil")
  index → item

- count.

  List = [12, 1, 1, 4, 12, 12, 14, 12, 12]
  list. count (12) # count the occurence.
  >>> 5

- len (ages) / len (list) → length of list.
  >>> 9

- ages. Sort () # sorts the list in ascending order.
- ages. reverse () # reverses the list.

- list_name [start index : end index + 1]. ... # Slicing.

  students [:] # all elements are printed.
  # default start index is 0.
  & default end index is len(list)

Crowd computing :-

By Trimmed mean. (remove 10% from start and 10% from end) (from sorted.)
→ Sort & then remove then take average.

Scanned with CamScanner

* A point to remember here is that there is multiplication of $\frac{10}{100}$ (10%) with length of list then answer is in float. So typecast into int by. final = int (0.1 * len(list))

first remove from beginning by.

List = List [final :] # slicing

then from end.

List = list[: len(list) - final]

Now, for mean, import libraries

- from statistics import mean.

then,

- mean(list)

* We can import scipy to do it easy way.

from scipy import stats.

List = [: . . .]

List.sort ( )

m = stats.trim_mean (list, 0.1) 10%

* Plotting. (matplot lib)

import matplotlib.pyplot at plt.

plt.plot ([1, 2, 3, 4])

plt.plot([y points], [x points], 'design')

plt.ylabel ("y-axis values")

plt.xlabel ("x-axis values")

Scanned with CamScanner

design = 'ro' 'r--' 'bs' 'gt' 'g^)

red dot — blue square — green triangle.

```
import statistics.
statistics.mean(List)
statistics.median(List)
```

—x—

Permutation / jumbed words.

```
import random
list = ['...', '', ...]
pick = random.choice(list)
```
→ choose any element from list

```
random.sample(word, len(word))
```

word is a string. This will randomly arrange the letters from 0 to len(word) of string word.

But this will return a list.

To join the list again,

We have to,

```
"".join(random.sample.(word, len(word)))
```

File handling:-

```
with open("file1.txt") as myfile:
    print(myfile.read())
```

```
with open("file1.txt", "r+") as myfile.
    myfile.write("I am fine")
```

★ then

```
myfile.close()
```
→ It has no indentation.

Scanned with CamScanner

Random :-

import random
random.randint (1, 5) # Both 1 & 5 are included.
~~random.randrange~~
random.randrange (1, 5) # 5 is NOT included.

random.random ( ) → generates decimal number for [0, 1).

random.randrange (1, 10, 2)
↳ common difference.

- This will generate random odd numbers.
- Slly we can randomly generate even nos too.

random.choice ([1, 2, 3, 4])
↳ List input. (chooses an element of list randomly.

Magic Square :-

$$Sum/M = \frac{n(n^2+1)}{2}$$

$n(3) = 15$
$n(4) = 34$

Steps :→
i) 1 is located at $(n/2, n-1)$
ii) Lets say that $(n/2, n-1)$ is (p, q) then next number which is 2 is located at $(p-1, q+1)$ position.
Anytime if row position becomes -1 then, make it $(n-1)$ and if column position becomes n then make it 0.
iii) If the calculated position already

contains a number, then decrement the colum by 2 and increment the row by 1.

4) If anytime row position becomes -1 and column becomes n, switch it to (0, n-2).

* inticlize the matrix to zero first. like.

```
square = [ ]
for i in range :
    dummy = [ ]
    for j in range :
        dummy.append (0)
    square.append (dummy)
```

& print this

OR.

```
magic_square = [[0 for i in range (3)] for
                j in range(3)].
```

Dobble Game (Spot the similarity).

```
Import String.
print (string.ascii_letter)
>>> ab..... zABC .... Z
card 1 = [0]x5
```

* When we choose randomly from list of letters. we must remove that letter from list. too.

Scanned with CamScanner

Birthday Paradox (Find twin).

First set random year, then ~~solo~~ check whether it is leap or not.

Then we choose month

if month is 2 then (1-29) if leap

else (1, 28)

and if month (Jan, March, May, July, Aug, Oct Dec)

then (1-31) else (1-30)

```
import datetime
var = datetime.date(year, month, day)
dd = datetime.date(year, month, day)
day_of_year = dd.timetuple().tm_yday
```

↓ variable

```
datetime.date.today()
>>> datetime.date(2021, 08, 11)

datetime.date.today().strftime("%Y")
>>> 2021

datetime.date.today().strftime("%B")
>>> August

datetime.date.today().strftime("%d")
>>> '31'

datetime.date.today().strftime("%W"))
>>> 22
```

22 wa week has year ka aisa batata hai

Scanned with CamScanner

weekday of week.

datetime.day.today().strftime("%w")

>>> 4 # kon sa day hai week ka, ye batayegga.

Daly of year :-

datetime.day.today().strftime("%j")

>>> 151

Day of week :-

datetime.day.today().strftime("%A")

>>> Thursday.

datetime.datetime.now()

>>> datetime.datetime(2018, 5, 31, 10, 48, 40, 824323)

☆ Guess the Movie Name :-

IMdB → Internet Moving ~~Dabo~~ Database.

make a list first then initialize turn 1, 2 then ~~check~~ create the an out the picked ~~word~~. word.

Scanned with CamScanner

Joining elements of list →

', '.joint (str(x) for x in list)

~~list.count~~ . str.count (letter)

↳ counts the occurence of letter in string.

Scanned with CamScanner

Week-5

Introduction to dictionaries.

key & # value.
↓ unique identifier for a given value ↓

Initializing dictionary :→ (By curly brackets)

```
var = { }
var['Key 1'] = value 1        → Adding new item
print(var)
    >>> {'key 1': value 1}
var['Key 2'] = value 2

print(var['Key 2'])
    >>> value 2.
```

var.keys() >>> dict_keys(['key 1', 'key 2'])
var.values() >>> dict_values(['value 1', 'value 2'])

or, for better look
list(var.keys()) >> ['key 1', 'key 2']
list(var.values()) >> [value 1, value 2]

list(var.items()) >> [('dollar', 60), ('euro', 70)]

☆ var.pop('key1') >>> remove specified key & return the corresponding value.

If key1 is not found, value1 is returned if given, otherwise Key error is raised.

☆ var['key 1'] = 5000
✓ updates the value.

☆ del conv_factor['yen']
del var['key 2']

Scanned with CamScanner

Set
'Key 1' = 50

e = 30
r = e * var['key 1']
30 * 50 ← value aayega key nhi
>>> r = 1500

Speech to text

- sudo pip install Speech Recognition

create an audio file
★★ and convert it to .wav extension

```
import speech_recognition as sr
AUDIO_FILE = ("name.wav")

# initialize recognizer
r = sr.Recognizer()

with sr.AudioFile(AUDIO_FILE) as source:
    audio = r.record(source)

★ try:
    print(r.recognize_google(audio))
★ except sr.UnknownValueError:
    print("Unknown audio/ not understandable")
★ except sr.RequestError:
    print("couldn't get results from
          Google speech recognition")
```

google Api (50 per day)

(except is an error type.

Monte Hall (3 doors & a twist)

3 door → (2 empty , 1 → prize)
after a choice is made then the other than chosen door with nothing (empty door) is opened and the player is asked to change (swap) or retain the intial choice.

★★ Swapping is better here as changes of winning are more.

?

★ Rock paper Scissors :-

0: rock 1: paper 2: scissor.

Here, the trick is we take any number, then ask user as the index e.g. 9513 index = 2
∴ 1 we got here now 1 % 3 = 1 → paper.

Slly for opponent ✓ then 6 if else statement

Initialize.
player 1 { 0: 'Rock', 1: 'Paper', 2: 'Scissor' }
player 2 { " }
Dictionary datatype.

Scanned with CamScanner

Sorting and Searching.

Binary search → Take mid → / compare → / right or left.

First Sorting required :→ Bubble sort
keep swapping the elements by its adjacent, untill all the elements are sorted.

$i = 0 ; i < n-1$

$j = 0 ; j < n-i-1$

** at last address(i), there will never be smaller element after the first loop.

```
def binarySearch (a, x):
    l = 0
    r = len(a) - 1
    flag = 0
    while (l <= r and flag = 0):
```

If no found then break loop instantly.

* # "//" is called integer division 3 // 2 = 1

```
        mid = (l + r) // 2
        If (x == a[mid]):
            flag = 1
            print (str(mid))
            return
        else :
            if (x < a[mid]):
                r = mid - 1
            else :
                l = mid + 1
    print ("No is not present")
```

Scanned with CamScanner

Substitution Cipher

(The science of secrecy)

String → It is used to represent texts in python.

```
import string.
x = "Hello"          # Enclose in double quote.

print (x[2:5])       >>> llo.
  len(x)             # length of string.

  x.upper()          # upper cased
  x.lower()          # lower case.

  x.replace("H", "J")   ✓
```

Make a dict of string.ascii_letters. then set the values to a letter previous than actual (a → Z , b = a)

(capital) ∵ a b c . . . z A B . . . Z

can be done by.

```
dict={ }
data = ""
for i in range(len(string.ascii_letters)):
    dict[string.ascii_letters[i]] =
        string.ascii_letters[i-1]
with open("text_file.txt") as f :
    While True :
        c = f.read(1)        # 1 is here to
        if not c :           read 1 by 1 each.
            print("khali file hai")
            break
        if c in dict :
            data = dict[c]
        else :
            data = c        → # is * * % _ - + etc.
        file.write(data)
```

Scanned with CamScanner

```
print(data)
file.close()
```

Tic tac toe

00	01	02
10	11	12
20	21	22

```
import numpy
board = numpy.array([['-','-','-'],['-','-','-'],['-','-','-']])
print(numpy.matrix(board))
```

It will convert the list into row & column format and print it.

Recursion :-

e.g. fact(6)

base condition giving is a must.
base or anchor case.

```
# recursive binary search.
def Binary_Search(l, x, start, end):
    if (start==end):
        if (l[start] == x):
            return start
        else:
            return -1
    else:
        mid = (start + end)//2
        if (l[mid] == x)
            return mid
        elif (l[mid] > x):
```

Base case.

return Binary_search (l, r, start, mid-1)

else:

return Binary-search (l, r, mid+1, end)

first sort it ★★ Binary search works on only sorted case.

★ fibonacci series ✓

Week-7

Snakes & ladders ✗ (Not on the board)

★ More & more abstraction should be there.
A good programming practice, to hide the implementation details. (Higher level overview, code should be organized such that it gives simple idea of code.

→ for opening, manipulating, & saving many different image file formats

```
from PIL import Image

img = Image.open('image1.jpg')
img.show()
```

We can input a raw string using.

```
s = raw_input('')
```

Ans → (2) + Week-7.

14 74 98 100

[6] [3] [5] [1] [1]

Scanned with CamScanner

comparing 2 lists
cmp(l1, l2)

Spiral Traversing.

By varying i & j of a matrix we can traverse in spiral manner.

1 2 3 4
12 13 14 5
11 16 15 6
10 9 8 7

* import turtle
* turtle.bgcolor ("black")

* Sewrat = turtle.Turtle()
* width = 5
* height = 7
  dot_distance = 25
* Sewrat.setpos(-250, 250.

* Sewrat.color = ("white")

  Sewrat.right(90) → Angle in degrees. → all for the traversing only right.

  Sewrat.forward(dot_distance)

at last
turtle.done()

---

turtle.dot( ) → Puts a dot.
turtle.penup() → Doesn't draw anything

turtle.back(5)
Turtle.bk(5)
turtle.backward(5)
} Sb chalta hai.

Scanned with CamScanner

GPS (global positioning system)

GPS gives information in latitudes & longitudes.

CSV file → (comma separated value) ~~latit, longitude~~
latitude, longitude.

```
import CSV
with open('route.csv', 'r') as f:
    reader = CSV.reader(f)

    for row in reader:
        lat = row[0]
        long = row[1]
        print (lat)
        print (long)
        print (lat + long)
```

→ Type cast into float.

```
import gmplot
or
from gmplot import gmplot        # create the map plotter.

gmap = gmplot.GoogleMapPlotter (latitud, longitude, zoom resolution)
gmap.coloricon = "https://www.googlemapsmarkers.com/v1/%s/"
with open('route.csv', 'r') as f:
    reader = CSV.reader(f)
    k = 0.
    for row in reader:
        lat = float (row[0])
        long = float (row[1])
```

Scanned with CamScanner

```
    if (k==0):
        gmap.marker(lat, long, 'yellow')
        k=1
    else:
        gmap.marker(lat, long, 'blue')

gmap.marker(lat, long, 'red')
gmap.draw("my map.html")
```

Week-8

Tuples :→ It is ordered & unchangeable (add or remove cant be done).
It allows duplicates.

ice-cream_flavours = ("vanilla", "chocolate", "butterscotch", "strawberry")

we can print whole by :-

```
* print(ice_cream_flavors)
```

or for individual elements :

```
* for i in ice_cream_flavours:
      | print(i)
```

or

```
print(ice_cream_flavours[2])
>>> butterscotch.
```

★★★ if we are making Tuple of 1 item, ADD , (comma) at last

```
var_tuple = ("apple",)
```

Index Error → This occurs when the index is out of range.

Scanned with CamScanner

'tuple' object does not support item assignment.
" " " item deletion.

del ice-cream-flavours [1] ✗

del ice-cream-flavours ✓ Whole at once.

☆ we can reassign tuple and here, the previous tuple get destroyed.

```
toy = (1, 3, 4)
toy = (1, 2, 3, 4, 1, 2, 3, 4)
toy.count(2)
>> 2
toy.index(5)
>>
toy.index(2)
>> 1
```

e.g. rainbow colours, pixels storing

Lottery Simulation (profit or loss)

coupons available / Bet on 1 / They will draw a coupon randomly & the coupon that appear matches then we get more money otherwise loose.

☆ You can define infinity in Python using float('inf')

☆☆ Try (import this)
☺ To get a poem.

Scanned with CamScanner

We can check memory usage of an object x using the command sys.getsizeof(x)

```
import matplotlib.pyplot as plt
    x = []
    y = []
    for each day of year
        x.appen[i+1]
    & y.append(account)

plt.plot(x,y)
plt.show()
```

Image Processing (Enhance your images)

The type() function returns the class of the argument the object belongs to.

Flipping the image. PIL (Python Imaging Library)

```
from PIL import Image

img = Image.open("imge source.png")
#Transposing (rows as column & col as rows)

transposed_img = img.transpose(Image.FLIP_LEFT_RIGHT)
# save it to a file in a human readable/understandable format.

transposed_img.save('final_mirrored.png')
print("Task completed")
```

Scanned with CamScanner

(Image enhancement Technique)
CLAHE = Contrast limited adaptive histogram equalization.

Image enhancement :—
(Histogram equalization / adaptive histogram equalization / clahe)

```
import CV2
img = CV2.read ('crime.png')
# preparation for CLAHE
clahe = CV2.create CLAHE ()

# convert to grey scale image (better for enhancement)

gray-img = CV2.cvtColor (img, cv2.COLOR_BGR2GRAY)
# Apply enhancement

enh_image = clahe.apply (gray_img

CV2.imwrite ('output.png', enh_img)
print ("done")
```

Anagrams :—

Hook's law :→ spring. (Force ∝ stretch.
Cat , ACT
Study , dusty.

ASCII → American Standard Code for Information Interchange.
ASCII values represent, character encoding.

Scanned with CamScanner

time.time () returns the current time in ms since midnight, Jan 1, 1970

```
x = [4, 3, 1, 2]
print ( sorted (x))
>>> [1, 2, 3, 4]
```

★ It also works on char, list & strings ✓

```
x = ['a', 'c', 'b']
print (sorted (x))
>> ['a', 'b', 'c']
```

```
x = 'python'        Print (sorted (x))
>>> ['h', 'n', 'o', 'p', 't', 'y']
```

If we sort the dictionary using sorted( ) then the keys are sorted.

```
# L = ["cccc", "bb", "a", "ddd"]
print (sorted ( L, key = len)
```

↳ sorted according to length

```
>>> ['a', 'bb', 'ddd', 'cccc']
```

★ How to check anagrams.

Sort both and compare, bas!.

Scanned with CamScanner

Facebook Sentiment Analysis.

positive connotation → I am proud of my country.

negative connotation → Life isn't fair as I expect it to be.

neutral connotation → Set your goals right.

Sentiment of a given sentence :-

i) Download facebook data. (over 10 years & more)

ii) Sentiment analysis (analyse the data)

'as' keyword is used to create the alias for module imported.

[eg. import math as mymath]

Facebook → settings → your Facebook Inform → Download your inf → create file (↓) zip

copy the html file & paste it in ms excel.

Now,

★ import pandas # it provides easy to use # data str. for data analysis.

★ import nltk # It is used to process human language.

nltk provides sentiment analysis of human data. sentiment analysis involves working out whether a piece of text is positive, negative or neutral.

★ We will use VADER (Valence Aware dictionary and Sentiment reasoner)

It also takes into account the intensity of the sentiment.

Scanned with CamScanner

Download VADER lexicon
Lexicon acts as a dictionary here.

Convert the excel sheet to data frame with the help of pandas. A data frame is a 2-D structure in form of a table.

ord( ) converts a character into its ascii notation.

```
import pandas as pd.
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
nltk.downloader.download('vader_lexicon')

file = '/path/datafile.xslsx'

xl = pdExcelFile(file)
dfs = xl.parse(xl.sheet_names[0])        # Parsing excel
                sheet to dataframe.
dfs = list(dfs['Timeline'])

Sid = SentimentIntensityAnalyzer()

ss = sid.polarity_scores(data)
```

Scanned with CamScanner

call function tuple ( ) for converting a list into a tuple.

Week-9

Stylometry with python.

↳ It is the quantity to study literacy cycle. It is based on observation that authors tend to write in relatively consistent, recognizable & unique ways.

An authors stylic signature can be analyzed by:—

Kilgariff's chi-squared method.

John Burrow's delta method

or plotting a graph of word length distribution

nltk can be used here too ✓

☆ nltk.word_tokenize(Dictionary where para are present).

letter words everything will be tokenized.

c.isalpha → It means whether it contains only alphabets.

☆ author_tokens[author] = ([token for token in tokens

if any(~~c.isalpha~~ c.isalpha() for c in token)])

☆ length_distribution [authors] = nltk.FreqDist (token_lengths)

length_distribution [authors].plot (15, title = author)

Scanned with CamScanner

NetworkX. (For plotting, visualizing graphs)

```
import networkx as nx
G = nx.Graph()          # will create empty graph.
G.add_node(1)
G.add_node(2)
G.add_node(3)
G.add_edge(1,2)
G.add_edge(2,3)
G.add_edge(1,3)

print(G.nodes())        >>> [1, 2, 3]
```

It will not visualise the whole graph.

```
print(G.edges())
                        >> [(1,2), (1,3), (2,3)]
```

so, import matplotlib.pyplot.

```
nx.draw(G)
plt.show()              >>>
```

we can add multiple nodes at once by list.

```
l = [1, 2, 3]
G.add_nodes_from[l]     → any kind of data.   >>>
nx.draw(G)
plt.show()
```

or.

```
☆☆ G = nx.complete_graph(10)
nx.draw(G)
plt.show()
```

Scanned with CamScanner

index (ele, beg, end) function returns the index of first occurence of element after beg. & before end.

Nodes Probability

nx.gnp_random_graph(50, 0.3)

nx.write_gexf (G, "saving-directory-name")

↓ Graph

gexf format for graphs.

Download ~~gefffy~~ gephi for visualizing graphs.

Graph

connected graph → atleast one path exists between any two nodes.

→ One format of representation of graph = edge list.

Reading facebook data :-

G = nx.read_edge list ("facebook-combined.txt")

Now compare shortest path link.

```
n = list(G.nodes())
spll = [ ]          # shortest path linked list.
for u in n:
    for v in n:
        if u != v :
            l = nx.shortest_path_length (G, u, v)
            print ("shortest path between u & v
                is ", l)
            spLL.append(l)
```

source target

```
min_spl = min(spll)
max_spl = max(spll)
```

Scanned with CamScanner

maximum & ~~minume~~ minimum path length can be found by min ( ) & ma( )

But for average :-

avg_spl = numpy.average (spll)

Area Calculation :-

Image → It consists of matrix of pixals. A pixel is basically a ~~clor~~ colour. A ~~coul~~ colour is defined by 3 attributes.

- ★ Amount of red
- ★ " blue
- ★ " grey

```
import numpy as np
from PIL import Image
width = 5
Height = 4
array = np.zeroes([height, width, 3], dtype = np.uint8)
```

bytes for R, G & B

data type

unsigned int

```
img = Image.fromarray(array)
```

```
img.save('test.png')
```
>>> → Black.

To convert image to RGB :-

```
from PIL import Image.
im = Image.open('.....png')
rgb_image = im.convert('RGB')
```

```
r, g, b = rgb_image.getPixel((1,1))
```

x y → location of pixel.

Scanned with CamScanner

import scipy.misc # we can read image through this.

imag = scipy.misc.imread("map-01.png")

Week-10

FLAMES Friend love Affection Marriage Enemy Sister.

String = "Ravish"
print (s.lower())
print (s.upper())

numpy → numerical python.
↓
For matrix operations.

Import numpy as np

a = np.array([1,2,3])

print (a)
>>> [1, 2, 3]

print(type(a)) >>> <class 'numpy.ndarray'>

print (a.shape)
>>> (3,)
↳ No of element in 1D/row

↳ If 2D array (2, 3)
↓ row ↳ colum.

a = np.zeros((2, 2))
↓ ↓ >>> [0 0
0 0]

Ily np.ones((i,j)) i j

The no. from which 2-D array will be filled

C = np.full((2,2), 6)

For random nos in array → np.random.random((2,2)) >>> [0 - 1 ke bich ka X 4 place]

x = np.array([[2,1],[1,4]])
print (x.dtype)
>>> int 32 where x is numpy array

x = np.array([[3.0, 4.3], [2.1, 1.9]])
print (x.dtype)
>>> float 64

x = np.array ([1, 2], dtype = np.int 64) # Forcing the datatype

we can do matrix addition / sub.
by:
* print (np.add (x,y))
* print (np.subtract (x,y))
or
* print (x-y) ✓ or print (x+y)
* print (x * y) → Item by item multiplication Not actual.

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = [19.$

* print (x/y)
* print (np.sqrt (x))
* print (np.
or
print (x.T) # transpose.
* print (np.sum (x)) → sum of element.

Scanned with CamScanner

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ Row wise

print (np.sum(x, axis = 0)) >>> [3, 7]

print (np.sum(x, axis = 1)) >>> [~~6~~]

↳ column wise [4, 6]

compression
- Loseless
- Loosy.

```
im = Image.open('lena.jpg')

pixelMap = im.load( )
I = numpy.asarray(im)
```

numpy.asanyarray → to show image as array

```
img = Image.new(im.mode, im.size)
```

→ copy image as previous

→ gray, etc.

```
pixelNew = img.load()
```

Scanned with CamScanner

Week-11.

Browser Automation.

Selenium library : → → → Download chrome driver (Type of webdriver)
different for different browsers.

```
from Selenium import webdriver.

browser = webdriver.Chrome("/Users..../Downloads/
                            chrome driver - 2")
                            # location of web driver.
browser.get("https://www.selenium.org")

elem = browser.find_element_by_link_text('Download')

elem.click().   # for clicking the link download.

search = browser.find_element_by_id('q')
search.send_keys('Download')
```

★ Found search bar.

Whats app automation

★ Day (Fun with calendar)

Libraries :-

calendar
datetime
pytz → timezone

time.time( )

current time.

datetime.now( )

>>> datetime.datetime (2018, 10, 4, 16, 10, 33, 558801)

import pytz

tz = pytz.timezone ('singapore')

print (datetime.now(tz))

>>> 2018-10-04 18:55:13.000001 08:00

pytz.all_timezones ✓ all list of 591 time zon.

import calendar

calendar.weekday(year, month, day)

>>> integer (0-6)

0 → Monday

1 → Tuesday.

$\boxed{\text{fabs()}}$ → absolute value.

Scanned with CamScanner