Section Overview

What You Will Learn

- Modules
- More built-in Python functions
- Module search path
- Python Standard Library

Modules

Modules

- Python modules are files that have a `.py` extension.
- They can implement a set of attributes (variables), methods (functions), and classes (types).

Modules

A module can be included in another Python program by using the `import` statement followed by the module name.

```
import time
time.method_name()
time.attribute_name
```

```
import time
print(time.asctime())
print(time.timezone)
```

```
Mon Nov 16 13:23:40 2015
21600
```

om

Modules

```
import module_name
module_name.method_name()

from module_name import method_name
method_name()
```

```
from time import asctime
print(asctime())
```

```
Mon Nov 16 13:23:40 2015
```

om

Modules

```
from module_name import method_name


from module_name import method_name1, method_nameN
from time import asctime, sleep
print(asctime())
sleep(3)
print(asctime())
```

```
Mon Nov 16 13:23:43 2015
Mon Nov 16 13:23:46 2015
```

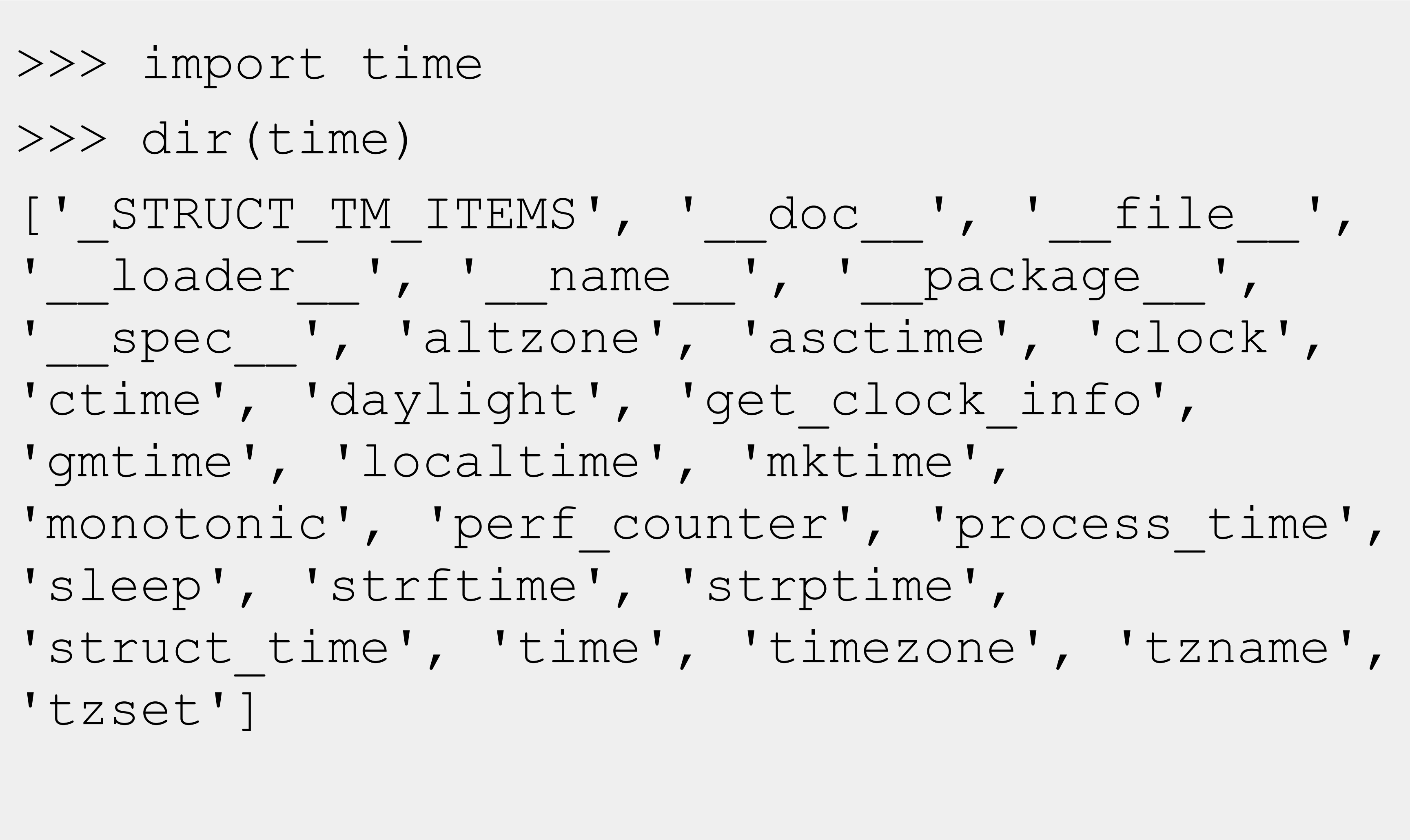
Modules

sleep()

time.sleep()

```
# Don't do this!
from time import *
from time import *
print(timezone)
print(asctime())
sleep(3)
print(asctime())
```

```
21600
Mon Nov 16 13:23:43 2015
Mon Nov 16 13:23:46 2015
```



```
>>> import time
>>> dir(time)
['_STRUCT_TM_ITEMS', '__doc__', '__file__',
'__loader__', '__name__', '__package__',
'__spec__', 'altzone', 'asctime', 'clock',
'ctime', 'daylight', 'get_clock_info',
'gmtime', 'localtime', 'mktime',
'monotonic', 'perf_counter', 'process_time',
'sleep', 'strftime', 'strptime',
'struct_time', 'time', 'timezone', 'tzname',
'tzset']
```

om

Module Search Path

sys.path - Returns the search path for modules.

```
import sys
sys.path
```

```
import sys
for path in sys.path:
    print(path)
```

```
/Users/jason
/Library/Frameworks/Python.framework/Versions/3.
4/lib/python34.zip
/Library/Frameworks/Python.framework/Versions/3.
4/lib/python3.4
/Library/Frameworks/Python.framework/Versions/3.
4/lib/python3.4/plat-darwin
/Library/Frameworks/Python.framework/Versions/3.
4/lib/python3.4/lib-dynload
```

```
import sys
sys.path.append('/Users/jason/python')
for path in sys.path:
    print(path)
```

```
/Users/jason
/Library/Frameworks/Python.framework/Versions/3.
4/lib/python34.zip
/Library/Frameworks/Python.framework/Versions/3.
4/lib/python3.4
/Library/Frameworks/Python.framework/Versions/3.
4/lib/python3.4/plat-darwin
/Users/jason/python
```

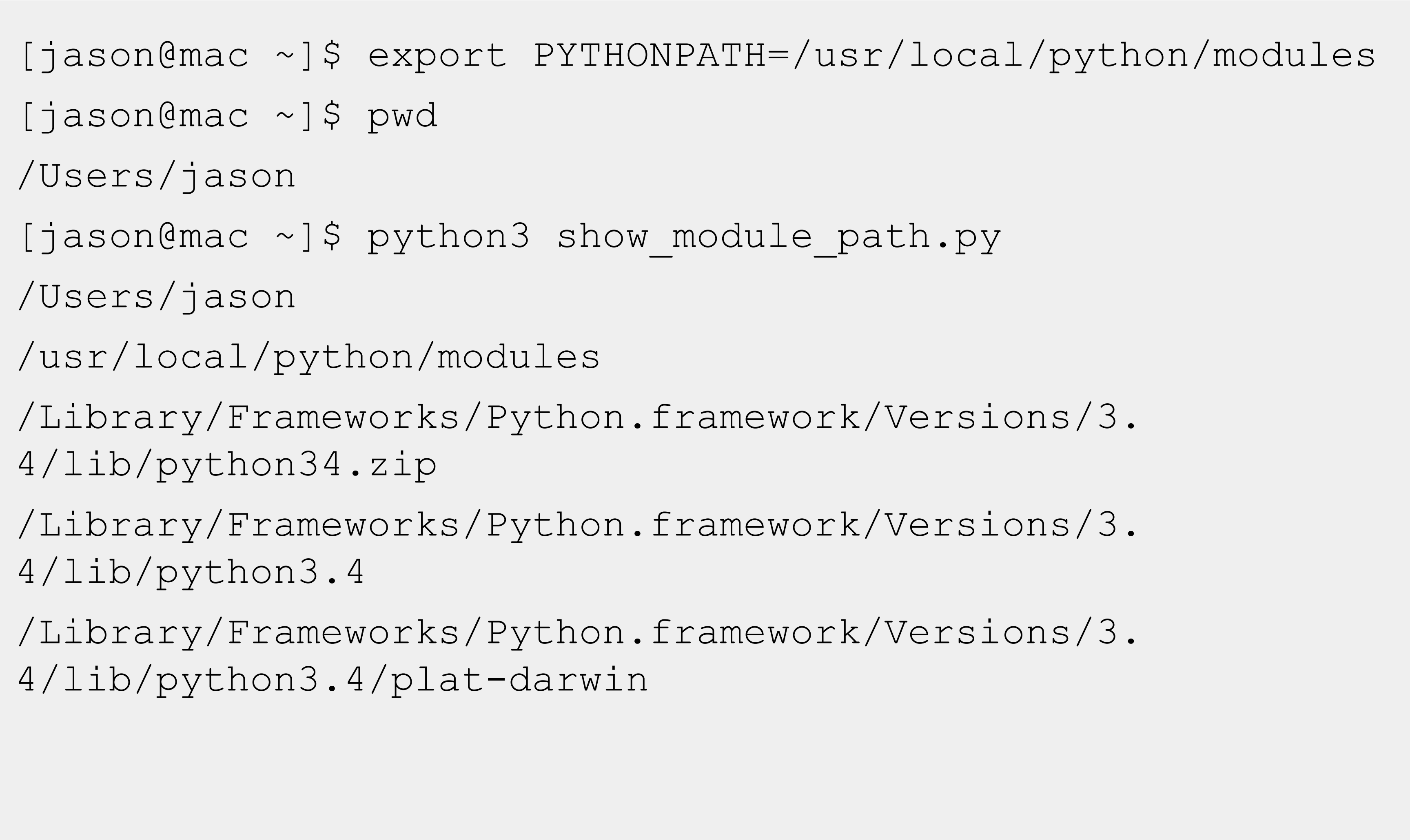
PYTHONPATH Environment Variable

Mac / Linux:

```
PYTHONPATH=path1:pathN
```

Windows:

```
PYTHONPATH=path1;pathN
```



```
[jason@mac ~]$ export PYTHONPATH=/usr/local/python/modules
[jason@mac ~]$ pwd
/Users/jason
[jason@mac ~]$ python3 show_module_path.py
/Users/jason
/usr/local/python/modules
/Library/Frameworks/Python.framework/Versions/3.
4/lib/python34.zip
/Library/Frameworks/Python.framework/Versions/3.
4/lib/python3.4
/Library/Frameworks/Python.framework/Versions/3.
4/lib/python3.4/plat-darwin
```

om

```
import say_hi
```

```
Traceback (most recent call last):
  File "test_say_hi.py", line 1, in <module>
    import say_hi
ImportError: No module named 'say_hi'
```

om

Python Standard Library

https://docs.python.org/3/library/

- Python is distributed with a large library of modules.
- Check the Python standard library before writing any of your own code!
- Just a few modules:
  - csv
  - logging
  - urllib.request
  - json

```
import sys
file_name = 'test.txt'
try:
    with open(file_name) as test_file:
       for line in test_file:
           print(line)
except:
    print('Could not open {}.'.format(file_name))
    sys.exit(1)
```

om

```
def say_hi():
    print('Hi!')
```

```
import say_hi
say_hi.say_hi()
```

```
Hi!
```

om

```
def say_hi():
    print('Hi!')

print('Hello from say_hi2.py!')
```

```
import say_hi2
say_hi2.say_hi()
```

```
Hello from say_hi2.py!
Hi!
```

om

```python
def say_hi():
    print('Hi!')

def main():
    print('Hello from say_hi3.py!')
    say_hi()

if __name__ == '__main__':
    main()
```

om

```
[jason@mac ~]$ python3 say_hi3.py
Hello from say_hi3.py!
Hi!
[jason@mac ~]$
```

Section Summary

Summary

- Python modules are files that have a .py extension and can implement a set of variables, functions, and classes.
- Use the `import module_name` syntax to import a module.

Summary

- The default module search path is determined by your Python installation.
- To manipulate the module search path modify `sys.path` or set the `PYTHONPATH` environment variable.

Summary

- The Python standard library is a large collection of code that can be reused in your Python programs.
- Use the `dir()` built-in function to find out what exists within a module.
- You can create your own personal library by writing your own modules.

Summary

- You can control how a Python program behaves based on whether it is run interactively or imported by checking the value of `__name__`.

```
if __name__ == '__main__':
    main()
```