

When the above code is executed, it produces the following result:

```
['__doc__', '__file__', '__name__', 'acos', 'asin', 'atan',  
'atan2', 'ceil', 'cos', 'cosh', 'degrees', 'e', 'exp',  
'fabs', 'floor', 'fmod', 'frexp', 'hypot', 'ldexp', 'log',  
'log10', 'modf', 'pi', 'pow', 'radians', 'sin', 'sinh',  
'sqrt', 'tan', 'tanh']
```

Here, the special string variable `__name__` is the module's name, and `__file__` is the filename from which the module was loaded.

The *globals()* and *locals()* Functions

The *globals()* and *locals()* functions can be used to return the names in the global and local namespaces depending on the location from where they are called.

- If *locals()* is called from within a function, it will return all the names that can be accessed locally from that function.
- If *globals()* is called from within a function, it will return all the names that can be accessed globally from that function.

The return type of both these functions is dictionary. Therefore, names can be extracted using the *keys()* function.

The *reload()* Function

When the module is imported into a script, the code in the top-level portion of a module is executed only once.

Therefore, if you want to reexecute the top-level code in a module, you can use the *reload()* function. The *reload()* function imports a previously imported module again. The syntax of the *reload()* function is this:

```
reload(module_name)
```

Here, *module_name* is the name of the module you want to reload and not the string containing the module name. For example, to reload *hello* module, do the following:

```
reload(hello)
```