

Introduction to Programming in Python

Built-in Functions and Libraries

Dr. Bill Young
Department of Computer Science
University of Texas at Austin

Last updated: June 4, 2021 at 11:04

Python provides many built-in functions that are directly available for processing items of the many Python data types.

In addition, there are many libraries of functions that you can use, but only if you *import* the library.

- os** interact with the operating system (e.g., change directory)
- sys** interact with the system (command-line args)
- math** access special math functions such as `log()`, `sin()`, `sqrt()`, `pi`
- random** random number generation
- datetime** clock and calendar functions

To use the functions from a library you have to import it.

Import Examples

```
>>> import os                # import module os
>>> os.name                  # what's my OS?
'posix'
>>> os.getcwd()              # get current working directory
'/u/byoung/cs303e/slides'
>>> import random            # import module random
>>> random.random()          # generate random value
0.36552104405513963          # between 0 and 1
>>> random.random()          # do it again
0.7465680663361102
>>> import math              # import module math
>>> math.sqrt( 1000 )         # square root of 1000
31.622776601683793
>>> math.pi                  # approximation of pi
3.141592653589793
import datetime              # import module datetime
>>> print(datetime.datetime.now()) # current time
2020-07-30 14:45:42.905190
```

Some Functions in the math Library

<code>floor(x)</code>	returns the largest integer no bigger than x
<code>ceil(x)</code>	returns the smallest integer no less than x
<code>exp(x)</code>	exponential function e^x
<code>log(x)</code>	natural logarithm (log to the base e of x)
<code>log(x, b)</code>	log to the base b of x
<code>sqrt(x)</code>	square root of x

Trigonometric functions, including:

<code>sin(x)</code>	sine of x
<code>asin(x)</code>	arcsine (inverse sine) of x
<code>degrees(x)</code>	convert angle x from radians to degrees
<code>radians(x)</code>	convert angle x from degrees to radians

To use a function from a library, import the library and then call the function using the module (library) name:

```
>>> import math
>>> math.floor( 3.2 )
3
>>> math.ceil( 3.2 )
4
>>> math.exp( 2 )           # e ** 2
7.38905609893065
>>> math.log( 7.389 )       # log base e
1.9999924078065106
>>> math.log( 1024, 2 )     # log base 2
10.0
>>> math.sqrt( 1024 )
32.0
>>> math.sin( math.pi )
1.2246467991473532e-16
>>> math.sin( 90 )
0.89399666636005579
>>> math.degrees( math.pi ) # pi radians is 180 deg.
180.0
>>> math.radians( 180 )     # 180 deg. is pi radians
3.141592653589793
```

Each of the Python types has an associated collection of *built-in* functions.

For example, *strings* have all of the following functions.

Function	Description
<code>x in s</code>	char x is in string s
<code>x not in s</code>	char x is not in string s
<code>s1 + s2</code>	concatenates two strings
<code>s * n</code>	repeat string s n times
<code>s[i]</code>	ith element of string (0-based)
<code>s[i:j]</code>	slice of string s from i to j-1
<code>len(s)</code>	number of elements in s
<code>min(s)</code>	return char in string with lowest ASCII value
<code>max(s)</code>	return char in string with highest ASCII value
<code>for loop</code>	traverse elements of string
<code><, <=, >, >=</code>	compares two strings
<code>==, !=</code>	compares two strings

We'll cover strings later in some detail.

Functions on Strings

```
>>> s1 = 'Hello,'
>>> s2 = ' World!'
>>> s3 = s1 + s2
>>> s3
'Hello, World!'
>>>
>>> 'a' in s3
False
>>> 'W' in s3
True
>>> s1*3
'Hello,Hello,Hello,'
>>> len( s3 )
13
>>> min( s3 )
','
>>> max( s3 )
'r'
>>> s1 < s2
False
>>> s2 < s3
True
>>>
```

Exercise: Generate Random Password

Suppose we'd like to write a program to generate a random password of length 6.

```
import random

def main():
    # Choose random characters from these characters
    uppers = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    lowers = "abcdefghijklmnopqrstuvwxyz"
    others = "0123456789!@#%&'()*_+={ } | <>?/ "
    choices = uppers + lowers + others
    numChoices = len( choices )
    char1 = choices[ random.randrange( 0, numChoices ) ]
    char2 = choices[ random.randrange( 0, numChoices ) ]
    char3 = choices[ random.randrange( 0, numChoices ) ]
    char4 = choices[ random.randrange( 0, numChoices ) ]
    char5 = choices[ random.randrange( 0, numChoices ) ]
    char6 = choices[ random.randrange( 0, numChoices ) ]
    password = char1 + char2 + char3 + char4 + char5 + char6
    print( "Password is:", password )

main()
```

```
> python Password.py
Password is: |EE0tj
> python Password.py
Password is: ww1|h5
> python Password.py
Password is: 7Cfg&?
> python Password.py
Password is: /b@37F
> python Password.py
Password is: lWgD?0
> python Password.py
Password is: *8u_+6
```

Can you think of ways to improve the passwords generated?