Week 6: Conditionals, Iteration

Data 8 Tutoring

# 1 Conditional Statements and Iteration

## Key Concepts

**Conditional Statements**

We can use conditional statements to write code and create functions that perform different operations based on certain conditions. As a reference, here is how conditional statements work:

if x > 10:

Do something

elif x > 5:

Do something

else:

Do something

**Iteration**

For loops in Python can potentially allow us to do two different operations. First, they allow us to iterate through arrays, manipulating each element as we wish. Alternatively, we can use for loops to repeat lines of code many times. Examples of how for loops can be used are below.

for item in some\_array:

print(item)

or

for i in np.arange(1000):

print(“Hello”)

## Practice Problems

**Question 1.** Examine the function, then answer the questions below. It has been written with a purposely vague name and arguments!

def mystery\_function(x):

if (x > 0):

return “Positive”

elif (x < 0):

return “Negative”

else:

return “Neither”

**1.1** What would mystery\_function(10) return?

**1.2** What does mystery\_function(-1) return?

**1.3** What does mystery\_function(0) return?

**Question 2.** The for loop statement below stores the length of each name in names in a new array called lengths.

lengths = make\_array()

names = make\_array(‘Bob’, ‘Sarah’, ‘Michael’, ‘Sam’)

for name in names:

lengths = np.append(lengths, len(name))

**2.1** For each iteration below, fill in the value of name as well as what lengths looks like.

Iteration 1: name = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ , lengths = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Iteration 2: name = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ , lengths = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Iteration 3: name = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ , lengths = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Iteration 4: name = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ , lengths = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2.2** Now, let’s say that instead of storing lengths, we want to store the name as long as the length of the name is greater than 4. Fill in the following for loop statement such that longer contains these names.

longer = make\_array()

for name in \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

if \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

longer = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2.3** What names would longer contain after the for loop executes?

**2.4** Finally, look at this last for loop below. What values does i take on throughout? How is i used as compared to the way name is used in the previous for loops?

counter = 0

for i in np.arange(1000):

counter = counter + 1

**Question 3.** Suppose you have an array called salaries, containing the salary information of 5 individuals. You would like to determine what percentage of the total salaries each individual's salary comprises. You want to output an array, proportion where the ith element of proportion corresponds to what percentage of the total salary salary.item(i) is.

For example, if salaries was equal to an array [1,2,3,1,3], then proportion.item(0) would be 0.1.

**3.1** Your friend writes some code, but it doesn’t work! Find the error that your friend made. What would the code output if executed as is? How would you fix it?

salaries = make\_array(25, 50, 100, 25, 100)

total = sum(salaries)

for salary in salaries:

proportion = make\_array()

percentage = salary/total

proportion = np.append(proportion, percentage)

**3.2** You fix the error described above, but in doing so, break something else. Again, find the error in the code below. What would the code output if executed as is? How would you fix it?

salaries = make\_array(25, 50, 100, 25, 100)

total = sum(salaries)

proportion = make\_array()

for salary in salaries:

percentage = salary/total

np.append(proportion, percentage)