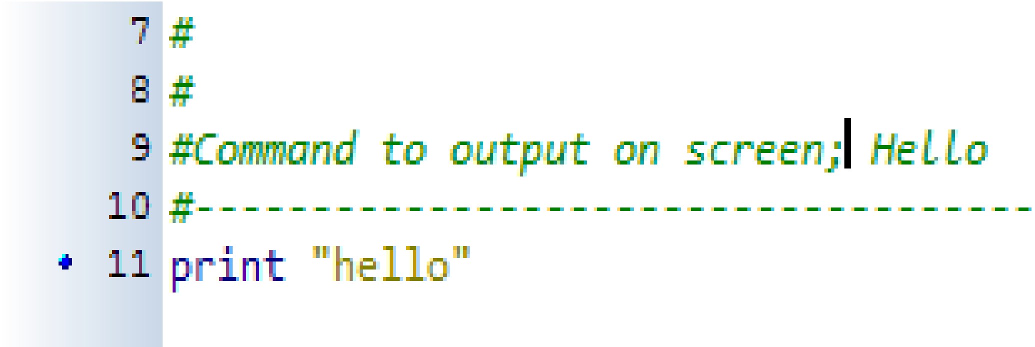
Python

Input / Output

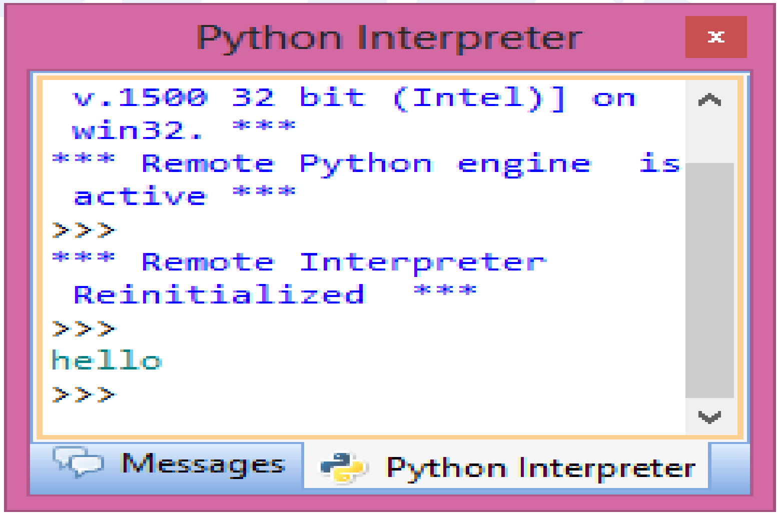
print Statement

Command:



```
#
#
#Command to output on screen; Hello
#-------------------------------------------------
print "hello"
```

Output:



```
 v.1500 32 bit (Intel)] on
 win32. ***
*** Remote Python engine  is
 active ***
>>>
*** Remote Interpreter
 Reinitialized  ***
>>>
hello
>>>
```

print Statement ; Features

- It auto converts the item into string for the object may be a integer or a float.
- It inserts spaces between items automatically.
- When a print statement ends with a comma, it continues the other item given (next print statement) in same line.

print Statement Examples

```
>>> a="a"
>>> b="b"
>>> a,b
('a', 'b')
>>> print (a,b)
a b
```

```
INPUT:
print (“Hello”),
Print (“How are you”)

OUTPUT:
hello how are you
```

print Statement ; Features

- It auto converts the item into string for the object may be a integer or a float.

- It inserts spaces between items automatically.

- When a print statement ends with a comma, it continues the other item given (next print statement) in same line.

Print - Formatting

- Modulo (%) is separator of format and parameters
- a= 234
- print(“%5d” % (a))

Print Continued..

```
# print integer and float value
print("Geeks : % 2d, Portal : % 5.2f" %(1, 05.333))

# print integer value
print("Total students : % 3d, Boys : % 2d" %(240, 120))

# print octal value
print("% 7.3o"% (25))

# print exponential value
print("% 10.3E"% (356.08977))
```

Print - Syntax

print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)

- Here, objects is the value(s) to be printed.
- The sep separator is used between the values. It defaults into a space character.
- After all values are printed, end is printed. It defaults into a new line.
- The file is the object where the values are printed and its default value is sys.stdout (screen).
- Print output goes to buffer and then sent to device mentioned in file. Flush is used here control this behavior.

Print - Example

```
print(1, 2, 3, 4)
print(1, 2, 3, 4, sep='*')
print(1, 2, 3, 4, sep='#', end='&')
```

Outputs:

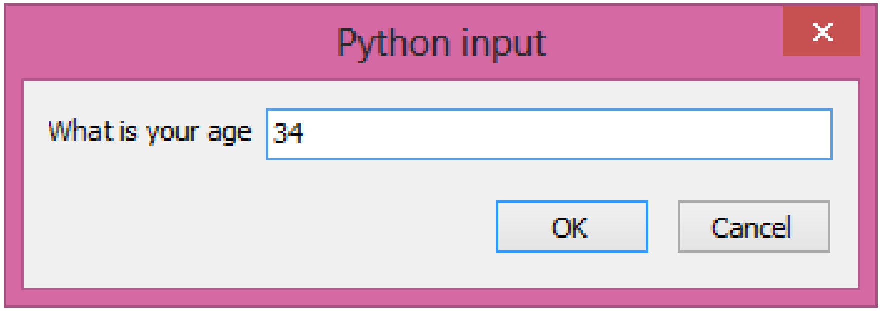
```
1 2 3 4
1*2*3*4
1#2#3#4&
```

Input

input([prompt])

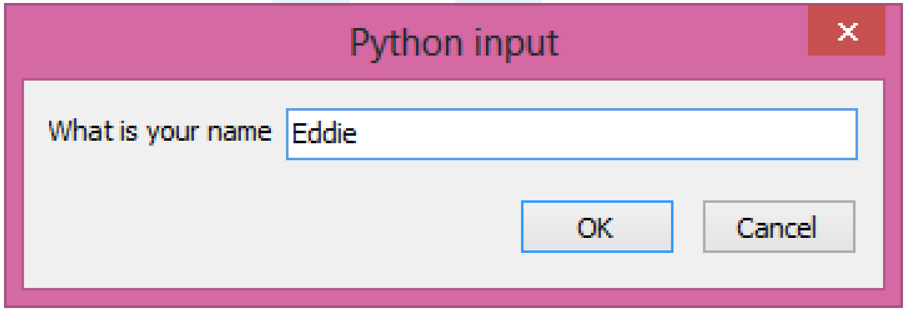
- [prompt] is string displayed to user on the screen
- This returns values in string

```
>>> name = input(“What is your name”)
>>> age = input(“What is your age”)
>>> print name, age
Eddie, 34
```



Python input

What is your age 34



Python input

What is your name Eddie