input and output statements

**Input()** :-

* **Read the data form key word is called dynamic data.**
* **Raw input() :- data is always treated as string type we required to use type casting function .**

Input()**:-**

* **input function not consider as a string what were type provide that kind data taken, no need type casting.**

**Raw input():-(“ enter some data “)**

* **In python -3 raw input function available only input function available.**
* **Python -2 and python -3 both are not same**
* **python -3 input function is raw input function python \_2 so that type casting not required.**

**Ex;-**

**x=int(input("enter no 4"))  
y=int(input("enter no 9"))  
print("sum",x+y)**

**output:-**

**enter no 1 4**

**enter no 9**

**sum 13**

**Ex1:-**

**Eno=input("enter emp no")  
ename=input("enter emp name")  
esal=float(input("enter emp sal"))  
eadd=input("enter emp address")  
married=bool(input("enter emp marries:[True|False"))  
print("emp no",eno)  
print("emp name",ename)  
print("emp sal",esal)  
print("emp add",eadd)**

**print("emp married",married)**

**output:-**

**enter emp no 4**

**enter emp name sekhar**

**enter emp address [lig \_141,ap hb colony ,arp]**

**enter emp sal 1546**

**emp no 4**

**name sekhar**

**emp sal 1546.0**

**Note=**

* **To the above example int ,float,bool,are the type conversation .**

Print()

**Ex:-**

**a,b,c=1,2,3  
print(a,b,c, sep=',')  
print(a,b,c,sep=':')  
print(a,b,c,sep='&')  
print(a,b,c,sep="$")**

**output:-**

**1,2,3**

**1:2:3**

**1&2&3**

**1$2$3**

* **In the above example by using separator function to separate the values with particular symbol**

**Ex:-**

**print("hello",end='\n')  
print("sekhar",end='\n')  
print("darling",end='\t')  
print("kavitha",end='\f')**

**output:-**

**hello**

**sekhar**

**darling kavitha**

**Form1:**

**syntax:**

print (string) or print()

* **To print the information without argument.**

**Ex:-**

**Print( )**

**Output: empty**

Form2:

**Syntax:**

**print(string)**

**Ex:-**

**>>> print(“hello sekhar”)**

**Output:-**

**hello sekhar**

**By using escape sequence character :-**

**Ex:-**

**>>> print("hello\n welcome\n python class")**

**Output:-**

**hello**

**welcome**

**python class**

We can use repeat ion operator \***:-**

**Ex:-**

**>>> print("hello"\*5)**

**Hellohellohellohellohello**

We can use plus (+) operator (+)**:-**

**Ex:**

**print("hari"+"nath"+"reddy")**

**output:**

**harinathreddy**

**From3:**

* **Print with variable no of argument.**

**Ex:-**

**>>a,b,c=12,34,35  
>>print(“variable: “,a,b,c)**

**Output:  
variable:12,34,35**

**Ex1:  
a,b,c=10,20,30  
print(a,b,c,sep=',')  
print(a,b,c,sep=':')**

**Output:10,20,30**

* **By default output values are separated by space if we want to specify separate operator by using “sep “ attribute.**

**Form4:**

* **Print with end attribute .**
* **Default end attribute is new line.**

**Ex:-**

**print("hello",end='')  
print("python",end='')  
print("class",end='')**

**output:**

**hellopythonclass**

**Form5:**

* **We can pass any objects like list, tuple, set.**

**etc. as argument to the print statement.**

**Ex:-**

**>>l=[10,20,30]  
>>t=(1,2,3)  
>>s={1,2,3}  
>>print(l)  
[10,20,30]  
>>print(t)  
print(1,2,3)  
>>print(s)**

**{1,2,3}**

**Form6 :**

**Syntax:**

**print (string ,variable list)**

* **We can supply string and any number for the argument .**

**Ex:**

**>>s="sekhar"  
>>d="2"  
>>s1="python"  
>>s2="loverboy"  
>>e=40  
>>f=30  
>>print("hello",s,"yourage is",d)  
>>print("your teaching",s1,"and",s2)  
>>print(s1 marks:"e,"s2 marks:"f)**

**output:**

**hello sekhar your age is 2**

**your teaching python and c language**

**python marks 40,c language 30**

**Form7 :**

**syntax:**

print(formatted string “%(variable list)

* **% i=int**
* **%d=int**
* **%f=float**
* **%s =string**

**Ex:-**

**>>a=10  
>>b=12.34  
>>c="sekhar"  
>>print("a value is :%d"%9)  
>>print("b value is:%f"%b)  
>>print("name is:%s"%c)**

**output:**

**a value is :9**

**b value is:12.340000**

**name is: sekhar**

**Ex:-**

**>>c="milky"  
>>list=[10,20,30,40]  
>>print("hello %s the list of items are %s"% (s, list))**

**output:-hello milky the list of item are [10,20,30,40]**

Form8:

* **By using replacement operator {}**
* **put in empty x at the time of printing replacement operator .**

**Ex:**

**>>name="sekhar"  
>>sal="20000"  
>>bb="prtti"  
>>print("hello {0} your salary friend {2}is waiting ().formate c name,sal,bb)  
>>print("hello {} your salary is waiting ".formate (name,sal,bb)  
>>print("hello{x}your salary is {y} and your friend {3}is waiting"format x=name y=sal,z=bb)**

**output:**

**hello sekhar your salary is 20000 and your friend prtti is waiting hello sekhar your salary is 20000 and your friend pretti is waiting hello sekhar your friend salary is 20000 and your friend prtti is waiting**