

# Built-in Python functions

In the previous article, you learned that Python has a basic function type, one is a built-in function, two are user-defined functions. Today, we will learn about a list of Python's built-in functions.

In the previous article, you learned that Python has a basic function type, one is a built-in function, two are user-defined functions. Today, we will learn about a list of Python's built-in functions.

Python interpreter has a number of functions available to use. These functions are called integrated functions. For example, `print()` is a function that prints the values provided to the function to the screen, the `list()` function creates a list in Python,

In Python 3.6 there are 68 Python functions built-in. Below is a list of those functions, along with a brief description of them. You can bookmark again to check it quickly when needed.

Press `Ctrl + F` on the browser and enter the name of the function you want to find, if you want to find it quickly.

Description function  
`abs()` Returns the absolute value of some  
`all()` Returns true when all iterable elements are true  
`any()` Tests any element of iterable as True  
`ascii()` Returns string contains a representation that can be printed in bin  
`bin()` Convert integers to a binary string  
`bool()` Returns a value to Boolean  
`bytearray()` Returns the byte size array provided by bytes  
`bytes()` Returns a constant byte object  
`callable()` Check if the object can call  
`chr()` Returns a character (a string) from Integer  
`classmethod()` Returns a method class for the compile  
`compile()` function Returns the Python code object  
`complex()` Create complex number  
`delattr()` Delete property from the dict  
`dict()` object Create Dictionary  
`dir()` Returns the property of the  
`divmod()` object Returns a Tuple of the Quotient and Remainder  
`enumerate()` Returns v object declared  
`eval()` Run Python code in  
`exec()` Execute the program created dynamically  
`filter()` Build iterator from True elements  
`float()` Returns decimal from number, format string  
`format()` Returns about representation formatted of the value  
`frozenset()` Returns the object frozenset does not change  
`getattr()` Returns the named attribute value of the globals object  
`globals()` Returns the dictionary of the current global symbol table  
`hasattr()` Returns the object even if the property is named or not  
`hash()` Returns the hash value of the help object  
`help()` Calls the Help System with  
`hex()` converts Integer to Hexadecimal  
`id()` Returns the identifier of the object  
`input()` icon Read and return the string in an int  
`line()` Returns an integer from the number or  
`isinstance()` string Check if the object There is an Instance of Class  
`not issubclass()` Check if the object is a Subclass of Class  
`not iter()` Returns the iterator for the len object  
`len()` Returns the length of the list object  
`list()` Create list in Python  
`locals()` Returns the dictionary of the current local symbol table  
`map()` Applies the function and returns a max list  
`max()` Returns the largest element  
`memoryview()` Returns the memory view of the argument  
`min()` Returns the element smallest  
`next()` Extract the next element from Iterator object  
`object()` Create an object that does not have (Featureless Object)  
`oct()` Convert integer to octal  
`open()` Returns object File  
`ord()` Returns Unicode code for pow Unicode characters  
`pow()` Returns x exponent y  
`print()` Print the provided object  
`property()` Returns property  
`range()` property Returns the integer string from the start number to the ending number  
`ramming repr()` Returns the printable representation of the reversed object  
`reversed()` Returns the reverse iterator of a round  
`round()` Rounds the set decimal  
`set()` Creates a set of new elements  
`setattr()` Set a value for a attribute of slice

object () Cut the specified object by range () sorted () Returns the sorted list staticmethod () Create static method from a function str () Returns an unofficial representation of an object sum () Add an entry to Iterable super () Allow reference to Parent Class by super tuple () Function Create a Tuple type () Return object type vars () Returns \_\_dict\_\_ attribute of class zip () Returns Tuple \_\_import \_\_ () iterator The advanced function, called by import

In the future, I will try to write a detailed article about these functions so that you understand the syntax, as well as the usage of each function.

If you want to know what this particular function does, there's an argument, you just need to enter the command:

```
print(ten_ham.__doc__)
```

Python will explain quite well about the function, you can read and do some examples to understand that function.

Don't forget to do Python exercises!

Next article: Python function is defined by the user

Last lesson: Python functions

You finished reading the article "**Built-in Python functions**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.