New York University
School of Continuing and Professional Studies
Division of Programs in Information Technology

Advanced Python

Executive Summary, Session 3

USER-DEFINED FUNCTIONS

- **argument** and **return value**

```
def myfunc(this_arg):
new_val = this_arg + 1
     return new_val
```

- **positional arguments**

```
def myfunc(arg1, arg2):
```

- **keyword arguments**

```
def myfunc(this=None, that=0):
```

- **arbitrary arguments: *args**

```
def myfunc(*args):
```

- **arbitrary keyword arguments: **kwargs**

```
def myfunc(**kwargs):
```

USING MODULES

- **import modname**: import a module's variables through its name

```
import mymod
print mymod.var
```

- **import modname as convenientname**: rename module in our code

```
import mymod as tt
print tt.var
```

- **from modname import varname**: import a variable from module

```
from mymod import var
```

- **from modname import ***: don't ever do this! ;)

NO

- **sys.path**: module search path

```
sys.path.append(newdir)
```

- **PYTHONPATH:** environment variable informing the **sys.path**
- **if __name__ == '__main__':** module / script "identity gate"

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

- **pip** module install program

MODULES

- **time**:

```
import time
time.sleep(10)
```

- **date** and **datetime**:

```
from datetime import datetime
dt = datetime.now()
```

- **timedelta**:

```
from datetime import timedelta
td = timedelta(hours=3)
```

EXCEPTIONS

- **UnboundLocalError**: reading a local variable before assigning

```
def dothis():
    x = x + 1    # is x a global?
```

- **IndentationError**: when an indent occurs before a block, or doesn't occur inside a block

```
for item in mylist:
print item
```

- **raise**: cause an exception to occur

```
raise ValueError
```

CLASSES

- **class** declaration

```
class MyClass(object):
    pass
```

- set object's attributes (__dict__)

```
x = MyClass()
x.var = 5
```

- get attribute value

```
print x.var
```

- **self** implicit argument to method

```
def dothis(self):
```

- __**init**__ constructor

```
def __init__(self):
```