#### Review

### by Deborah R. Fowler

updated from python 2 to 3 ie. print() versus print



#### **KEY CONCEPTS**

- variables
- truth statements
- looping functions
- I/O
  - lists
    - classes/objects



# Today:

- E3 strings lists
- Review

```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [N
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more inf
>>> mylist = ["cavalier", "dog", "spaniel"]
>>> print(mylist[1])
dog
>>> for item in mylist:
        print(item)
```

cavalier dog spaniel >>>

Recall that lists can contain strings

randomNumbers1.py - C:\Users\Deborah\Desktop\SRCWebSite\PythonResources\programmingPDF\Class10-Review\random	浸 IDLE Shell 3.10.2
File Edit Format Run Options Window Help	File Edit Shell Debug Options Window
<pre>1 # Suppose we want to make strings that contain dog, cat and + and 2 # Create a function called makeList 3 # Start by defining our list 4 # Step 1 - make a list which contains dog, cat and + and - 5 6 import random 7 8 def makeList(): 9 mylist = ["cat","dog"] 10 operator = [" + "," - "] 11 result = random.choice(mylist) 12 print (result)</pre>	<pre>&gt;&gt;&gt; Python 3.10.2 (tags/v3 (AMD64)] on win32 Type "help", "copyrigh &gt;&gt;&gt; = RESTART: C:\Users\Del Class10-Review\randomN cat cat cat dog dog &gt;&gt;&gt;&gt;</pre>
for i in range(0,5): test = makeList() 16 17 18 19 20 Introducing random	

```
IDLE Shell 3.10.2
👔 randomNumbers2.py - C:\Users\Deborah\Desktop\SRCWebSite\PythonResources\programmingPDF\Class10-Review\randomNu...
                                                                                  File Edit Shell Debug Option
File Edit Format Run Options Window Help
                                                                                       Python 3.10.2 (
  # Suppose we want to make strings that contain dog, cat and + and -
   # Create a function called makeList
                                                                                       t (AMD64)] on w
                                                                                       Type "help", "c
 3
  # Start by defining our list
                                                                                  >>>
  # Step 2 - take those random values and combine them
                                                                                       = RESTART: C:\U
 5
                                                                                       \Class10-Review
  import random
 6
                                                                                       cat +
                                                                                       dog +
 8
  def makeList():
                                                                                       dog -
 9
       mylist = ["dog", "cat"]
                                                                                       dog +
10
       operator = [" + "," - "]
11
                                                                                       cat +
       resultName = random.choice(mylist)
                                                                                  >>>
12
       resultOp = random.choice(operator)
13
       result = resultName + resultOp
14
       # print (result)
15
       return result
16
17
  for i in range(0,5):
18
       test = makeList()
19
       print (test)
20
21
22
23
                               Combine the random choices
24
```

🕞 randomNumbers3.py - C:\Users\Deborah\Desktop\SRCWebSite\PythonResources\programmingPDF\Class10-Review\randomNu	DLE Shell 3.10.2
File Edit Format Run Options Window Help	File Edit Shell Debug Options Window Help
<pre>1 # Suppose we want to make strings that contain dog, cat and + and - 2 # Create a function called makeList 3 # Start by defining our list 4 # Step 3 - what if we continued to combine them?</pre>	<pre>Python 3.10.2 (tags/v3.10.2:a58ebc (AMD64)] on win32 Type "help", "copyright", "credits &gt;&gt;&gt;</pre>
5 6 import random 7	<pre>= RESTART: C:\Users\Deborah\Deskto Class10-Review\randomNumberSteps\r dog -</pre>
8 def makeList(): 9 mylist = ["dog","cat"] 10 operator = [" + "," - "]	dog - dog + dog - dog +
<pre>11 resultName = random.choice(mylist) 12 resultOp = random.choice(operator) 13 result = resultName = resultOn</pre>	dog + dog - dog + dog +
13 result = resultName + resultOp 14 return result 15	dog – dog + dog + cat + dog –
<pre>16 finalResult="" 17 for i in range(0,5): 18</pre>	dog - dog + dog + cat + dog - dog - dog + dog + cat + dog -
<pre>10 test = makeList() 19 print(test) 20 finalResult = finalResult + test</pre>	
<pre>21 print(finalResult) 22 print (finalResult) 22</pre>	
$\begin{array}{c} 23\\ 24\\ 25 \end{array}$	

🕞 randomNumbers4.py - C:\Users\Deborah\Desktop\SRCWebSite\PythonResources\programmingPDF\Class10-Review\randomNu — 🛛	🛛 🔂 IC	DLE Shell 3.10.2
File Edit Format Run Options Window Help	File	Edit Shell Debug Options Window Help
<pre>1 # Suppose we want to make strings that contain dog, cat and + and - 2 # Create a function called makeList 3 # Start by defining our list 4 # Step 4 - let's handle the end of the string and put this into a function</pre>	>>>	Python 3.10.2 (tags/v3.10.2:a58ebcc, C AMD64)] on win32 Type "help", "copyright", "credits" on
5 6 import random 7		<pre>= RESTART: C:\Users\Deborah\Desktop\SF ass10-Review\randomNumberSteps\randomN cat - dog - cat - cat - cat</pre>
<pre>8 def makeList(last): 9 mylist = ["dog", "cat"] 10 operator = [" + "," - "] 11 resultName = random.choice(mylist)</pre>	,,,,	
<pre>12 resultOp = random.choice(operator) 13 if last != 1: 14 result = resultName + resultOp</pre>		
<pre>15 else: 16 result = resultName</pre>		
17 return result 18		
<pre>19 def finalList(): 20  finalResult="" 21  last = 0</pre>		
<pre>22 for i in range(0,5): 23 if i == 5-1:</pre>		
<pre>24 last = 1 25 test = makeList(last) 26 finalRegult = finalRegult + test</pre>		
<pre>26 Iffalkesult = Iffalkesult + test 27 print (finalResult) 28</pre>		
29 finalList()		
End case and make it a	a fu	Inction

```
IDLE Shell 3.10.2
🙀 randomNumbers5.py - C:\Users\Deborah\Desktop\SRCWebSite\PythonResources\programmingPDF\Class10-Review\randomNu...
File Edit Format Run Options Window Help
                                                                                      File Edit Shell Debug Options Window Help
1 # Suppose we want to make strings that contain dog, cat and + and -
                                                                                          Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 20
2 # Create a function called makeList
                                                                                          t (AMD64)] on win32
3 # Start by defining our list
                                                                                          Type "help", "copyright", "credits" or "licens
4 # Step 5 - What is dog and cat were functions?
                                                                                     >>>
5
                                                                                         = RESTART: C:\Users\Deborah\Desktop\SRCWebSite
6
  import random
                                                                                          \Class10-Review\randomNumberSteps\randomNumber
 7
                                                                                          cat() + cat() - cat() - dog() + dog()
8 def makeList(last):
                                                                                     >>>
      mylist = ["dog()", "cat()"]
9
10
      operator = [" + "," - "]
11
      resultName = random.choice(mylist)
12
      resultOp = random.choice(operator)
13
      if last != 1:
14
           result = resultName + resultOp
15
       else:
16
           result = resultName
17
       return result
18
19
   def finalList(num):
20
       finalResult=""
                                                         What if dog and cat were functions?
21
      last = 0
22
      for i in range(0,num):
23
           if i == num - 1:
               last = 1
24
25
           test = makeList(last)
26
           finalResult = finalResult + test
27
      print (finalResult)
28
29 finalList(5)
30
31
```

32 33

```
📄 randomNumbers6.py - C:\Users\Deborah\Desktop\SRCWebSite\PythonResources\programmingPDF\Class10-Review\randomNu... 🦳
                                                                                         IDLE Shell 3.10.2
                                                                                File Edit Format Run Options Window Help
                                                                                           Edit Shell Debug Options Window Help
1 # Suppose we want to make strings that contain dog, cat and + and -
                                                                                           Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [N
2 # Create a function called makeList
                                                                                           t (AMD64)] on win32
 3 # Start by defining our list
                                                                                           Type "help", "copyright", "credits" or "license()" for more in
 4 # Step 6 - What if we evaluated the result? Well minus doesn't work with strings >>
5 # for now just change that to a plus
                                                                                           = RESTART: C:\Users\Deborah\Desktop\SRCWebSite\PythonResources
                                                                                           \Class10-Review\randomNumberSteps\randomNumbers6.pv
                                                                                           Resulting string is: cat() + dog() + cat() + dog() + dog()
7 import random
8
                                                                                            Result:
9 def makeList(last):
10
      mylist = ["dog()", "cat()"]
                                                                                            meow arf arf meow arf arf arf arf
11
      operator = [" + "," + "]
12
      resultName = random.choice(mylist)
13
      resultOp = random.choice(operator)
14
      if last != 1:
15
          result = resultName + resultOp
16
      else:
17
           result = resultName
                                                                                                   In python eval takes on a string and
18
      return result
19
                                                                                                   returns the evaluation of the string
  def dog():
      return " arf arf "
X3 def cat():
                                                                                                   So eval("1 + 1") would be 2
      return " meow "
26 def finalList(num):
      finalResult=""
27
28
      last = 0
29
      for i in range(0,num):
30
          if i == num - 1:
31
              last = 1
32
          test = makeList(last)
          finalResult = finalResult + test
33
34
      print ("Resulting string is: ", finalResult, "\n", "Result: ", "\n")
35
      print (eval(finalResult))
                                                   Introducing
36
                                                                               eval
37 finalList(5)
38
                                                                                Ln: 30 Col: 19
```

```
IDLE Shell 3.10.2
                                                                                                                                                        🙀 randomNumbers7.py - C:\Users\Deborah\Desktop\SRCWebSite\PythonResources\programmingPDF\Class10-Review\randomNu... —
                                                                                  File Edit Shell Debug Options Window Help
File Edit Format Run Options Window Help
                                                                                      Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1
1 # Suppose we want to make strings that contain dog, cat and + and -
2 # Create a function called makeList
                                                                                      929 64 bit (AMD64)] on win32
                                                                                     Type "help", "copyright", "credits" or "license()" for more informat
 3 # Start by defining our list
 4 # Step 7 - what if we added variables
                                                                                     ion.
                                                                                 >>>
                                                                                     = RESTART: C:\Users\Deborah\Desktop\SRCWebSite\PythonResources\progr
 6 import random
                                                                                     ammingPDF\Class10-Review\randomNumberSteps\randomNumbers7.py
                                                                                     Resulting string is: dog( robin ) + dog( robin ) + cat( kermit ) +
 8 def makeList(last):
                                                                                     dog( robin ) + cat( kermit )
       mylist = ["dog( ","cat( "]
 9
                                                                                      Result:
10
      variables = ["kermit", "robin"]
11
       operator = [" + "," + "]
12
                                                                                      bark bark meow bark meow
       resultName = random.choice(mylist)
      resultVariable = random.choice(variables)
                                                                                 >>>
13
14
       resultOp = random.choice(operator)
15
       if last != 1:
16
           result = resultName + resultVariable + " )" + resultOp
17
      else:
18
           result = resultName + resultVariable + " )"
19
       return result
   def dog(parm):
       if parm == 10:
          return " arf arf "
      else:
           return " bark "
   def cat(parm):
       if parm == 10:
           return " meow "
      else:
           return " hiss "
33 def finalList(num):
       finalResult=""
34
35
      last = 0
36
       for i in range(0,num):
37
          if i == num - 1:
38
               last = 1
39
           test = makeList(last)
40
           finalResult = finalResult + test
      print ("Resulting string is: ", finalResult, "\n", "Result: ", "\n")
41
42
       kermit = 10
43
       robin = 15
                                                                  Adding variables
44
      print (eval(finalResult))
45
46 finalList(5)
47
48
```

浸 randomNumbers8.py - C:\Users\Deborah\Desktop\SRCWebSite\PythonResources\programmingPDF\Class10-Review\randomNumberSteps\randomNumbers8.py (3.10.2) File Edit Format Run Options Window Help 1 # Suppose we want to make strings that contain dog, cat and + and -2 # Create a function called makeList 3 # Start by defining our list 4 # Step 8 - What if we replaced dog and cat with sin and cos? 6 import random 7 from math import sin, cos, pi 9 # TIP: sin and cos are in radians not in degrees - could convert \*pi/180 however interesting patterns results with just \* pi 10 print ("result of 90 should be one however \* pi give interesting patterns", sin(90 \* pi/180)) 12 def makeList(last): 13 mylist = ["sin( ","cos( "] 14 variables = ["x", "y"] IDLE Shell 3.10.2 – 🗆 🗙 15 operator = [" + "," + "] File Edit Shell Debug Options Window Help resultName = random.choice(mylist) Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bi 17 resultVariable = random.choice(variables) 18 t (AMD64)] on win32 resultOp = random.choice(operator) Type "help", "copyright", "credits" or "license()" for more information. 19 if last != 1: >>> result = resultName + resultVariable + ")" + resultOp 21 22 = RESTART: C:\Users\Deborah\Desktop\SRCWebSite\PythonResources\programmingPDF else: \Class10-Review\randomNumberSteps\randomNumbers8.py result = resultName + resultVariable + " )" 23 result of 90 should be one however \* pi give interesting patterns 1.0 return result 24 Resulting string is: sin(y) + cos(y) + cos(x) + cos(x) + sin(y)25 def dog(parm): Result: 26 if parm == 10: 27 -1.1372552906974924return " arf arf " 28 >>> else: 29 return " bark " 31 def cat(parm): 32 if parm == 10: return " meow " What if cat and dog were 34 else: 35 return " hiss " 36 37 def finalList(num): replaced by sin and cos? 38 finalResult="" 39 last = 040 for i in range(0,num): 41 if i == num - 1: 42 last = 143 test = makeList(last) 44 finalResult = finalResult + test 45 print ("Resulting string is: ", finalResult, "\n", "Result: ", "\n") 46 x = 1047 v = 1548 print (eval(finalResult)) 49

50 finalList(5)

51 52

#### In-class Exercise

- Create a string that is comprised of the words cat, dog and operators + and –
- Next try a cat and dog function,
- Remember to change to + and + for cat and dog
- Evaluate your string

Now change it to two other animals and hand in to Dailies/FirstnameAnimalSounds



## Recursion

🕞 recursionExample0.py - C:/Users/Deborah/Desktop/SRCWebSite/PythonResources/programmingPDF/Class10-Review/recursion —	🕞 I	DLE Shell 3.10.2
File Edit Format Run Options Window Help	File	Edit Shell Debug Options Window Help
<pre>1 # Remember to include title/description/author/date in your top block com 2 # 3 # Recursion Example 4 # 5 # Author: Deborah R. Fowler</pre>	T	Python 3.10.2 (tags/v3.10.2:a58ebcc, 29 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" on.
6 # Date: Oct 8 2018	///	= RESTART: C:/Users/Deborah/Deskton/
7 # 8 # Description: an example of recursion 9		mmingPDF/Class10-Review/recursionExa Result of recursion 1 Result of recursion 2
10		Result of recursion 3
11	>>>	
12 # A simple example of recursion - similar to the factorial		
13 # example found at https://www.python-course.eu/recursive_functions.php		
15 def testrecursion(x.currentLevel.maxLevel):		
16 if (currentLevel == maxLevel):		
17 return x;		
18 else:		
<pre>19 return testrecursion(x+1,currentLevel+1,maxLevel) 20</pre>		
21 result = testrecursion(1,0,0)		
<pre>22 print("Result of recursion", result) 23</pre>		
24 result = testrecursion(1,0,1)		
<pre>25 print("Result of recursion", result) 26</pre>		
27 result = testrecursion(1,0,2)		
28 print("Result of recursion", result)		
29		

```
🙀 recursionExample.py - C:\Users\Deborah\Desktop\SRCWebSite\PythonResources\programmingPDF\Class10-Review\recursionEx... 🦳 🗌
                                                                                                                             \sim
```

File Edit Format Run Options Window Help

```
1 # Remember to include title/description/author/date in your top block comment
2 #
 3 # Recursion Example
  #
 4
  # Author: Deborah R. Fowler
 6 # Date: Oct 8 2018
  # Description: an example of recursion
 8
 9
10
11
12 # A simple example of recursion - similar to the factorial
13 # example found at https://www.python-course.eu/recursive functions.php
14
15 def testrecursion(x, currentLevel, maxLevel):
       print("I'm entering", x, currentLevel, maxLevel)
16
17
       if (currentLevel == maxLevel):
           print("I'm in the if",x,currentLevel, maxLevel)
18
19
           return x;
20
       else:
21
           print("I'm in the else", x, currentLevel, maxLevel)
22
           return testrecursion(x+1,currentLevel+1,maxLevel)
23
24  ##result = testrecursion(1,0,0)
                                                IDLE Shell 3.10.2
25 ##print("Result of recursion", result)
                                                File Edit Shell Debug Options Window Help
26
27 ##result = testrecursion(1,0,1)
                                                    Python 3.10.2 (tags/v3.10.2:a58ebc
28 ##print("Result of recursion", result)
                                                    4 bit (AMD64)] on win32
29 ##
                                                    Type "help", "copyright", "credits
30 result = testrecursion(1,0,2)
                                                >>>
31 print("Result of recursion", result)
                                                    = RESTART: C:\Users\Deborah\Deskto
32
                                                    gPDF\Class10-Review\recursionExamp
                                                    I'm entering 1 0 2
                                                    I'm in the else 1 0 2
                                                    I'm entering 2 1 2
                                                    I'm in the else 2 1 2
                                                    I'm entering 3 2 2
                                                    I'm in the if 3 2 2
                                                    Result of recursion 3
```

>>>

Recursion is when a function calls itself

#### This can be very useful

File Edit Format Run Options Window Help 1 # Remember to include title/description/author/date in your top block comment 2 # 3 # From http://interactivepython.org/runestone/static/pythonds/Recursion/pythondsintro-VisualizingRecursion.html 6 import turtle 8 myTurtle = turtle.Turtle() 9 myWin = turtle.Screen() 10 Python Turtle Graphics  $\times$ 11 def drawSpiral(myTurtle, lineLen): \_ if lineLen > 10: 12 myTurtle.forward(lineLen) 13 myTurtle.right(90) 14 15 drawSpiral(myTurtle,lineLen-5) 16 17 myTurtle.speed(0) 18 drawSpiral(myTurtle,100) 19 myWin.exitonclick() 20 21 22 23

🔀 recursionExampleTurtleTree.py - D:\SCAD\_ALL\BuildWebSite\SRCWebSite\ClassHandouts\VSFX160-Handouts\VSFX160-PRIVATE\_SRC\RandomArt\recur... — 🛛 🛛 🔀

File Edit Format Run Options Window Help



```
🌛 recursionKoch.py - D:\SCAD_ALL\BuildWebSite\SRCWebSite\ClassHandouts\VSFX160-Hando... 🚽 🚽
                                                                           File Edit Format Run Options Window Help
# From http://openbookproject.net/thinkcs/python/english3e/recursion.html
# 18.1
import turtle
def koch(t, order, size):
    .....
      Make turtle t draw a Koch fractal of 'order' and 'size'.
       Leave the turtle facing the same direction.
    .....
                            # The base case is just a straight line
    if order == 0:
        t.forward(size)
    else:
        koch(t, order-1, size/3) # Go 1/3 of the way
        t.left(60)
        koch(t, order-1, size/3)
        t.right(120)
        koch(t, order-1, size/3)
        t.left(60)
        koch(t, order-1, size/3)
turtle.pu()
turtle.setpos(-400,0)
turtle.pd()
koch(turtle,3,800)
turtle.exitonclick()
```

```
76 Python Turtle Graphics
```

Homework:

- Dailies Exercise due Monday
- Work on the Random Art (E3) Exercise

#### **KEY CONCEPTS**

- variables
- truth statements
- looping functions
- I/O
  - lists
    - classes/objects

#### **KEY CONCEPTS**

Continue to keep up with your reading in the online textbook

If any of these key concepts are not clear – see me!