

Basic py coding 1

```
print("prgm started: ")
```

```
x=5
```

```
if x>10:
```

```
    print("x is grater: ")
```

```
else:
```

```
    print("x is smaller: ")
```

```
print("prgm ended: ")
```

prgm to chech which nor is grater in the given variable

```
a,b,c=10,20,30
```

```
print("prgm started: ")
```

```
if a>b and a>c:
```

```
    print("a is grater: ",a)
```

```
elif b>c:
```

```
    print("b is grater: ",b)
```

```
else:
```

```
    print("c is gratest: ",c)
```

```
print("prgm ended: ")
```

prgm to print D,FC,Fail based on ueer input

```
marks=int(input("enter marks: "))
```

```
print("prgm started")
```

```
if marks>=85:
```

```
    print("distinction")
```

```
elif marks>=60 and marks<=85:
```

```
    print("First class")
```

```
elif marks>=35 and marks<=60:
```

```
    print("seconfd class")
```

```
else:
```

```
    print("fail")
```

```
print("prgm ended")
```

prgm to print number is single, double or tribble digit from user i/p

```
num=int(input("enter number: "))
```

```
if num>0 and num<9:
```

```
    print("single digit: ")
```

```
elif num>10 and num<99:
```

```
    print("double digit: ")
```

```
else:
```

```
    print("tribble digit: ")
```

```
value
value=int(input("enter value: "))
if value:
    print("hi good morning: ")
if value>=100:
    print("python coding is awesome: ")
else:
    print("i am in esle block: ")
```

nested statements

if condition starts executing based on the true value of another if condition.

```
syntax:
if condition:
    if condition:
        stam 1
        stam 2
    else:
        stam 1
        stam 2
else:
    if condition:
        stam 1
        stam 2
    else:
        stam 1
        stam 2
```

EX:

```
gender=input("enter gender: ")
age=int(input("enter age: "))
if gender=="male":
    if age>=18:
        print("nalla boy: ")
    else:
        print("child: ")
elif gender=="female":
    if age>=18:
        print("nalla girl: ")
    else:
        print("child: ")
else:
    print("get lost you ediot: ")
```

EX:

w p to check the give n nor is divisible by 3 or 7 or both 3 & 7 are nt

```
num=int(input("enter number: "))
if num % 3==0:
    if num % 3==0 and num % 7==0:
        print("divisible by both 3 and 7:",num)
    else:
        print("divisible by 3 only:",num)
else:
    if num % 7==0:
        print("divisible by 7:",num)
    else:
        print("not divisible by 3 & 7: ",num)
```

or

```
num=int(input("enter number: "))
if num % 3==0:
    print("divisible by 3 only:",num)
elif num % 7==0:
    print("divisible by 7:",num)
elif num % 3==0 and num % 7==0:
    print("divisible by both 3 & 7: ",num)
else:
    print("not divisible by 3 & 7: ",num)
```

**w p to accept user input like loan ammount, existing loan, creadit score,
emp type then based on the given condition need to print loan is approved or not.**

```
loanamount=float(input("enter loan amount: "))
existingloan=float(input("enter existing loan amt: "))
creditscore=int(input("enter credit score: "))
emptype=input("enter employee type (salaried /self-emp): ").lower()
salary=float(input("enter sal amt: "))
if creditscore >=700 and existingloan<=50000 and emptype in ("salaried","self-emp"):
    if loanamount <=salary*10:
        print("loan approved: ")
    else:
        print("not eligible:")
else:
    print("criteria did not met: ")
```

or, CHATGPT

```
loan_amount = float(input("Enter Loan Amount: "))
existing_loan = float(input("Enter Existing Loan Amount: "))
credit_score = int(input("Enter Credit Score: "))
```

```
emp_type = input("Enter Employment Type (Salaried / Self-Employed): ").lower()
salary = float(input("Enter Monthly Salary: "))
```

Conditions

```
# if credit_score >= 700 and existing_loan <= 50000 and emp_type in ("salaried", "self-employed"):
#     if loan_amount <= salary * 10:
#         print("Loan Approved ✓")
#     else:
#         print("Loan Rejected ✗ - Loan amount too high compared to salary")
# else:
#     print("Loan Rejected ✗ - Eligibility criteria not met")
```

Match case statements.

it is a another type of branching statement in python. it will execute based on the patter matching given by the user and comparing with the case blocks.

syntax:

match variable:

case value 1:
statements

case value 2:
statements

case value 3:
statements

case value n:
statements n

- case nor r value can be any type int,str,special char,float etc.
- there should only unique case value.
- duplicate are not allowed.
- default case block should present at the last & it is optional.
- match statement include at least one case statement.

Ex 1:

```
option=input("enter the option: ")
```

match option:

```
case 'A':
    print("move left")
```

```
case 'B':
    print("move right")
```

```
case 'C':
    print("move up")
```

```
case 'D':
    print("move down")
```

```
case _:
    print("invaied option")
```

wpt builda simple calculator using match case.

```
print("simple calculation:")

option=int(input("enter the option:"))
a=int(input("enter the num for a: "))
b=int(input("enter the num for b: "))

match option:
    case 1:
        print("add",a+b)
    case 2:
        print("sub",a-b)
    case 3:
        print("mul",a*b)
    case 4:
        print("div",a/b) # division gives quotient.
    case 5:
        print("modulus",a%b) # Modules give reminder

    case _:
        print("get lost ediot, & prepare well: ")
```

wp to display food menu

```
food_meanu={"1.IDLY":35,"2.DHOSA":45,"3.UPPIT":40,"4.PALOW":55}
print(food_meanu)
option=int(input("enter your option: "))
quantity=int(input("enter your quantity: "))

match option:
    case 1:
        print("your ordered IDLY",quantity,"plates")
        print("total price Rs:",quantity*35)
    case 2:
        print("you ordered DHOSA",quantity,"plates")
        print("total price Rs:",quantity*45)
    case 3:
        print("you ordered UPPIT",quantity,"plates")
        print("total price Rs:",quantity*40)
    case 4:
        print("you ordered PALOW",quantity,"plates")
        print("total price Rs:",quantity*55)
    case _:
        print("no orders:",quantity=None)

print("thank you for order Visit again")
```

looping statements

this statements will be use when a certain task want's to do repeatedly for several nor of times.

there are 2 types.

1) for loop:

this statement iterates over the sequence of values perent in a collection time like(touple,string etc) and also over another iterating object.

syntax:

for vaarname in range() or collection type:

syntax:

```
range(par1, par2, par3)
    s,      e,      step
    o,      user defined, optional
    Dv=0,   run upto n-1, default value is 1
```

Ex:

```
for i in range(0,10,1):
    print(i,end=" ")
```

w p t print 1st 10 natural numbers

```
for i in range(1,11):
    print(i,end=" ")
1 2 3 4 5 6 7 8 9 10 #o/p
```

wpt print the sum of 1st 10 natural number

```
sum=0
for i in range(1,11):
    sum+=i
print(sum)
```

or

```
count=0
for i in range(1,11):
    count=count+i
print(count)
```

wpt print sum of 1st 10 even and odd

```
even=0
odd=0
for i in range(1,11):
    if i%2==0:
        even+=i
```

```
    else:
        odd+=i
print(even)
print(odd)
```

o/p

30

25

wpt print 1st 10 natural numbers in reverse.

```
for i in range(10,0,-1):
    print(i,end=" ")
```

o/p

10 9 8 7 6 5 4 3 2 1

```
print(" ")
```

```
for i in range(10,0,-2):
    print(i,end=" ")
```

o/p

10 9 8 7 6 5 4 3 2 1

10 8 6 4 2

for reverse mention -1 IMP.

NESTED FOR LOOP.

writing for loop inside another for loop.

syntax:

```
for var in range(-----):
    for var in range(-----):
        ststatement 1:
        ststatement 2:
```

Ex:

```
for row in range(3):
    for column in range(3):
        print(row,column)
```

0 0

0 1

0 2

1 0

1 1

1 2

2 0

2 1

2 2

CONTINUE(keyword)

it is KW in py used to skip the values.

continue KW can be used only inside the looping statements.

wpt skip values which is divisible by 3 while printing no from 1 to 10.

```
for i in range(1,10):
    if i%3==0:
        continue
    print(i,end=" ")
```

o/p

1 2 4 5 7 8

wpt skip values which is divisible by 5 while printing no from 1 to 50.

```
for i in range(1,20):
    if i%5==0:
        continue
    print(i,end=" ")
```

o/p

1 2 3 4 6 7 8 9 11 12 13 14 16 17 18 19

Break.

it is the KW used to break the iteration process when ever it is required.

Ex:

```
for i in range(1,20):
    if i==10:
        break
    print(i,end=" ")
```

WHILE Loop.

it is a while statement helps to do the task for several nor of times untile the condition iis reached.

here manually need to give the initialization, condition, updation.

syntax:

initialisation

while condition:

 stmt 1

 stmt 2

ITERABLE.

Iterable = something that contains multiple values and gives them one by one when we loop.

Examples of iterables in Python

All these are iterables:

Type	Example
String	"hello"
List	[10, 20, 30]
Tuple	(5, 6, 7)
Dictionary	{"a": 1, "b": 2}
Set	{3, 4, 5}
Range	range(5)

syntax:

for var in collection:

Ex:

```
s="flower"
for i in s:
    print(i,end=" ")
```

o/p

```
# f l o w e r
print(" ")
```

Ex:

```
for i in ('f','l','o','w','e','r'):
    print(i,end=" ")
```

o/p

```
# f , l , o , w , e , r
print(" ")
```

Ex:

```
for i in [1,2,3,4,5,6]:
    print(i,end=" ")
```

o/p

```
1 2 3 4 5 6
```

wpto check the entered integer is odd.

```
num=int(input("enter the number: "))
if num%3==0:
    print("num is odd: ",num)
else:
    print("num is even: ",num)
```

or.

```
num=int(input("enter the number: "))
if num%2!=0:
    print(f"{num} is oddd:")
else:
    print(f"{num} is even :")
```

consider 2 i/p & check whether both are pointing to same memory location

```
a=int(input("enter number a: "))
b=int(input("enter number b: "))
if a is b:
    print(" same location :")
else:
    print(" different location: ")
```

o/p

enter number a: 10
enter number b: 12
different location:

enter number a: 10
enter number b: 10
same location :

When to use for loop

Use for loop when you know exactly how many times you want to repeat something.

Example:

```
for i in range(5):
    print("Hi",i,end=" ")
```

o/p

Hi 0 Hi 1 Hi 2 Hi 3 Hi 4

Runs 5 times

The number of repetitions is known

When to use while loop

Use while loop when you don't know the exact number of times the loop should run, and it should stop only when a condition becomes false.

Example:

```
password = " "
while password != "1234":
    password = input("Enter password: ")
print("welcome")
```

using loops

```
password = input("Enter password: ")
if password == "1234":
    print("welcome")
else:
    print("wrong password")
```

wpt print n natural number using WHILE loop

```
n=int(input("enter n: "))
i=1 # initialization
while i<=n: # condition
    print(i,end=" ")
    i=i+1 # increment
```

o/p

```
enter n: 5
1 2 3 4 5
```

in while loop initialization 1st
condition check 2nd
increment 3rd

```
n = int(input("Enter a number: "))
i = 1
total = 0

while i <= n:
    print(i)
    total = total + i # adding i to total
    i = i + 1

print("Sum =", total)
```

o/p

```
Enter a number: 5
1
2
3
4
5
Sum = 15
```

wpt print 1st 10 natural even number using while loop

```
n=int(input("enter n: "))
```

```
i=1
```

```
while i<=n:
```

```
    if i%2==0:
```

```
        print(i)
```

```
        i=i+1
```

or.

```
i=2
```

```
while i<=10:
```

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

```
    i=i+1
```

o/p

2 3 4 5 6 7 8 9 10

```
print("ji ") last
```