

## Chapter 2:-

### Algorithm:-

Ek algorithm ek work ko complete kerne ya kisi problem ko solve kerne k liye ek achi trh se defined instruction ka ek set ya collection h , in simple words kisi problem ko step by step solve kerna algorithm kehlata h.

Algorithm koi bhi achi trh se defined computational process h, jo input k roop m kuch value ya vakue k set ko leti h aur output k roop m kuch value ya value k set ka utpaadan kerti h.

Step 1:- BEGIN

Step 2:- var a,b

Step 3:- c=a+b

Step 4:- write c

Step 5:- END

### Characteristics:

Iski nimann vishestaye h-

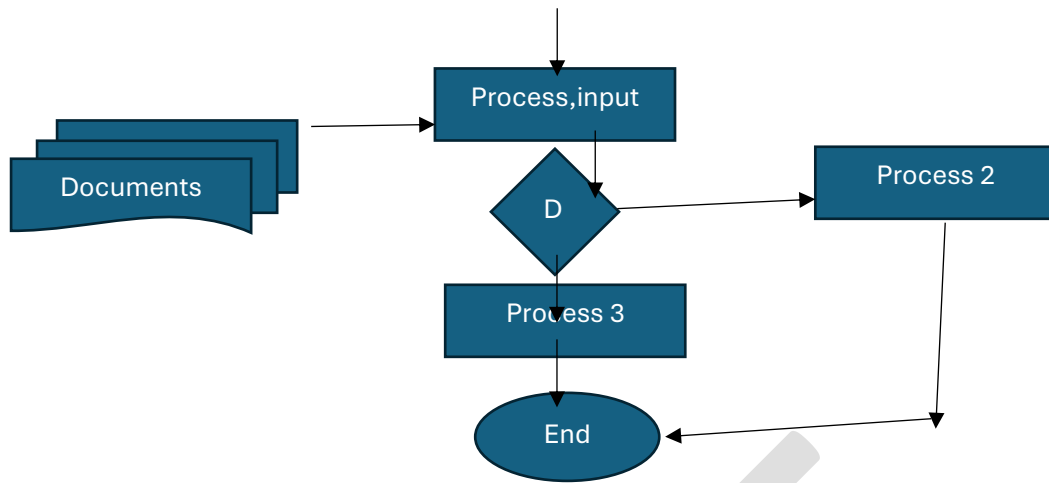
1. Algorithm kisi bhi programming lang per depend nhi karti isliye ye programming lang se independent h.
2. Ya general English lang h jise program k aim ko shi sabit kerne k liye steps m likha gya h jise asani se samjha ja skta hai.
3. Yah or koi programming code nhi jo program ko more read ya write k liye complexity paida karta h.

### Flow Chart:-

Ek flow chart process m shamil steps ka ek graphical representation h ek flow chart us logical order ko bhi show kerta jisme steps ko execute kiya jata hai. Ek flow chart m boxes hote h jinhe symbols or arrow kaha jata h jinhe flow lines bhi kaha jata h .



start



Flow chart kisi program ko design kerne ka ek graphical representation hota iski help se kisi program ko solve krne ka diagram ready kerte h isme various type k symbols ka use kerte h –

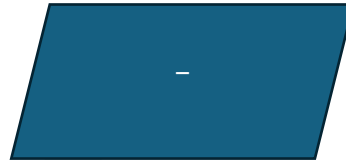
**Oval =>Start/stop**

**Parrallogram =>Input or output**

**Rectangle=> Processing**

**Decision box=>conditional box**

**Flow lines =>indicate control flow**



**Profit of flow chart:**

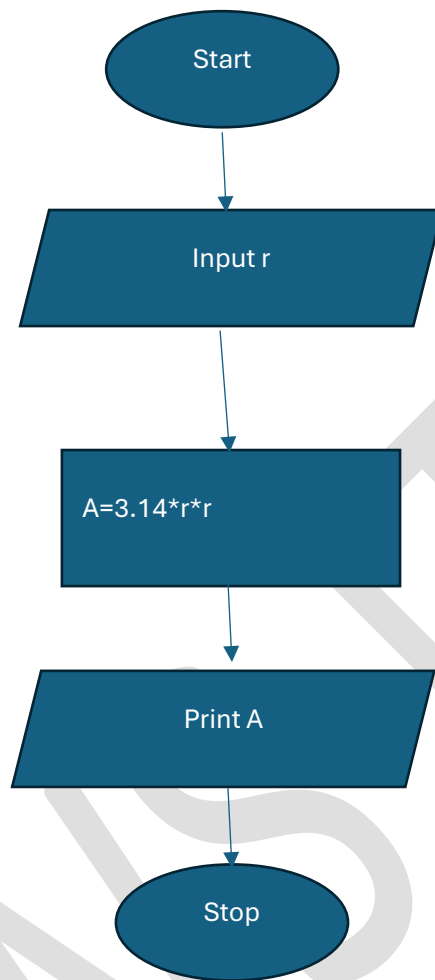
1. Flow chart ki help se problem ka achi trh se analysis kiya ja skta h isliye isme laagat or time ki km barbadi hoti h.
2. Program flow chart ek ache program documentation ke roop m kaam kerta h jiski jarurt various aim k liye hoti h jisse cheeze adhik kushal bnti h.
3. Flow chart debugging process m help kerta h
4. flow chart system analysis or program development phase k daruan ek guide k roop m work kerta h.
5. flow chart ki help se operation program ka maintance easy hota h .

**Limitation of Flow chart:-**

1. kabhi kabhi program logic bahot complex ho jata h.iss case m flow chart complex or ugly ho jata h.
2. yadi changes possible h to flow chart ko puri trh se re-drawing ki need hoti h.
3. flow chart ke symbols ko type nhi kiya ja skta h isliye flow chart ka reproduction ek problem bn jati h.

**Examples:-**

**Q1:- Draw the flow chart of additon of two numbers.**  
**Sol:-**



**Q2: Draw the flow chart of simple interest [ $si = p \cdot r \cdot t / 100$ ].**

**Q3: Draw the flow chart of area of circle.**

**Q4: Draw the flow chart of circumference of circle.**

**Q5: Draw the flow chart of area of rectangle.**

**Q6: Draw the flow chart of perimeter of rectangle.**

**Q7: Draw the flow chart to find the number is even or odd.**

**Q8: Draw the flow chart to find the numbers is greater of smaller**

**Q9: Draw the flow chart to check the year is leap or not**

**Q10: write a program to find the table**

**Q11: Fibonnaicc series ka Flow chart banao**  
**0,1,1,2,3,5,8,13,21,34,55,.....**

**#Star Pattern:**

**For i in range(0,5):**

**Print("\*",end="")**

*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*

## **Chapter 3:-**

### **Introduction to Python**

Python ek most popular programming language hai yah 1991 main Guido Van Rossum ke dawara develop kiya gya hai. Python ka use web development ,software development,Robotics, IOT (Internet Of Things), aur Ethical Hacking AI,Machine learning (ML), etc jagah pr use kiya jata hai.

Python ka syntax English language ke saman hai jaise C,C++ etc se alg hoti hai kyuki isme hum semicolon (;) braces ({} ) ka use nhhi kerte hai. Python ek interpreted language hai kyuki iske program ko check kerne k liye interpreter ki jarurt hoti h.

### **Features of Python:-**

1. Python support various type of plateform like MAC, WINDOWS,LINUX , RESPBERRY PIE etc.
2. Python interpreter pe chalta hai.
3. We can use python like POP's (Procedural oriented programing style) way , or Oops (Object oriented programming style) way or in Functional way.
4. Python ka use server pr web application create kerne ke liye kiya jata hai, iska use data base ko manage kerne ke bhi kiya jata hai.
5. Python ka use big data ko handle kerne ke liye kiya jata hai.
6. Python ka use production ke liye ready software development ke liye kiya jata hai.

### **Interpreter**

Intepreter High level language lo low level language main ya binary language main convert kerta hai jo machine ke dawara easily samjha ja skta hai, interpreter code ko line by line convert kerta hai.

### **Programing language**

**Programming language computer language hai jiska use computer program ko design kerne aur likhne ke liye kiya jata hai.**

**Jis trh hume baat cheet kerne ke liye ek language ki awshayakta hoti hai usi prakar computer main kisi program ko design ya code kerne ke liye ek language ki awshayakta hoti hai jise programming language kehte hai.**

**Classification of computer language:-**

**Computer language ko 3 bhago main bata gya hai-**

- 1. Low level language or Machine language**
- 2. Assembly level language**
- 3. High level language**

**Machine language=>**

**Machine level language ya low level lanaguge ya binary level language bhi kaha jata hai jo input aur output ke beech roop main binary number kin information ka use kerti hai machine lang main code badlne ke liye kisi mediator ki awshayakta nhi hoti hai.**

**Profit:**

**Yah computer ka fast tatha efficient use kerte hai,iske liye kisi translator ki awshayakta nhi hoti kyuki yah sidhe computer dawara smjha jata hai.**

**Loses:**

- 1.binary code or hexa code actual main un-readable hote hai**
- 2. machine lang ko debug kerne mushkil hai.**
- 3. machine lang main koi mathematical functional available nhi hoti hai**

**Assembly level language=>**

**Assembly level lang ve hoti hai jo machine lang ki tulna main iski vishehtaon lo badhati hai. Assembly lang machine lang ki tulna**

mian thoda fast or easy hai iske liye assembler ka use kiya jata hai jo assembly lang to low level lang main convert karta hai.

**Profit:**

Yah machine lang ki tulna main thoda fast hai.

**Loses:**

Isme implement karna katin hota hai, implement ke dauraan MNEMONICES (like com, serial) yad rakhna bahot mushkil hai.

**High level language:**

High level lang ek programming lang hai isme manav asani se likh sakta hai jo samjha ja skta hai high level lang ka mtlb samany English se hai jo manav samjh ke bahut ke karib hai.

High level lang programming lang ka use kerke programming code bahot asani se or sahi tarike se likha ja skta hai.

Program code ko binary information ko change kerne ke liye compiler ya interpreter ka use kerte hai. Kuch high level lang jaise C, C++, JAVA, PYTHON, PHP etc.

**Profit:**

1. High level language ka use kerne ka reason yah samjhne ya lagoo kerne main asan hai.
2. Machine lang tatha assembly lang ki tulna main yah fast hoti hai.
3. Yah Machine per depend nhi hoti hai isko hum kisi bhi machine per chala skte hai.

**Loses:**

1. Implement ke dauran application IDE (Integrated development environment) ko samjhne ke liye awshyak skill ki requirement hoti hai.
2. Application ke saare component ko chalane ke liye memory space awshyak hai.



## **Shortforms-**

**AlGol-Algorithm language**

**BASIC-Beginners all purpose symbolic instruction code**

**CoBol- Common business oriented language**

**Fortran-Formula Translator**

**List-List processor**

## **Linker :-**

**Source program -> assembler -> object code -> linker -> Executive code -> Loader**

**Linker ek system software hai. Jo binary lang main prapt code ko machine per chalne layak machine code main badal deta hai.**

**Jo computer m lang hai use source lang kehte hai. Ya hum use source code ke naam se bhi jante hai.aur iss source code ko jis lang main bdla jata hai use target lang kehte hai.**

**Iss prapt code ko object code kehte hai. Object code ko hi linker machine code main change karta hai. Application extension (.exe) prakar ki file ko linker hi develop karta hai.**

## **Loader:-**

**Loader OS ka ek hissa hota hai.jo disk per memory main rhne wali ek executable yogya file banata hai or lata hai or use start karta hai. linker ke baad aage ka kaam loader per hi aata hai. Linker ke dawara generate kiye gye machine code ko memory main load kerne ka kaam loader ka hi hota hai.**

**Loader program ke virtual address ko physical address main change ker deta hai. Jiske parinam swroop file तथा folder taiyar ho jate hai.**

“dono main antar yah hai ki linker program ko execution yogya banata hai jiski loader execution ke liye linker se prapt files ko main memory m load karta hai.

### **Testing:**

Testing program m kisi website ya software ka last step hota hai. Jiske dawara hum apne project ki all functionality ko check kerte hai. Testing ka use program ke result se related sabhi cheezo ko safe kerne k liye kiya jata hai aur goal ko thik kerne ke liye jo bhi anivary lgta hai uske according hum use thik ker skte hai. Kisi bhi project ko banane k baad testing bahot important hoti h. kuki iske dawara hum pratek cheez ko shi se check ker skte hai, yadi kisi cheez m koi problem aati h to uska pta chal jata hai.

Testing various type ki hoti hai-

1. **Functional testing**-> iss prakar ki testing ke dwara kisi project ki all functionality check ki jati hai.
2. **Performance testing**-> is testing ka use uski speed ko check kerne k liye kiya jata hai.
3. **Stress testing**-> yah wah testing hoti hai jiska use yah check kerne ke liye kiaya jata hai ki system pratikool paristithiyo m kaisa vyavhaar karta hai .
4. **Integration testing**-> yah wah testing hai. Jiske component ke ek group ko utpadan kerne k liye tatha software k beech interaction tatha test integration testing dwara kiya jata hai.

### **Debugging:-**

Debugging wah process hai jisme program main error aati hai, aur unhe thik kiya jata hai . dusre shabdo m hum keh skte hai ki kisi program ko compile kerte time koi error dikhai deti h or usko thik kerne ki process ko debugging kaha jata hai isme various typebki error aati hai.

1. **Syntax error**:- syntax error gaalt syntax likhne se generate hoti hai. Ise hum gramitical error bhi kehte hai.

For example=> jb hum kisi function ko define kerte time (:) lagana bhul jate hai to iss prakar ki error ko syntax error kehte hai , jaise hum koi program likhte time if condition ka use kerte hai tb if  $x > 0$  likhte hai or uske baad m colon (:) lagana bhul jate hai to iss error ko syntax error kehte hai.

2. **Runtime error:-** sabhi syntax error semantic error ko compiler dwara pta lagaya jata hai jo ek msg generate karta hai jisse hume error ka type or uski location pta chal jati h .

Runtime error ve error hoti hai jo code execution ke daruan aate h ye bahut hanikarak hoti h kuki ye chalte chalte program ko ye rok deti hai.

3. **Semantic error:-** program statement ke improper use k karan jo error aati h usko semantic error kehte hai.

4. **Logical error:-** logical error ek error hoti hai jo ek program ke source code m jisse ki result m error ya unexpected changes dikhai deta hai. Ye error tb aati h jb software specification ki value nhi ki jati hai.

**Testing & debugging m difference:-**

Testing	Dubugging
1. Testing humesha known condition ke saath start hota h.	Debugging unknown condition k saath start hota h
2. Yah program ko failure sabit karta h.	Yah programmer ka sanket h.
3. Yah error ya spashath sudhdhta ka pradarshan h.	Yah humesha ek deductive process k roop m mana jata h.
4. Testing kisi outsider dwara ki jati h.	Debugging kisi insider dwara ki jati h
5. Bahut sare test bina kiye design gyat kiye jate h.	isme design knowledge ka hona bahut awshyak h.
6. Testing ka purpose bug ko find kerna h	Debugging ka purpose bug ka karan search kerna h.

<b>7. Testing certain niyojit design or nirdharit kiya jana chahiye.</b>	<b>Debugging ki avdhi or avdhi ki process itni constant nhi ho skti.</b>
--	--

### **Documentation:-**

Sabhi bade software development project chahe jo bhi ho , badi matra m redocument banate h jaise files ko ek cupboard m rakhna or yadi badi files ho to unhe room m bhar dete h.iss document k nirmaan m software process laagat ka ek acha ratio kharch hota h, iske alawa documentation k dwara hum kisi bhi document ko asani se prapt kerte h.

**Document two type k hote h-**

- i) User Document**
- ii) Technical Document**

#### **1.User Document=>**

User document ek product ya service k liye documentation ko refer karta h jo last user ko pradan kiya jata h .user documentation product ya service ka use kerne k liye last usr ki help k liye design kiya gya h. ise aksar user helper k roop m mana jata h.

**2.Technical document=>** engineering m technical document kisi bhi prakar k documentation m ek technical product handling aur functionality ko describe karta h. technical documentation ka purpose product ki inner or outer dependencies ko samjhne k liye user k liye prapt jaankari pradan karta h.

### **Aim for documentation:-**

- 1. Documentation k dwara dhoka dhadhi or durvyavahar (misbehavior) ko km kiya ja skta h**

2. Customer ki need ka anupalan.
3. Union and state & local regulation ka anupalan.
4. Trained new employees.

### **Documentation:-**

Software documentation written ya graphical representation h jo computer software k saath hota h ya source code m embedded hota h, yah batata h ki software kaise operate hota h ya iska use kaise kiya jata h or alg alg bhumikaon m logo k liye alg alg cheeze ho skti h.

### **Comments in python:-**

Comments ka use python code ko samjhne k liye kiya ja skta h.

Comments code ko more readable banata h or code ko test kerte time execution ko rokne k liye kiya ja skta h.

Comments start with #, and python will ignore the code

Example: #This is a comment or #print("This statement will not work.")

### **Literals:-**

Literal ek variable m ya constant m diya gya raw data h.

Python m various type k literal h-

1. Numeric literals
2. String literals
3. Boolean literals
4. Special literals

### **Constant:-**

Ek constant ek type ka variable h jiska value change nhi ho skta. Constant ko container k roop m sochna helpful hota h jisme aisi info hoti jise baad m change nhi kiya ja skta.

### Variable:-

Variables data ke values ko store kerne k liye container h, ek variable define kerne k liye python k pass koi order nhi h jb hum phli baar ise ek value dete h to ek variable banaya jata h. variable ko kisi special type k saath define kerne ki need nhi hoti or set hone k baad bhi type change ker skte h.

### Example:

X=5,y="Sumit"

### Python Keywords:

Python keyword ko reserved word k name se bhi jana jata h inko hum variable, class, function etc k name k roop m use nhi kr skte h.

Inn keywords ka ek special meaning hota h or ye programming m special purpose k liye use kiye jate h. in other hands, python keyword ve words hote h jinka means compiler ya interpreter ko program k execution se phele pta hota h.

Python (version 3.6) m total 33 keywords hote h-

False	def	if	Raise	break	not
None	del	Import	return	class	of
True	Elif	In	Try	continue	pass
and	except	Is	While	for	
as	else	lambda	With	from	
assert	finally	nonlocal	Yield	global	

### Python Identifiers:-

Variables name ko identifiers k name se jana jata h

For example: ek integer type ka variable lete h jiski value 20 h  
tb variable ka name jo NUM h usko identifier kehte h.

NUM=20 (where NUM is a identifier and 20 is the value of it.)

### Python Data Type:-

Variable k type ko hi data type k name se jana jata h.

Jaise integer variable , string variable, tuple, list, dictionary etc.

Python data type ko 2 category m bata gya h

1. Mutable data type (changeable)
2. Immutable data type (not-changeable)

Immutable data type => ye various type k hote h

1. Numeric
2. String
3. Tuple

#### 1.Numeric data type in python:

a) Integer: python m int number ko koi upper limit nhi hoti iska mtlb hum isme badi se badi value le skte h , jo hamara system memory allow karta h , ex- n=10.

b) Float: float value vah value hoti h jo decimal point rakhti h,  
ex- n=12.56

c) complex numbers: complex number real and imaginary part se milker banta h, python complex number ko support krti h or ise cnum se represent kerte h.

ex- 3+4i,10-5j

#### 2.String data type in python:

Character ke ek sequence ko string kehte h, python m string ko represent kerne k liye Double or single quotes (inverted comma),(“ ”, ' '), ka use kerte h.

Ex- S="This is a string."

### 3.Tuple data type in python:

Tuple data type python m change nhi hote h. Element ke collection ka order h jo small bracket k ander likhe jate h or each element ko comma se seprate kiya h.

Ex- tuple=(5,6,4,3,8,9,10) , tuple2=("amit","Sumit","prachi")

### Mutable Data type in python:-

- A) List: list bhi tuple ki trh element k collection ka order h but list ko hum change ker kte h ,tuple ko nhi. List ko square bracket k andr and element ko comma lga ker seprate kiya jata h.

ex- List=[10,20,30,40,50,60]

- B) Dictionary: value and key ke collection ko dictionary kehte h, key unique hoti h iska mean yah h ki key duplicate nhi ho skti jbki value duplicate ho skti h.

Ex- Dict={

```
1:"Amit",  
2:"Prachi",  
"Name1": "Shreya",  
"Name2": "Amit",  
"Name3": "Shreya"  
}
```

- C) Set: Item k unordered and un-indexed collection ko set kehte h ,set k element ko curly brackets k andr and comma lga k value ko seprate kerte h.



**Ex- Set={6,2,5,4,8,3,2,5,9}**

**#14/02/2025**

**Slicing:-** Computer programming me ek string paramparagat k roop ke characters ka ek order hai. Mostly programming language string k roop me her pher kerne k liye built-in function provide kerti h, yani hum string ko add ker sakte h , find ker sakte h, string se sub-string nikal sakte h etc.in simple words koi bhi sentence ya characters ka collection ek string h.

Slicing ka matlab kisi string k ek hisse se sub string nikalna hota h. hum slicing syntax ka use kerke character ka ek sequence return ker sakte h. string k ek hisse ko return kerne k liye, ek colon dawara alg kiye gye start or end index define kare

**Syntax- str.obj[begin:end:step]**

**Begin=>** starting index jaha object ki slicing start hoti h.

**End=>** yah slicing end kerne ki process h jisme hamari slice -1 pr end hoti h.

**Step=>** character k beech ke distance ko batata h.

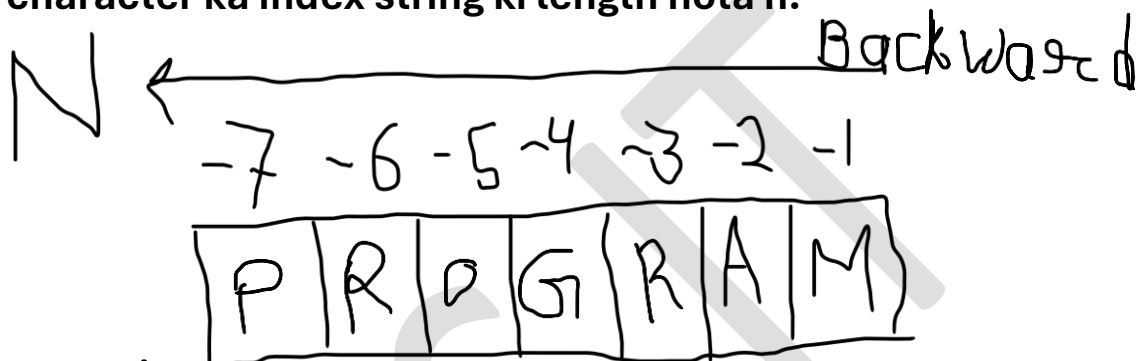
**Indexing:-**

Python k string data me specific character tak seedhe numeric index ya key value ka use kerke pahucha ja skta h. iss process ko indexing kehte h.

**Python me indexing 2 type ki hoti h-**

- 1. Positive index**
- 2. Negative index**

1. **Positive index:** positive index ka means left se start hokr right m hota h iska mtlb yah h ki forward direction m hota h. string m phele character ka index m 0 (Zero) hota h and isi trh agle index m 1 hota h or last character ka index string ki length se 1 km hota h.
2. **Negative index:** iska means right se left start hota h string k phele character ka index -1 and second ka -2 isi trh last character ka index string ki length hota h.



**Concatenation:**

Kisi bhi language m string ka merging ka combination hota h 2 string ko merge kerne ki process ko concatenation kehte h. Iske liye hum (+) operator ka use kerte h.

Ex:

Str1="Python"

Str2="programming"

print(str1+str2)

Output:

Python programming.

## Chapter 3:-

### **Operator ,Expression and Statement:**

#### **Operator:**

Operator ka mtlb variable ya value pr operation ko perform karana hota h. python m different type k operator hote h-

#### **1. Arithmetic Operator:-**

Arithmetic operator ve operator hote h jinka use arithmetic operation ko perform kerne k liye kiya jata h. jaise- +, -, \*, / etc.

Operator	Name	Example
+	Addition	x+y
-	Subtraction	x-y
*	Multiplication	x*y
/	Division	x/y
%	Modulus	x%y
**	Exponent	x**y
//	Float division	x//y

#### **2. Assignment operator:-**

Assignment operator ka use variable ki value ko assign kerne k liye kiya jata h, isko hum equal(=) sign se represent kerte h-

Operator	Example	Same as
=	X=3	X=3
+=	X+=3	X=X+3
-=	X-=3	X=X-3
*=	X*=3	X=X*3
/=	X/=3	X=X/3
%=	X%=3	X=X%3
**=	X**=3	X=X**3
&=	X&=3	X=X&3
^=	X^=3	X=X^3
=	X =3	X=X 3

#### **3. Comparison Operator:-**

**Comparison operator ka use 2 value ko compare kerne k liye kiya jata h.**

Operator	Name	Example
<b>==</b>	<b>Equal</b>	<b>x==y</b>
<b>!=</b>	<b>Not equal</b>	<b>x!=y</b>
<b>&gt;</b>	<b>Greater then</b>	<b>x&gt;y</b>
<b>&lt;</b>	<b>Less then</b>	<b>x&lt;y</b>
<b>&gt;=</b>	<b>Greater than or equal</b>	<b>x&gt;=y</b>
<b>&lt;=</b>	<b>Less than or equal</b>	<b>x&lt;=y</b>

#### **4. Logical Operator:-**

**Logical operator ka use conditional statement ko combine kerne k liye kiya jata h.**

Operator	Description	Example
<b>And</b>	<b>Returns true if both statement are true</b>	<b>x&lt;5 and x&lt;10</b> <b>(Dono condition true honi chaiye)</b>
<b>or</b>	<b>Return true if one of the statement are true</b>	<b>x&lt;5 or x&lt;10</b> <b>(Dono m se koi ek condition true honi chaiye)</b>
<b>Not</b>	<b>Return the result, return false if the result is true</b>	<b>not(x&lt;5 and x&lt;10)</b>

#### **5. Identity Operator:-**

**Inka use object ki tulna kerne k liye kiya jata h, Not if they are equal, but if they are actually the object with the same memory location.**

Operator	Description	Example
<b>is</b>	<b>Return true if both variables are the same object</b>	<b>x is y</b>
<b>is not</b>	<b>Return true if both variables are not same object</b>	<b>X is not y</b>

## 6. Membership Operator:-

Is operator ka use ek object m ek sequence m element k presence ko test kerne k liye kiya jata h.

It is used to check if a value or element is present in a sequence or not.

Operator	Description	Example
In	Return true if a sequence with the specified value is presented in the object.	x in y
not in	Return true if a sequence with the specified is not presented in the object.	x not in y

## 7. Bitwise operator:-

Iska use binary number ko compare kerne k liye kiya jata h.

Operator	Name	Description
&	AND	Sets each bit to 1 if both bits are 1.
	OR	Set each bit to 1 if one of two bits is 1.
^	XOR	Sets each bit to 1 if only one of two bits is 1.
~	NOT	Invert all the bits.

#17/02/2025

**Conditional Statements:-** inn statement ka use yah check kerne k liye kiya jata hai, ki kya kuch condition shi h? if conditions ko follow kiya ja rha h to statement k ander ka code chalaya jata hai.

**If:**

If keyword check krta h ki kya koi condition true h yadi ha to if k ander ka code chalaya jata h otherwise pura code block chodh diya jata h.

**Ex-**

**a=30**

**if(a>18):**

**print("This is Working.")**

**if-else:**

This is pretty much similar to the if statement the only difference is that another block of code is written inside the else statement which is run when the condition of the "if" statement is found to be false.

**Ex-**

**a=30**

**if(a>18):**

**print("IF block is working.")**

**else:**

**print("Else block is working.")**

**elif:**

python m elif ek keyword h, yadi last condtions true nhi thi to yah block run (execute) hoga.

**Ex-**

**a=30**

**if(a>18):**

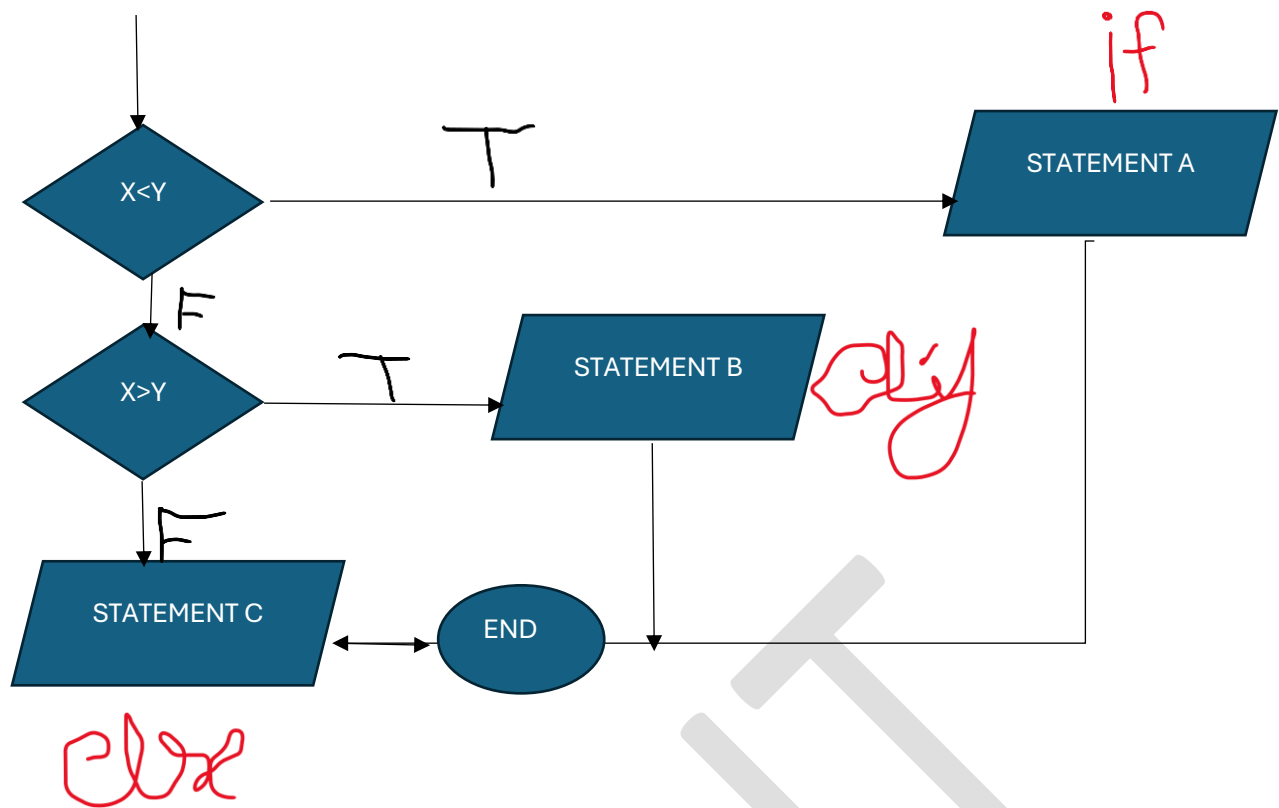
**print("IF block is working.")**

**elif(a<18):**

**print("Elif block is working.")**

**else:**

**print("Else block is working. That is a is equal to 18.")**



**Loop:-**

Python has two type primitive loop command.

1. While loop
2. For loop

1. **While loop:-** while loop k saath hum statement k ek set ko tb tk execute ker skte h jb tk ki koi condition shi ho. While loop phele condition ko check karega yadi use condition shi lgti h to statement ko execute kerta h or yah fir se next execution k liye condition ko firse check karega ki kya condition firse shi h?

Yah process tb tk jari rhta h jb tk ki condition false n ho jaye. While loop ko ready kerne k liye koi relevant variable ko define kerne ki need hoti h jise `i=1` pr set kerte h

Ex- print i as long as i is less than 6:

`i=1`

`while(i<6):`

`print(i)`

`i++; #i=i+1`

**1,2,3,4,5**

## 2. For loop:

Ek for loop ka use ek sequence (set,list,tuple,dictionary or string) per iterating ke liye kiya jata h. yah other OOPS language m paye jane wale method ki trh work karta h. loop k liye hum statement k ek set ko execute ker skte h.

Ex- print each fruit in a fruit list:

```
Fruits=["apple","mango","banana","cherry","grapes"]
```

```
for x in Fruits:
```

```
    print(x)
```

Note: the for loop does not require an indexing variable to set beforehand.

### The Break statement:

Break statement ka use loop k flow ko rokne k liye kiya jata h, reek keyword k saath hum ek loop ko firse chalne k liye rok skte h bhale hi condition shi ho.

Ex-

```
for i in range(1,11):
```

```
    if(i==6):
```

```
        break
```

```
    else:
```

```
        print(i)
```

### The continue statement:

Continue keyword ka use loop ke ek particular iteration ko rokne k liye kiya jata h. ex-

```
for i in range(1,11):
```

```
    if(i==5 and i==8):
```

```
        continue
```

```
    else:
```

```
        print(i)
```



### **The pass Statement:**

Python programming m pass statement ek null(empty) statement h. python , ek comment or pass keyword k beech yah difference h ki interpreter comments ko puri trh ignore ker deta h jbki pass keyword ko ignore nhi ker skta.

Suppose hamare paas ek loop ya function h jo abhi tk execute nhi hua h but hum ise future m execute kerna chahte h to hum pass keyword ka use ker skte h, kuki kisi bhi loop k pass empty body ni ho skta so interpreter ek error uthayega, isliye hum pass keyword ka use ek body banana k liye kerte h jo kuch bhi nhi kerta h, hence pass execute hone pr kuch nhi hota isliye iska result (NOP) No Operation hota h.

### **The assert keyword:**

Code debug hona chahiye, kerte time assert keyword ka use kerte h. assert keyword apko test kerne deta h ki kya aapke code m koi condition true h if no to program ek Assertion Error raise karega, yadi code wrong h to hum ek message bhi likh skte h.

#18/02/2025

## **Chapter 5:**

### **Sequence data type**

**List:** Python m list ek ordered collection h. yadi hum ek single entity k roop m alg alg elements k group ko represent kerna chahte h tab hume list ki need hoti h. list m order preserved hota h or duplicate elements ki bhi anumati hoti h.

#### **Properties of list:**

1. Insertion order preserved hota h
2. Duplicate value allow hote h
3. Heterogeneous object ki anumati h, iska mtlb isme mixture element le skte h. ex: number,string etc.
4. Ek list k element ko index dawara access kiya ja skta h
5. List dynamic h, kyuki hum apni need k anusar isko ghta or bada skte h.
6. List object mutable hote h,mtlb hum list k content ko change ker skte h

7. List m element ko [] square bracket m Rakha jata h or each element ko comma se separate kiya jata h.

#### Creation of List Object:

1. Create a Empty List:-

```
List=[]  
print(List)  
Output: []
```

2. Create a list with element:-

```
List=[5,10,15,20,25]  
print(List)  
Output: [5,10,15,20,25]
```

3. Creating list with dynamic input:-

```
List=eval(input("Enter the list:"))  
print(List)  
Output: Enter the list: [10,20]  
[10,20]
```

4. Creating a multi-dimensional list:-

```
List=[["Apple","Banana"],["BMW","Bently"],["Burger","Pizza"]]  
print(List)
```

#### Accessing Element on the list:-

1. Accessing of element from list using positive index:-

```
List=["Apple","Banana","Mango"]  
print(List)  
print("List[0]:",List[0])  
print("List[1]:",List[1])  
print("List[2]:",List[2])
```

Output:

```
["Apple","Banana","Mango"]  
List[0]:"Apple"
```

**List[1]:"Banana"**

**List[2]:"Mango"**

**2. Accessing element from list using loop:**

**List=["Apple","Mango","Banana"]**

**for I in List:**

**print(i,":",List[i])**

**output:**

**0:Apple**

**1: Mango**

**2:Banana**

**Adding element in a list:-**

**1. Append:**

Python m built-in function **append()** ka use kerke list m element ko add kiya ja skta h,isle dawara list m ek bar m ek hi element list k last m add hota h.

**2. Insert:**

Insert function ka use position ke saath value ko add kerne k liye kiya jata h, insert function m 2 argument (index,value) hota h.

**3. Extend:**

Iss function ka use list k last m ek time m multiple elements ko add kerne k liye kiya jata h.

**Removing element from a list:-**

**1. Remove:**

Iss function ka use di gyi list m element ko search kerna or phele matching element ko hatana hota h, yadi item kayi baar list m hota h to yah fjirst matching element ko remove karega.

**2. Pop:**

Iss function ka use kerke list m kisi element ko nikalne or return kerne k liye bhi kiya jata h but default roop se yah list k specific position se element ko remove kerne k liye , list k kewal last element ko remive kerne k liye use kiya jata h.

### Difference between remove and pop method:

Remove	Pop
1. remove() ka use hum list se special element ko hatane k liye kerte h.	1. pop() ka use hum list se last element ko hatane k liye kerte h.
2. yah kisi nhi value ko return nhi karta h.	2.yah element ko return ekarta h.
3. yadi special element available nhi h to hume error milti h.	3.yadi list blank h to ek error milti h.
4. syntax: list.remove(element)	4.syntax: list.pop()
5. ex: list=[10,20,30] list.remove(10) print(list) Output: [20,30]	5.ex: list=[10,20,30] list.pop() print(list) output: [10,20]

### Ordering element of list:-

1. reverse(): iss function ka use hum list k element ko reverse order m kerne k liye kiya jata h, iss function m koi argument nhi hota or nahi yah value ko return karta h. yah kewal element ko reverse karta h.

ex:

```
list=[10,20,30,40,50]
```

```
print(list)
```

```
list.reverse()
```

```
print(list)
```

output:

```
[10,20,30,40,50]
```

```
[50,40,30,20,10]
```

2. sort(): iss function ka use specific order m di gayi list k element ko sort kerti h.

syntax: list.sort(key=.... , reverse....)

key= function jo sort comparison k liye ek key k roop m work kerti h.

reverse=if true then list will be reversed.

**Note:-**

**For number: default natural sorting order as it is order rhta h.**

**For string: default natural sorting order alphabet order hota h.**

**Ex:**

```
x=[2,5,3,50,12,67]
```

```
x.sort()
```

```
print(x)
```

```
output: [2,3,5,12,50,67]
```

**Using mathematical operator for list object:-**

1. **Concatenation operation:** iss operator ka use do list ko ek list m convert kerne k liye kiya jata h. iske liye (+) operator ka use kerte h.

```
x=[10,20,30]
```

```
y=[40,50,60]
```

```
z=x+y
```

```
print(z)
```

```
output: [10,20,30,40,50,60]
```

- 2.

3. **Repetition operator:** hum repetition operator ka use list ki specified number m element ko repeat kerne k liye kerte h.

```
x=[10,20,30]
```

```
z=x*3
```

```
print(z)
```

```
output: [10,20,30,10,20,30,10,20,30]
```

4. **List comprehension:**

List comprehension new list banana k liye ek acha tarika provide kerta h. list comprehension m koi condition true hob hi skti h or nhi bhi.

**Ex:**

```
list=[]
```

```
for a in range(1,11):
```

```
    list.append(a**2)
```

```
print(list)
```

```
output: [1,4,9,16,25,36,49,64,81,100]
```

## Tuples:

Tuple python object ka ek collection h jo object comma dawara alg hote h. tuple list k saman hote h but inme main difference yah hota h ki list mutable hoti h, jbki tuple immutable hote h.iska means yah h ki list k content ko hum change ker skte h but tuple l content ko change nhi ker skte h, isliye tuple ko read only version bhi kaha jata h

Yadi hamara data fix h or kabhi change nhi hota tb hum tuple ka use kerte h. tuple ke liye hum parenthesis (open/round brackts) ka use kerte h or element ko comma lgakr separate kete h. isme parenthesis optional hoti h.

### Properties of tuple:

1. Insertion order preserved rhta h, duplicate elements allow hote h.
2. Tuple positive or negative index ko support kerta h.
3. Tuple k element ko small bracket m likha jata h or ye element comma dawara separate hote h, parenthesis optional h.

### Profit of tuple:

1. Tuple list ki tulna m fast hote h.
2. Yadi hume data ko change nhi kerna hota h tb hume list k bajay tuple use kerna chahiye kuki yah hamare data ko accidental changes se bachata h.
3. Tuple ko dictionary k roop m use ker skte h jbki list ko nhi use ker skte h.

### Create a tuple:

```
Tpl=(10,20,30,40,50)
print(Tpl)
print(type(Tpl))
output: (10,20,30,40,50)
<class 'tuple'>
```

### Accessing element from tuple:

```
Tpl=(10,20,30,40)
print(Tpl)
print("Tpl[0]:",tpl[0])
print("Tpl[1]:",tpl[1])
```

```
print("Tpl[2]:",tpl[2])
print("Tpl[3]:",tpl[3])
```

output:

```
(10,20,30,40)
Tpl:10
Tpl:20
Tpl:30
Tpl:40
```

**Tuple packing and Unpacking:**

Packing ko hum values new tuple m rakhte h , jbki unpacking m hum unn values ko single variable m nkalte h.

Ex:

```
a=10
b=2.2
c="Hello"
d=2+4j
e=True
tpl=(a,b,c,d,e)
```

Tuple unpacking, tuple packing ki reverse process h hum ek tuple ko unpack ker skte h or various variables ko isko value assign ker skte h, Tuple unpacking k time variables ki sankhaya values k Barabar honi chahiye.

Ex:

```
Tple=(10,2.2,"Hello",2+4j,True)
a,b,c,d,e=Tple
print("a=",a,"b=",b,"c=",c,"d=",d,"e=",e)
```

**Difference between Tuple and List:**

**Tuple:**

1. Tuple parenthesis ( ) k andr comma se alg kiye gye value ka ek roop h jisme parenthesis optional hoti h.
2. Tuple object immutable h.
3. Tuple m content fix hota h
4. Ise dictionary k roop m bhi use ker skte h.
5. Ex: tpl=(1,2,3) or tple=1,2,3

**List:**

1. List [] square bracket k andr comma se alg kiye gye value ka ek group h jisme brackets compulsory hoti h.

2. List object mutable hoti h.
3. List m content fix nhi hota h.
4. List ko dictionary k roop m use nhi ker skte h.
5. Ex: list=[1,2,3]

#### **Tuple Comprehension:**

Comprehension kisi object ya item pr looping ya iterating kerke unhe container m assign kerkr kaam kerte h but tuple assignment prapt kerne m unable h kuki ek baar tuple banne k baad ise add ya subtract nhi kiya ja skta.

Python m tuple comprehension ko support nhi karta h because some matter in which to function create and execute which is very expensive in python.

Ex:

```
Tpl=(x*2 for x in range(1,5))
```

```
for x in Tpl:
```

```
    print(x)
```

```
print(type(Tpl))
```

output:

```
2
```

```
4
```

```
6
```

```
8
```

```
Class '<generator>'
```

Yaha hume tuple object nhi mill rha, hume tuple generator mill rha h.

#### **Dictionary:**

Hum single entity k roop alg alg object k group ko represent kerne ke liye tuple, list and set ka use kerte h. yadi hum key value ke pair k roop m object k group ka represent kerte h tb hum dictionary ka use kerte h. ek item m ek key or value ek pair k roop define ki jane wali value h. dictionary k element ko curly brackets (braces) ke andr likha jata h or usko comma k dawara separate kiya jata h. dictionary m key unique hoti h jbk value duplicate ho skti h

#### **Syntax:**

```
Var={<key>:<value>}
```

#### **Properties of Dictionary:**

1. Dictionary m key duplicate nhi hoti jbk value ho skti h.
2. Key or value dono k liye heterogenous object ki anumati hoti h.
3. Insertion order preserved nhi h.



4. Dictionary mutable h.
5. Dictionary dynamic h.
6. Indexing or slicing concept implement nhi hote.

Creating a dictionary:-

```
Dct[1]="Ankita"  
Dct[2]="Prachi"  
Dct[3]="Amit"
```

Output:

```
{1:"Ankita",2:"Prachi",3:"Amit"}
```

Ex:

```
Student['ID']=CWS/03/01  
Student['Name']="Sumit"  
Student['Mob']=9118926216  
Student['Add']="Naini"  
print(Student)
```

output:

```
{'ID':CWS/03/01,'Name':"Sumit",'Mob':9118926216,'Add':"Naini"}
```

To Access data from dictionary:

```
Std={'ID':CWS/03/01,'Name':"Sumit",'Mob':9118926216,'Add':"Naini"}  
print(Std['ID'])  
print(Std['Name'])  
print(Std['Mob'])  
print(Std['Add'])
```

output:

```
CWS/03/01
```

```
"Sumit"
```

```
9118926216
```

```
"Naini"
```

**Updating Element in a dictionary:**

```
Dct={'Name':'Sumit','id':100}
print("Reference id value is:",id(Dct))
Dct['id']=200
print(Dct)
```

**Deleting element in a dictionary:**

**Dictionary se element ko delete kerne k liye following rule hote h-**

1. **pop():**- yah method provide ki gyi key k saath item ko hta deta h or value ko return karta h.

```
d={1:100,2:200,3:300,4:400,5:500}
print(d.pop(4))
print(d)
```

output : d={1:100,2:200,3:300,5:500}

2. **popitem():**- yah method ek arbitrary item ki value ko nikalta h or return karta h.

ex: d.popitem(5)

3. **clear():**- yah method ek baar m sabhi item ko hta deta h.

ex:  
d.clear()  
print(d)

4. **del keyword:**- del keyword individual item ya puri dictionary ko hta deta h.

ex:  
del d[2]  
print(d)

**Dictionary concatenation:-**

1. **using update():**- python m update() ka use kerke ek list ko dusri list m merge ker diya jata h but isme dusri dictionary ko pheli dictionary m mila diya jata h.

ex:  
dict1={'a':10,'b':20}  
dict2={'c':30,'d':40}

```
dict2.update(dict1)
print("dict2:",dict2)
output: dict2={'a':10,'b':20,'c':30,'d':40}
```

## Chapter-6:-

### Functions:-

Ek function code ka ek block hota h jo tbhi chalta h j bise call kiya jata h.

Ap kisi function m parameter k roop m known data pass ker skte h.ek function pariranamsavroop data return ker skta h.

### Creating a function:-

Python m function def keyword ka use kerke define kiya jata h.

Ex:

```
def my_function():
    print("Hello world")
```

### Calling a function:-

Kisi function ko call kerne k liye , bracket k phele function name use kare.

Ex:

```
my_function()
```

### Arguments or parameter:

Information ko parameter k roop m works ko define kiya ja skta h.

Bracket k andr function name k baad parameter define kiye jate h, hum jitne chahe utne parameter add ker skte hbs unhe comma dekr alg kare

Ex:

```
def my_fun(fname,lname):
    print("Hello"+fname+lname)
```

```
my_fun("Shreya","Mishra")
```

output: Hello Shreya Mishra

### Top-Down Approach:-

Top down approach m ek complex algorithm ko chote chote pieces m break ker diya jata h, jise module kaha jata h. inn module ko orr chote

pcs m break kiya jata h or tb tk kiya jata h jb tk khandit nhi ho skte .iss process ko modularisation kaha jata h.

Hence modularisation process k time apko algorithm ki akhandta or maulikta ko humesha banaye rakhna chahiye. Ek bdi problem ko chote pcs m break kerke , top down approach algorithm ko design kerte time aamtaur pr hone wali complexity kôkm kerta h.iske alawa iss approach m code m each function unique h or other work ko independently kerta h. c programming lang m top down approach ka kafi use kiya jata h.

**Recursion function:-**

Python function recursion ko bhi accept keta h jiska means h ki ek defined function khud ko call ker skta h.

Recursion ek general mathematical or programming assumption h jiska mtlb h ki ek function khud ko call kerta h ,eska means yah h ki result tk pahuchne k liye ap data k madhayam se loop ker skte h.

**Ex:**

```
def recursive():
```

```
....
```

```
    recursive()
```

```
....
```

```
recursive()
```

**profit of function:-**

1. function ka use kerke hum ek program m ek hi code ko bar bar likhne se bach skte h.
2. hum kisi bhi program m kisi bhi jgh pr function call kara skte h.
3. function ki help se hum bade program ko easily handle ker skte h.
4. reusability function m available hoti h.

**Type of function:-**

**Function 2 type k hote h.**

1. User define function
2. Built in function

1. User define function:

Jo function use k dawara define kiye jate h unko user define function kehte h.

Ex:

```
def add(a,b):  
    print(a+b)  
add(5,10)
```

## 2. Built-in function:

Jo function python software ke saath automatic roop se work kerte h ,built in function kehte h. ex: print(),input(),type(),id(),dir() etc.

Built-in function different type k hote h.

### a. Library function:-

Python m standard function ki badi library h jo generally programming work k liye use ki jati h , isme following function hote h-

1. input(): iska use user se input lene k liye kiya jata h .
2. print(): screen p output ko show kerne k liye kiya jata h
3. eval(): isme expression ko pass karaya jata h or program k andr python code ko run kerta h, dusre sabdo m hu keh skte h ek string ki value prapt kerne k liye eval() ka use kerte h.

### b. String function:-

String function k saath hum string ko modify or manipulate ker skte h-

- 1.count() : Returns the number of times a specified value occurs in a strung.
2. capitalize() :converts the first character to upper case
- 3.find() :search the string for a specified value and returns the position of where it was found.
4. lower() : converts the string into a lower case
5. islower(): returns true if all character in the string are lower case.
- 6.istitle(): returns true if the string follow the rules of a title.
- 7.isupper() : Returns true if the all character are in upper case
8. replace(): returns a string where a specified value is replaced with a specified value.
- 9.swapcase():swap cases lower case becomes upper case and vice versa

10. `rfind()`: search the string for a specified value and returns the last position where it was found.
11. `strip()`: returns a trimmed version of the string.
12. `lstrip()`: returns a left trim version of the string.
13. `rstrip()`: returns a right trim version of the string.
14. `title()`: converts the first character of each word to upper case
15. `split()` : splits the string at the specified separators and returns a list.
16. `isalnum()`: returns True if all character in the string are alphanumeric .
17. `isalpha()`: returns true if all character in the string are alphabet.
18. `isdecimal()`: returns true if all character in the string are decimal.
19. `isdigit()`: returns true if all character in the string are digits.
20. `startswith()`: return true if the string starts with the specified value.
21. `endswith()`: return true if the string ends with the specified value.
22. `encode()`: returns an encoded version of string.

**c. Library and Numeric functions:-**

1. `input()`: allowing user input
2. `eval()`: evaluates and executes an expression.
3. `print()`: prints to the standard output devices.
4. `max()`: returns the largest item in an iterable.
5. `min()`: returns the smallest item in an iterable.
6. `pow()`: returns the value x to the power of y.
7. `round()`: round a number.
8. `int()` returns the integer.
9. `random()`: returns a random a value.
10. `ceil()`: rounds a number up to the nearest integer.
11. `floor()`: rounds a number down to the nearest integer.
12. `sqrt()`: returns the square root of any number.

#### **d. Date and time functions:-**

- 1. date: return date**
- 2. time: return time**
- 3. datetime: return date and time**
- 4. time: microsecond resolution m do time, dates ya datetime period**
- 5. tzinfo: returns the time zone.**

#### **Chapter 7:- File processing**

**File:-** file system storage pr ek name placed h jo baad m access k liye data record ko store karta h. yah ek non volatile memory ya hard disk m continuous storage ko enable karta h. jb hum kisi file ko read ya write kerte h tb hume use sbse phele open karna hota h or jb hum ise read ya write ker lete h to file ko close kerne ki need hoti h. file ko open kerne k liye open() or close krne ke liye close() ka use kerte h.

File hamare data ko store kerne k liye comman permanent storage area h. file ka use kerne k liye hum file I/O operation ko samjhna hota , iska mtlb h ki data file m kaise read or write kiya jata h. iss process ko file handling kehte h.

**Python m file 2 types ki hoti h-**

- 1. Text file**
- 2. Binary file**

**Text file:**

Text file ve file hoti h jinhe human asani se samjh lete h ye mostly letter, digit or special character k roop m hoti h. ek text file jisme each byte ASCII code k according ek character ka represent karta h. ASCII file kbhi kbhi plain text file kehlati h or mostly hum inhi file ka use data store kerne k liye kerte h. for example:- Amit.txt

**Text file bhi 2 type ki hoti h-**

- 1. Plain text file**
- 2. Rich text file**

## Binary file:

Binary file ve file hoti h jo data ko bit k sequence m store kerti h jo 6 bit or kbhi kbhi 16 bit ka bhi hota h. ye bits custom data ko show kerti h. or ye files alg alg type k data ko store ker skti h.

For example: audio,text,etc. ek saath file m store ker skti h normally hum file ka use binary data jaise image, video, file, audio, etc ko store kerne k liye kiya jata h.

## Difference between text and binary file:-

### Text file:

1. Text file m bit character ko display karta h.
2. Text file jaldi kharab nhi hoti or kisi small change se effect nhi karta.
3. Text file m plain text ke roop m data ko store kiya jata h.
4. Text file m file format ka use kiya jata h or ise open kerne k liye simple text editor ka use kiya jata h.
5. Text file user ke liye read kerne m easy hoti h.

### Binary file:

1. Binary file m bit custom data ko display karta h.
2. Binary file ek small change se bhi damage ho jati h.
3. Binary file m alg alg method se data ko store kiya jata h.
4. Binary file m aisa nhi hota.
5. Binary file read kerne m easy nhi h.

## Different mode m files ko open and close kerna:-

1. File opening ke mode: python m koi bhi operation kerne se phele hume sabse phele file ko open kerna hota h, file ko open kerne k liye ek built-in function `open()` ka use kerte h.  
Syntax: `file_object=open(file_name,access_mode)`  
`File_name` ek string h jo file k name ko define karta h.  
`Access_mode` program ko batata h ki file ko kis mode m open kerna h by default yah `<r>` hota h.

## File ko access kerne k nimn mode hote h-

1. `<r>`:- read operation ke liye existing file ko open karta h.
2. `<w>`:- write operation k liye ek existing file ko open karta h.
3. `<a>`:- append operation k liye ek existing file ko open karta h.



4. <r+>:- file m read and write k liye <r+> ka use kiya jata h.iske dawara file ka recent data nhi erase kiya jata h.
5. <w+>:- file m write and read k liye iska use kiya jata h yah existing file ko overwrite karta h.
6. <a+>:- file se data ko add and read k liye iska use kiya jata h yah existing data ko over-ride nhi karta.

## 2. File ko close kerna:-

File pr operation pura hone pr file ko close kerna padta h

Isle liye close() ka use kiya jata h.

Syntax- fileobj.close()

File object ke properties:-

Ek bar jab hum file open kerte h to hume file object mil jata h or iske guno ka use kerke uss file se related various details received ker skte h.

- a. Fileobject.name :- iske dawara open ki gyo file ka name present hota h.
- b. Fileobject.mode :-yah wah mode batata h ki jisme file open ki jati h.
- c. Fileobject.close :- yah batata h ki file close h ya nhi.
- d. Fileobject.readable :- yah batata h ki file readablw h ya nhi.
- e. Fileobject.writeable :- yah batata h ki file writeable h ya nhi.

## Reading from a file:-

Jb hum kisi file ko read kerna chahte h to hume sabse phele use open karta hota h fir files ko read krne k baad close kerne ki need hoti h, isliye python m ek file operation following order m hota h.

1. Open a file
2. Read a file
3. Close a file

Hum nimnlikhit read method ka use kerke text file se character data read ker skte h-

### 1.read(n):-

File Ko Banane Ke liye maximum n character read karega.

## **2.readable():-**

Yah file stream se read kiya ja skta h to yah shi h.

## **3.readline():-**

File se ek line read kerke return kerta h.

## **4.readlines():-**

File se lines ki ek list return kerta h.

## **Writing onto a file:-**

Python m ek file m likhne k liye file ko open krna hota hor access mode m 'w' likhne ki jarurt hoti h

Example:

```
File.open("Sumit.txt","w")
```

```
File.write("First Line\n")
```

```
File.write("Second Line\n")
```

```
File.write("Third Line\n")
```

```
File.close()
```

## **File functions:-**

### **1. Open():-**

Python m ek file ko open kerne k liye open() ka use kiye jata h,yah built-in function h, yah function ka file object hota h, jise handle bhi kaha jata h kuki iska use file ko read or mdify kerne k liye kiya jata h.

Syntax: fileobj=open("<file name>","<access mode>")

### **2. Close():-** iss function ka use file pr work complete hone k bad file ko close kerne ki need hoti h.

Syntax: fileobj.close()

### **3. Read():-** file k content ko ko read kerne k liye read function ka use kerte h.

Syntax: fileobj.read()

### **4. Read(n):-** file se maximum n character read karega.

### **5. Readline():-** yah function file se ek line ko read kerta h.

6. **Readlines():**- yah function file se data read kerke uski list return karta h, or ise limit bhi kiya ja skta h ki kitni line ki hume need h.
7. **Write():**- iss function ka use file m data ko likhne k liye kiya jata h.
8. **Writelines():**- yah kisi list k item ko file m write kerti h, jaha text ko daalte h wah file mode and stream position pr nirchar kerti h.
9. **Tell():**- yah ek file stream m current file position ko return karta h, iska mtlb hota h ki yah function file object ki current location return karta h.
10. **Seek():**- yah ek file stream m current file position ko set karta h current file position ko seek method se change ker skte h .  
**Syntax-** fileobj.seek(<offset>)  
**Offset=>** current file stream set kerne k liye position ko represent kerne wala number hota h

#### **Command line argument:-**

Jab hum program ko start hone se phele input dena chahte h tb command line argument diya jata h. command kine argument user dawara pradan kiye gye input kerne ka ek tarika h. python script ko execute kerne k time command argument se jo argument pass hota h usko command line argument kehte h.

#### **Chapter 8:- Scope and Module:-**

Python m ek name ek container h, jaha object ke name mapped kiye jate h. inka use situations m confusions se bachne k liye kiya jata h jaha ek hi name various namespace m exist kerte h, inhe module, function, classes etc dawara banaya jata h. python m hum each name k mapping k roop m ek namespace le skte h, isme related object ko define kiya jata h.

Jb python interpreter se start hoti h or jab tk hum bahar nhi niklte tb tk sabhi nirmat namo wala ek namespace banaya jata h. yhi reason h ki built-in function jaise id(), print() etc program k kisi bhi hisse m available hote h.

Module m various function or classes ho skti h. kisi function ko call kerne pr ek local namespace banaya jata h. jisme sabhi name define hote h.

#### **Scope:-**

Ek scope hierarchical order ko define karta h jisme name se object ki mapping prapt kerne k liye namespace ki khoj ki jati h , yah ek context h,

jisme variables hote h or jisme se unhe reference diya jata h yah variable ki pahuchni or jeevankal ko define karta h.

Python m scope ek important topic h jo variable ki visibility or accessibility ko define karta h . scope k according , python m variables ko three types ke scope m divide kiya ja skta h.

1. Global scope: global scope m variables program ke sabhi hisson m accessible hote h , ye variables program k shuruat m define kiye h or program k last tak accessible rhte h ,
2. Local scope: local scope m variables sirf us function ya block m accessible hote h jism eve define kiye gye h . ye variables function ya block ke bahar accessible nhi hote h.
3. Nonlocal scope: nonlocal scope m variables sirf us function ya block m accessible hote h jism ve define kiye kiye gye h, lekin ye variables outer functions ya block m bhi accessible hote h.

Example

```
x="Sumit" #global
```

```
def house():
```

```
    #x="Sumit" is not accessible inside the function
```

```
    x="manoj" #nonlocal
```

```
    #x="manoj " is accesble inside the function
```

```
    x="prachi" #local
```

```
House()
```

```
#x="prachi" is not accessible outside the function
```

```
print(x)
```

```
#x="manoj is accessible outside the function"
```

LEGB Rule:-LEGB ka full form "Local Enclosed Global Built-in" hota h. jab hum function ya program ke andar variable ko access kerte h to python LEGB Rule ko follow kerke uski value ko find karta h.

**LEGB Rule** ek mnemonic device hai jo python mein variable scope ko samjhne mein help karta hai. Yeh rule k anusar,python m variable scope ko nimnlikhit order m resolve karta h –

1. **L-> Local:** sabse phele ,python local scope m variable ko search karta h. yeh scope current function ya method ka scope hota hai.
2. **E-> Enclosing:** Agar local scope m variable nhi milta h,to python enclosing scope m variable ko search karta h. yeh scope outer function ya methods ka scope hota h.
3. **G->Global:** agar enclosing scope m variable nhi milta h , to python global scope m variable ko search karta h. yeh scope program ke sabhi function aur methods ke liye common hota h.
4. **B-Built-in:**Agar global scope m variable nhi milta h, to python built-in scope m variable ko search karta h.yeh scope python ke built-in function or variables ka scope hota h.

**Example:**

```
x=10 #global variable
def outer():
    x=20 #Enclosing variable
    def inner():
        x=30 #Local variable
        print(x)
    inner()
    print(x)
outer()
print(x)
```

**Module/Library/packages:-**

Ek module file ko define karta h jo python code ko rakhta h.

Each program jo python m create kerte h module jaisa behave karta h.In other words, module ek functions, class, variables ka group hai jo python program (.py) m save kiya jata h. Large program ko small program m break kerne k liye module ka use kerte h.

Hum sabse jyada use kiye jane wale function ko ek module m define ker skte h or use import bhi ker skte h. iske alawa inki definition ko alg alg program copy ker skte h.

Moduler ka means h ki alg, small, more manageable subtask ko, ek bade programming work ko break kerna

Module ek pre-written code ka collection hai jo kisi bhi program mein import kiya ja sakta hai. Modules Python mein code ko organize aur reusable banane mein madad karte hain.

**Module ke fayde:**

1. **Code Reusability:** Modules code ko reusable banate hain, jisse code ko duplicate nahi karna padta.
2. **Code Organization:** Modules code ko organize karte hain, jisse code ko padhna aur samajhna aasan hota hai.
3. **Easy Maintenance:** Modules code ko maintain karna aasan hota hai, kyunki modules ko alag-alag update kiya ja sakta hai.

**Module ko import karna:**

Python mein module ko import karne ke liye import statement ka upyog kiya jata hai:

```
import module_name
```

**Example:**

```
import math
```

**Module ke functions aur variables ko access karna:**

Module ke functions aur variables ko access karne ke liye module ka naam dot (.) operator ke saath upyog kiya jata hai:

```
module_name.function_name()
```

**Example:**

`math.sqrt(4)`

`math.pow(2,3)`

**Module ke types:**

1. **Built-in Modules:** Python mein built-in modules hote hain, jaise ki `math`, `time`, `os`, `calendar`, `datetime` etc.
2. **External Modules:** External modules Python mein install kiye ja sakte hain, jaise ki `requests`, `numpy`, `pandas` etc.
3. **Custom Modules:** Custom modules Python mein create kiye ja sakte hain, jaise ki apne program ke liye ek module create karna.

**Module commands:-**

2 type ki module command python m hoti h-

1. `Dir()`
2. `Reload()`

1. `Dir()`: `dir()` python ka built in function , jo string ko sorted list ko return karta h pr module k dawara object name rakhta h, list sabhi variables k name or function rakhta h jo module m define h.  
Jaise: `dir(math)`
2. `Reload()`: python interpreter ek session daruan kewal ek bar module ko import karta h yah cheezo ko more efficient banata h.

**Example:**

`Import time`

`from imp import reload`

`import example`

`time.sleep(30)`

`reload(example)`

`time.sleep(30)`

`reload(example)`

`print("This is testing")`

## Chapter 9: Numpy Basics:-

**Numpy:-** numpy ka full form Numeric python hai, numpy python k saath general purpose k liye ek array processing package h. yah ek library hai jisme multi-dimensional array mathematical and logical operation kiya ja skta h, iski nimn vishtaye hoti h-

1. Yah ek powerful n-dimensional object h.
2. C/c++ or fortran code ko integrate kerne k liye yah ek tool h.
3. Linear algebra , fourier transform or random numbers k liye yah useful hota h.
4. Numpy ko generic data ke efficient multi-dimensional container k roop m bhi iska use kiya jata h.
5. Numpy ka use kerke manmane dhang se data ke type ko define kiya ja skta h, jo numpy ko widly or teji se database ki wide form ke saath integrate kerne ki anumati deta h.

**SciPy:-** scipy ek free or open source python library hai . jiska use scientific computing or technical computing ke liye kiya jata h. scipy m linear algebra , integration, interpolation ,special functions or science and engineering m common work ke liye yah ek module h.

Scipy Numpy array k object pr bnta h numpy step ka hissa hota h scipy dawara use ki jane wali fundamental data ek multi-dimensional array h, jo numpy module dawara provide ki jati h.

**Differnce between numpy and scipy:-**

NumPy (Numerical Python) aur SciPy (Scientific Python) dono hi Python ke liye libraries hain jo numerical aur scientific computing ke liye upyog kiye jate hain. Inke beech mukhya antar nimnalikhit hain:

### ### NumPy:

1. Multi-dimensional arrays: NumPy multi-dimensional arrays ke liye support provide karta hai.
2. Basic mathematical operations: NumPy basic mathematical operations jaise ki addition, subtraction, multiplication, aur division ke liye support provide karta hai.



**3. Linear algebra operations:** NumPy linear algebra operations jaise ki matrix multiplication, eigenvalue decomposition, aur singular value decomposition ke liye support provide karta hai.

**4. Random number generation:** NumPy random number generation ke liye support provide karta hai.

### **### SciPy:**

**1. Scientific computing:** SciPy scientific computing ke liye upyog kiya jata hai, jaise ki signal processing, image processing, aur optimization.

**2. Specialized functions:** SciPy specialized functions jaise ki Fourier transforms, wavelet transforms, aur filter design ke liye support provide karta hai.

**3. Optimization and minimization:** SciPy optimization and minimization ke liye support provide karta hai, jaise ki linear programming, quadratic programming, aur nonlinear least squares.

**4. Statistics and random processes:** SciPy statistics and random processes ke liye support provide karta hai, jaise ki hypothesis testing, confidence intervals, aur random process simulation.

### **### Antar:**

**1. Focus:** NumPy ka focus basic numerical computing par hai, jabki SciPy ka focus scientific computing par hai.

**2. Functionality:** NumPy mein basic mathematical operations aur linear algebra operations ke liye support hai, jabki SciPy mein specialized functions aur optimization ke liye support hai.

**3. Complexity:** SciPy mein NumPy ki tulna mein adhik complex algorithms aur techniques ke liye support hai.

### **### Upyog:**

**1. Numerical computing:** NumPy numerical computing ke liye upyog kiya jata hai, jaise ki data analysis, machine learning, aur scientific simulations.

**2. Scientific computing: SciPy scientific computing ke liye upyog kiya jata hai, jaise ki signal processing, image processing, aur optimization.**

**In antaron ke aadhar par, aap apne project ke anusar NumPy aur SciPy ka upyog kar sakte hain.**

**Creating Numpy Array:-**

CMSIT