

Boolean data type and expressions

```
In [1]: a=True
        type(a)

Out[1]: bool

In [5]: # relational operators
a=True
b=False
print(a and b)
print(a or b)
print(not(a))

False
True
False

In [7]: a=int(input())

# both conditionals will be true
if a%2==0 :
    print("even")
if a==6 :
    print("I am six")

6
even
I am six

In [13]: age=int(input())

if age<18 :
    print('child')
elif age>=18 and age<=40 :
    print('adult')
else :
    print('old')

67
old

In [15]: if True or True:
        if False and True or False:
            print('A')
        elif False and False or True and True:
            print('B')
        else:
            print('C')
    else:
        print('D')

B
```

Loops

while loop

```
In [22]: i=1
        n=int(input())
        while i<=n :
            print(i)
            i+=1

5
1
2
3
4
5
```

for loop

```
In [26]: str='abcdef'

for i in str :
    print(i)

a
b
c
d
e
f

In [31]: n=int(input())

# range -> (start, end, step)
# # [start, end)

for i in range(0, n, 2) :
    print(i)

8
0
2
4
6

In [36]: for i in range(5, 0, -1) :
        print(i)

5
4
3
2
1
```

Functions

```
In [3]: # factorial

n=int(input())
fact=1

for i in range(1, n+1, 1) :
    fact = fact * i

print(fact)

7
5040

But if we want to use this code factorial for multiple number of times, then we should write a function which uses the piece of code required amount of times

the piece of code below shows how we define a function in python

In [13]: def fact(a) :
        if a<=1 :
            return 1
        return a*fact(a-1)

x=int(input())
y=int(input())
# xCy
print(fact(x)//(fact(y)*fact(x-y)))

4
2
6

In [21]: # more is 'not' a keyword, it signifies a tuple of indefinite length
def sum(a, *more) :
    s=0
    s=s+a
    # print(type(more))
    for i in more :
        s=s+i
    return s
print(sum(1, 2, 3, 4, 5, 6, 7))

28

In [27]: def calc(a, b) :
        return a+b, a-b, a*b, a/b

print(calc(2, 5))
#print(type(calc(20, 5)))

(7, -3, 10, 0.4)

In [31]: # only a copy of data is passed, not the actual reference in the memory
def change(a) :
    a=45

a=34
change(a)
print(a)

34

In [33]: def power(x, y=2):
        ans = 1
        for i in range(y):
            ans = ans * x
        return ans
print(power(3), end=" ")
print(power(3,3))

9 27
```