

Chapter 2

Introduction to Python

2.1 Vocabulary

2.2 Introduction

2.3 Summary

2.4 Example Python Problems

Example 1:

Programming is essentially simplifying steps of a task to an alien that doesn't understand how humans think. You need to be specific, detailed, and consistent in the format. The computer has a few basic commands it can do, so you need to be creative about how you'll lay out the steps. With this in mind how would you explain to a computer how to get pizza from Commons?

Solution (outline):

- You would first need to give the computer directions to Commons
- Scan its ID to enter Commons from the west facing door
- Proceed to the first scanning station, scan its ID, and wait in line
- Proceed to the pizza station which is on the left side of the building
- Look at the types of pizza there are and decide which one it wants
- Ask the Commons employee for the pizza it wants
- Take its box and follow the signs to exit
- Scan out and exit through the east facing doors

Example 2:

The `type()` function tells you what type of data a certain variable is. Predict what Python will return for the following commands:

- a.) `type("hello world")`
- b.) `type(747)`
- c.) `type(3.14)`

Solution:

- a.) `string`
- b.) `integer`
- c.) `float`

Example 3:

Python is a simple programming language and doesn't naturally come with all of the functions you might need. However, there is no reason to worry. Because Python is so ubiquitous, you can find downloadable packages of code to perform things you might otherwise sink hours into programming. One of these packages is `numpy` which performs many mathematical calculations that Python doesn't come equipped with. Here are some examples of `numpy` functions:

```
# exponential
numpy.exp(x)

# square root
numpy.sqrt(x)

# natural log
numpy.log(x)
```

In order to use these functions you must first import the package into your notebook which can simply be done with the `import` function:

```
import numpy
```

Example 4: Write code to do the following and print them:

- a.) e^8
- b.) $\sqrt{55 + 31}$
- c.) $\ln(664/41)$

Solution:

```
# part a
print(numpy.exp(8))

# part b
print(numpy.sqrt(55+31))

# part c
print(numpy.log(664/41))
```

Example 5: "If" statements are rather straightforward. You tell Python what

to do *if* a particular circumstance is true. For example:

```
# pick a random integer between 1 and 10
x = np.random.randint(10)

# print "okie dokie" for if x is small, but print "nokie dokie"
# for larger values
if 4>x:
    print("okie dokie")
else:
    print("nokie dokie")
```

In this example we are using the `numpy.random.randint` function to generate a random number between 0-9 and storing it in the variable `x`. This is because the `numpy.random.randint` function begins counting at 0 and is not inclusive of the last digit. The `if` statement then tells Python that if the randomly generated number is less than four print `okie dokie` and if not print `nokie dokie`. You can run this code multiple times and see the changes.

Example 6: Now try it yourself. Imagine you’re playing the lottery and your numbers are 3, 22, 55, and 31 and numbers are selected between 0 and 150, you win if any of your numbers are selected. Write an “`if`” statement that returns “I’m loaded!!!” if your numbers are not selected, print “better luck next time”.

- Hint: the “`==`” symbols are used to show equivalency.
- “`Or`” is a function that can be used in conjunction with “`if`” statements.

Solution:

```
# print a random integer between 1 and 56
x = numpy.random.randint(57)

# check to see if x is a winning number
if x==3 or x==22 or x==55 or x==31:
    print("I'm loaded!")
else:
    print("Better luck next time!")

# check what number was chosen
print(x)
```

Example 7: Imagine you are cleaning up at the end of your birthday party. You are tired and hungry from a long day of having fun. You are wondering how many slices of pizza are left in each of the 5 pizza boxes, and you decide to write an algorithm that will look through each pizza box and count the number of remaining slices.

Follow the comments below to understand what each line of this code does:

```
# We are importing the package numpy which has built in functions  
# that will make coding easier and more efficient.  
import numpy  
  
# We are defining the total number of pizza boxes that we need  
# to examine.  
totalBoxes = 5  
  
# Start counting the number of pizza slices. Since we haven't  
# started counting, we will start at zero.  
count = 0  
  
# Here we are initializing the for loop. We are telling Python  
# to look at each box in the range of totalBoxes.  
for box in range(totalBoxes):  
    # We've indented here because we are working inside the for loop.  
  
    # The numpy.random.randint() function is selecting a random  
    # number between 0 and 16 (we use 17 because the numpy random  
    # function is not inclusive of the last digit). This will  
    # represent the number of slices left in this box.  
    slices = numpy.random.randint(0, high=17)  
  
    # We are telling Python that if there is at least one slice in  
    # the box, we should increase the count and print the number of  
    # slices found.  
    if slices > 0:  
        count += slices  
        print('I have found {} slices!'.format(count))  
  
    # "Else" tells Python what to do if there is no pizza in this box.  
    else:  
  
        #Print is indented inside the else statement because you want to  
        # print ("Noooooooo!!!") if the else statement needs to be used.  
  
        print("Noooooooo!!!")
```

2.5 Conceptual Exercises for Learning Python

1. How does Python read code?
2. Compare and contrast the function of brackets and parentheses.
3. How is a tuple different than a list, and given their respective scenarios

what characters should be used?

4. Why is indentation important when using writing code in Python, and how can it affect your cell outcome?
5. Describe the four most common data types (int, float, bool, str) that variables can be assigned to and describe each one.

2.6 Coding Exercises for Learning Python

1. Define two variables, one called `varOne` that represents the value 20 and one called `varTwo` that represents the value 5:
 - a. Creating a new variable called `addVar`, write a code that will add `varOne` and `varTwo`.
 - b. Creating a new variable called `divVar`, write a code that will divide `varOne` and `varTwo`.
 - c. Finally, multiply your two new values (from parts a + b), and assign this to the variable `multVar`. Hint: use basic addition, division, and multiplication commands rather than making your own function.
2. Using the variable name `L1`:
 - a. Create a list of five different food items and print the list.
 - b. Write a line of Python code that will print only the 3rd and 5th food in your list from part a.

Using variable name `L2`:

- c. Create a second list with three sport names and print the list.
- d. Write a line of Python code that will sort your list from part c alphabetically.

3. Create a vector from 5 to 15 (non-inclusive) in increments of 2. Then write a line of code that will print the square of each value in the vector.
4. Create a matrix of zeros with 3 rows and 6 columns. Then, create a 2x4 matrix where the first row lists ice cream flavors and the second row lists soda flavors.
5. Write a for loop that will start at the lowest temperature you experienced this week and end at the highest temperature you experienced (in one degree increments). Inside your loop, use an `if...else` statement that prints whether each temperature is even or odd.

