Formatting output or formatting string
In python string formatting is done in 3 ways

1. Old style string formatting
2. New style string formatting
3. F-string

String formatting is used to format output.

Old-Style string formatting
Old style string formatting is also called c-style string formatting
In old-style string formatting the string contain character and replacement fields/formatting fields/formatting specifiers
Each formatting field is replace with value
Formatting string uses % operator to define replacement values.
String objects have one unique built-in operation: the % operator (modulo). This is also known as the string *formatting* or *interpolation* operator. Given format % values (where *format* is a string), % conversion specifications in *format* are replaced with zero or more elements of *values*.

Example:
```
# write a program to add two numbers
# first number :10
# second number :20
# output : sum of 10 and 20 is 30

n1=int(input("First number"))
n2=int(input("Second number"))
n3=n1+n2
print("sum of",n1,"and",n2,"is",n3)
print("sum of %d and %d is %d"%(n1,n2,n3))
```
Output:
```
First number10
Second number20
sum of 10 and 20 is 30
sum of 10 and 20 is 30
>>>
```

Example:
```
# write a program to find area of triangle

base=float(input("Enter base"))
```

```
height=float(input("Enter height"))
area=0.5*base*height
print("area of triangle with base=%.2f and height=%.2f is
%.2f"%(base,height,area))
```
Output:
```
Enter base1.5
Enter height2.0
area of triangle with base=1.50 and height=2.00 is 1.50
```

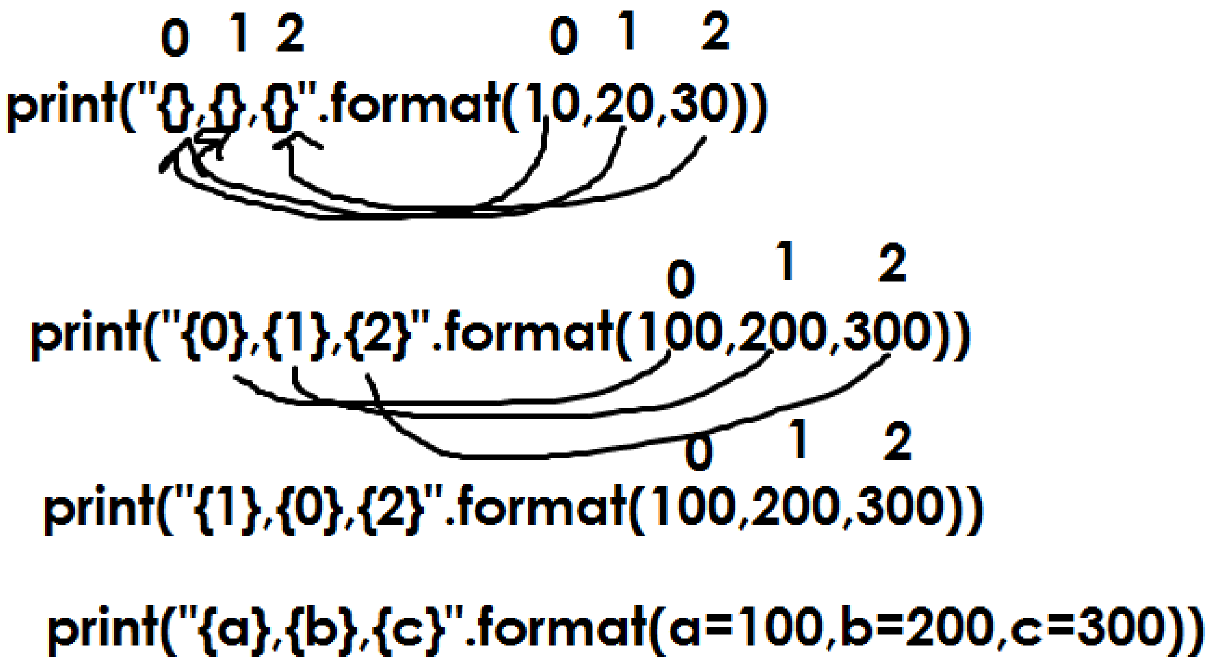
Formatting fields/characters/specifiers

%d	➔ decimal integer
%o	➔ octal integer
%x	➔ hexadecimal integer
%s	➔ string
%f	➔ float in fixed notation
%e	➔ float in exponent notation

Example:
```
a=65
b=0o45
c=0xab
print("a=%d,b=%o,c=%x"%(a,b,c))
print("%d,%o,%x"%(a,a,a))
```
Output:
```
a=65,b=45,c=ab
65,101,41
>>>
```

New style string formatting
new string string formatting is done using **format** method of string class or type.

The string contain formatting fields or replacement fields, formatting fields/replacement filed is represented using {}, this is identified with name or position of argument/values



```
print("{},{},{}".format(10,20,30))

print("{0},{1},{2}".format(100,200,300))

print("{1},{0},{2}".format(100,200,300))

print("{a},{b},{c}".format(a=100,b=200,c=300))
```

Example:

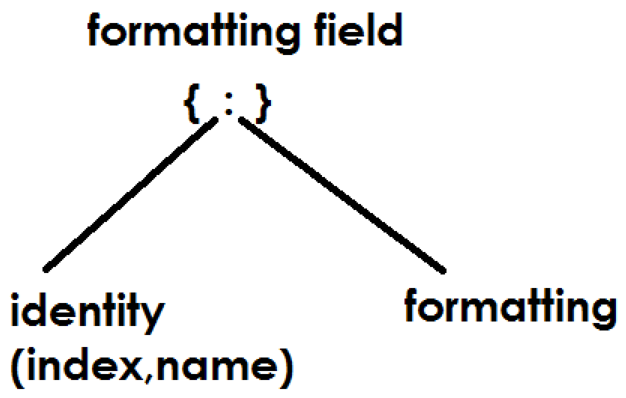
```
a=10
b=5
print("sum of {} and {} is {}".format(a,b,a+b))
print("{} is sum of {},{}".format(a+b,a,b))
print("sum of {0} and {1} is {2}".format(a,b,a+b))
print("sum of {x} and {y} is {z}".format(x=a,y=b,z=a+b))
```

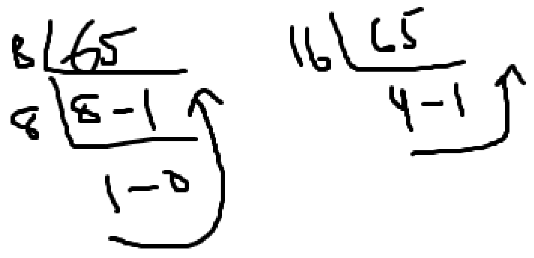
Output:

```
sum of 10 and 5 is 15
15 is sum of 10,5
sum of 10 and 5 is 15
sum of 10 and 5 is 15
>>>
```

Formatting characters

d ➔	decimal integer
o ➔	octal integer
x ➔	hexadecimal integer
b ➔	binary integer
f ➔	float in fixed
e ➔	float in expo
s ➔	string





```
print("{x:d},{y:o},{z:x}".format(x=65,y=65,z=65))
```

Example

```
print("{:d},{:o},{:x},{:b}".format(65,65,65,65))
print("{:f}".format(65))
```

Output

```
65,101,41,1000001
65.000000
>>>
```

Example:

```
# write a program to find result of a student
name=input("Enter name")
sub1=int(input("Enter subject1"))
sub2=int(input("Enter subject2"))
print("name={}\nsubject1={}\nsubject2={}\nresult={}".format(name,sub1,sub
2,"fail" if sub1<40 or sub2<40 else "pass"))
```

Output:

```
Enter namenaresh
Enter subject130
Enter subject220
name=naresh
subject1=30
subject2=20
result=fail
>>>
```