SWINBURNE UNIVERSITY OF TECHNOLOGY
SARAWAK CAMPUS

Lecture 8

Python: Module

KNOWING

Introduction

- Modules are a simple way to organize a program which contains program code, variables, etc.
- All these definitions and statements are contained in a single Python file.
- The name of the module is the name of the file name with .py extension.

Introduction (cont.)

- To define a module, you can use Python IDLE, Notepad ++ or any suitable text editor.

Introduction (cont.)

- Modules are not loaded unless we executed in Python interpreter or call within a program.
- In Python, there are modules in standard library, current directory or directories containing .py files (in fact each file with .py extension is a module).

Example: Create a module

File Edit Format Run Options Window Help

```
#factorial.py

def factcal(n):
    fact = 1
    while n> 0:
        fact *= n
        n = n - 1
        if (n<=1):
            break
    else:
        print("Input a correct number:")
        return
    return fact

def factdata(n):
    result = []
    while n>0:
        result.append(n)
        n = n - 1
        if (n==0):
            break
    else:
        print("Input a correct number")
        return
    return result
```

From .. Import statement

- From..import statement is used to import selective names(s) or all names that a module defines.

Example: from .. Import statement

File Edit Format Run Options Window Help

```
#factorial.py

def factcal(n):
    fact = 1
    while n> 0:
        fact *= n
        n = n - 1
        if (n<=1):
            break
    else:
        print("Input a correct
        return
    return fact

def factdata(n):
    result = []
    while n>0:
        result.append(n)
        n = n - 1
        if (n==0):
            break
    else:
        print("Input a correct number")
        return
    return result
```

File Edit Format Run Options Window Help

```
import factorial

ans_factcal = factorial.factcal(5)
ans_factdata = factorial.factdata(5)

print(ans_factcal)
print(ans_factdata)
```

```
120
[5, 4, 3, 2, 1]
>>>
```

KNOW ING

Modules Path

- Python comes with numerous modules.
- Python has many standard modules as library. These are aimed to add efficiency or to access operating system primitives.
- Some of the modules depend upon the operating system
- sys is a Python standard module which is very useful. It is built into every Python interpreter.

The dir() function

- The built-in function dir() is used to get the names (a sorted list of strings), a module is defined. Check it into Python shell.

The dir() function

- The built-in function dir() is used to get the names (a sorted list of strings), a module is defined. Check it into Python shell.

```
>>> import factorial, sys
>>> dir(factorial)
['__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__',
'__package__', '__spec__', 'factcal', 'factdata']
>>>
```

KNOW ING

The dir() function

```
>>> dir(sys)
['__displayhook__', '__doc__', '__excepthook__', '__interactivehook__', '__loade
r__', '__name__', '__package__', '__spec__', '__stderr__', '__stdin__', '__stdou
t__', '_clear_type_cache', '_current_frames', '_debugmallocstats', '_getframe',
'_home', '_mercurial', '_xoptions', 'api_version', 'argv', 'base_exec_prefix', '
base_prefix', 'builtin_module_names', 'byteorder', 'call_tracing', 'callstats',
'copyright', 'displayhook', 'dllhandle', 'dont_write_bytecode', 'exc_info', 'exc
epthook', 'exec_prefix', 'executable', 'exit', 'flags', 'float_info', 'float_rep
r_style', 'get_coroutine_wrapper', 'getallocatedblocks', 'getcheckinterval', 'ge
tdefaultencoding', 'getfilesystemencoding', 'getprofile', 'getrecursionlimit', '
getrefcount', 'getsizeof', 'getswitchinterval', 'gettrace', 'getwindowsversion',
 'hash_info', 'hexversion', 'implementation', 'int_info', 'intern', 'is_finalizi
ng', 'maxsize', 'maxunicode', 'meta_path', 'modules', 'path', 'path_hooks', 'pat
h_importer_cache', 'platform', 'prefix', 'set_coroutine_wrapper', 'setcheckinter
val', 'setprofile', 'setrecursionlimit', 'setswitchinterval', 'settrace', 'stder
r', 'stdin', 'stdout', 'thread_info', 'version', 'version_info', 'warnoptions',
'winver']
>>>
```

KNOW ING