Learning Python

Session 14 - Modules

Modules - Example

```
# Fibonacci numbers module
# save this in a file called fibo.py
#Modules are regular python programs
def fib(n):    # write Fibonacci series up to n
    a, b = 0, 1
    while b < n:
        print b,
        a, b = b, a+b

def fact(n):
    if n <= 0: return 1;
    return fact(n-1);
```

Modules - Example

```
>>> import fibo
>>> fibo.fib(1000)
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
>>> fibo.__name__
'fibo'

#To use it often
>>> fib = fibo.fib
>>> fib(500)
1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

Modules - Example

We can import few functions only.

```
>>> from fibo import fib, fact
>>> fib(500)
1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

Or we can import all

```
>>> from fibo import *
>>> fib(500)
1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

Modules - Differentiating

```
#save it as modls.py
if __name__  == "__main__":
        print "Called as standalone";
else:
        print "called as " + __name__;
```

```
$ python modls.py
Called as standalone

>>> import modls
called as modls
```

Modules - Search Path

1. *If we import spam module*
2. *First built-in modules are searched*
3. *Then, all the path in variable sys.path are searched*
4. *Current Program can modify sys.path*

Compiled Python

1. *It creates a .pyc for each .py*
   a. *If can't it ignores*
2. *To make it faster to load not run*
3. *It is based on modification time*
4. *.pyc contents are platform independent*

Packages

```
sound/                          Top-level package
      __init__.py               Initialize the sound package
      formats/                  Subpackage for file format conversions
              __init__.py
              wavread.py
              wavwrite.py
              aiffread.py
              aiffwrite.py
              auread.py
              auwrite.py
              ...
      effects/                  Subpackage for sound effects
              __init__.py
              echo.py
              surround.py
              reverse.py
              ...
      filters/                  Subpackage for filters
              __init__.py
              equalizer.py
              vocoder.py
              karaoke.py
              ...
```

1. Load a subpackage
   a. import sound.effects.echo
   b. Absolute names to invoke:
   c. sound.effects.echo.echofilter()

2. Or you also load in the following way
   a. from sound.effects import echo
   b. Then you can call:echo.echofilter()

3. Or you could
   a. from sound.effects.echo import echofilter
   b. echofilter()

Packages

1. *A way of organizing the code in the form of A.B*
2. *In A.B, A is package and B is another package*
3. *In A.B, B is called subpackage*
4. *Code is organized in the form of Folders*

Packages

__init__.py

__init__.py:

1. Every package folder should have a file named __init__.py
2. Even if empty

Packages & Sub Packages

1. A package folder can have other packages called subpackages
2. A package can import modules defined in parent or child in following manner:
   i. *from . import echo*
   ii. *from .. import formats*
   iii. *from ..filters import equalizer*

Built-in Functions

abs()	divmod()	input()	open()	staticmethod()
all()	enumerate()	int()	ord()	str()
any()	eval()	isinstance()	pow()	sum()
basestring()	execfile()	issubclass()	print()	super()
bin()	file()	iter()	property()	tuple()
bool()	filter()	len()	range()	type()
bytearray()	float()	list()	raw_input()	unichr()
callable()	format()	locals()	reduce()	unicode()
chr()	frozenset()	long()	reload()	vars()
classmethod()	getattr()	map()	repr()	xrange()
cmp()	globals()	max()	reversed()	zip()
compile()	hasattr()	memoryview()	round()	__import__()
complex()	hash()	min()	set()	
delattr()	help()	next()	setattr()	
dict()	hex()	object()	slice()	
dir()	id()	oct()	sorted()	

https://docs.python.org/2/library/functions.html

Useful Python Packages

PyGame - Game Development
PyGtk - Bindings for the cross-platform Gtk toolkit.
Matplotlib - Production quality output in a wide variety of formats
NumPy, SciPy - Includes modules for graphics and plotting, optimization, integration, special functions, signal and image processing, genetic algorithms ...
Django - High-level web framework.
Beautiful Soup - HTML/XML parser
pandas - Python Data Analysis Library, R Alternative

More at https://wiki.python.org/moin/UsefulModules

Questions?