

SETS: Please submit these.

Namespaces

A namespace is a collection of identifiers belonging to a module, or to a function, or other structure in Python.

Each module has its own namespace

Ex. Module1.py

Module2.py

Every module has a special attribute called `--name--`, which is a string containing the name of the module.

→ If a module is run as a program (rather than being imported by another program), then its name becomes '__main__'. Thus

```
if __name__ == '__main__':
```

```
    # stuff
```

```
#end if
```

asks: "if this module is running as a program, then"

Also, Functions have their own name space.

EX NameSpace.py

any file, ending in .py, containing ³
Python code, constitutes a module

file: Prog.py

module: Prog

namespace: Prog

we have a 1-1 correspondence

file $\leftrightarrow$ module $\leftrightarrow$ namespace,

but not same concept.

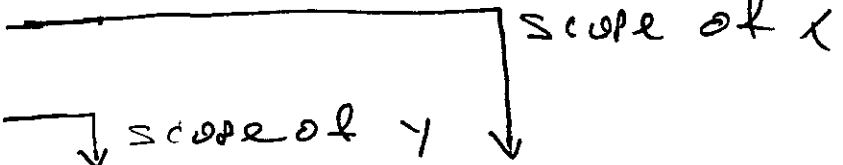
Scope

The scope of an identifier is the region of a program where it can be accessed.

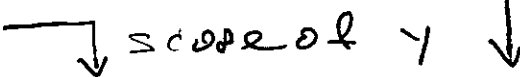
Local scope : within a function definition

```
def f():
```

```
    x = 3
```



```
    y = 2
```

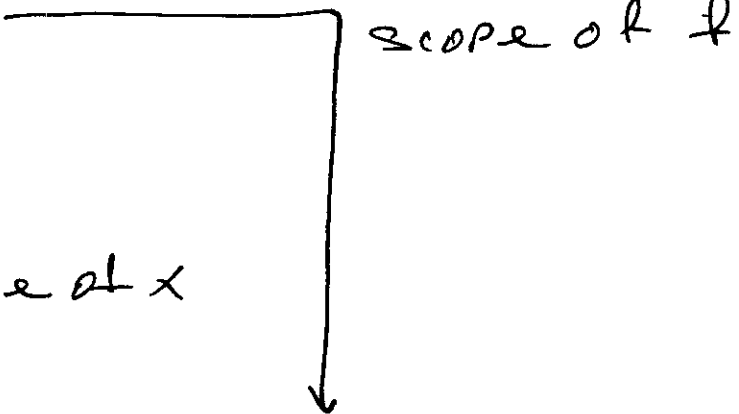


```
#end f()
```

Global scope : within current module

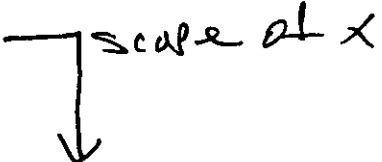
```
# module.py
```

```
def f():
```



```
    ...
#end f()
```

```
x = 6
```

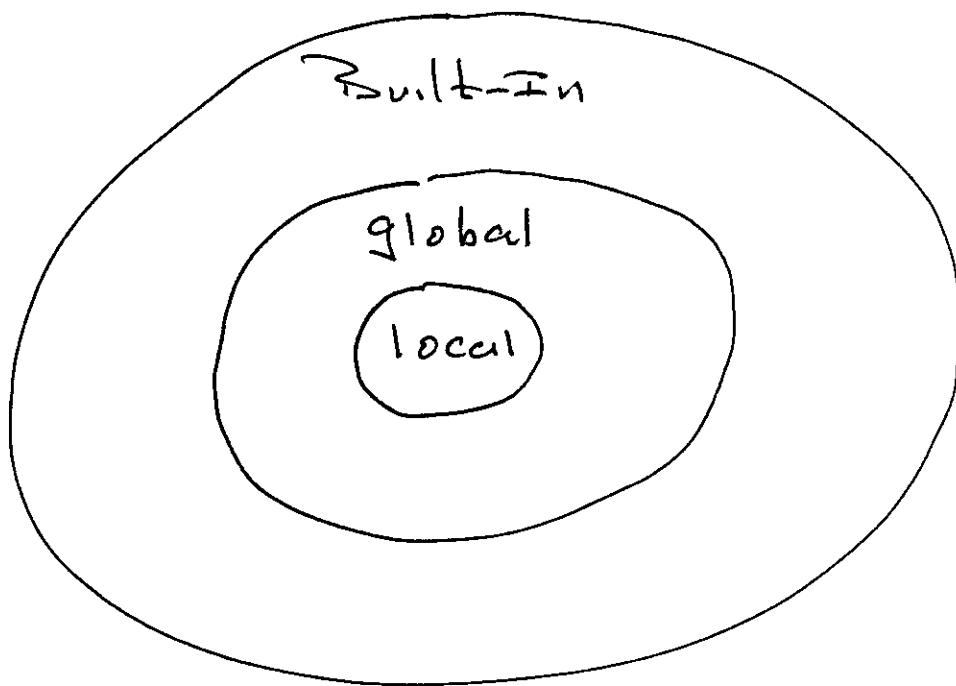


```
#end module
```

Built-In Scope : encloses all modules,
and includes all built-in functions

```
Print()
sum()
range()
!
```

when there are name collisions,
the innermost scope takes precedence.



Interactive :

```
Print = 'happy'
```

`del Print` # removes the binding of name
Print to global scope.

Variations on 'import' :

- import math,
x = math.sqrt(10)
- from math import sqrt
x = sqrt(10)
- from math import * #everything
x = sqrt(10)
- import math as m
x = m.sqrt(10)
- from math import sqrt as s
x = s(10)

next time

17

chap. 14 files .

Quiz Review .

① write a fn that prints out
a list of lists of lists .

```
def printList(L) :
```

```
for i in range(len(L))
```

```
    Print('L')
```

```
    for i in range(len(L))
```

```
        Print('...L')
```

```
        for i in range(len(L[i]))
```

```
            for k in range
```

```
                Print('...', L[i][L[i]])
```

```
        Print('...']')
```

```
    Print('[]')
```