References ("ref:") are from http://docs.python.org/py3k/reference/

Except where otherwise indicated, each item separately quoted in a "Prints" comment appears on a separate line in the output.

Values and Variables

Variables are not declared;
Variables can be assigned any type of value at any time using =.

```
average = ( first + second ) / 2
```

Operators

add + ; subtract - ; multiply * ; power **

- Truncating (round-down) division: `//`
- Normal division: `/`
- String concatenation uses `+`

Comparison (==,!=,<,<=,>,>=) checks object content, (not *addresses*) for all standard types.

Logic operators: **`and`**, **`or`**, **`not`**

"if" statements

```
if volume > 100:
      print( "Big!" )
      big = big + 1
elif volume > 25:
      print( "medium." )
else:
      print( "small." )
      small = small + 1
```

Common functions

```
int( "52" ) # The integer 52
int( 98.6 ) # The integer 98
str( 52 ) # The string "52"
float( 52 ) # The float 52.0

x = 42
y = 24
print( x )
# Prints "42" on its own line

print( x, y )
# Prints "42 24" on one line

print( str(x) + "|" + str(y) )
# Prints "42|24"

n = int( \
      input( "Number, please: " ))
# Reads in literal string;
# int() converts it
```

Importing Packages

To use code from another Python file...

```
import math

if x >= 0:
      print( "Square root of", \
             x, " is ", \
             math.sqrt( x ) )
else:
      print( "Negative number!" )
```

Alternative (beware of name conflicts.)

```
from math import *

if x >= 0:
      print( "Square root of", \
             x, " is ", sqrt( x ) )
else:
      print( "Negative number!" )
```

String - type name "`str`"

Use double <u>or</u> single quotes.
There is no separate character type.
To make a multi-line string, **`"""use 3 double (or single) quotes."""`**
Indexing with brackets (**`s[i]`**) works.

Defining Your Own Functions

ref: compound_stmts.html#function-definitions

```
def order( val1, val2 ):
    """State which value naturally
       comes first.
    """
    if val1 < val2:
        print(val1, "comes first")
    else:
        print(val2, "comes first")


def sum3( a, b, c ):
    """Add 3 numbers."""
    return a + b + c
```

The string that follows the header is used for documentation generation.

```
order( "joe", "black" )
# Prints "black comes first"

order( 13, 21 )
# Prints "13 comes first"


print( sum3( 1, 5, 9 ) )
# Prints "15"
```