The Python Dir

We've already seen 'dir' function. Let's look at it in more detail

What is dir? Is it something you say when someone says or does something stupid? Not in this context! No, here on Planet Python, the dir keyword is used to tell the programmer what attributes and methods there are in the passed in object. If you forget to pass in an object, dir will return a list of names in the current scope. As usual, this is easier to understand with a few examples.

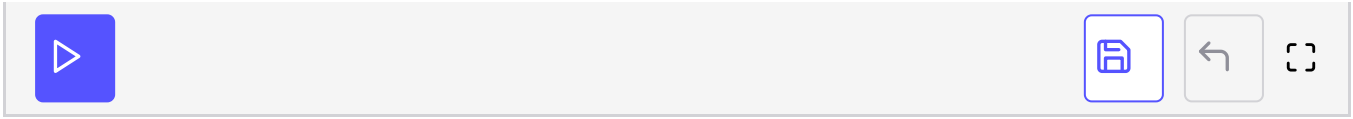
```
x = "test"
y = 7
z = None

print(dir("test"))
# ['__add__', '__class__', '__contains__', '__delattr__',
# '__doc__', '__eq__', '__ge__', '__getattribute__',
# '__getitem__', '__getnewargs__', '__getslice__', '__gt__',
# '__hash__', '__init__', '__le__', '__len__', '__lt__',
# '__mod__', '__mul__', '__ne__', '__new__', '__reduce__',
# '__reduce_ex__', '__repr__', '__rmod__', '__rmul__',
# '__setattr__', '__str__', 'capitalize', 'center',
# 'count', 'decode', 'encode', 'endswith', 'expandtabs',
# 'find', 'index', 'isalnum', 'isalpha', 'isdigit', 'islower',
# 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower',
# 'lstrip', 'replace', 'rfind', 'rindex', 'rjust', 'rsplit',
# 'rstrip', 'split', 'splitlines', 'startswith', 'strip',
# 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

Since everything in Python is an object, we can pass a string to dir and find out what methods it has. Pretty neat, huh? Now let's try it with an imported module:

```
import sys
print(dir(sys))
#  ['__displayhook__', '__doc__', '__egginsert', '__excepthook__',
# '__name__', '__plen', '__stderr__', '__stdin__', '__stdout__',
# '_getframe', 'api_version', 'argv', 'builtin_module_names',
# 'byteorder', 'call_tracing', 'callstats', 'copyright'
```

```
# 'displayhook', 'dllhandle', 'exc_clear', 'exc_info',
# 'exc_traceback', 'exc_type', 'exc_value', 'excepthook',
# 'exec_prefix', 'executable', 'exit', 'exitfunc',
# 'getcheckinterval', 'getdefaultencoding', 'getfilesystemencoding',
# 'getrecursionlimit', 'getrefcount', 'getwindowsversion', 'hexversion',
# 'maxint', 'maxunicode', 'meta_path', 'modules', 'path', 'path_hooks',
# 'path_importer_cache', 'platform', 'prefix', 'setcheckinterval',
# 'setprofile', 'setrecursionlimit', 'settrace', 'stderr', 'stdin',
# 'stdout', 'version', 'version_info', 'warnoptions', 'winver']
```



Now, that's handy! If you haven't figured it out yet, the **dir** function is extremely handy for those 3rd party packages that you have downloaded (or will soon download) that have little to no documentation. How do you find out about what methods are available in these cases? Well, **dir** will help you figure it out. Of course, sometimes the documentation is in the code itself, which brings us to the builtin help utility.