

பொதுவடிவம்:

Simple if கூற்று:

```
if <condition>:  
    statements-block1
```

- வயதை சரிபார்த்து வாக்களிக்க தகுதியா என அச்சிடும் பைத்தான் நிரலை எழுதுக.

```
நிரல்: x=int(input("Enter your age :"))  
if x>=18:  
    print("You are eligible for voting")
```

வெளியீடு 1: Enter your age :34

You are eligible for voting

வெளியீடு 2: Enter your age :16

>>>

if..else கூற்று:

```
if <condition>:  
    statements-block 1  
else:  
    statements-block 2
```

- உள்ளிடப்பட்ட எண் ஒற்றைப்படை எண் அல்லது இரட்டைப்படை எண்ணா என்பதைக் கண்டறியும் நிரலை எழுதுக.

```
நிரல்: a = int(input("Enter any number :"))  
if a%2==0:  
    print(a, " is an even number")  
else:  
    print(a, " is an odd number")
```

வெளியீடு 1: Enter any number :56

56 is an even number

வெளியீடு 2: Enter any number :67

67 is an odd number

பின்னலான if..elif..else கூற்று:

```
if <condition-1>:  
    statements-block 1  
elif <condition-2>:  
    statements-block 2  
else:  
    statements-block n
```

- If..elif..else கூற்றை பயன்படுத்தி கொடுக்கப்பட்ட மூன்று எண்களில் மிகச் சிறிய எண்ணைக் கண்டுபிடிப்பதற்கான நிரல் எழுதுக.

```
நிரல்: a=int(input("Enter Number 1:"))  
b=int(input("Enter Number 2:"))  
c=int(input("Enter Number 3:"))  
if a<b and a<c:  
    print(a,"is Smallest")  
elif b<a and b<c:  
    print(b,"is Smallest")  
else:  
    print(c,"is Smallest")
```

வெளியீடு:

Enter Number 1: 4

Enter Number 2: 7

Enter Number 3: 2

2 is Smallest

While மடக்கு:

```
while <condition>:  
    statements block 1  
[else:  
    statements block2]
```

- while மடக்கை பயன்படுத்தி 10 லிருந்து 15 வரை அனைத்து எண்களையும் அச்சிடும் நிரல். i=10

```
while (i<=15):  
    print (i,end='\t')  
    i=i+1
```

வெளியீடு: 10 11 12 13 14 15

For மடக்கு:

```
for counter_variable in sequence:  
    statements-block 1  
[else: # optional block  
    statements-block 2]
```

range ()ன் தொடரியல்:

```
range(start,stop,[step])
```

- for மடக்கு பயன்படுத்தி ஒற்றை இலக்க இரட்டைப்படை எண்ணை அச்சிடும் நிரல். நிரல் :for i in range (2,10,2):

```
print (i, end=' ')
```

வெளியீடு: 2 4 6 8

பயனர் வரையறுக்கும் செயற்கூறு:

```
def <function_name ([parameter1, parameter2...])> :  
    <Block of Statements>  
    return <expression / None>
```

பெயரில்லாத செயற்கூறுகள்:

```
lambda [argument(s)] :expression
```

- லாம்டா செயற்கூறுகள்:

நிரல் :sum = lambda arg1, arg2: arg1 + arg2

```
print ('The Sum is :', sum(30,40))
```

```
print ('The Sum is :', sum(-30,40))
```

வெளியீடு: The Sum is : 70
The Sum is : 10

பைத்தானில் List-ஐ உருவாக்குதல்:

```
Variable = [element-1, element-2, element-3 ..... element-n]
```

List - ல் ஒரு உறுப்பை அணுகுதல்

```
List_Variable = [E1, E2, E3 ..... En]
```

```
print (List_Variable[index of a element])
```

Sort() செயற்கூறு:

```
List.sort(reverse=True/False,key=myFunc)
```

Tuples உருவாக்குதல்

```
# வெற்று Tuples
```

```
Tuples_Name = ()
```

```
# n எண்ணிக்கை உறுப்புகளுடன் Tuples
```

```
Tuples_Name = (E1, E2, E2 ..... En)
```

```
# அடைப்புக்குறி இல்லாத Tuple உறுப்புகள்
```

```
Tuples_Name = E1, E2, E3 ..... En
```

இனக்குழுவை வரையறுத்தல்

```
class class_name:
    statement_1
    statement_2
    .....
    .....
    statement_n
```

- இனக்குழுவை வரையறுத்து அதன் உறுப்பு மாறிகளை அணுகுதல்

நிரல்: class Sample: #class variables

```
x, y = 10, 20
S=Sample() # class instantiation
print("Value of x = ", S.x)
print("Value of y = ", S.y)
print("Value of x and y = ", S.x+S.y)
```

வெளியீடு: **Value of x = 10**
 Value of y = 20
 Value of x and y = 30

ஆக்கி வழிமுறையின் பொதுவடிவம்:

```
def __init__(self,[args.....]):
    <statements>
```

1. உருவாக்கப்பட்ட பொருள்களின் எண்ணிக்கையை கொண்டிருக்கும் இனக்குழு மாறியை விளக்குதல்.

நிரல்: class Sample:

```
num=0
def __init__(self, var):
    Sample.num+=1
    self.var=var
    print("The object value is = ", var)
    print("The count of object created = ", Sample.num)

S1=Sample(15)
S2=Sample(35)
S3=Sample(45)
```

வெளியீடு: **The object value is = 15**
 The count of object created = 1
 The object value is = 35
 The count of object created = 2
 The object value is = 45
 The count of object created = 3

CREATE - கட்டளை

```
CREATE TABLE <table-name>
(<column name><data type>[<size>]
<column name><data type>[<size>].....);
```

ALTER கட்டளை

தொடரியல் 1: ALTER TABLE <table-name> ADD <column-name><data type><size>;
தொடரியல் 2: ALTER <table-name> MODIFY<column-name><data type><size>;
தொடரியல் 3: ALTER <table-name> RENAME old-column-name TO new-column-name;

DROP கட்டளை: **DROP TABLE table-name;**

TRUNCATE – கட்டளை: **TRUNCATE TABLE table-name;**

INSERT கட்டளை: **INSERT INTO <table-name> [column-list] VALUES (values);**

UPDATE கட்டளை: **UPDATE <table-name> SET column-name = value, column-name = value,... WHERE condition;**

DELETE – கட்டளை: **DELETE FROM table-name WHERE condition;**

SELECT கட்டளை: SELECT <column-list>FROM<table-name>;

DISTINCT சிறப்புச் சொல்லுடன் SELECT கட்டளை: SELECT * DISTINCT place FROM student;

ALL என்ற சிறப்புச் சொல்: SELECT ALL Place FROM Student;

BETWEEN என்ற சிறப்புச் சொல்:
 SELECT Admno, Name, Age, Gender FROM Student WHERE Age BETWEEN 18 AND 19;

NOT BETWEEN என்ற சிறப்புச் சொல்
 SELECT Admno, Name, Age FROM Student WHERE Age NOT BETWEEN 18 AND 19;

IN சிறப்புச் சொல்
 SELECT Admno, Name, Place FROM Student WHERE Place IN ("Chennai", "Delhi");

NOT IN சிறப்புச் சொல்
 SELECT Admno, Name, Place FROM Student WHERE Place NOT IN ("Chennai", "Delhi");

ORDER BY clause
 SELECT <column-name> FROM <table-name> ORDER BY <column1>,<column2>,...ASC| DESC ;

WHERE clause
 SELECT * FROM Student WHERE Age>=18 ORDER BY Name;

GROUP BY clause
 SELECT <column-names> FROM <table-name> GROUP BY <column-name>HAVING [condition];

HAVING clause
 SELECT Gender , count(*) FROM Student GROUP BY Gender HAVING Place = 'Chennai';

CSV- ன் Reader செயற்கூறு: csv.reader(fileobject, delimiter, fmtparams)

பைத்தானை இயக்குவதற்கான தொடரியல்

Python <filename.py> -<i> <C++ filename without cpp extension>

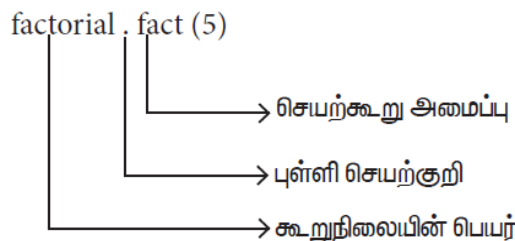
- Python - கட்டளை வரியிலிருந்து பைத்தான் நிரலை செயல்படுத்துவதற்கான சிறப்புச் சொல்
- filename.py - செயல்படுத்த வேண்டிய பைத்தான் நிரலின் பெயர்
- -i - உள்ளீட்டு முறைமை
- C++ filename without cpp extension - தொகுக்கப்பட்டு செயல்படுத்தப்பட வேண்டிய C++ நிரலின் பெயர்.

கூறுநிலையிலிருந்து செயற்கூறுகளை அணுகுவதற்கான தொடரியல்

<module name> . <function name>

எடுத்துக்காட்டாக:

```
>>> factorial.fact(5)
120
```



பைத்தானில் OS கூறுநிலை:

os.system ('g++' + <variable_name1> '-<mode>' + <variable_name2>)

பைத்தானில் getopt கூறுநிலை:

<opts>,<args>=getopt.getopt(argv, options, [long_options])