

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

CHAPTER-4

Network Programming and GUI using Python

- Network Programming:
- Protocol, Sockets,
- Knowing IP Address,
- URL, Reading the Source Code of a Web Page,
- Downloading a Web Page from Internet,
- Downloading an Image from Internet,
- A TCP/IP Server, A TCP/IP Client,
- A UDP Server, A UDP Client,
- File Server, File Client,
- Two-Way Communication between Server and Client,
- Sending a Simple Mail.
- GUI Programming:
- Event-driven programming paradigm;
- creating simple GUI;
- buttons, labels, entry fields, dialogs;
widget attributes - sizes, fonts, colors ,layouts,nested frames

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

Q-1 What is Socket in network programming?

Detail :-

- Python provides two levels of access to network services.
- At a low level, you can access the basic socket support in the underlying operating system, which allows you to implement clients and servers for both connection-oriented and connectionless protocols.
- Python also has libraries that provide higher-level access to specific application-level network protocols, such as FTP, HTTP, and so on.

What is Sockets?

- Sockets are the endpoints of a bidirectional communications channel. Sockets may communicate within a process, between processes on the same machine, or between processes on different continents.
- Sockets may be implemented over a number of different channel types: Unix domain sockets, TCP, UDP, and so on. The socket library provides specific classes for handling the common transports as well as a generic interface for handling threst.

The *socket* Module

- To create a socket, you must use the `socket.socket()` function available in `socket` module, which has the general syntax –
- `s = socket.socket (socket_family, socket_type,`

`protocol=0)`

`Server`

`Socket`

`Method`

Sr.No.	Method & Description
1	<code>s.bind()</code> This method binds address (hostname, port number pair) to socket.
2	<code>s.listen()</code> This method sets up and start TCP listener.
3	<code>s.accept()</code> This passively accept TCP client connection, waiting until connection arrives (blocking).

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

Client Socket Methods

Sr.No.	Method & Description
1	s.connect()

This method actively initiates TCP server connection.

1 Word Question – Answer

SR.NO	QUESTION	ANSWER
1	To create Socket , you must use _____ module	Socket
2	_____ Method bind address to the socket.	Bind()
3	_____ method will accept the client connection.	Accept()

Q-2 IP Address in python.

- **IP (Internet Protocol)** -Address is the basic fundamental concept of computer networks which provides the address assigning capabilities to a network.
- Python provides *ipaddress* module which is used to validate and categorize the IP address according to their types(*IPv4 or IPv6*).
- This module is also used for performing wide range of operation like arithmetic, comparison, etc to manipulate the IP addresses.

Detail :

- **Validating IP Addresses**

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

□ **IPv4** : It is a 32-bit number typically written in decimal digits formatted as four 8-bit numbers separated by dots, is used to identify the network interface of a machine.

□ The `ip_address()` function throws an error if the range value exceeds from 0 to 255.

```
# Import module  
import ipaddress
```

```
# Example of valid IPv4 address  
print (ipaddress.ip_address(u'175.198.42.211'))
```

```
# Invalid IPv4 address raises error  
print (ipaddress.ip_address(u'175.198.42.270'))
```

Output :
`175.198.42.211`

➤ **IPv6** : It is represented by eight groups of four hexadecimal digits separated by colons, where each group represents 16 bits, two octets also known as hextet.

➤ The `ip_address()` function throws an error if the range value exceeds from 0 to FFFF.

```
# Import module  
import ipaddress
```

```
# Example of valid IPv6 address  
print
```

```
➤ (ipaddress.ip_address(u'2001:0db8:85a3:2bfe:070d:8a2e:0370:7334'))
```

➤ For validating IP addresses python uses `ip_address()` function provided by the `ipaddress` module which raises error if values of IP exceeds the range of the type of the IP address.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

```
# Invalid IPv6 address raises  
errorprint
```

```
(ipaddress.ip_address(u'2001:0db8:85a3:0ff0:0000:8a2e:0370:7334'))
```

Output :

```
2001:db8:85a3:2bfe:70d:8a2e:370:7334
```

- **Operations on IP Address-**

Various operations like arithmetic, comparison, type, etc can be performed on the IP addresses with the help of *ipaddress* module. Some operations are listed below:

- **Type Check operation:** The *type()* method takes various formats of IP addresses as input and recognizes whether it is IPv4 or IPv6 address, indicating the category of the IP address.

```
# Import module  
import ipaddress
```

```
# IPv4 address  
print(type(ipaddress.ip_address(u'175.198.42.211')))  
print(type(ipaddress.ip_address(u'5.0.0.1')))
```

```
# IPv6 address  
print(type(ipaddress.ip_address(u'2001:0db8:85a3:2bfe:070d:8a2e:0  
370:7334')))  
print(type(ipaddress.ip_address(u'0000:f0f0::7b8a:ffff')))
```

1 Word Question – Answer

SR.NO	QUESTION	ANSWER
1	_____ module support various operations performed on IP address.	ipaddress
2	_____ is represented by eight groups of four hexadecimal digits	IPv6
3	IP stands for _____	Internet Protocol

Q-3 Explain Protocol in detail.[2-3 marks]

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

Detail :-

- The Internet Protocol is designed to implement a uniform system of addresses on all of the Internet-connected computers everywhere and to make it possible for packets to travel from one end of the Internet to the other.
- A program like the web browser should be able to connect to a host anywhere without ever knowing which maze of network devices each packet is traversing on its journey.
- There are various categories of internet protocols.
- These protocols are created to serve the needs of different types of data communication between different computers in the internet.
- Python has several modules to handle each of these communication scenarios.
- The methods and functions in these modules can do the simplest job of just validating a URL or also the complex job of handling the cookies and sessions.
- In this chapter we will look at the most prominent python modules used for internet protocols.

Protocol	Python Module Name	Description
TTP	urllib.request	Opening the HTTP URL
HTTP	urllib.response	Create a reponse object for a url request
HTTP	urllib.parse	To break Uniform Resource Locator (URL) strings up in components like (addressing scheme, network location, path etc.),

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

HTTP	urllib.robotparser	It finds out whether or not a particular user agent can fetch a URL on the Web site that published the robots.txt file.
FTP	ftplib	implements the client side of the FTP protocol. You can use this to write Python programs that perform a variety of automated FTP jobs, such as mirroring other FTP servers.
POP	poplib	This module defines a class, POP3, which encapsulates a connection to a POP3 server to read messages from a email server
IMAP	imaplib	This module defines three classes, IMAP4, IMAP4_SSL and IMAP4_stream, which encapsulate a connection to an IMAP4 server to read emails.
SMTP	smtplib	The smtplib module defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP listner daemon.
Telnet	telnet	This module provides a Telnet class that implements the Telnet protocol to access a server thorough teleent.

1 Word Question – Answer

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

SR.NO	QUESTION	ANSWER
1	_____ is designed to implement a uniform system of addresses	Internet protocol
2	_____ Protocol can be used to open HTTP url.	HTTP
3	_____ protocol helps to support different file operations	FTP

Q-4 What is URL ? How to read source code of webpage.

Detail :

- URL stands for Uniform resource locator.
- The requests module can help us build the URLs and manipulate the URL value dynamically.
- Any sub-directory of the URL can be fetched programmatically and then some part of it can be substituted with new values to build new URLs.

What is urllib?

- urllib is a Python module that can be used for opening URLs. It defines functions and classes to help in URL actions.
- With Python you can also access and retrieve data from the internet like XML, HTML, JSON, etc. You can also use Python to work with this data directly.
- In this tutorial we are going to see how we can retrieve data from the web. For example, here we used a guru99 video URL, and we are going to access this video URL using Python as well as print HTML file of this URL.

Steps :-

- Import urllib
- Define your main function
- Declare the variable webUrl
- Then call the urlopen function on the URL lib library
- The URL we are opening is guru99 tutorial on youtube
- Next, we going to print the result code

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

- Result code is retrieved by calling the getcode function on the webUrl variable we have created
- We going to convert that to a string, so that it can be concatenated with our string “result code”
- This will be a regular HTTP code “200”, indicating http request is processed successfully

“how to get source code of website in python” :-

```
import requests
url = input('Webpage to grab source from: ')
html_output_name = input('Name for html file: ')
req = requests.get(url, 'html.parser')
```

```
with open(html_output_name, 'w') as f:
f.write(req.text)
f.close()
```

1 Word Question – Answer

SR.NO	QUESTION	ANSWER
1	_____ is a Python module that can be used for opening URLs .	urllib
2	URL stands for _____	Uniform Resource Locator
3	_____ module can help us build the URLs and manipulate the URL value dynamically.	request

Q-5 Explain TCP/IP server and TCP/IP client.

Detail :-

- Python is one of the most popular object-oriented scripting languages with a programmer-friendly syntax and a vast developer community.
- Here, we'll showcase how to write a TCP server and client in Python and

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

implement them using classes.

TCP/IP Client and Server

- Sockets can be configured to act as a *server* and listen for incoming messages, or connect to other applications as a *client*. After both ends of a TCP/IP socket are connected, communication is bi-directional.
- Hence, we'll keep our focus only on the workflow and example code of the Python TCP server and client.
- The sample contains the source code for a TCP server and client. For practice, you can extend it to build a small chat system or a local attendance tracking system.

Synchronous socket entities

- **TCPServer class** – It follows the (Internet) TCP protocol that allows continuous streams of data between the server and client.
- The **socketserver** module has more classes to handle sockets.
- Now let's see the example of the Python TCP Server.

Python-TCP-Server.py

```
import socketserver
```

```
class
```

```
    Handler_TCPServer(socketserver.BaseRequestHandler):
```

```
        """
```

```
        The TCP Server class for demonstration.
```

```
        Note: We need to implement the Handle method to exchange data  
        with TCP client.
```

```
        """
```

```
        def handle(self):
```

```
            # self.request - TCP socket connected to the client
```

```
            self.data = self.request.recv(1024).strip()
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

```
print("{}  
sent:".format(self.client_address[0]))  
print(self.data)  
# just send back ACK for data arrival confirmation  
self.request.sendall("ACK from TCP Server".encode())  
  
if __name__ == "__main__":  
    HOST, PORT = "localhost", 9999  
  
    # Init the TCP server object, bind it to the localhost on 9999 port  
    tcp_server = socketserver.TCPServer((HOST, PORT),  
    Handler_TCPServer)  
  
    # Activate the TCP server.  
    # To abort the TCP server, press  
    Ctrl-C.tcp_server.serve_forever()
```

In the next example code, you'll see the Python TCP client module code to communicate with the TCP server.

Python-TCP-Client.py

```
import socket  
  
host_ip, server_port = "127.0.0.1", 9999  
data = " Hello how are you?\n"  
  
# Initialize a TCP client socket using SOCK_STREAM  
tcp_client = socket.socket(socket.AF_INET,  
socket.SOCK_STREAM)  
  
try:  
    # Establish connection to TCP server and exchange data  
    tcp_client.connect((host_ip, server_port))  
    tcp_client.sendall(data.encode())  
  
    # Read data from the TCP server and close the connection  
    received = tcp_client.recv(1024)
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

finally:

```
tcp_client.close()
```

```
print ("Bytes Sent: {}".format(data))
```

```
print ("Bytes Received: {}".format(received.decode()))
```

Execution of Python TCP Server and Client modules

- You can run both the server and client in separate Python instances. We recommend that you use Python version 3 for executing the above modules.
- Next, you would first run the server module followed by the client. See below the output of both the client and the server.
- Python 3.5.1 (v3.5.1:37a07cee5969, Dec 6 2015, 01:54:25) [MSC v.1900 64 bit (AMD64)] on win32
- Type "copyright", "credits" or "license()" for more information.
 - RESTART:
 - C:\Users\Techbeamers\AppData\Local\Programs\Python\Python35\Python-TCP- Server.py
 - 127.0.0.1 sent: b'Hello how are you?'

1 Word Question – Answer

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

SR.NO	QUESTION	ANSWER
1	_____ class allows continuous streams of data between the server and client.	TCPserver
2	_____ module has more classes to handle sockets.	socketserver
3	_____ method can be implement to exchange data with TCP client.	handle

Q-6 Explain UDP server and UDP client.

Detail :-

- UDP or user datagram protocol is an alternative protocol to its more common counterpart TCP.
- UDP like TCP is a protocol for packet transfer from 1 host to another, but has some important differences.
- UDP is a connection-less and non-stream oriented protocol.
- It means a UDP server just catches incoming packets from any and many hosts without establishing a reliable pipe kind of connection.

Create UDP sockets

A udp socket is created like this

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

User Datagram Client and Server

- The user datagram protocol (UDP) works differently from TCP/IP.
- Where TCP is a *stream oriented* protocol, ensuring that all of the data is transmitted in the right order, UDP is a *message oriented* protocol.
- UDP does not require a long-lived connection, so setting up a UDP socket is a little simpler.
- On the other hand, UDP messages must fit within a single packet

Simple UDP Server program

```
import socket

sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM) # For UDP

udp_host = socket.gethostname() # Host IP
udp_port = 12345 # specified port to connect

#print type(sock) =====> 'type' can be used to see type
# of any variable ('sock' here)

sock.bind((udp_host,udp_port))

while True:
    print "Waiting for client..."
    data,addr = sock.recvfrom(1024) #receive data from client

    print "Received Messages:",data," from",addr
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

Output of the above script is as follows. Keep it running and then fire up the client.py module.

```
Waiting for client...
```

Simple UDP Client program

This is the udpclient.py script:

```
import socket

sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM) # For UDP

udp_host = socket.gethostname() # Host IP
udp_port = 12345 # specified port to connect

msg = "Hello Python!"
print "UDP target IP:", udp_host
print "UDP target Port:", udp_port
```

Our udpserver.py is up and running, so now we try to run the udpclient.py script,

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

And here is what happened to our server after the client sends the request:

```
~/Desktop/Assignment3$ python udpserver.py
Waiting for client...
Received Messages: Hello Python! from ('192.168.43.217', 55734)
Waiting for client...
```

1 Word Question – Answer

SR.NO	QUESTION	ANSWER
1	UDP stands for _____.	User Datagram Protocol
2	_____ is a connection-less and non-stream oriented protocol.	UDP
3	_____ does not require a long-lived connection	UDP
4	UDP messages must fit within a	Single packet

Q-7 Explain sending E-mail using SMTP.

- Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers.
- Python provides **smtplib** module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.
- Here is a simple syntax to create one SMTP object, which can later be used to send an e-mail –

```
import smtplib
smtpObj = smtplib.SMTP( [host [, port [, local_hostname]] ] )
```


SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

- Here is the detail of the parameters –
- **host** – This is the host running your SMTP server. You can specify IP address of the host or a domain name like tutorialspoint.com. This is optional argument.
- **port** – If you are providing *host* argument, then you need to specify a port, where SMTP server is listening. Usually this port would be 25.
- **local_hostname** – If your SMTP server is running on your local machine, then you can specify just *localhost* as of this option.
- An SMTP object has an instance method called **sendmail**,

Example

Here is a simple way to send one e-mail using Python script. Try it once –

```
import smtplib
```

```
sender = 'from@fromdomain.com'  
receivers = ['to@todomain.com']
```

```
message = """From: From Person <from@fromdomain.com>  
To: To Person <to@todomain.com>  
Subject: SMTP e-mail test
```

```
This is a test e-mail message.
```

```
"""
```

```
try:
```

```
    smtpObj = smtplib.SMTP('localhost')  
    smtpObj.sendmail(sender, receivers, message)  
    print "Successfully sent email"
```

```
except SMTPException:
```

```
    print "Error: unable to send email"
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

- Here, you have placed a basic e-mail in message, using a triple quote, taking care to format the headers correctly.
- An e-mail requires a **From**, **To**, and **Subject** header, separated from the body of the e-mail with a blank line.

1 Word Question – Answer

SR.NO	QUESTION	ANSWER
1	SMTP stands for_____.	Simple Mail
2		sendmail
	SMTP object has an instance method called _____	
3	Python provides _____ module to work with mail.	smtplib
4	Sendmail() have _____, _____ and _____ parameters.	Sender Receivers, message

Q-8 Explain File server and File client

- A simple file transfer server written in Python 3, that allows the user to download files located on the server.
- The server can handle and serve multiple clients at the same time and send files in the same/child directories.

Usage

- Run the server.py file, entering the port you wish for the server to run on.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)

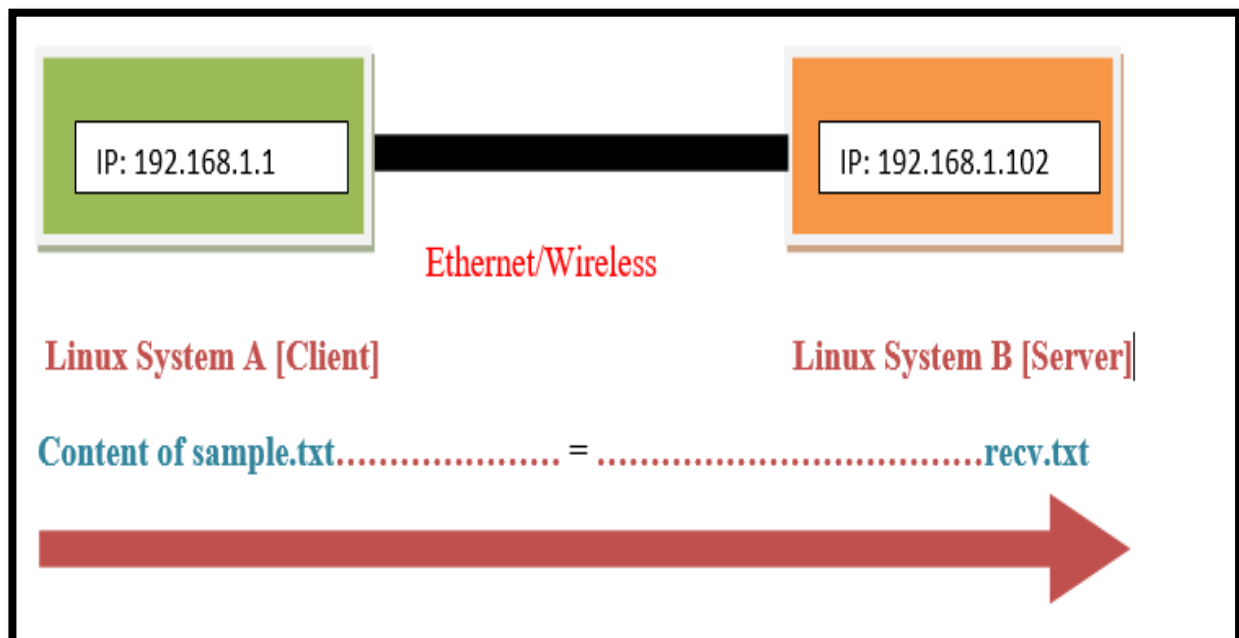


2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

- Now, users can run the client.py file and connect to the server by entering the IP address and port displayed on your server.
- A user can enter the name of a file on the server (in the same folder as the server.py) and download that file - the file will appear in the same directory as the client.py file, named from_server+filename.
- file transfer is based on server client model to use socket programming in python3+.

- **Basic Set up Diagram:**



- Here is the basic set up diagram to run this program.

File requirements:

- We need **server.py** and this file should be present at server system. In our case server.py should be at B_server system.
- Another two files **client.py** and **sample.txt** should be present at client system. In our case those two files should be present at A_client system.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

- File transfer is the process of copying or moving a file from a computer to another over a network or Internet connection.
- In this tutorial, we'll go step by step on how you can write client/server Python scripts that handles that.
- The basic idea is to create a server that listens on a particular port, this server will be responsible for receiving files (you can make the server sends files as well).
- On the other hand, the client will try to connect to the server and send a file of any type.
- Here is the code to send a file from a local server to a local client.

➤ # server.py

```
import socket    # Import socket module

port = 60000     # Reserve a port for your service.
s = socket.socket() # Create a socket object
host = socket.gethostname()# Get local machine names
s.bind((host, port)) # Bind to the port
s.listen(5)      # Now wait for client connection.

print 'Server listening ... '
while True:

    conn, addr = s.accept()    # Establish connection with client.
    print 'Got connection from', addr
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

data = conn.recv(1024)

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

```
print('Server received', repr(data))
```

```
filename='mytext.txt'
```

```
f = open(filename,'rb')
```

```
l = f.read(1024)
```

```
while (l):
```

```
    conn.send(l)
```

```
    print('Sent ',repr(l))
```

```
    l = f.read(1024)
```

```
f.close()
```

```
print('Done sending')
```

```
conn.send('Thank you for connecting')
```

```
conn.close()
```

client.py

```
import socket # Import socket module
```

```
s = socket.socket() # Create a socket object
```

```
host = socket.gethostname() # Get local machine name
```

```
port = 60000 # Reserve a port for your service.
```

```
s.connect((host, port))
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



**2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001**

**3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001**

```
s.send("Hello server!")
```

```
with open('received_file', 'wb') as f:
```

```
    print 'file opened'
```

```
    while True:
```

```
        print('receiving data...')
```

```
        data = s.recv(1024)
```

```
        print('data=%s', (data))
```

```
        if not data:
```

```
            break
```

```
        # write data to a file
```

```
        f.write(data)
```

```
f.close()
```

```
print('Successfully get the file')
```

```
s.close()
```

```
print('connection closed')
```

Server listening....

```
Got connection from ('192.168.56.10', 62854)
```

```
('Server received', "Hello server!")
```

```
('Sent ', "1 1234567890\n")
```

```
('Sent ', "4567890\n105")
```

```
('Sent ', "300 1234567890\n")
```

Done sending

Output on a local client:

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

file opened

receiving data...

data=1 1234567890

2 1234567890

...

103 1234567890

104 123

receiving data...

data=4567890

105 1234567890

106 1234567890

...

299 1234567890

receiving data...

data=300 1234567890

Thank you for connecting

receiving data...

data=Successfully get the file

connection closed

1 **Word Question – Answer**

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

SR.NO	QUESTION	ANSWER
1	_____ is the process of copying or moving the file across internet	File transfer
2	FTP stands for _____.	File Transfer Protocol
3	_____ method can be used to close the currently opened file	close()

Q-9 Explain two way communication between client and server.

- Sockets are the endpoints of a bidirectional communications channel. Sockets may communicate within a process, between processes on the same machine, or between processes on different continents.
- So we will design both server and client model so that each can communicate with them. The steps can be considered like this.
 1. Python socket server program executes at first and wait for any request
 2. Python socket client program will initiate the conversation at first.
 3. Then server program will response accordingly to client requests.
 4. Client program will terminate if user enters “bye” message.
- Server program will also terminate when client program terminates, this is optional and we can keep server program running indefinitely or terminate with some specific command in client request.

A Simple Server

A socket object is then used to call other functions to setup a socket server.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

- Now call **bind(hostname, port)** function to specify a *port* for your service on the given host.
- Next, call the *accept* method of the returned object. This method waits until a client connects to the port you specified, and then returns a *connection* object that represents the connection to that client.

A Simple Client

- Let us write a very simple client program which opens a connection to a given port 12345 and given host.
- This is very simple to create a socket client using Python's *socket* module function.
- The **socket.connect(hostname, port)** opens a TCP connection to *hostname* on the *port*. Once you have a socket open, you can read from it like any IO object.
- When done, remember to close it, as you would close a file.
- The following code is a very simple client that connects to a given host and port, reads any available data from the socket, and then exits
- Now run this server.py in background and then run above client.py to see the result.

```
# Following would start a server in background.
```

```
$ python server.py &
```

```
# Once server is started run client as follows:
```

```
$ python client.py
```

This would produce following result –

Got connection from ('127.0.0.1', 48437)

Thank you for connecting

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001

1 Word Question – Answer

SR.NO	QUESTION	ANSWER
1	_____ are the endpoints of a bidirectional communications channel	sockets
2	_____ mehod can be used to specify a <i>port</i> for your service on the given host.	bind()
3	_____ method waits until a client connects to the port you specified	accept

Q-10 Explain how to download webpage from internet.

- Python provides several ways to do just that in its **standard library** .
- The urllib.request method is a Python module for fetching URLs.
- It offers a very simple interface, in the form of the urlopen function.
- This is capable of fetching URLs using a _____ variety of protocols.

urllib.request.urlopen() , open the URL url,
which can be either a string or a Request object.

```
from urllib.request import urlopen
html = urlopen("http://net-
informations.com/about.htm")

print(html.read())
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

`downloadFile(URL_from_file)`

If you want to obtain the contents of a web page into a variable, just read the response of `urllib.request.urlopen`:

```
import urllib.request
...
url = 'http://example.com/'
response = urllib.request.urlopen(url)
data = response.read() # a `bytes` object
text = data.decode('utf-8') # a `str`; this step can't be used if data is binary
```

The easiest way to download and save a file is to use the `urllib.request.urlretrieve` function:

```
import urllib.request
...
# Download the file from `url` and save it locally under
`file_name`:
urllib.request.urlretrieve(url, file_name)
import urllib.request
...
# Download the file from `url`, save it in a temporary directory and get
the# path to it (e.g. '/tmp/tmpb48zma.txt') in the `file_name` variable:
file_name, headers = urllib.request.urlretrieve(url)
```

1 Word Question – Answer

SR.NO	QUESTION	ANSWER
-------	----------	--------

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

1	_____ method is a Python module for fetching URLs.	urllib.request
2	_____ module provide a way to download a file is over HTTP	Urllib
3	_____ method can open the URL url, which can be either a string or a Request object.	urlopen()

Q-11 Explain how to download image from internet.

- The **standard Python library** for accessing websites via your program is **urllib**. It is also used by the requests module.
- Through urllib, we can do a variety of things: **access websites, download data, parse data, send GET and, POST requests.**
- We can download our image using just a few lines of code:

```
import urllib.request
```

```
# setting filename and image  
URL      filename      =  
'sunshine_dog.jpg' image_url =  
          "https://cdn.pixabay.com/photo/2020/02/06/09/39/summer-  
          4823612_960_720.jpg"
```

```
#calling urlretrieve function to get resource  
urllib.request.urlretrieve(image_url, filename)
```

- We used the **urlretrieve** method to copy the required web resource to a local file.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

- It is important to note that on some systems and a lot of websites, the above code will result in an error: **HTTPError: HTTP Error 403: Forbidden.**
- This is because a lot of websites don't appreciate random programs accessing their data.
- Some programs can attack the server by sending a large number of requests. This prevents the server from functioning.

PYTHON GUI PROGRAMMING :-

Q-1 Explain Event Driven Programming.

- Anything that happens in a user interface is an *event*.
- We say that an event is *fired* whenever the user does something – for example, clicks on a button or types a keyboard shortcut.
- Some events could also be triggered by occurrences which are not controlled by the user – for example, a background task might complete, or a network connection might be established or lost.
- Our application needs to monitor, or *listen* for, all the events that we find interesting, and respond to them in some way if they occur.
- To do this, we usually associate certain functions with particular events. We call a function which performs an action in response to an event an *event handler* – we *bind* handlers to events.
- Event-driven programming focuses on events. Eventually, the flow of program depends upon events.

□ Until now, we were dealing with either sequential or parallel

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



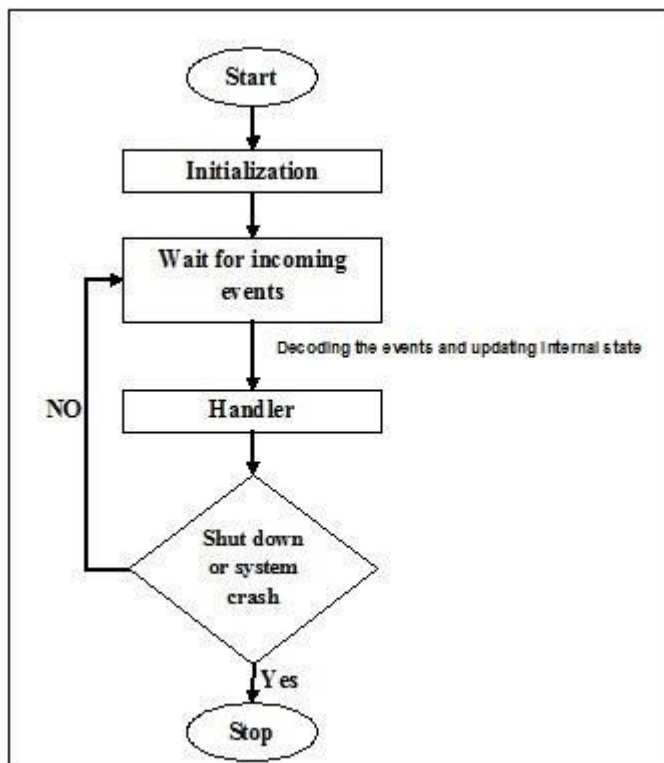
2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

execution model but the model having the concept of event-driven programming is called asynchronous model.

- Event-driven programming depends upon an event loop that is always listening for the new incoming events.
- The working of event-driven programming is dependent upon events.
- Once an event loops, then events decide what to execute and in what order.

- Following flowchart will help you understand how this works –



1 **Word Question – Answer**

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

SR.NO	QUESTION	ANSWER
1	Anything that happens in a user interface is an_____.	event
2	Event-driven programming depends upon _____	eventloop
3	The model having concept of event-driven programming is called_____.	asynchronous model

Q-2 Explain How to create simple GUI.

- **Gui stands for Graphical User Interface.**
- `tkinter` provides us with a variety of common GUI elements which we can use
 - to build our interface – such as buttons, menus and various kinds of entry fields and display areas.
 - We call these elements *widgets*. We are going to construct a *tree* of widgets for our GUI – each widget will have a parent widget, all the way up to the *root window* of our application.
 - For example, a button or a text field needs to be *inside* some kind of containing window.
 - The widget classes provide us with a lot of default functionality.
 - They have methods for configuring the GUI's appearance – for example, arranging the elements according to some kind of *layout* – and for handling various kinds of user-driven events.
 - Once we have constructed the backbone of our GUI, we will need to customise it by integrating it with our internal application class.
 - **Our first GUI will be a window with a label and two buttons:**

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

```
from tkinter import Tk, Label, Button
```

```
class MyFirstGUI:
```

```
    def __init__(self, master):
```

```
        self.master = master
```

```
        master.title("A simple GUI")
```

```
        self.label = Label(master, text="This is our first GUI!")
```

```
        self.label.pack()
```

```
        self.greet_button = Button(master, text="Greet" command=self.greet)
```

```
        self.greet_button.pack()
```

```
        self.close_button = Button(master, text="Close",  
command=master.quit)
```

```
        self.close_button.pack()
```

1 **Word Question – Answer**

SR.NO	QUESTION	ANSWER
1	GUI stands for_____.	Graphical User Interface
2	_____module provides us with a variety of common GUI elements	tkinter

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

Q-3 Explain Python GUI – tkinter.

- Python offers multiple options for developing GUI (Graphical UserInterface).
- Out of all the GUI methods, tkinter is the most commonly used method.
- It is a standard Python interface to the Tk GUI toolkit shipped with Python.
- Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

- Importing tkinter is same as importing any other module in the Python code. Note that the name of the module in Python 2.x is 'Tkinter' and in Python 3.x it is 'tkinter'.

import tkinter

- There are two main methods used which the user needs to remember while creating the Python application with GUI.

1. mainloop(): There is a method known by the name mainloop() is used when your application is ready to run.

mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.

m.mainloop()

```
import tkinter
m =
tkinter.Tk()'''
widgets are added here
m.mainloop()
```

where m is the name of the main window object

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

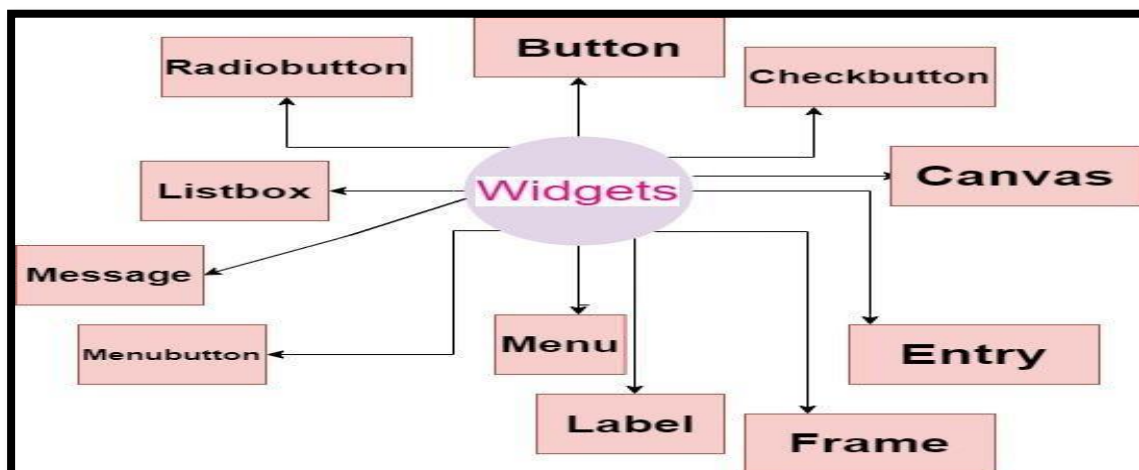
3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

1 Word Question – Answer

SR.NO	QUESTION	ANSWER
1	_____ is the fastest and easiest way to create the GUI applications.	tkinter
2	_____ is an infinite loop used to run the application	mainloop()

Q-4 Explain Widgets in Python GUI.

- Widgets are the bread and butter of the Python GUI framework Tkinter.
- They are the elements through which users interact with your program.
- Each **widget** in Tkinter is defined by a class. Here are some of the widgets available:
- There are various controls, such as **buttons**, **labels**, **scrollbars**, **radiobuttons**, and **text boxes** used in a GUI application.
- These **little components** or controls of **Graphical User Interface (GUI)** are known as **widgets** in Tkinter.
- These are **19 widgets** available in Python Tkinter module.



SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

Widget Class	Description
Label	A widget used to display text on the screen
Button	A button that can contain text and can perform an action when Clicked
Entry	A text entry widget that allows only a single line of text
Text	A text entry widget that allows multiline text entry
Frame	A rectangular region used to group related widgets or provide Padding between widgets

1 Word Question – Answer

SR.NO	QUESTION	ANSWER
1	Each _____ in tkinter is defined by a class	widget
2	little components or controls of Graphical User Interface (GUI) are known as _____ in Tkinter.	widget

Q-5 Explain Dialogues and Message boxes.



SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

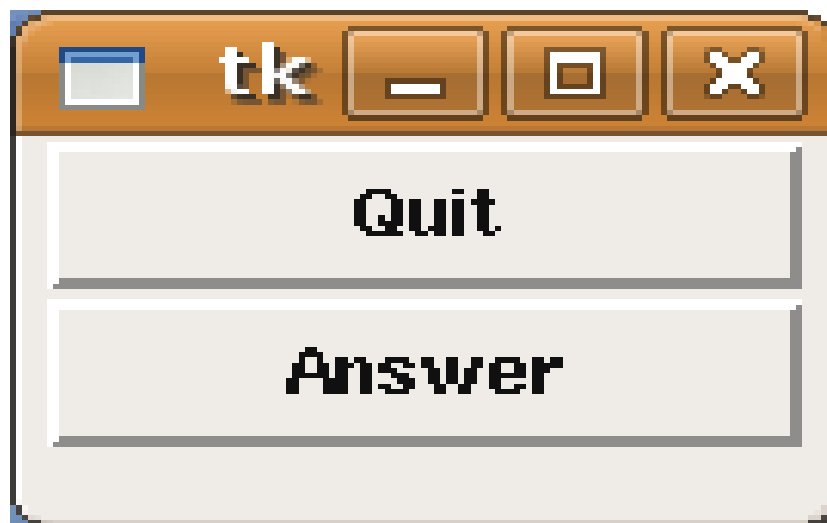
(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

- Tkinter (and TK of course) provides a set of dialogues (dialogs in American English spelling), which can be used to display message boxes, showing warning or errors, or widgets to select files and colours.
- There are also simple dialogues, asking the user to enter string, integers or float numbers.
- Let's look at a typical GUI Session with Dialogues and Message boxes.
- There might be a button starting the dialogue, like the "quit" button in the following window:



SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



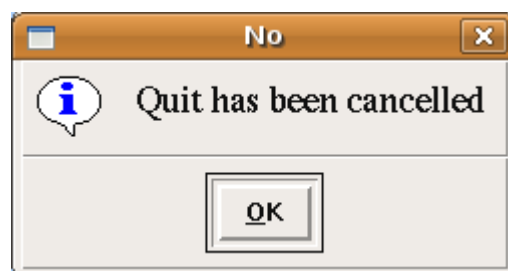
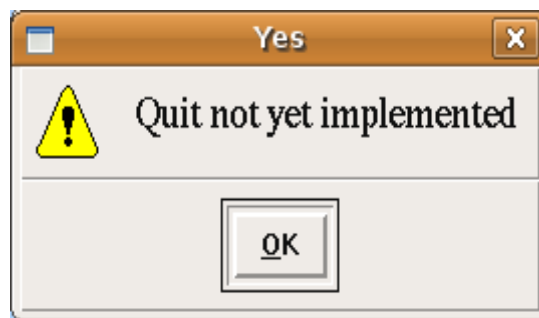
2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

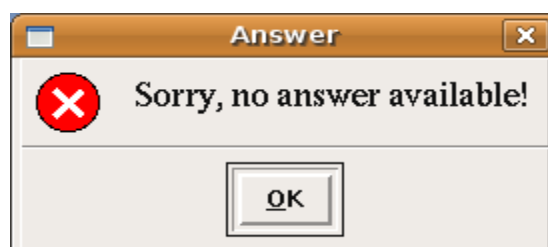
Pushing the "quit" button raises the Verify window:



Let's assume that we want to warn users that the "quit" functionality is not yet implemented. In this case we can use the warning message to inform the user, if he or she pushes the "yes" button:



Let's go back to our first Dialogue with the "quit" and "answer" buttons. If the "Answer" functionality is not implemented, it might be useful to use the following error message box:



SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

- Python script, which implements the previous dialogue widgets:

```
import tkinter as tk
from tkinter import messagebox as mb

def answer():
    mb.showerror("Answer", "Sorry, no answer
available")def callback():
    if mb.askyesno('Verify', 'Really quit?):
        mb.showwarning('Yes', 'Not yet
implemented')
    else:
        mb.showinfo('No', 'Quit has been cancelled')

tk.Button(text='Quit', command=callback).pack(fill=tk.X)
tk.Button(text='Answer', command=answer).pack(fill=tk.X)
tk.mainloop()
```

Message Boxes

- The message dialogues are provided by the 'messagebox' submodule oftkinter.
- 'messagebox' consists of the following functions, which correspond todialog windows:
- askokcancel(title=None, message=None, **options)
Ask if operation should proceed; return true if the answer is ok
- askquestion(title=None, message=None, **options)Ask a question
- askretrycancel(title=None, message=None, **options)
Ask if operation should be retried; return true if the answer is yes
- askyesno(title=None, message=None, **options) Ask
a question; return true if the answer is yes
- askyesnocancel(title=None, message=None, **options)

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

Ask a question; return true if the answer is yes, None if cancelled.

- `showerror(title=None, message=None, **options)`Show an error message
- `showinfo(title=None, message=None, **options)`Show an info message
- `showwarning(title=None, message=None, **options)`Show a warning message

Open File Dialogue

- There is hardly any serious application, which doesn't need a way to read from a file or write to a file. Furthermore, such an application might have to choose a directory.
- Tkinter provides the module `tkFileDialog` for these purposes.

```
import tkinter as tk
from tkinter import filedialog as fd
def callback():
    name= fd.askopenfilename()
    print(name)
errmsg = 'Error!'
tk.Button(text='File Open',
          command=callback).pack(fill=tk.X)
tk.mainloop()
```

- The code above creates a window with a single button with the text "FileOpen".
- If the button is pushed, the following window appears:

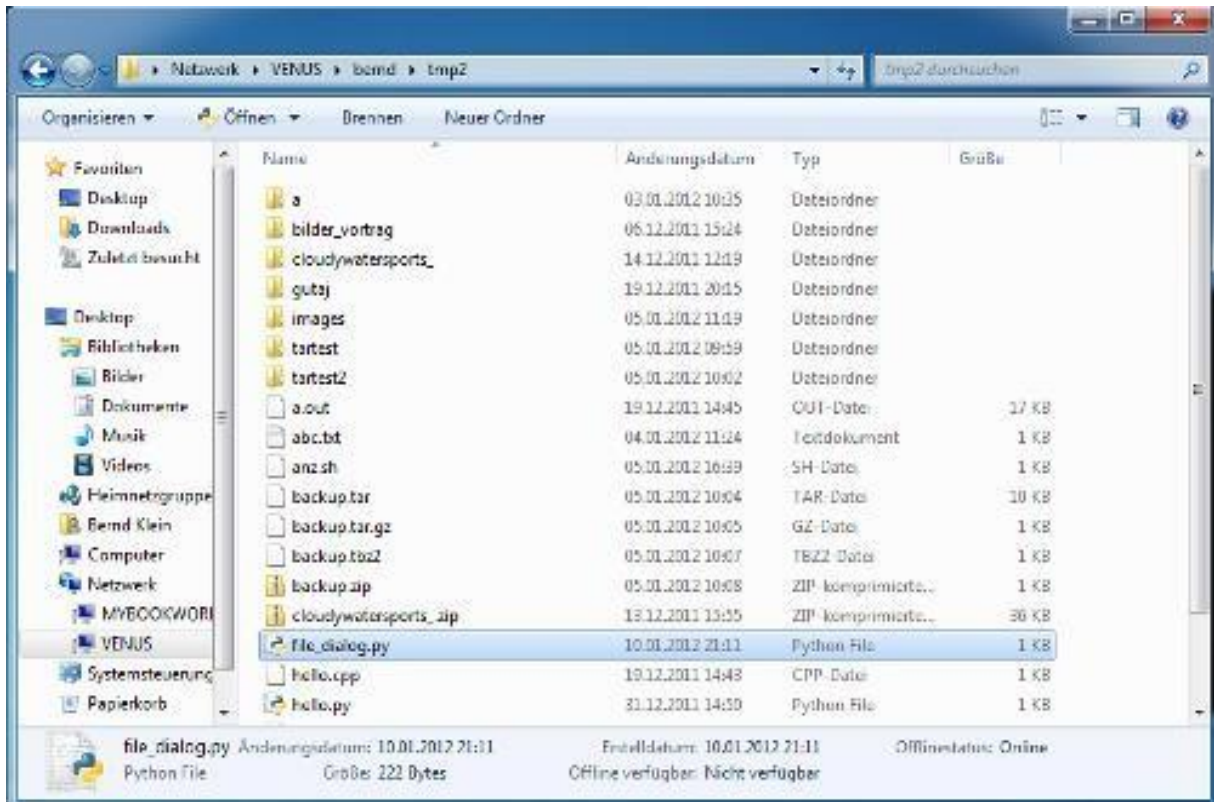
SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001



➤ The above example was created using a gnome desktop under Linux. If we start the same program under Windows 7, it looks like this:

Choosing a Colour

- There are applications where the user should have the possibility to select a colour.
- Tkinter provides a pop-up menu to choose a colour. To this purpose we have to import the 'tkinter.colorchooser' module and have to use the method askColor:

```
result = tkinter.colorchooser.askcolor ( color, option=value, ...)
```

- If the user clicks the OK button on the pop-up window, respectively, the return value of askcolor() is a tuple with two elements, both a representation of the chosen colour, e.g. ((106, 150, 98), '#6a9662')
The first element return[0] is a tuple (R, G, B) with the RGB representation in decimal values (from 0 to 255).
- The second element return[1] is a hexadecimal representation of the chosen colour.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

If the user clicks "Cancel" the method returns the tuple (None, None).

The optional keyword parameters are:

color	The variable color is used to set the default colour to be displayed. If color is not set, the initial colour will be grey.
title	The text assigned to the variable title will appear in the pop-up window's title area. The default title is "Color".
parent	Make the pop-up window appear over window W. The default behaviour is that it appears over the root window.

Let's have a look at an example:

```
import tkinter as tk
from tkinter.colorchooser import askcolor
```

```
def callback():
    result = askcolor(color="#6A9662",
                      title = "Bernd's Colour Chooser")
    print(result)
root = tk.Tk()
tk.Button(root,
          text='Choose Color',
          fg="darkgreen",
          command=callback).pack(side=tk.LEFT, padx=10)
tk.Button(text='Quit',
          command=root.quit,
          fg="red").pack(side=tk.LEFT, padx=10)
tk.mainloop()
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

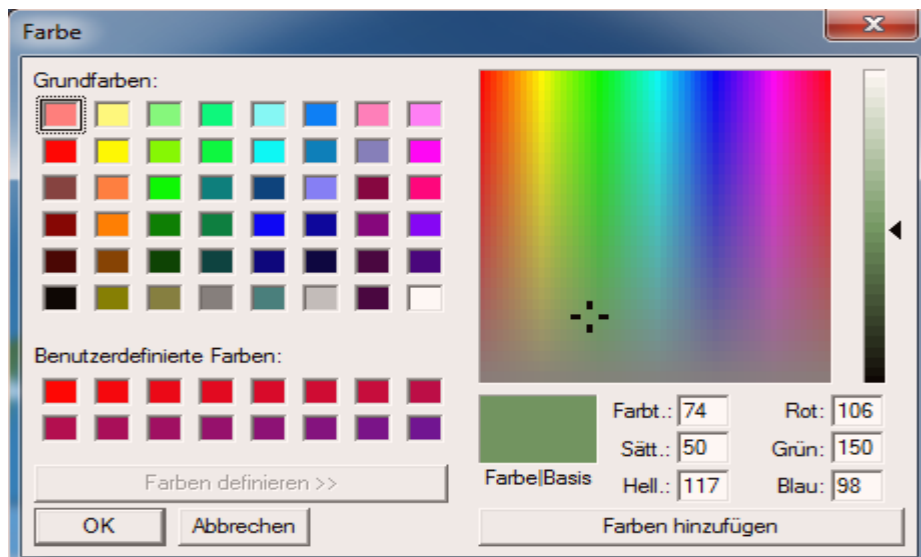
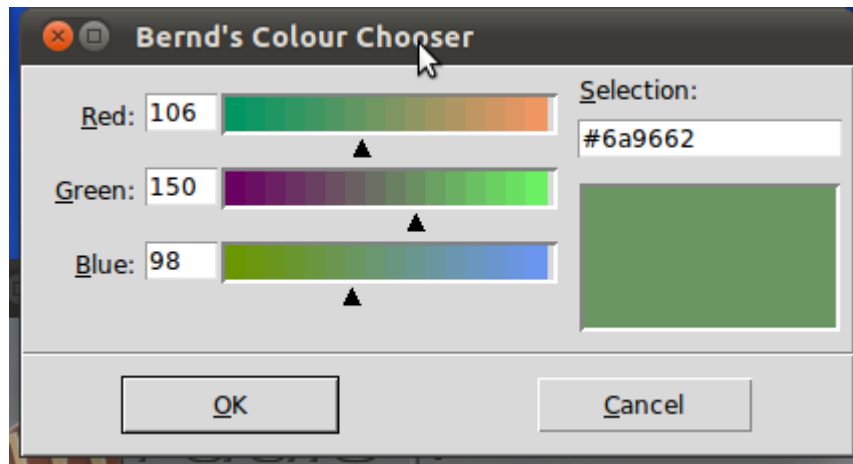
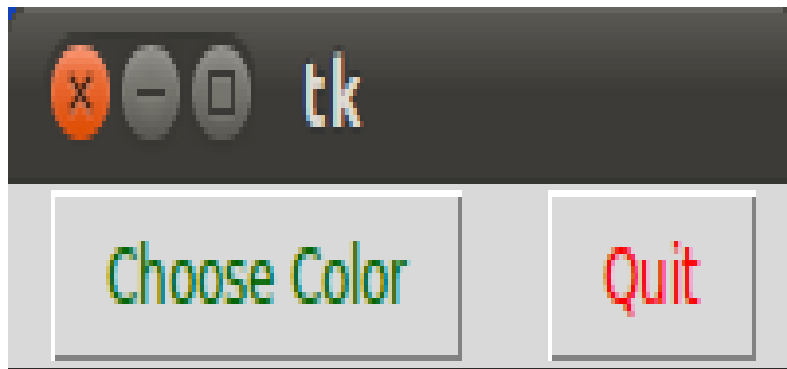
(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

The look and feel depends on the operating system (e.g. Linux or Windows) and the chosen GUI (GNOME, KDE and so on). The following windows appear, if you use Gnome:



SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

- Using the same script under Windows 7 gives us the following result.

Q-6 Explain Widget Attributes

- In this part ,we talk about standard widget attributes, including cursors,reliefs, colours, and fonts.
- The standard widget attributes are keywords used in widget constructors.

o Tkinter widget state

- The state attribute defines the state of the widget. It can have the following values: NORMAL, ACTIVE, and DISABLED.

We demonstrate the state attribute on the Label widget.

```
label1 = Label(self, text=txt, state=NORMAL)
```

o Tkinter Background colours

- The background colours of widgets can be set with background attribute. It can be abbreviated to bg.
- Likewise, the foreground colours of widgets can be set with foreground attribute. It can be abbreviated to fg.

In the code example, we create label with background colour.

```
lbl1 = Label(frame, bg='SlateGray3', width=15, height=10)  
lbl1.pack(side=LEFT, padx=3)
```

o Width and height

- The width and height attributes set the width and height of the widget. In the example, we control the size of buttons with width and height.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

```
btn2 = Button(frame, text='Button', width=5 ,height =4 )
```

o Tkinter fonts

- Tkinter has a tkinter.font module for working with fonts. It has some built- in fonts such as TkToolTipFont, TkDefaultFont, or TkTextFont. The font is setwith the font attribute.

Here we use a built-in font name.

```
label3 = Label(self, text=txt, font=('Times', '18', 'italic'))
```

o Tkinter cursors

- The cursor is a small icon that shows where the mouse pointer is located.The cursor in Tkinter is set with the cursor attribute.

In the example, we have four labels. Different cursors are used for eachof the labels.

```
lbl4 = Label(frame, bg='DarkSeaGreen4', width=15, height=10,  
cursor='pencil')
```

1 Word Question – Answer

SR.NO	QUESTION	ANSWER
1	A_____is a small icon that shows where the mouse pointer is located.	cursor
2	_____attribute defines the state of the widget	state
3	Tkinter has a_____module for working with fonts.	tkinter.font

Q-7Explain Layout Manager in tkinter.

- When we design the GUI of our application, we decide what widgets we willuse and how we will organize those widgets in the application.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

- To organize our widgets, we use specialized non-visible objects called layout managers.
- There are two kinds of widgets: **containers and their children.**
- The containers group their children into suitable layouts.
- Tkinter has three built-in layout managers: the pack, grid, and place managers.
- The place geometry manager positions widgets using absolute positioning. The pack geometry manager organizes widgets in horizontal and vertical boxes. The grid geometry manager places widgets in a two dimensional grid.

Absolute positioning

- In most cases, programmers should use layout managers.
- There are a few situations where we could use absolute positioning.
- In absolute positioning, the programmer specifies the position and the size of each widget in pixels.
- The size and the position of a widget do not change if we resize a window. Applications look different on various platforms, and what looks OK on Linux, might not look OK on Mac OS.
- Changing fonts in our application might spoil the layout. If we translate our application into another language, we must redo our layout.

```
absolute.py  
#!/usr/bin/env  
python3
```

```
.....
```

ZetCode Tkinter tutorial

In this script, we lay out
images using absolute
positioning.

```
from PIL import Image,
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

```
ImageTk from tkinter
import Tk, BOTH
from tkinter.ttk import Frame, Label, Style
```

```
class Example(Frame):
```

```
    def __init__(
        self):
        super().__init__()
```

```
        self.initUI()
```

```
    def initUI(self):
```

```
        self.master.title("Absolute
        positioning")    self.pack(fill=BOTH,
        expand=1)
```

```
        Style().configure("TFrame", background="#333")
```

```
        bard = Image.open("bardejov.jpg")
```

```
        bardejov=
        ImageTk.PhotoImage(bard)
        label1=Label(self, image=bardejov)
```

In this example, we place three images using absolute positioning. We use the place geometry manager.

```
from PIL import Image, ImageTk
```

- We use Image and ImageTk from the Python Imaging Library (PIL) module.

```
    style = Style()
    style.configure("TFrame", background="#333")
```

We configure our frame to have a dark gray background using styles.

```
bard = Image.open("bardejov.jpg")
bardejov = ImageTk.PhotoImage(bard)
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT. (AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

We create an image object and a photo image object from an image in the current working directory.

```
label1 = Label(self, image=bardejov)
```

We create a Label with an image. Labels can contain text or images.

```
label1.image = bardejov
```

We must keep the reference to the image to prevent image from being garbage collected.

```
label1.place(x=20, y=20)
```

The label is placed on the frame at x=20 and y=20 coordinates.

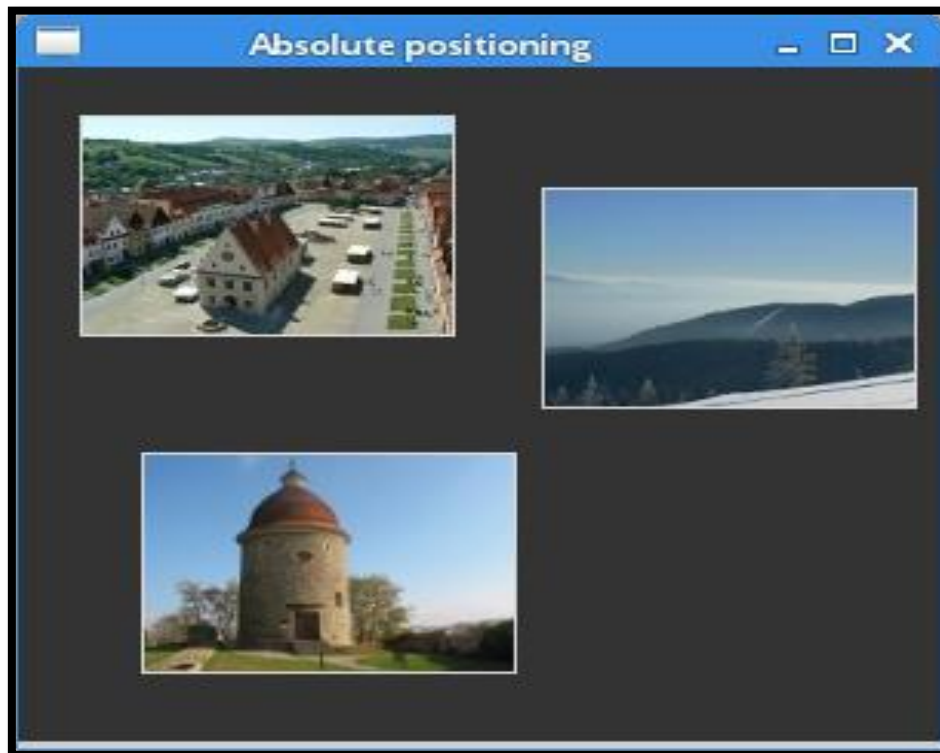


Figure: Absolute positioning

1 Word Question – Answer

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

SR.NO	QUESTION	ANSWER
1	The containers group their children into_____.	suitable layout
2	the position and the size of each widget can be specified in _____	pixels
3	Tkinter has three built-in layout managers: _____,_____ &_____.	pack Grid place managers.

Q-8 Explain Button , Label ,Entryfield.

Button: To add a button in your application, this widget is used. The general syntax is:

w=Button(master, option=value)

- master is the parameter used to represent the parent window.
- There are number of options which are used to change the format of theButtons.
- Number of options can be passed as parameters separated by commas. Some of them are listed below.
 - **activebackground:** to set the background color when button is under the cursor.
 - **activeforeground:** to set the foreground color when button is under the cursor.
 - **bg:** to set the normal background color.
 - **command:** to call a function.
 - **font:** to set the font on the button label.
 - **image:** to set the image on the button.
 - **width:** to set the width of the button.
 - **height:** to set the height of the button.

CheckButton: To select any number of options by displaying a number of options to a user as toggle buttons.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

The general syntax is:

w = CheckButton(master, option=value)

- There are number of options which are used to change the format of this widget.
- Number of options can be passed as parameters separated by commas. Some of them are listed below.
 - **Title:** To set the title of the widget.
 - **activebackground:** to set the background color when widget is under the cursor.
 - **activeforeground:** to set the foreground color when widget is under the cursor.
 - **bg:** to set the normal background.
 - **command:** to call a function.
 - **font:** to set the font on the button label.
 - **image:** to set the image on the widget.

```
from tkinter import *
```

```
master = Tk()
```

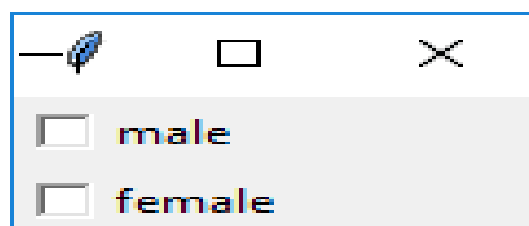
```
var1 = IntVar()
```

```
Checkbutton(master, text='male', variable=var1).grid(row=0,  
sticky=W)
```

```
var2 = IntVar() Checkbutton(master, text='female',  
variable=var2).grid(row=1, sticky=W)
```

```
mainloop()
```

Output :-



SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

Entry: It is used to input the single line text entry from the user..
For multi-line text input, Text widget is used.

The general syntax is:

w=Entry(master, option=value)

- master is the parameter used to represent the parent window.
- There are number of options which are used to change the format of the widget.
- Number of options can be passed as parameters separated by commas. Some of them are listed below.
 - **bd:** to set the border width in pixels.
 - **bg:** to set the normal background color.
 - **cursor:** to set the cursor used.
 - **command:** to call a function.
 - **highlightcolor:** to set the color shown in the focus highlight.
 - **width:** to set the width of the button.
 - **height:** to set the height of the button.

```
from tkinter
import *
master = Tk()
Label(master, text='First Name').grid(row=0)
```

```
Label(master, text='Last Name').grid(row=1)
```

```
e1 = Entry(master)
```

```
e2 = Entry(master)
```

```
e1.grid(row=0, column=1)
```

```
e2.grid(row=1, column=1)
```

```
mainloop()
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

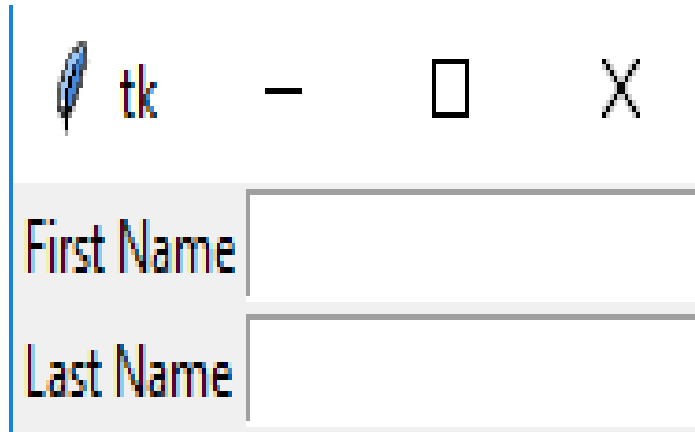
(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

Output :-



Frame: It acts as a container to hold the widgets. It is used for grouping and organizing the widgets.

The general syntax is:

w = Frame(master, option=value)

- master is the parameter used to represent the parent window.
- There are number of options which are used to change the format of the widget.
- Number of options can be passed as parameters separated by commas. Some of them are listed below.
 - **highlightcolor:** To set the color of the focus highlight when widget has to be focused.
 - **bd:** to set the border width in pixels.
 - **bg:** to set the normal background color.
 - **cursor:** to set the cursor used.
 - **width:** to set the width of the widget.
 - **height:** to set the height of the widget.

from tkinter

import * root =

Tk()

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

```
frame = Frame(root)
```

```
frame.pack()
```

```
bottomframe = Frame(root)
```

```
bottomframe.pack( side = BOTTOM )
```

```
redbutton = Button(frame, text = 'Red', fg='red')
```

```
redbutton.pack( side = LEFT)
```

```
greenbutton = Button(frame, text = 'Brown', fg='brown')
```

```
greenbutton.pack( side = LEFT )
```

```
bluebutton = Button(frame, text = 'Blue', fg = 'blue')
```

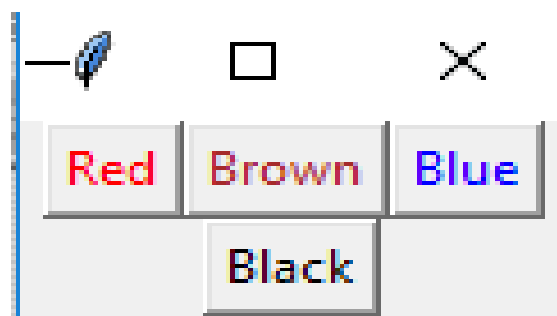
```
bluebutton.pack( side = LEFT )
```

```
blackbutton = Button(bottomframe, text = 'Black', fg = 'black')
```

```
blackbutton.pack( side = BOTTOM)
```

```
root.mainloop()
```

Output:-



Label: It refers to the display box where you can put any text or image which can be updated any time as per the code.

The general syntax is:

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

`w=Label(master, option=value)`

- master is the parameter used to represent the parent window.
- There are number of options which are used to change the format of the widget.
- Number of options can be passed as parameters separated by commas. Some of them are listed below.
 - **bg**: to set the normal background color.
 - **bg** to set the normal background color.
 - **command**: to call a function.
 - **font**: to set the font on the button label.
 - **image**: to set the image on the button.
 - **width**: to set the width of the button.
 - **height** to set the height of the button.

```
from tkinter

import * root =

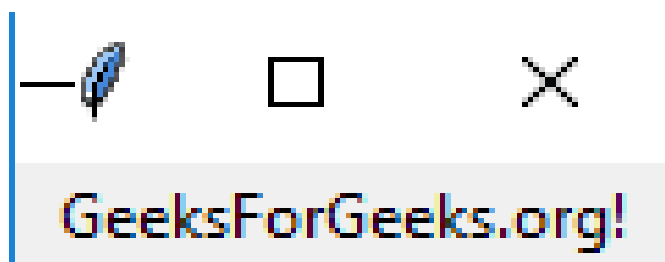
Tk()

w = Label(root, text='GeeksForGeeks.org!')

w.pack()

root.mainloop()
```

Output :-



SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

Listbox: It offers a list to the user from which the user can accept any number of options.

The general syntax is:

w = Listbox(master, option=value)

- master is the parameter used to represent the parent window.
- There are number of options which are used to change the format of the widget.
- Number of options can be passed as parameters separated by commas. Some of them are listed below.
 - **highlightcolor:** To set the color of the focus highlight when widget has to be focused.
 - **bg:** to set the normal background color.
 - **bd:** to set the border width in pixels.
 - **font:** to set the font on the button label.
 - **image:** to set the image on the widget.
 - **width:** to set the width of the widget.
 - **height:** to set the height of the widget.

```
from tkinter
import *
top = Tk()
Lb = Listbox(top)
Lb.insert(1, 'Python')
Lb.insert(2, 'Java')
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001

```
Lb.insert(3, 'C++')  
Lb.insert(4, 'Any  
other') Lb.pack()  
top.mainloop()
```

Output :-

