

Python Built-in Functions

abs()

abs() is one of the most popular Python built-in functions, which returns the absolute value of a number. A negative value's absolute is that value is positive.

```
>>> abs(-7)
7
```

all()

all() function takes a container as an argument. This Built in Functions returns True if all values in a python iterable have a Boolean value of True. An empty value has a Boolean value of False.

```
>>> all(['*', '!', '"'])
False
```

any()

all(), it takes one argument and returns True if, even one value in the iterable has a Boolean value of True.

```
>>> any((1,0,0))
True
```

ascii()

ascii(), is important Python built-in functions, returns a printable representation of a [python object](#) (like a string or a [Python list](#)). Let's take a Romanian character.

```
>>> ascii('ș')
'"\\u0219"'
```

bin()

bin() converts an integer to a binary string. We have seen this and other functions in our article on [Python Numbers](#).

```
1. >>> bin(7)
'0b111'
```

bool()

bool() converts a value to Boolean.

```
1. >>> bool(0.5)
```

True

chr()

chr() Built In function returns the character in python for an ASCII value.

```
>>> chr(65)
```

'A'

compile()

compile() returns a Python code object. We use Python in built function to convert a string code into object code.

```
1. >>> exec(compile('a=5\nb=7\nprint(a+b)',",','exec'))
```

Here, 'exec' is the mode. The parameter before that is the filename for the file from which the code is read.

Finally, we execute it using exec().

complex()

Python complex() function creates a complex number.

```
>>> complex(3)
```

(3+0j)

divmod()

Takes two numbers and gives the quotient and remainder

```
>>> divmod(7,3)
```

(2,1)

eval()

This Function takes a string as an argument, which is parsed as an expression.

```
1. >>> x=7
```

```
2. >>> eval('x+7')
```

14

exec()

exec() runs Python code dynamically.

```
1. >>> exec('a=2;b=3;print(a+b)')  
5
```

float()

This Python Built IN function converts an int or a compatible value into a float.

```
>>> float(2)  
2.0
```

format()

used to format string/numbers

```
>>>format("hello world",".5s")  
'hello'
```

help()

To get details about any module, keyword, symbol, or topic, we use the help() function.

```
1. >>> help()
```

hex()

Hex() Python built-in functions, converts an integer to hexadecimal.

```
>>> hex(16)  
'0x10'
```

id()

Python id() returns an object's identity.

```
>>> id(orange)  
100218832
```

input()

Input() Python built-in functions, reads and returns a line of string.

```
1. >>> input("Enter a number")  
Enter a number7  
'7'
```

Note that this returns the input as a string. If we want to take 7 as an integer, we need to apply the int() function to it.

int()

Python int() converts a value to an integer.

```
1. >>> int('7')  
7
```

len()

It returns the length of an object.

```
1. >>> len({1,2,3})  
3
```

max()

Python max() returns the item, in a sequence, with the highest value of all.

```
1. >>> max(2,3,4)  
4
```

min()

Python min() returns the lowest value in a sequence.

```
1. >>> min(3,5,1)  
1
```

ord()

ord() returns an integer that represents the Unicode point for a given Unicode character. This is complementary to chr().

```
1. >>> ord('A')  
65
```

pow()

Python pow() takes two arguments- say, x and y. It then returns the value of x to the power of y.

```
1. >>> pow(3,4)  
81
```

round()

Python round() rounds off a float to the given number of digits (given by the second argument).

```
1. >>> round(3.777,2)
3.78
```

reversed()

Reverses the content of iterable

```
>>> reversed([1,2,3])
[3,2,1]
```

sorted()

Prints the sorted version of iterable

```
>>> sorted([5,9,1])
[1,5,9]
```

```
>>> sorted("Python")
['P', 'h', 'n', 'o', 't', 'y']
```

Note: character sorting is done based on ASCII value)

sum()

The function sum() takes an iterable as an argument, and returns the sum of all values.

```
1. >>> sum([3,4,5])
12
```

type()

type() function is used to check the type of object we're dealing with

```
type(5.15)
<class 'float'>
```

zip()

Python zip() returns us an iterator of tuples.

```
1. >>> list(zip([1,2,3],['a','b','c']))
[(1, 'a'), (2, 'b'), (3, 'c')]
```

Inbuilt mathematical Functions in Python(import math)

Function	Example	Description
ceil(X)	>>> math.ceil(10.23) 11	Round X to nearest integer and returns that integer.
floor(X)	>>> math.floor(18.9) 18	Returns the largest value not greater than X
exp(X)	>>> math.exp(1) 2.718281828459045	Returns the exponential value for e^x
log(X)	>>> math.log(2.71828) 0.999999327347282	Returns the natural logarithmic of x (to base e)
log(x,base)	>>> math.log(8,2) 3.0	Returns the logarithmic of x to the given base
sqrt(X)	>>>math.sqrt(9) 3.0	Return the square root of x
Sin(X)	>>> math.sin(3.14159/2) 0.9999999999991198	Return the sin of X, where X is the value in radians
asin(X)	>>> math.asin(1) 1.5707963267948966	Return the angle in radians for the inverse of sine
cos(X)	>>> math.cos(0) 1.0	Return the sin of X, where X is the value in radians
aCos(X)	>>> math.acos(1) 0.0	Return the angle in radians for the inverse of cosine
tan(X)	>>> math.tan(3.14/4) 0.9992039901050427	Return the tangent of X, where X is the value in radians
degrees(X)	>>> math.degrees(1.57) 89.95437383553924	Convert angle X from to radians to degrees
Radians(X)	>>> math. radians(89.99999) 1.5707961522619713	Convert angle x from degrees to radians