Engineering Sketch Pad (ESP)

Training Session 3.1
Python Tutorial

John F. Dannenhoffer, III
jfdannen@syr.edu
Syracuse University

Bob Haimes
haimes@mit.edu

Marshall Galbraith
galbramc@mit.edu

Massachusetts Institute of Technology

Overview

- Language Overview
  - File Structure / Imports / Comments
  - Simple Variables
  - Expressions / Order of Operations / Assignments
  - If / Else
  - For / While / Break / Continue
- Functions
- Tuples / Lists / Dictionaries
- Variable Scope
- Classes

Example Program 1 (1)

```
# ********************************************************************
# *                                                                  *
# * Example1 -- examples for Introduction to Python                  *
# *                                                                  *
# *              Written by John Dannenhoffer @ Syracuse University *
# *                                                                  *
# ********************************************************************

from __future__ import print_function
from math import sqrt, floor



# ********************************************************************
# *                                                                  *
# *   main - main program                                            *
# *                                                                  *
# ********************************************************************

def Example1():
    """
    example program for Example1

    inputs:
        none
    output:
        none

    Written by John Dannenhoffer
    """
```

Example Program 1 (2)

```
    # simple assignments
    i   = 7                         # 7
    j   = 3                         # 3
    ONE = 1                         # 1
    TWO = 2                         # 2
    a = 7.0                         # 7.0
    b = 3.0                         # 3.0
    c = 2.0                         # 2.0

#    order of operations (highest to lowest):
#       (...), [...], {...}    tuple, list, dictionary creation
#       '...'                  string conversion
#       S[i], S[i:j]           indexing and slicing
#       s.attr                 attributes
#       f(...)                 function calls
#       +x, -x, ~x             unary operators
#       x**y                   power (Right-to-left)
#       x*y, x/y, x//y, x%y    multiplication, division, floor, modulo
#       x+y, x-y               addition, subtraction
#       x<<y, x>>y             bit shifting
#       x&y                    bitwise and
#       x^y                    bitwise exclusive or (xor)
#       x|y                    bitwise or
#       x<y, x<=y, x>y, x>=y   comparison, identity, sequence membership test
#       x==y, x!=y
#       x<>y
#       x is y, x is not y
#       x in S, x not in S
#       not x                  logical negation
#       x and y                logical and
#       x or y                 logical or
#       lambda args: expr      anonymous function
```

Example Program 1 (3)

```
# expressions
k = i + j                       # k=10
print('k = ' + repr(k))

k = i - j                       # k=4
print('k = ' + repr(k))

k = i * j                       # k=21
print('k = ' + repr(k))

k = i / j                       # k=2     (in python 2)
                                # k=2.333 (in python 3)
print('k = ' + repr(k))

k = floor(i / j)                # k=2.0
print('k = ' + repr(k))

k = i // j                      # k=2
print('k = ' + repr(k))

k = i % j                       # k=1
print('k = ' + repr(k))

k = pow(i, 2)                   # k=49
print('k = ' + repr(k))

k = i ** 2                      # k=49
print('k = ' + repr(k))

k += 3                          # k=52
print('k = ' + repr(k))
```

Example Program 1 (4)

```
c = a + b                       # c=10.0
print('c = ' + repr(c))

c = a - b                       # c=4.0
print('c = ' + repr(c))

c = a * b                       # c=21.0
print('c = ' + repr(c))

c = a / b                       # c=2.3333
print('c = ' + repr(c))

c = pow(a, 2)                   # c=49.0
print('c = ' + repr(c))

c = a ** 2                      # c=49.0
print('c = ' + repr(c))

d = 1/2 * a                     # d=0.0    (in python 2)
                                # d=3.5    (in python 3)
print('d = ' + repr(d))

d = 0.5 * a                     # d=3.5
print('d = ' + repr(d))

d = 1/2. * a                    # d=3.5
print('d = ' + repr(d))

d = ONE/float(TWO) * a          # d=3.5
print('d = ' + repr(d))
```

Example Program 1 (5)

```
# quadratic formula
a = 4.
b = 8.
c = 2.

NROOT = 8
roots = [None] * NROOT          # array of length NROOT

roots[0] = (-b + sqrt(b**2 - 4*a*c)) / (2*a)
roots[1] = (-b - sqrt(b**2 - 4*a*c)) / (2*a)
```

Example Program 1 (6)

```
# simple logic block
if (a > b and c == 2):
    d = 1
elif (b != 3):
    d = 2                       # d=2.0 (this gets executed)
else:
    d = 3
print('d = ' + repr(d))

# sample loop
for i in range(0, NROOT):
    roots[i] = float(i + 1)

    if (i == 4):
        continue                # skip to next i and top of loop
    elif (i > 6):
        break                   # exit loop

    # print out:
    #       roots[0] = 1.0
    #       roots[1] = 2.0
    #       roots[2] = 3.0
    #       roots[3] = 4.0
    #       roots[5] = 6.0
    #       roots[6] = 7.0
    #
    print('roots[' + repr(i) + '] = ' + repr(roots[i]))

return
```

Example Program 1 (7)

```
# get help on the functions in this file
help(Example1)

# run the tests
print('Run Example1:\n')
Example1()

print('\nDone')
```

Example Program 2 (1)

```
# ********************************************************************
# *                                                                  *
# * Example2 -- examples for Introduction to Python                  *
# *                                                                  *
# *              Written by John Dannenhoffer @ Syracuse University *
# *                                                                  *
# ********************************************************************

from __future__ import print_function
from math import sqrt

# ********************************************************************

def Example2():
    """
    example program for Example 2

    inputs:
        none
    output:
        none

    Written by John Dannenhoffer
    """
```

Example Program 2 (2)

```
# functions are called by value
leg1 = 3
leg2 = 4
hyp  = hypot2(leg1, leg2)
print('hyp = ' + repr(hyp))

val1 = 3
val2 = 4
val1, val2 = swapInt(val1, val2)
print('val1 = ' + repr(val1) + '   val2 = ' + repr(val2))
```

Example Program 2 (3)

```
# quadratic formulae
ans = quadform( 2, 2, -12)
print('ans = ' + repr(ans))

ans = quadform( 2, 2,  12)
print('ans = ' + repr(ans))

ans = quadform( 4, -12, -12)
print('ans = ' + repr(ans))

ans = quadform( 2, 0, -18)
print('ans = ' + repr(ans))

ans = quadform( 2, 0, 0)
print('ans = ' + repr(ans))

ans = quadform( 0, 2, -6)
print('ans = ' + repr(ans))

ans = quadform( 0, 2, 0)
print('ans = ' + repr(ans))

ans = quadform(0, 0, 3)
print('ans = ' + repr(ans))
```

Example Program 2 (4)

```
# tuples, lists, and dictionaries
tuple1 = (1, 2, 3)                # tuples are immutable
print('tuple1='+repr(tuple1))     # outputs (1, 2, 3)
print(' ')

tuple2 = appendToSeq(tuple1)
print('tuple1='+repr(tuple1))     # outputs (1, 2, 3)
print('tuple2='+repr(tuple2))     # outputs (1, 2, 3, 9, 9, 9)
print(' ')

try:
    tuple3 = changeFirst(tuple1) # causes TypeError since tuple is immutable
except TypeError:
    print('error in changeFirst(tuple1)')
print(' ')
```

Example Program 2 (5)

```
list1  = [1, 2, 3]                # lists  are   mutable
print('list1 ='+repr(list1))      # outputs [1, 2, 3]
print(' ')

list2  = appendToSeq(list1)       # list1 and list2 point to same memory
print('list1 ='+repr(list1))      # outputs [1, 2, 3, 9, 9, 9]
print('list2 ='+repr(list2))      # outputs [1, 2, 3, 9, 9, 9]
print(' ')

list3  = changeFirst(list1)       # list1, list2, and list3 all point to same memory
print('list1 ='+repr(list1))      # outputs [0, 2, 3, 9, 9, 9]
print('list2 ='+repr(list2))      # outputs [0, 2, 3, 9, 9, 9]
print('list3 ='+repr(list3))      # outputs [0, 2, 3, 9, 9, 9]
print(' ')

list4  = list1[:]                 # list4 is a separate list
print('list1 ='+repr(list1))      # outputs [0, 2, 3, 9, 9, 9]
print('list2 ='+repr(list2))      # outputs [0, 2, 3, 9, 9, 9]
print('list3 ='+repr(list3))      # outputs [0, 2, 3, 9, 9, 9]
print('list4 ='+repr(list4))      # outputs [0, 2, 3, 9, 9, 9]
print(' ')

list4[2] = 22                     # only list4 changes
print('list1 ='+repr(list1))      # outputs [0, 2, 3, 9, 9, 9]
print('list2 ='+repr(list2))      # outputs [0, 2, 3, 9, 9, 9]
print('list3 ='+repr(list3))      # outputs [0, 2, 3, 9, 9, 9]
print('list4 ='+repr(list4))      # outputs [0, 2, 22, 9, 9, 9]
print(' ')
```

Example Program 2 (6)

```
dict1  = {'Name': 'John', 'Age': 21, 'Favs': (1, 17, 47)}
print('dict1 ='+repr(dict1))     # output {'Age': 21, 'Favs': (1, 17, 47), 'Name': 'John'}

dict1['Age'] = 22                # updates Age
dict1['School'] = 'Syracuse'     # appends School
print('dict1 ='+repr(dict1))     # output {'Age': 22, 'Favs': (1, 17, 47), 'Name': 'John', 'School': 'Syrac

del dict1['Favs']                # removes entry with key Favs
dict1.clear()                    # removes all entries in dict
del dict1                        # deletes entire dictionary

dict2  = {'Name': 'John', 'Age': 21, 'Favs': (1, 17, 47)}
print('len ='+repr(len(dict2)))  # outputs 3
print('str ='+str(dict2))        # outputs {'Age': 21, 'Favs': (1, 17, 47), 'Name': 'John'}
```

Example Program 2 (7)

```
# pseudo-2D array (stored across the rows)
NROW = 3
NCOL = 4

mat1 = [None] * (NROW*NCOL)

for i in range(0, NROW):
    for j in range(0, NCOL):
        mat1[(i)*NCOL+(j)] = i * 10 + j

# square the diagonal entries
squareDiag(mat1, NROW, NCOL)

# print out matrix
k = 0
for i in range(0, NROW):
    for j in range(0, NCOL):
        print('mat1('+repr(i)+','+repr(j)+') = '+repr(mat1[k]))
        k += 1
print(' ')
```

Example Program 2 (8)

```
# variable scope
one = 1
two = 2
print('in Example2: one='+repr(one)+'   two='+repr(two))

varscope()

print('in Example2: one='+repr(one)+'   two='+repr(two))

return
```

Example Program 2 (9)

```
def hypot2(x,
           y):
    """
    hyponentuse of right triangle

    inputs:
        leg1
        leg2
    output:
        hypeneuse
    """

    return sqrt(x * x + y * y)
```

Example Program 2 (10)

```
def swapInt(a,
            b):
    """
    swap two numbers

    inputs:
        num1
        num2
    output:
        (num2, num1)
    """

    return (b, a)
```

Example Program 2 (11)

```
def quadform(a,
             b,
             c):
    """
    quadratic formula: a*x^2+b*x+c=0

    inputs:
        a
        b
        c
    output:
        (-3)          no roots
        (-2)          complex roots - not returned
        (-1, x1)      linear equation
        ( 0, x1, x2)  roots of quadratic
    """

    # quadratic
    if (abs(a) > 1e-16):
        det = b * b - 4 * a * c
        if (det < 0):
            return -2

        x1 = (-b + sqrt(det)) / (2 * a)
        x2 = (-b - sqrt(det)) / (2 * a)

        return (0, x1, x2)
```

Example Program 2 (12)

```
    # linear
    elif (abs(b) > 1e-16):
        x1 = -c / b
        return (-1, x1)

    # no solution
    else:
        return -3
```

Example Program 2 (13)

```
def appendToSeq(myseq):
    """
    append 9,9,9 to sequence

    inputs:
        myseq
    output:
        myseq
    """

    myseq += (9, 9, 9)
    return myseq
```

Example Program 2 (14)

```
def changeFirst(myseq):
    """
    change first element to 0

    inputs:
        myseq
    output:
        myseq
    """

    myseq[0] = 0
    return myseq
```

Example Program 2 (15)

```
def squareDiag(A,
               nrow,
               ncol):
    """
    square diagonal elements

    inputs:
        A
        nrow
        ncol
    output:
        none
    """

    for i in range(0, min(nrow,ncol)):
        A[(i)*ncol+(i)] = pow(A[(i)*ncol+(i)], 2)
```

Example Program 2 (16)

```
def varscope():
    """
    demonstrate variable scope

    inputs:
        none
    output:
        none
    """

    one = -1
    two = -2

    print('in varscope: one='+repr(one)+'   two='+repr(two))

    def funscope():
        one = 11
        two = 22

        print('in funscope: one='+repr(one)+'   two='+repr(two))

    funscope()

    print('in varscope: one='+repr(one)+'   two='+repr(two))
```

Example Program 2 (17)

```
# get help on the functions in this file
help(Example2)

# run the tests
print('Run Example2:\n')
Example2()

print('\nDone')
```

Example Program 3 (1)

```
# ********************************************************************
# *                                                                  *
# * Example3 -- examples for Introduction to Python                  *
# *                                                                  *
# *              Written by John Dannenhoffer @ Syracuse University *
# *                                                                  *
# ********************************************************************

from __future__ import print_function
from   copy     import deepcopy

# ********************************************************************

# global definitions
PI = 3.14159265358979311599796 35

# ********************************************************************

def Example3():
    """
    example program for Example 3

    inputs:
        none
```

Example Program 3 (2)

```
# get an array of NBOX Boxes
NBOX = 10
myBoxes = [Box() for count in range(NBOX)]

# set the size of the Boxes
i = 0
for thisBox in myBoxes:
    thisBox.length = i
    thisBox.height = i*i
    thisBox.update()
    i += 1

# print box info
for i in range(0, len(myBoxes)):
    print('myBoxes[ ' + format(i, '5d') + ']' +
          '  .length='   + format(myBoxes[i].length,  '8.2f') +
          '  .height='   + format(myBoxes[i].height,  '8.2f') +
          '  .perim='    + format(myBoxes[i].perim(), '8.2f') +
          '  .area='     + format(myBoxes[i].area(),  '8.2f') +
          '  .centroid=' + repr(  myBoxes[i].centroid))
print('')

# make another reference to myBoxes
myBoxes1 = myBoxes

# make a deep copy of myBoxes
myBoxes2 = deepcopy(myBoxes)
```

Example Program 3 (3)

```
# change one of the Boxes
myBoxes[3].length = 0     # note: this will not update centroid

# print box info
for i in range(0, len(myBoxes)):
    print('myBoxes[ ' + format(i, '5d') + ']' +
          '  .length='   + format(myBoxes[i].length,  '8.2f') +
          '  .height='   + format(myBoxes[i].height,  '8.2f') +
          '  .perim='    + format(myBoxes[i].perim(), '8.2f') +
          '  .area='     + format(myBoxes[i].area(),  '8.2f') +
          '  .centroid=' + repr(  myBoxes[i].centroid))
print('')

for i in range(0, len(myBoxes1)):
    print('myBoxes1[' + format(i, '5d') + ']' +
          '  .length='   + format(myBoxes1[i].length,  '8.2f') +
          '  .height='   + format(myBoxes1[i].height,  '8.2f') +
          '  .perim='    + format(myBoxes1[i].perim(), '8.2f') +
          '  .area='     + format(myBoxes1[i].area(),  '8.2f') +
          '  .centroid=' + repr(  myBoxes1[i].centroid))
print('')

for i in range(0, len(myBoxes2)):
    print('myBoxes2[' + format(i, '5d') + ']' +
          '  .length='   + format(myBoxes2[i].length,  '8.2f') +
          '  .height='   + format(myBoxes2[i].height,  '8.2f') +
          '  .perim='    + format(myBoxes2[i].perim(), '8.2f') +
          '  .area='     + format(myBoxes2[i].area(),  '8.2f') +
          '  .centroid=' + repr(  myBoxes2[i].centroid))
print('')
```

Example Program 3 (4)

```
# get an array of NCIRCLE Circles
NCIRCLE = 5
myCircles = [Circle(i+4) for i in range(NCIRCLE)]

# print circle info
i = 0
for thisCircle in myCircles:
    print('myCircles[' + format(i, '5d') + ']' +
          '  .radius='   + format(thisCircle.radius,   '8.2f') +
          '  .diam='     + format(thisCircle.diam(),   '8.2f') +
          '  .circum='   + format(thisCircle.circum(), '8.2f') +
          '  .area='     + format(thisCircle.area(),   '8.2f'))
    i += 1
```

Example Program 3 (5)

```
class Box:
    """
    A box anchored at origin
    """

    # length        -> length (in x)
    # height        -> height (in y)
    # perim()       -> perimeter
    # area()        -> area
    # update()      -> None               (updates centroid)
    # centroidc     -> (xcent, ycent)

    def __init__(self, *args):
        if (len(args) < 1):
            self.length = 0.0
        else:
            self.length = float(args[0])
        if (len(args) < 2):
            self.height = 0.0
        else:
            self.height = float(args[1])
        self.centroid = (0.0, 0.0)

    def perim(self):
        return 2 * (self.length + self.height)

    def area(self):
        return self.length * self.height

    def update(self):
        self.centroid = (self.length/2.0, self.height/2.0)
```

Example Program 3 (6)

```
class Circle:
    """
    A circle centered at origin
    """

    # radius        -> radius
    # diam()        -> diameter
    # circum()      -> circumference
    # area()        -> area

    def __init__(self, *args):
        if (len(args) < 1):
            self.radius = 0.0
        else:
            self.radius = float(args[0])

    def diam(self):
        return (2.0      * self.radius)

    def circum(self):
        return (2.0 * PI * self.radius)

    def area(self):
        return (      PI * self.radius * self.radius)
```

Example Program 3 (7)

```
# get help on the functions in this file
help(Example3)
help(Box)
help(Circle)

# run the tests
print('Run Example3:\n')
Example3()

print('\nDone')
```

Getting More Information

- There are lots of online Python tutorials
- One paticularly useful one is at `http://www.tutorialspoint.com/python`