

# Python

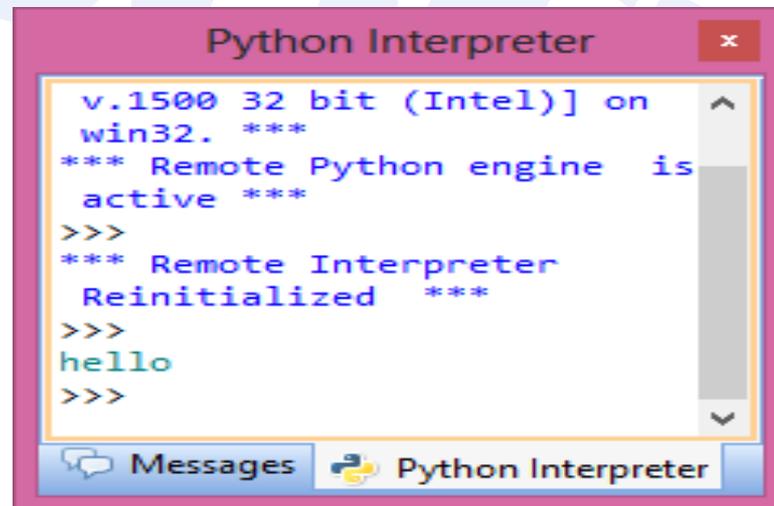
Input / Output

# print Statement

Command:

```
7 #  
8 #  
9 #Command to output on screen;| Hello  
10 #-----  
• 11 print "hello"
```

*Output:*

A screenshot of a Python Interpreter window. The window title is "Python Interpreter" with a close button (X) in the top right corner. The main area contains the following text:

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

At the bottom of the window, there is a status bar with two buttons: "Messages" and "Python Interpreter".

# 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
```

A screenshot of a Python input dialog box. The title bar is pink and contains the text "Python input" and a close button (X). The main area is light gray and contains the text "What is your age" followed by a text input field containing the number "34". Below the input field are two buttons: "OK" and "Cancel".A screenshot of a Python input dialog box. The title bar is pink and contains the text "Python input" and a close button (X). The main area is light gray and contains the text "What is your name" followed by a text input field containing the name "Eddie". Below the input field are two buttons: "OK" and "Cancel".