



Why Python?

Programmer continue to focus on Python for many reasons, for example:

1. Python is easy to learn.
2. Python is an incredibly efficient language: your programs will do more in fewer lines of code than many other languages would requires.
3. Python's syntax will also help you write "clean" code: your code will be easy to read, easy to debug, and easy to extend.

Python are used for many purpose:

1. Developing games.
2. Build web applications.
3. Solve business problems.
4. Python is heavily used in scientific fields for academic research and applied work.

Typing in Python?

- Python is case sensitive: to python print, Print and PRINT are different command.
- Spaces matter at the beginning of lines, but not elsewhere. Python uses indentation of lines.

Printing Function:

The print function requires parenthesis around its arguments. In the following program, its only argument is the string 'Hi there'. Anything inside quotes will be printed exactly as it appears.

```
print('Hi there')
```

Moreover, in the following program, the first statement will output 3+4, while the second will output 7.

```
print('3+4')  
print(3+4)
```

To print several things at once, separate them by commas. Python will automatically insert spaces between them. Below is an example and the output is produces.

```
print('The value of 3+4 is', 3+4)  
print('A', 1, 'XYZ', 2)
```

```
The value of 3+4 is 7  
A 1 XYZ 2
```

Optional print arguments:



There are two optional arguments to the print function. They are very useful for making your output look organized.

1. **sep:** python will insert a space between each of the arguments of the print function. There is an optional argument called “sep”, short for separator that you can use to change the space to some thing else. For example using sep=':' would separate the arguments by a colon and sep='##' would separate the arguments by two pound sings. Here is an example of using sep to make the output look organized.

```
print ('The value of 3+4 is', 3+4, '.')  
print ('The value of 3+4 is ', 3+4, '.', sep='')
```

```
The value of 3+4 is 7 .  
The value of 3+4 is 7.
```

2. **end:** the print function will automatically advance to the next line. For instance, the following will print two lines:

```
print('On the first line')  
print('On the second line')
```

```
On the first line  
On the second line
```

There is an optional argument called end that you can use to keep the print function from advancing to the next line. Here is an example:

```
print('On the first line', end='')  
print('On the second line')
```

```
On the first lineOn the second line
```

Getting input:

The input *function* is a simple way for your program to get information from people using your program. Here is an example.

```
name = input('Enter your name: ')  
print('Hello, ', name)
```

The basic structure is

Variable name = input (message to user)



The above works for getting text from the user. To get numbers from the user to use in calculations, we need to do something extra. Here is an example:

```
num = eval(input('Enter a number: '))
print('Your number squared:', num*num)
```

The *eval* functions converts text entered by the user into a number. One nice feature of this is you can enter expression, like $3*12+5$, and *eval* will compute the for you.

Variables:

Variable is used to remember a value from one part of a program that is can be used in another part of the program. Variable can also be used to stores the value that the user enters so that we can do a calculation with it in the program.

```
temp = eval(input('Enter a temperature in Celsius: '))
print('In Fahrenheit, that is', 9/5*temp+32)
```

In the example below, we perform a calculation and need to use the result of the calculation in several places in the program. If we save the result of the calculation in a variable, then we only need to do the calculation once. This also helps to make the program more readable.

```
temp = eval(input('Enter a temperature in Celsius: '))
f_temp = 9/5*temp+32
print('In Fahrenheit, that is', f_temp)
if f_temp > 212:
    print('That temperature is above the boiling point.')
if f_temp < 32:
    print('That temperature is below the freezing point.')
```

Read the code below and try to figure out what the values of x and y will be after the code is executed.

```
x=3
y=4
z=x+y
z=z+1
x=y
y=5
```

After these four lines of code are executed, the value of x is 4, y is 5 and z is 8. One way to understand something like this is to take it one line at a time. This is an especially useful technique



for trying to understand more complicated chunks of code. Here is a description of what happens in the code above.

1. x starts with the value 3 and y starts with the value 4.
2. In line 3, a variable z is created to equal x+y, which is 7.
3. Then the value of z is changed to equal one more than it currently equals, changing it from 7 to 8.
4. Next, x is changed to the current value of y, which is 4.
5. Finally, y is changed to 5. Note that this does not affect x.
6. So at the end, x is 4, y is 5, and z is 8.

Variable names:

There are just a couple of rules to follow when naming your variables.

- Variable names can contain letters, numbers, and the underscore.
- Variable names cannot contain spaces.
- Variable names cannot start with number.
- Case sensitive. For instance, temp and Temp are different variables.

Exercises:

1. **Print a box like the one below**

```
*****  
*****  
*****  
*****
```

2. **Print a box like the one below.**

```
*****  
*                                     *  
*                                     *  
*****
```

3. **Print a triangle like the one below.**

```
*  
**  
***  
****
```

4. **Write a program that computes and prints the result of $512-282/47*48+5$.**
5. **Ask the user to enter a number. Print out the square of the number, then use sep optional argument to print it out in full sentence that ends in a period. Sample output is shown below.**



```
Enter a number: 5  
The square of 5 is 25.
```

6. Ask the user to enter a number x , use the sep optional argument to print out x , $2x$, $3x$, $4x$, and $5x$, each separated by three dashes, like below.

```
Enter a number: 7  
7---14---21---28---35
```

7. Write a program that asks the user for a weight in kilograms and converts it to pounds. There are 2.2 pounds in a kilogram.
8. Write a program that asks the user to enter three numbers using three separate input statements. Then create variables called total and average that hold the sum and average of the three numbers and print out the values of total and average.
9. A lot of cell phones have tip calculators. Write one, ask the user for the price of the meal and the percent tip they want to leave. Then print both the tip amount and the total bill with the tip included.