Python Basics

Python is an interpreter, not a compiler

⇒ Your program (or you, at the console) are "in conversation" with the Interpreter

- Objects, functions, types are created and removed on demand.
- You can (not easy!) do interactive debugging

Data Types

int, long, float, complex, string, bool (sort of)

Variables

x, y, b2, __symmetric__ – k2 — special

- variables can be primitive data (1, "ab", ...)
- " " " tuples, lists, dictionaries
- " " " object references

mutable vs immutable

Operations on variables

```
x + y
x.method()      # x += y  same as
                  x = x.__iadd__(y)
```

suggests overloading?

Tuples

```
x = (5, 6, "end")
a, b, c = x      # what happens?
```

Lists

```
x = [ 8, 9e10, x ]
x[2] ......
```

Dictionary

```
x = { 'first': 5, "second": 0, "third": 2 }
x['second']
```

Slicing

x[14 : 20]

refers to the segment of list x,

x[14] x[15] ... x[18] x[19]

Since lists are mutable, we can replace or delete a slice in a list

ex. del x[14 : 20]

x[14 : 20] = [5, 4]

Comparison

x > y , x >= y , etc

x == y , x != y

x is y , x is not y

x in y , x not in y ? dictionary, sequence

P and Q , P or Q , not P

chains of compare a < b < c

List & Dictionary Operations

Lists + , concatenate ; * , repeated +

L.count( ), L.index( ), L.append( ), ...

Dictionaries:

D.keys( ), D.values( ), ...

D.iteritems( )
D.iterkeys( )
D.itervalues( )

} return a function of a special type called an iterator.

Program Control

if expression :
while expression :
for expression :
try :
≡ and more

} can either be a one-liner or followed by a compound statement -- can have break / continue

for is the most overload of these. (not like C, C++, Java ...)

ex: for x in "this is a string" :
    print x

    for i in range(0,n): x[i] = None

in Java this would be:
for (i=0; i<n; i++) x[i] = null;

Functions

def funcName (parameters):
    statements

In Python, the parameters can be positional, keyword, optional, default, and any variation of these.

ex: def divide (p,q): return p/q
    divide (8,2)
    divide (q=2, p=8)

Watch out! See example on page 60!