Learn Python Through Data Hacking

Reference Card

Basic Statements

```
name = value                Variable Assignment
print(item)                 Printing
if condition:               Conditional
   statements
else:
  statements
while condition:            Loop on condition
   statements
for x in sequence:          Iterate over data
   statements
def func(x, y, z):          Define a function
   statements
   return result
import module               Library import
try:                        Exceptions
   statements
except Exception as e:
   statements
```

Basic Python Types

```
None                        Nothing
True, False                 Booleans
123                         Integer
12.34                       Float
'Hello'                     String
"World"                     String
[1, 2, 3]                   List
('ACME', 50, 91.1)          Tuple
{'a' : 2, 'b': 3}           Dictionary
```

String Operations

```
len(s)                  # Length
s[n]                    # Indexing
s[start:end]            # Slice
s.strip([chrs])         # Strip chars
s.upper()               # Case convert
s.lower()               # Case convert
s.split([sep])          # Split on sep
s.startswith(prefix)    # Test start
s.endswith(suffix)      # Test end
sep.join(parts)         # Join parts
```

List Operations

```
items = []              # Empty list
len(items)              # Length
items[n]                # Indexing
items[start:end]        # Slice
items.append(x)         # Add to end
items.insert(n, x)      # Insert at n
items.remove(x)         # Remove x
items.sort()            # Sort items
for x in items:         # Iteration
   statements
[expr for x in items    # List
      if condition]     # Comprehension
```

Dictionary Operations

```
map = { }               # Empty dict
map[key] = value        # Assign item
v = map[key]            # Lookup item
del map[key]            # Delete item
key in map              # Test exists
map.keys()              # List of keys
map.values()            # List of vals
map.items()             # List of items
```

Operators

```
x + y           # Math
x - y
x * y
x / y
x ** y

x < y           # Tests
x <= y
x > y
x >= y
x == y
x != y

x in y       # Membership
x not in y
```

Files

```
# Read file all at once
f = open('filename','r')
data = f.read()
f.close()

# Read file line-by-line
f = open('filename', 'r')
for line in f:
    statements
f.close()

# Write to a file
f = open('filename','w')
f.write(text)
f.close()
```

Parsing CSV Data

portfolio.csv

```
Name,Shares,Price
"AA",100,32.20
"IBM",50,91.10
"CAT",150,83.44
"MSFT",200,51.23
"GE",95,40.37
"MSFT",50,65.10
"IBM",100,70.44
```

```
# Read data and perform a calculation
import csv
f = open('portfolio.csv', 'r')
total = 0.0
for row in csv.DictReader(f):
    shares = int(row['Shares'])
    price = float(row['Price'])
    total += shares * price
f.close()

# Turn the file into a list of dicts
f = open('portfolio.csv', 'r')
portfolio = list(csv.DictReader(f))
f.close()

# Sort data by price
portfolio.sort(key=lambda s: s['Price'])

# Filtering data
more100 = [s for s in portfolio
             if s['Shares'] > 100]
```

Parsing XML Data

portfolio.xml

```
<portfolio>
    <stock>
        <name>AA</name>
        <shares>100</shares>
        <price>32.20</shares>
     </stock>
     <stock>
         <name>IBM</name>
         <shares>50</shares>
         <price>91.1</price>
     </stock>
     ...
</portfolio>
```

```
# Parse XML and perform a calculation
from xml.etree.ElementTree import parse
doc = parse('portfolio.xml')

total = 0.0
for stock in doc.findall('stock'):
    shares = int(stock.findtext('shares'))
    price  = float(stock.findtext('price'))
    total += shares * price
```

Downloading Data From the Web

```
# Python 2
import urllib
u = urllib.urlopen('http://www.python.org')
data = u.read()

# Python 3
import urllib.request
u = urllib.request.urlopen('http://www.python.org')
data = u.read()
```