

GETTING STARTED: PROGRAM INPUT AND THE RAW_INPUT () BUILT IN FUNCTION

The easiest way to obtain user input from the command line is with the raw input () built-in function. It reads from standard input and assigns the string value to the variable you designate. You can use the int() built-in function to convert any numeric input string to an integer representation.

```
>>> user = raw_  
input('Enter login name: ') Enter login name: root  
>>> print 'Your login is:', user  
Your login is: root
```

The earlier example was strictly for text input. A numeric string input (with conversion to a real integer) example follows below:

```
>>> num = raw_  
input('Now enter a number: ') Now enter a number: 1024  
>>> print 'Doubling your number: %d' % (int(num) * 2)  
Doubling your number: 2048
```

The int() function converts the string num to an integer so that the mathematical operation can be performed.

COMMENTS

Comments As with most scripting and Unix-shell languages, the hash or pound (#) sign signals that a comment begins from the # and continues until the end of the line.

```
>>> # one comment ...
```

```
print 'Hello World!' # another comment
```

Hello World!

There are special comments called documentation strings, or "doc strings" for short. You can add a "comment" at the beginning of a module, class, or function string that serves as a doc string, a feature familiar to Java programmers:

```
def foo():  
    "This is a doc string."  
    return
```

True Unlike regular comments, however, doc strings can be accessed at runtime and be used to automatically generate documentation.