

1) Programs on regular expressions

```
import re

s='Welcome to cyber security 2nd year'

match =re.search(r'cyber',s)

print('Start index:',match.start())

print('End index:',match.end())
```

OUTPUT:-

Start index: 11

End index: 16

2) To find all the characters in the string that fall within the range of 'a' to 'm' using find all the methods

```
import re  
  
string="The quick brown fox jumps over the lazy dog"  
  
pattern="[a-m]"  
  
result=re.findall(pattern,string)  
  
print(result)
```

OUTPUT:-

```
['h', 'e', 'i', 'c', 'k', 'b', 'f', 'j', 'm', 'e', 'h', 'e', 'l', 'a', 'd', 'g']
```

3) To check if a list of strings are matched or unmatched

```
import re

regex=r'The'

strings=['The quick brown fox', 'The lazy dog','A quick brown fox']

for string in strings:

    if re.match(regex,string):

        print(f'matched:{string}')

    else:

        print(f'Not matched:{string}')
```

OUTPUT:-

matched:The quick brown fox

matched:The lazy dog

Not matched:A quick brown fox

4) To check if the string ends with "World!"

```
import re  
  
string="Hello World!"  
  
pattern=r"World!$"   
  
match=re.search(pattern,string)  
  
if match:  
  
    print("Match found!")  
  
else:  
  
    print("Match not found.")
```

OUTPUT:-

Match found!

5) To search for the pattern "brown.fox" within the string.

```
import re

string="The quick brown fox jumps over the lazy dog."

pattern=r"brown.fox"

match=re.search(pattern,string)

if match:

    print("Match found!")

else:

    print("Match not found.")
```

OUTPUT:-

Match not found.

6) python programming using numbers in pyhton

```
a=4
```

```
b=4.5
```

```
c=2j
```

```
print (type(a))
```

```
print (type(b))
```

```
print (type(c))
```

Output:

```
<class 'int'>
```

```
<class 'float'>
```

```
<class 'complex'>
```

7) using arithmetic operations on int type

```
res=5+3
```

```
print (res)
```

```
res=5-3
```

```
print (res)
```

```
res=5*3
```

```
print (res)
```

```
res=15/4
```

```
print (res)
```

```
res=15//4
```

```
print (res)
```

```
res=15%4
```

```
print (res)
```

```
res=2**3
```

```
print (res)
```

```
res=abs(-10)
```

```
print (res)
```

```
res=round(3.14159,2)
```

```
print (res)
```

Output:

```
8      3
```

```
2      8
```

```
15     10
```

```
3.75   3.14
```

```
3
```

8) Using arithmetic operations on complex type

```
res=(3+4j)+(1+2j)
```

```
print (res)
```

```
res=(5+6j)-(2+3j)
```

```
print (res)
```

```
res=(2+3j)*(1+4j)
```

```
print (res)
```

```
res=(8+6j)/(2+3j)
```

```
print (res)
```

```
res=(1+1j)**2
```

```
print (res)
```

```
res=abs(3+4j)
```

```
print (res)
```

```
res=(3+4j).conjugate()
```

```
print (res)
```

```
real=(3+4j).real
```

```
imag=(3+4j).imag
```

```
print (real)
```

```
print (imag)
```

Output:

(4+6j)	2j	4.0
(3+3j)	5.0	
(-10+11j)	(3-4j)	
(2.6153846153846154-0.9230769230769231j)	3.0	

9) Using built in functions

```
a=2
```

```
print(float(a))
```

```
b=5.6
```

```
print(int(b))
```

```
c='3'
```

```
print(type(int(c)))
```

```
d='5.6'
```

```
print(type(float(d)))
```

```
e=5
```

```
print(complex(e))
```

```
f=6.5
```

```
print(complex(f))
```

Output:

```
2.0
```

```
5
```

```
<class 'int'>
```

```
<class 'float'>
```

```
(5+0j)
```

```
(6.5+0j)
```

10) Generating random integers

```
import random  
  
x=random.randint(1,100)  
  
print x
```

Output:

32

11) Using arithmetic operations

a=1.6

b=5

c=a+b

print(c)

Output:

6.6

12) Generating floating point integer

```
import random
```

```
x=random.uniform(1,100)
```

```
print(x)
```

Output:

92.43244874215033

13)Nan, inf and -inf functions

```
import math
```

```
n=math.nan
```

```
print(n)
```

```
x=float('inf')
```

```
x=float('-inf')
```

```
print(x)
```

```
print(y)
```

Output:

```
nan
```

```
inf
```

```
-inf
```

Programs on Array module

14) Create an array in python

```
import array as arr  
a=arr.array('i',[1,2,3])  
print(a[0])  
a.append(5)  
print(a)
```

Output:

```
1  
array('i', [1, 2, 3, 5])
```

15) Adding elements to an array

```
import array as arr  
a=arr.array('i',[1,2,3])  
print("integers array before instertion :",*a)  
a.insert(1,4)  
print("integer array before instertion:",*a)
```

Output:

integers array before instertion : 1 2 3

integer array before instertion: 1 4 2 3

16) Accesing array items

```
import array as arr  
a=arr.array('i',[1,2,3,4,5,6])  
print(a[0])  
print(a[3])  
b= arr.array('d',[2.5,3.2,3.3])  
print(b[1])  
print(b[2])
```

Output:

```
1  
4  
3.2  
3.3
```


17) Removing elements from array

```
import array  
  
arr=array.array('i',[1,2,3,1,5])  
  
arr.remove(1)  
  
print(arr)  
  
arr.pop(2)  
  
print(arr)
```

Output:

```
array('i', [2, 3, 1, 5])  
  
array('i', [2, 3, 5])
```

18) Slicing an array

```
import array as arr  
l=[1,2,3,4,5,6,7,8,9,10]  
a=arr.array('i',l)  
sliced_array=a[3:8]  
print(sliced_array)  
sliced_array=a[5:]  
print(sliced_array)  
sliced_array=a[:]  
print(sliced_array)
```

Output:

```
array('i', [4, 5, 6, 7, 8])  
array('i', [6, 7, 8, 9, 10])  
array('i', [1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
```

19) Searching element in an array

```
import array  
arr=array.array('i',[1,2,3,1,2,5])  
print(arr.index(2))  
print(arr.index(1))
```

Output:

1

0

20) Updating elements in an array

```
import array  
arr=array.array('i',[1,2,3,1,2,5])  
arr[2]=6  
print(arr)  
arr[4]=8  
print(arr)
```

Output:

```
array('i', [1, 2, 6, 1, 2, 5])  
array('i', [1, 2, 6, 1, 8, 5])
```

21) Counting elements in array

```
import array  
  
arr=array.array('i',[1,2,3,4,2,5,2])  
  
count=arr.count(2)  
  
print("Numbers of occurrence of 2:",count)
```

Output:

Numbers of occurrence of 2: 3

22) Reversing elements in an array

```
import array  
  
arr=array.array('i',[1,2,3,4,5])  
  
arr.reverse()  
  
print("reversed array:",*arr)
```

Output:

reversed array: 5 4 3 2 1

23) Extend elements from array

```
import array as arr  
a=arr.array('i',[1,2,3,4,5])  
a.extend([6,7,8,9,10])  
print(a)
```

Output:

```
array('i', [1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
```

24)Program to display checkbox

```
from tkinter import *

class Mycheck:

    def __init__(self, root):

        self.f = Frame(root, height=350, width=500)

        self.f.propagate(0)

        self.f.pack()

        self.var1 = IntVar()

        self.var2 = IntVar()

        self.var3 = IntVar()

        self.c1 = Checkbutton(self.f, bg='yellow', fg = 'green', font=('Georgia', 20, 'underline'),
text='Java', variable= self.var1, command=self.display)

        self.c2 = Checkbutton(self.f, text='Python', variable= self.var2, command=self.display)

        self.c3 = Checkbutton(self.f, text='.NET', variable= self.var3, command=self.display)

        self.c1.place(x=50, y=100)

        self.c2.place(x=200, y=100)

        self.c3.place(x=350, y=100)

    def display(self):

        x = self.var1.get()

        y = self.var2.get()

        z = self.var3.get()

        str = ""

        if x==1:

            str += 'java'

        if y==1:
```



```
str+= 'python'

if z==1:

    str+= '.NET'

lbl = Label(text=str, fg='blue').place(x=50, y=150, width=200, height=20)

root = Tk()

mb = Mycheck(root)

root.mainloop()
```

Output:



25) Program to display Label

```
from tkinter import *

class MyButtons:

    def __init__(self, root):

        self.f = Frame(root, height=350, width=500)

        self.f.propagate(0)

        self.f.pack()

        self.b1 = Button(self.f, text='Click Me', width=15, height=2, command=self.buttonClick)

        self.b2 = Button(self.f, text='Close', width=15, height=2, command=quit)

        self.b1.grid(row=0, column=1)

        self.b2.grid(row=0, column=2)

    def buttonClick(self):

        self.lbl = Label(self.f, text="welcome to python" , width=20, height=2, font=('Courier', -30,
        'bold underline '),fg='blue')

        self.lbl.grid(row=2, column=0)

root = Tk()

mb=MyButtons(root)

root.mainloop()
```

Output:



26) Program to display buttons

```
from tkinter import *

class MyButton:

    def __init__(self,root):

        self.f = Frame(root, height=400 , width=500)

        self.f.propagate(0)

        self.f.pack()

        self.b1 = Button(self.f, text='Red' , width=15 , height=2,command=lambda:
self.buttonClick(1))

        self.b2 = Button(self.f , text='Green' , width=15, height=2,command=lambda:
self.buttonClick(2))

        self.b3 = Button(self.f , text='Blue' , width=15, height=2,command=lambda:
self.buttonClick(3))

        self.b1.pack()

        self.b2.pack()

        self.b3.pack()

        def buttonClick(self,num):

            if num==1:

                self.f["bg"] = 'red'

            if num==2:

                self.f["bg"] = 'green'

            if num==3:

                self.f["bg"] = 'blue'

root=Tk()

mb=MyButton(root)

root.mainloop()
```

Output:



27) Program to display radiobuttons

```
from tkinter import *

class Myradio:

    def __init__(self , root):

        self.f = Frame(root, height=350, width=500)

        self.f.propagate(0)

        self.f.pack()

        self.var = IntVar()

        self.r1 = Radiobutton(self.f, bg='yellow', fg= 'green' ,

                               font=('georgia' , 20, 'underline') , text='male', variable=self.var, value=1,

                               command=self.display)

        self.r2 = Radiobutton(self.f, text='Female' , variable=self.var, value=2,

                               command=self.display)

        self.r1.place(x=50, y=100)

        self.r2.place(x=200, y=100)

    def display(self):

        x = self.var.get()

        str = "

        if x==1:

            str += 'You selected: Male '

        if x==2:

            str += 'You selected: Female '

        lbl= Label(text=str, fg='blue').place(x=50, y=150, width=200, height=20)

root = Tk()

mb= Myradio(root)

root.mainloop()
```

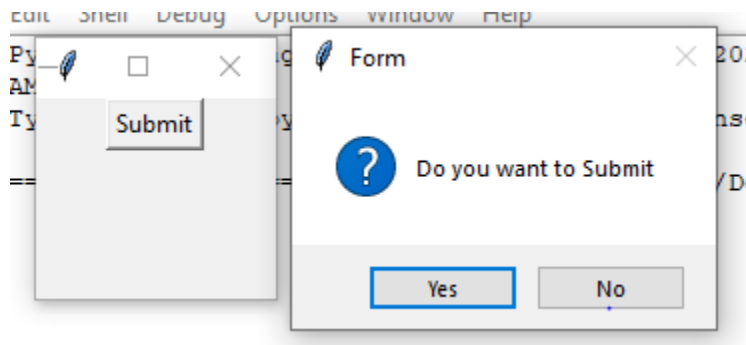
Output:



28) Program to display Dialouge box:

```
from tkinter import *  
  
from tkinter import messagebox  
  
main = Tk()  
  
def Submit():  
    messagebox.askquestion("Form", "Do you want to Submit")  
  
main.geometry("100x100")  
  
B1 = Button(main, text="Submit", command=Submit)  
  
B1.pack()  
  
main.mainloop()
```

Output:



29) Program to display Radio Button 2 :

```
import tkinter as tk

from tkinter import messagebox

def show_choice():

    choice = language.get()

    if choice:

        messagebox.showinfo("Choice", f"You chose: {choice}")

    else:

        messagebox.showwarning("Choice", "Please select a language.")

root = tk.Tk()

root.title("Language Choice")

language = tk.StringVar()

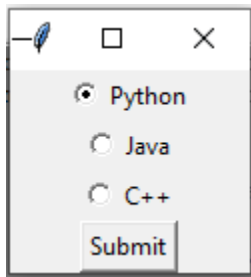
for lang in ["Python", "Java", "C++"]:

    tk.Radiobutton(root, text=lang, variable=language, value=lang).pack()

tk.Button(root, text="Submit", command=show_choice).pack()

root.mainloop()
```

Output :



30) Program to display creation of home

```
from tkinter import *

root=Tk()

c=Canvas(root, bg="#091e42", height=700, width=1200)

c.create_polygon(600,250,700,200,800,250,800,400,600,400,width=2, fill="yellow",
outline="red")

c.create_line(600,250,800, 250, width=2, fill="red")

c.create_rectangle(650,275,750,375, fill="red")

x1,y1=0,350

x2,y2=200,450

for i in range(3):

    c.create_arc(x1,y1,x2,y2, start=0, extent=180, fill="green")

    x1+=200

    y1+=200

c.create_arc(800,350,1000,450, start=0, extent=180, fill="green")

c.create_arc(1000,350,1200,450, start=0, extent=180, fill="green")

id=c.create_text(600,600, text="My Happy Home!!", font=('Helvetica', 30, 'bold'),
fill="magenta")

c.pack()

root.mainloop
```

Output:



31) Program to display menu

```
from tkinter import *

class MyMenuDemo:

    def __init__(self, root):

        self.menubar = Menu(root)

        root.config(menu=self.menubar)

        self.filemenu= Menu(root, tearoff=0)

        self.filemenu.add_command(label="New", command=self.donothing)

        self.filemenu.add_command(label="Open", command=self.donothing)

        self.filemenu.add_command(label="Save", command=self.donothing)

        self.filemenu.add_separator()

        self.filemenu.add_command(label="exit", command=root.destroy)

        self.menubar.add_cascade(label="file", menu=self.filemenu)

        self.editmenu = Menu(root, tearoff=0)

        self.editmenu.add_command(label='cut', command=self.donothing)

        self.editmenu.add_command(label='copy', command=self.donothing)

        self.editmenu.add_command(label='paste', command=self.donothing)

        self.menubar.add_cascade(label="edit", menu=self.editmenu)

    def donothing(self):

        pass

root = Tk()

root.title("A Menu Example")

obj= MyMenuDemo(root)

root.geometry('600x350')

root.mainloop()
```

Output:

