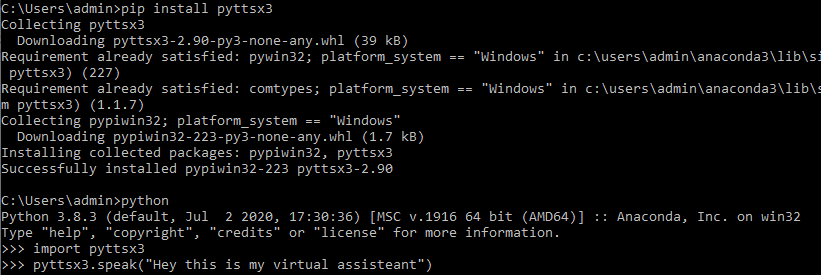
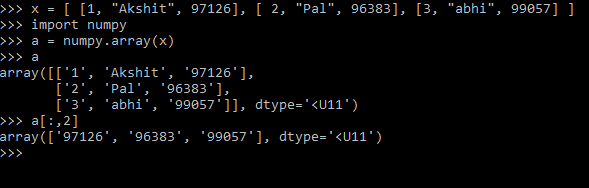
**Session 01**

* We need programming language, so we can communicate with our OS, because OS only knows programming language.
  + But if OS don’t know python language then we can not do anything, for this we have to install python interpreter in our OS.
  + Here we are using Python 3.8 version of python.
  + We are using anaconda distribution for installing python interpreter.
    - * While installing click on add path.
    - It will install python interpreter.
    - Jupyter notebook
    - And lots of library.
* For storing value inside ram we have to create one box, for this we use variable so we can get value using variable whenever we require.
* We have 2 way to communicate to OS.
* For using base windows, linux commands we have to use system function.
  + There are some function known as builtin function, which comes with python, we don’t have to import them.
    - Eg, print()
    - Dir(\_\_builtins\_\_)
  + System() is not a builtin function, so we have to tell our language where is function located.
    - It is inside OS module (program file – contains lots of function).
    - We have to import this module for using this.
  + SO, for this we have to use os.system() for running.
    - Os.system(“notepad”)
    - Os.system(“chrome yahoo.com”)
* If you want to tell your OS to speak instead of print we have to use speak function.
  + It will come from pyttsx3 module.
    - For this we have to install library.
    - We have a pip command to install library in python.
  + Pip install pyttsx3
    - Pyttsx3.speak(“How are you”)



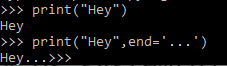
**Session 02**

* We have three way to run commands – programs.
  + Offline
    - Using notepad or vim.
  + Online
    - Your command prompt – shell – CMD.
  + IDE
* When you have any file stored in your H.D. which have some code it is known as program file.
  + When you run your program it will known as process.
    - Here your entire code load inside your ram.
* When you create variable in other language you have to tell datatype of variable.
  + In python we don’t have to define datatype.
  + Python dynamically find datatype of variable, it is known as Type Inference.
* Anything inside quotes (“”)are known as string.
  + In pyton we don’t have difference between “ “ or ‘ ‘ .
    - Between “ “ or ‘ ‘ if you write escape sequence it will process this in python.
      * In different language ‘ ‘ will not process escape sequence.
* In online interpreter, python shell come with REPL.
  + Read, evaluate, print, loop.
* List datatype
  + X= [ “Akshit” , “Pal” , “Abhi”]
    - X[0]
      * Behind the seen x.\_\_getitems\_\_(0)
    - X[-1]
    - Len(x)
    - X[0:2] == x[:2]
      * [ “ Akshit ”, “ Pal ”]
      * It will exclude last number.
      * It is known as slicing operator.
  + Y= [ [ 1, ”Akshit” , 97126] , [ 2, “pal”, 96383] , [ 3, “Abhi”, 99067] ]
    - Y[2][2]
      * 99067
* List only works on row wise, it will not work column wise.
* We have array datatype for getting column wise data.
* It will come from numpy.
  + A = numpy.array(x)



**Session 03**

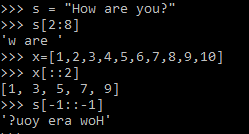
* Jupyter notebook
  + To launch notebook.
  + In jupyter file of code is known as notebook.
* Escape sequence
  + Between “ “ or ‘ ‘ if you write escape sequence it will process this in python.
    - In different language ‘ ‘ will not process escape sequence.
  + You write print(r” Welcome here”)
    - It will print as it is as raw string.
  + Help(print)
    - It will always end with new line.
    - Print( “mdfbd” , end=’\n’)



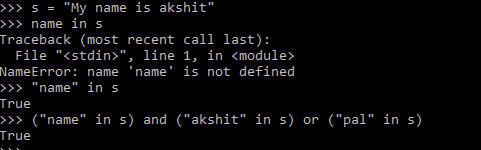
* If – else
  + If x > 2:  
     print(“HI”)  
    else:  
     print(“not”)
    - Here first x>2 evaluated.
      * If condition match it will return True else False…
      * It will always return Boolean.
    - So, if True  
       do this.

**Session 04**

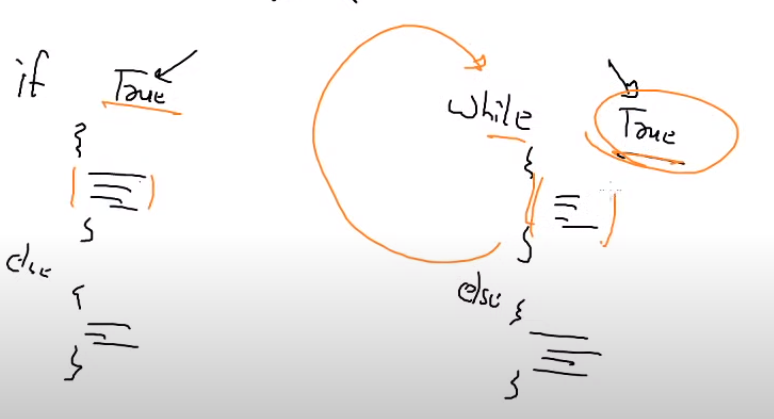
* String is also like a list.
* String = ‘’How are you?’’
  + String[2:8]  
    “w are “
  + By default it will jump with 1.
    - String[2:8] == string[2:8:1]



* + String [-1: :-1]
    - To print reverse string.
* In, and, or



* While loop
  + When you want to run same thing, some finite number of times then we use while loop.

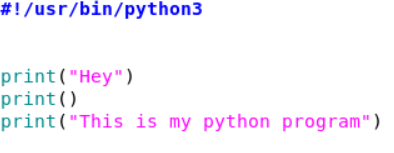


* You can compare while loop with if – else.
* There is only one difference between them.
  + While run multiple times, but if run only one time.
* We can use break for exiting from of loop.

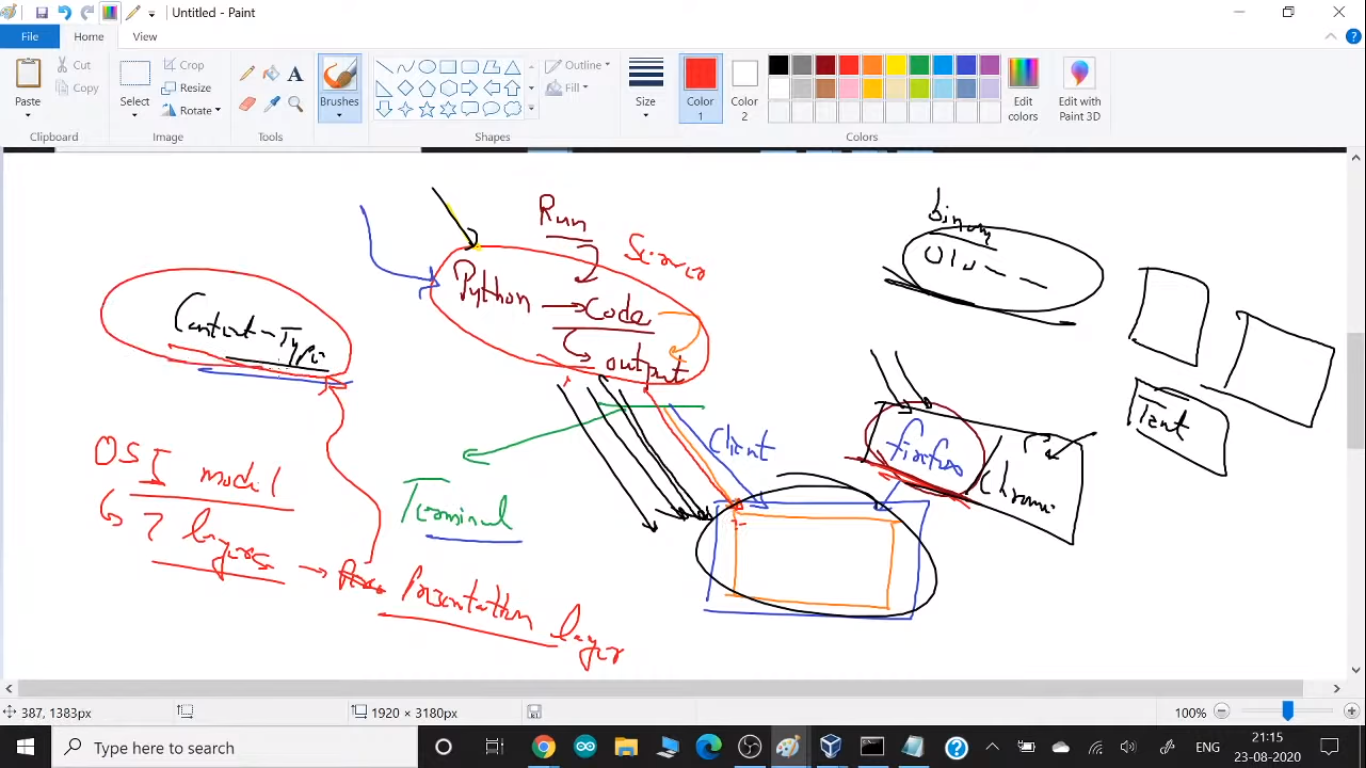
**Session 05**

**CGI**

* If you want run code from different laptop without local or remote login (You don’t know login – password), run a code and see output in your local system.
* You have to send packet for running a code… so you have to send a request.
  + Here we need some program in that system, which will take your request.
  + So, on your behalf this program will go and run a program and will send output to you.
    - Without remote login
    - Without going there
    - Without knowing login—pass
    - On your behalf
  + So, you can think it as a Interface.
  + That system provides you one gate – Interface.
    - This is a common gateway for any programming language, for running any code.
  + This program is known as **Common Gateway Interface (CGI).**
* This program is giving you service so it is known as server and here you are known as client.
* Web program or web services are giving you this type of CGI services.
  + Concept name – CGI
  + For Implement concept – enable webserver
  + That system – webserver
* Installing – configuring webserver
  + CGI facility is already enabled in Apache software.
    - You have to write your code inside /var/www/cgi-bin.
  + Yum install httpd
  + Systemctl start httpd
* Connectivity
  + You have configured webserver.
  + Client need web client program.
    - chrome
  + client have to use https protocol.
* If you want to run your code as a webApp as a CGI, then you have to deploy your code in /var/www/cgi-bin.
* For accessing this you have to use <http://192.168.43.128/cgi-bin/mycode.py>
  + If instead of you someone else (webserver) want to run your program you have to make file executable.
    - Chmod +x mycode.py
  + Now you have to tell which interpreter or compiler to run.
    - Extension is only for you, for your reference.
    - For this you have to use #! this symbol.
      * This is known as shebang / hashbang.
      * This symbol will tell which interpreter to use.



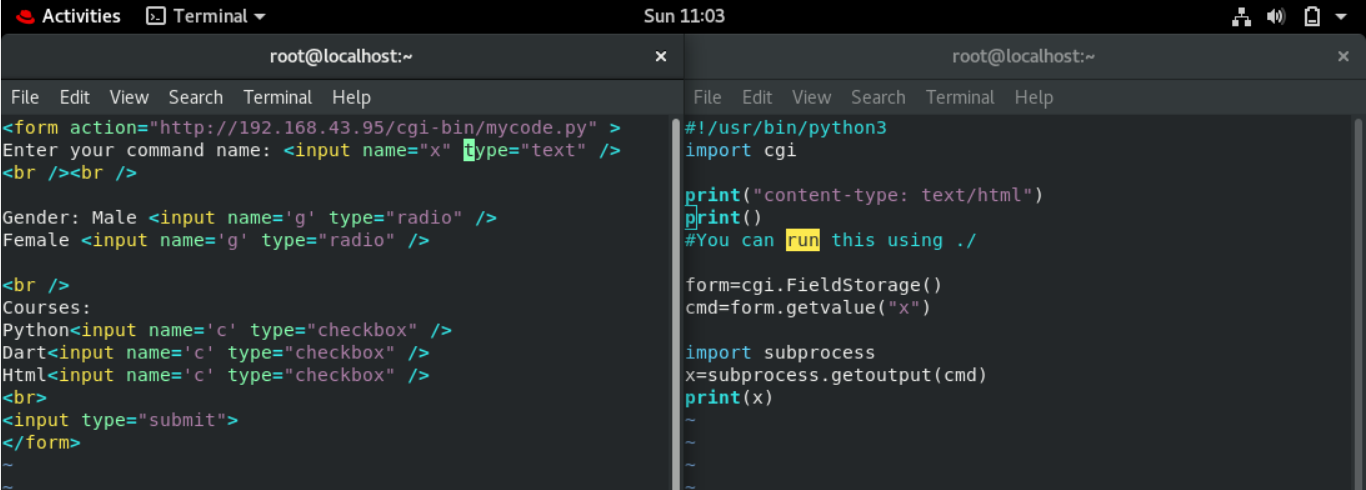
* + - * + Now you can also run this code using ./mycode.py
    - Now you have to do one more thing.
    - Here your output is going to the client browser and printed there.

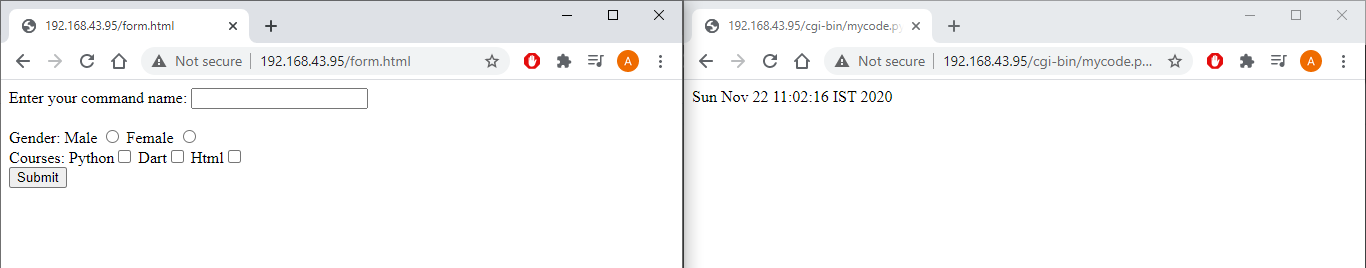


* + Here output is printing on the terminal or on top of client browser.
    - Here you have to tell which format you want print output on top of browser.
    - Here python output go to the client using network, It will convert into the binary.
    - So, your browser don’t know convert this binary to the image, video or normal text.
  + This is the meaning of presentation layer.
  + Before printing output browser look into first line and search for content-type.
    - Content-type is a header, to tell your browser is header is till here and after that it is body part. For telling this we have to use one line before body.
* Sometime OS module not work. So instead of that you can use subprocess instead of OS module.
  + Import subprocess
  + X = Subprocess.getoutput(“date”)
  + Print(x)
* <http://192.168.43.128/cgi-bin/mycode.py>
  + This URL is going to OS and run something for you.
  + This is interface between client and your program.
  + Here you have to manage, what application you want to run.
  + This is known as Application Program Interface (API).
* In today’s world always all the URL server giving you API.

**Session 06**

* Here our website / page is static, they are only running date command, but we need dynamic webApp.
  + For this they have to ask / prompt you in the webApp.
  + For this we have to use fieldstorage() function.
* In html some tags don’t have closing tags.
  + Eg, <br>
  + So, instead of this we can write like this.
    - <br />
    - This is the best way to write this type of tags.
* For taking input we have to use <input> tag.
  + By default input type is text, for password we can write type=”password”
  + Input type=”submit”
  + You have to use name =”x” variable for holding data.
    - But still it will not send data via URL.
    - If you want to send data, all the box would part of big box – form tag.
  + You can also send data from one page to another page, for this you have to use action=” URL/path”





**Session 07**

* Redirection
  + When you type <http://www.gmail.com> then it will redirect you to the different URL of Gmail.
  + In header of the webserver it is mentioned that when someone come to this URL then redirect you to this different URL.
  + You can check this thing using curl command.
    - Curl -I <URL>
    - Print(“location: http://www.google.com”)

**Questions**

* How to read data from ram?
* How to whatsapp anybody?
* How to reverse a string?
  + String = ‘’How are you?’’
  + String[-1: : -1]
* What is the difference between checkbox and radiobox?
  + Radio box is just a variable, while checkbox is a list.
    - So, you can store only one value in one variable, while in checkbox you can give multiple choices.
    - We use this concept in radio box, so when you select first option it will overwrite and last data removed.
  + So, you can say you can select only one button, it’s a functionality of variable. It’s not because of radio box.
  + So, there is mainly 2 difference.

1. Radio box stores data in variable, checkbox store in list.
2. You can unselect in checkbox, but in radio box you cannot unselect selected choice.



**Content**

* **Session 01**
  + Programming language
  + Installing python
  + Anaconda distribution
  + Variables
  + Builtin function
    - Dir(\_\_builtins\_\_)
  + Modules – program file
  + Os.system()
  + Pip install pyttsx3
  + Pyttsx3.speak(“Hey how are you”)
* **Session 02**
  + Program file
  + Program – process
  + Type Inference
  + List datatype
  + Slicing operator
  + Numpy
* **Session 03**
  + Jupyter notebook
  + Escape sequence
  + If – else
* **Session 04**
  + String
  + In, And, Or
  + while
  + break
* **session 05**
  + Common Gateway Interface (CGI)
  + Installing webserver – httpd
  + /var/www/cgi-bin
  + Execute file
  + Stop firewall
  + Shebang – hashbang ( #! )
  + Header – body
  + Presentation layer
  + Subprocess module
  + Application program interface ( API)
* **Session 06**
  + Static-dynamic webapp
  + Html
  + Form – input – radio button-checkbox
  + Subprocess
  + Fieldstorage()