

Description:

Your role as an assistant computer programmer means that you must learn to be proficient in many types of programming and coding. For this assignment, you will have to continue learning Python by applying and extending your prior knowledge to understanding for loops, while loops and the specific differences between them. Because you are in training, you will be working on several modules and continue to build upon your skills. You will work on various skills in Python and submit them to the head computer programmer to make sure your codes are both sufficient and efficient. For this task, you will learn and utilize for loops and while loops in Python by designing your own program.

Students will be able to:

- Differentiate between for loops and while loops.
- Differentiate between mathematical symbols and Python relational operators.
- Debug given code that has only logic errors.
- Parse given Python codes with loops and apply all prior lesson goals to generate an original code that utilizes variable assignments, both built-in and user-defined functions, and both for and while loops.
- Apply logic and problem solving skills to design a program.

Students will understand:

Python is a widely known, high-level coding language. Prior to this lesson, students mastered declaring variables, utilizing built-in functions and implementing user defined functions. Also, students learned about the bool variable type and how to breakdown a given code with conditional statements. Through this lesson, students will understand how to breakdown a given code with both for loops and while loops. Students will continue learning how to design their own program by utilizing for loops, while loops and user-defined functions.

Key Definitions & Concepts: [1]

- **Iteration:** repeated execution of a set of statements using either a recursive function call or a loop
- **Indefinite loop:** a loop in which the terminating condition is never satisfied

Standards: [Copied from: 2]

1B.AP.10: Create programs that include sequences, events, loops and conditionals.

2.AP.13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

Background Information**Prior Knowledge:**

- Variable assignment

- Expression evaluation
- Defining functions
- Utilizing math module
- Conditional statement execution

Science Practices: [Copied from: 3]

- Constructing explanations and designing solutions
- Obtaining, evaluating and communicating information

Core Ideas: [Copied from: 4]

- Developing possible solutions
- Optimizing the design solution

Cross Cutting Concepts: [Copied from: 5]

- Patterns
- Structure and function

Possible Preconceptions/Misconceptions:

During initialization of a variable within a for loop or a while loop, most begin as 0 instead of 1. Without prior knowledge or practice in computer science. Hence, expect students to struggle in correcting the semantic error for Program 1 within the benchmark of the lesson. Also, students may struggle to conceptualize the difference between a for loop and a while loop. Hence, students frequently experience runtime errors when incorrectly coding with while loops. Reference the answer key for the engagement and the benchmark lesson for specific details in addressing this misconception.

Lesson Plan - 5E(+) Model

Engage: [6]

Engage the students with an anchoring event: a video clip about using loops in Python programming. This clip talks about how both for loops and while loops are used in computer programming by using a the same problem statement for each loop, then shows the students how to write the loops, and concludes with proper Python syntax and some basic explanation. The purpose of this clip is to get students thinking about the importance of computer programming and to engage them in a class discussion of “what is the difference between for loops and while loops, and how are they used?.” The following is a link to the website: [Python Loops](#). While the students are watching the video, they should be filling in the *Loops* worksheet independently. This should take less than 5 minutes to complete.

Explore:

Part I: Introduction

The students will be engaged in a whole class discussion after completing the worksheet, *Loops*, about how that video clip connects with the lesson plan’s goal and objectives. Teachers will use students’ responses to lead into the exploration’s benchmark of the lesson by reviewing their answers and correcting any misconceptions as necessary. This section should take less than 10 minutes to complete.

Part II: Benchmark Lesson: Interacting with For and While Loops

The instructor will hand out the worksheets titled *Investment Programs* and give each pair of students access to tablets or computers that run IDLE and the latest update of Python. The students will upload or copy the code from the *Investment Program Script* handout into the script mode of IDLE and complete the worksheet in pairs. By interacting with the given code, students will be able to answer the questions about what the for loops and the while loops are producing as an output. They will also be able to correct the syntax errors in the given code. The purpose of this worksheet is to introduce loops in Python and to emphasize the importance of syntax when writing code. The worksheet forces students to engage with loops, which serves as a precursor to understanding the difference between utilizing a for loop versus a while loop and to understanding the purpose of the syntax associated with each loop. Students will be engaged in a whole class discussion to correct and elaborate on the questions in the worksheet titled, *Investment Programs*. Here, teachers should expect to explain Python syntax rules when writing loops. Teachers should also expect to explain the difference between the keywords “for” and “while.” Also, teachers should use this discussion to correct students’ responses on their worksheets and address any misconceptions as necessary. The students should be given 10 minutes to complete the worksheet in pairs, and the discussion should last no longer than 10 additional minutes. Hence, this section of the lesson should last less than 20 minutes.

Part III: Investigation Lesson: Programming

Students will work individually on the worksheet *Investigation and Programming*. This worksheet provides the students with the RAFT of the unit, which is written in the description section of this lesson plan. This worksheet contains a prompt that asks students to design their own program from scratch. This activity has them utilize built-in functions, variable assignments, user-defined functions and both for and while loops. Since this curriculum is geared towards understanding the necessities of coding a SeaGlide, the students are explicitly taught about the above programming concepts. Hence, the worksheet does not provide the students with a skeleton of the code required to fully answer the prompt. Rather, the students are expected to have learned everything that is necessary for them to successfully complete this coding challenge individually. The students are responsible for recalling proper utilization of built-in functions, user-defined functions, and of variable assignments. Students will have to utilize this prior knowledge and extend it into completing the given prompt by creating an individual code that utilizes a while loop that calls user-defined functions. The purpose of this part of the exploration is to have students create a Python code from scratch so they can work through a real-world example by developing the skills they have learned from the beginning of this curriculum. The remaining class time should be allotted for the students to complete writing this program independently. If the students do not complete the program within the remaining 20 minutes, it is recommended to have the students finish the program for homework and submit their codes prior to the next class meeting.

Explain:

Throughout the exploration, the students will engage in discussions that inquire their understanding and knowledge of the information at-hand. Thus, teachers will be informally asking students to explain all topics and relevant connections throughout the entirety of this lesson. The worksheets include “why” and “how” questions to provide the teacher with their individual explanations and to also give the students an opportunity to verbalize their understandings. The exploration also includes questions that prompt students to defend their responses and explain why they chose a particular solution.

Elaborate:

The elaboration of this lesson is the investigation lesson section. Computer programming is currently a prominent and relevant career path that requires developers to tackle problems by debugging current code and writing new codes with helpful functions and other tools such as conditionals and loops. The *Investigation and Programming* worksheet introduces the students to these programming skills. This student-designed program forces the students to think like a programmer by engaging in a real-world scenario.

Evaluate:

This lesson is designed with having both informal and formal evaluations throughout its entirety. The informal evaluations occur throughout the exploration because of the leading and open-ended questions and class discussions. This allows teachers to gauge surface-level student understanding. By surveying the students during completion of the worksheets, teachers will be able to hear and address any misconceptions or misunderstandings as necessary. The formal evaluation of this lesson is the *Investigation and Programming* worksheet. This individual activity forces students to write their own code from beginning to end by pulling on information learned throughout this lesson and both prior lessons.

Enrich:

This lesson could be extended by introducing how to utilize libraries. Since most functions contain complicated conditionals and nested loops, the next logical step is to have students interact and explore Python code that includes functions with libraries that store and hold values. Further, libraries are most commonly used in industry when programmers realize that certain parts of codes are utilized in a multitude of programs. Hence, programmers tend to write a code containing functions that can be called and utilized in other program codes. Having students to learn libraries primes them to developing necessary skills that programmers use daily.

****All associated documents are attached below****

****Reference *Annotated Bibliography* on the very last page of this packet****

Name: _____ Date: _____

Loops [6]

1. Why are for and while loops used in programming? Use evidence from the video to support your response.

2. For Questions 2-3, use the following terms to fill in the blanks. Some terms may be used once, multiple times or never.

for
while
start

finish
increment
> (greater than)

< (less than)
= (equal to)
!= (not equal to)

Code 1:

```
max_age = 30
_____ age _____ max_age:
    print (age)
    age += 1
```

Code 2:

```
_____ age in range (_____, _____, _____):
    print (age)
```

3. Explain the difference between a for loop and a while loop. Use evidence from the video to support your response.

Name: _____ Date: _____

Investment: Script
Program 1

```
##Title:      Investments
##Description: Python program that returns the number of years
##            required to reach a target balance.

print("This function returns the amount of years it will take to reach a target balance.\n")

def InvestTarget(target, deposit, rate):
    years = 1
    balance = deposit
    while balance <= target:
        years = years + 1
        balance = deposit + balance*(1 + rate)
        print("Year: %d " %years)
        print("Interest: %.3f" %rate)
        print("Deposit: $%.2f" %deposit)
        print("New Balance: $%.2f \n" %balance)

    return years

e = InvestTarget(100000,11000,0.065)
```

Program 2

```
##            Python program that returns the account balance
##            after a given number of years.

print("This function returns an account balance when given a number of years to save.\n")

def InvestYears(years, deposit, rate):
    balance = deposit
    for year in range (1 , years):
        balance = deposit + balance*(1 + rate)
        print("Year: %d " %year)
        print("Interest: %.3f" %rate)
        print("Deposit: $%.2f" %deposit)
        print("New Balance: $%.2f \n" %balance)

    return balance

f = InvestYears(30,11000,0.065)
```

Name: _____ Teacher Copy ONLY _____ Date: _____

Investment: Script (Without Errors)

```
##Loops Explore
##Python Lesson #5

##Title:      Investments
##Description: Python program that returns the number of years
##            required to reach a target balance.

print("This function returns the amount of years it will take to reach a target balance.\n")

def InvestTarget(target,deposit,rate):
    years = 0
    balance = deposit
    while balance <= target:
        years = years + 1
        balance = deposit + balance*(1 + rate)
        print("Year: %d " %years)
        print("Interest: %.3f" %rate)
        print("Deposit: $%.2f" %deposit)
        print("New Balance: $%.2f \n" %balance)

    return years

e = InvestTarget(100000,11000,0.065)

##            Python program that returns the account balance
##            after a given number of years.

print("This function returns an account balance when given a number of years to save.\n")

def InvestYears(years, deposit, rate):
    balance = deposit
    for year in range (1 , years + 1):
        balance = deposit + balance*(1 + rate)
        print("Year: %d " %year)
        print("Interest: %.3f" %rate)
        print("Deposit: $%.2f" %deposit)
        print("New Balance: $%.2f \n" %balance)

    return balance

f = InvestYears(30,11000,0.065)
```


5. Debug and rewrite the code so that it produces the proper outputs. Provide your corrections by writing your code below. (Hint: after you make code changes, run the code by testing at least three inputs). Explain WHY you made those changes.

Name: _____ Date: _____

Investigation and Programming

Directions:

Your role as an assistant computer programmer means that you must learn to be proficient in many types of programming and coding. For this assignment, you will have to continue learning Python by applying and extending your prior knowledge to understanding for loops, while loops and the specific differences between them. Because you are in training, you will be working on several modules and continue to build upon your skills. You will work on various skills in Python and submit them to the head computer programmer to make sure your codes are both sufficient and efficient. For this task, you will learn and utilize for loops and while loops in Python by designing your own program.

Prompt:

A nearby Recreational Center hired you to work on a project team to organize rotating activities to be offered during the weekends. They want your project team to devise a list of activities that the Recreational Center should run and to create a Python program that displays this information to the patrons. Your job is to think of three activities that the Recreational Center will rotate through. They provided the following rules for you to follow:

- At least THREE user-defined functions - one function for each activity that you propose.
- Each user defined function must have at least two print statements including materials the patron will need and a friendly wish (ex: have fun! Or safe travels! etc.)
- Outside of a while loop
 - Initialize your variable (i.e. test expression)
 - Print statements that show the patron the list of activities you are proposing, including which key they need to press in order to select it
 - Ask the patron what activity they would like to engage in
- While loop (since patrons can choose to attend any number of your proposed activities)
- Inside of the while loop
 - Conditional statements to call the functions (i.e. one if block would call your first activity, etc.)
- You must ensure that a patron's selection is valid
- You CANNOT use any of the activities listed in the sample output below
 - Notice: the blue is produced by the program code, and the black is typed by the patron

Sample Output:

Welcome to the Recreational Center! Here's our weekend line-up:

1. Bike Ride (Enter 1).
 2. Triathalon (Enter 2).
 3. Hike (Enter 3).
- Enter q to quit.

What would you like to do?

1

Here's a bicycle.
Be sure to utilize the trail maps.
Ride safe!

What else would you like to do?

2

You will have to complete a 1 mile swim,
then a 12 mile bike, then a 3.2 mile run.
Best of luck!

What else would you like to do?

3

Here's a trail map and some suggested trip plans.
Please complete our survey concluding your hike.
We would love to hear about the quality of our planned trips!
Have fun!

What else would you like to do?

4

Enter a valid response.
What would you like to do?
q
Thank you, good bye!

Name: _____ ANSWER KEY _____ Date: _____

Loops [6]

1. Why are for and while loops used in programming? Use evidence from the video to support your response.

*Student responses will vary but look for something along these lines: for loops and while loops are used in programming when there is a repetition of code in a program. For example, the video states how frustrating it would be to write a code that adds 1 to a value to age every single time.

2. For Questions 2-3, use the following terms to fill in the blanks. Some terms may be used once, multiple times or never.

for
while
start

finish
increment
> (greater than)

< (less than)
= (equal to)
!= (not equal to)

Code 1:

```
max_age = 30
while age ≤ max_age:
    print (age)
    age += 1
```

Code 2:

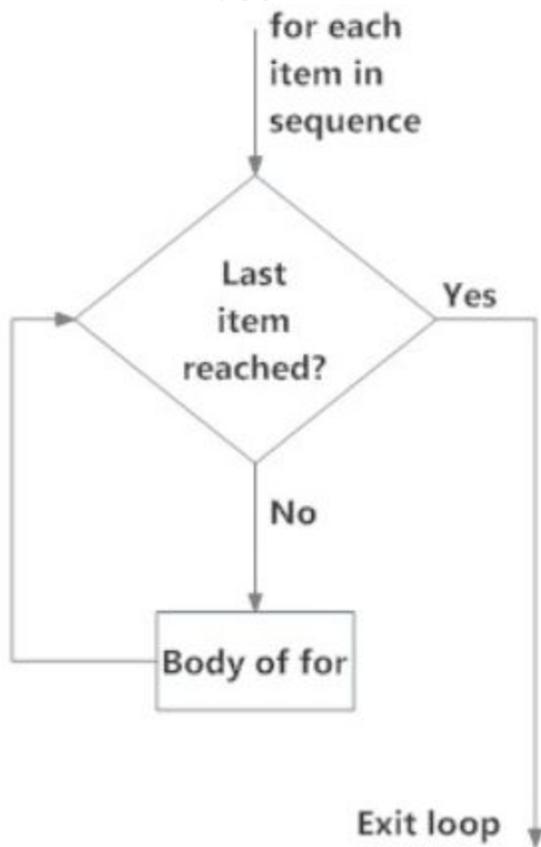
```
for age in range (start, finish, increment):
    print (age)
```

3. Explain the difference between a for loop and a while loop. Use evidence from the video to support your response.

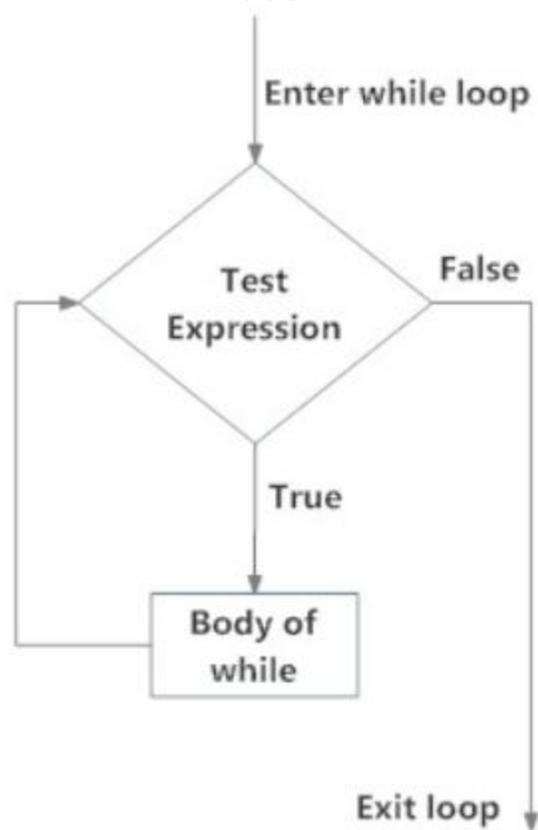
*Student responses will vary but look for something along these lines: for loops are used when the programmer wants to repeat a piece of code a specific number of times. An example of this in the video is when age + 1 is printed 30 times. However, while loops serve most useful when a boolean condition is being tested. Usually, the code is executed as long as a certain condition is met. An example of this in the video is when age + 1 is printed as long as age is less than max age (where max age is equal to 30).

Teacher's Discussion Guide for *Loops*

Flowchart of for loop [7]



Flowchart of while loop [8]



Both for loops and while loops are used for repeating sections of code. Recall the debugging activity from lesson #1. In that lesson, the online debugging activity had a command called "repeat." This command, in block-coding, is the equivalent command for using for and while loops in python. Then what's the difference? For loops are used for repeating sections of code that will run n times. While loops are used for repeating sections of code that will run until a defined condition is no longer met.

Learning loops is one of the most daunting tasks for beginner programmers because there is no defined method of choosing which loop to utilize per code or problem statement. As the engagement video shows, a given problem can be successfully completed by both a for loop and a while loop. So when and how does a programmer determine when to use a for loop rather than a while loop, and vice versa? Experienced programmers know this answer by having years of practice and debugging. For beginner programmers, the best way to learn when to use which loop type is gained through practice and physically seeing the usefulness of each loop in action.

Name: _____ ANSWER KEY _____ Date: _____

Investment Programs [9]

1. Run Program 1. What was the first set of information resulting as an output value?

Year: 2

Interest: 0.065

Deposit: \$11000.00

New Balance: \$22715.00

2. In Program 1, the variable “years” is initialized at 1. In the while loop, “years” is incremented by 1.

- a. How does this affect the output?

The output values begin with year 2 instead of year 1.

- b. What should the variable “years” have been initialized at? (i.e. how would a programmer ensure that the first resulting set of information is for “year 1” instead of “year 2?”) Defend your response.

The variable “years” should have been initialized at 0 because year 0 is the initial start year. Year 1 would be one year after the initialization year.

Discussion Notes: Initialization refers to defining a constant or variable values that are used in the code for executing a computer program. Initialization plays a key role in programming as the variables that are used for writing the code occupy a certain amount of memory in the CPU. If the memory values are not defined by the user at the start of the code’s execution, the CPU will set the variable value to anything that is acceptable in computer programming language, this is usually termed as garbage value (in this case 0). If a garbage value is set for a variable, then the whole logic of the program changes and will result in an incorrect value as the output. Some compilers will not even set a garbage value for the variable and this results to a null value for the variable which can also result in a compile time error. Thus, initialization is very important. If variables are not initialized, then at least the variable values must be overwritten to erase the garbage data and have a valid value for the variable which will ensure that the program gives the desired output.

3. In Program 2, the for loop’s range was set to “(1, years)”.

- a. How does this affect the output?

The output values begin with year 1, but end one year short of the target year.

- b. What should the range of the for loop be changed? Defend your response.

The range of the loop should be changed to “(1, years + 1)”.

4. What purpose do the variables ‘e’ in Program 1 and ‘f’ in Program 2 serve?

“e” is a variable. “e = InvestTarget(100000, 11000, 0.065)” is a variable assignment. This assignment is utilized to call the InvestTarget function. The numbers within the variable assignment (ORDER MATTERS) are the parameters that are passed into the argument of the InvestTarget function. This is to say that 100000 is the “target”; 11000 is the “deposit”, and 0.065 is the “rate”. Notice: parameters are within function calls, and arguments are within the function.

Similarly: "f" is a variable. "f = InvestYears(30, 11000, 0.065)" is a variable assignment. This assignment is utilized to call the InvestYears function. The numbers within the variable assignment (ORDER MATTERS) are the parameters that are passed into the argument of the InvestYears function. This is to say that 30 is the "years"; 11000 is the "deposit", and 0.065 is the "rate". Notice: parameters are within function calls, and arguments are within the function.

5. Debug and rewrite the code so that it produces the proper outputs. Provide your corrections by writing your code below. (Hint: after you make code changes, run the code by testing at least three inputs). Explain WHY you made those changes.

See the handout *Investment: Script (Without Errors)* for the proper code, and also reference the Discussion Notes above for the reasoning.

Name: _____ ANSWER KEY _____ Date: _____

Investigation and Programming

```
##Loops Investigation
##Python Lesson 5

##Title:          Functions, Conditionals and While Loops
##Description:    Python program that prints out specific messages
##               depending the user's chosen activity / activities
##               that they want to plan. The program does not stop
##               running until specified by the user.

##Define the activities offered by the Recreational Center

def BikeRide():
    print("Here's a bicycle.")
    print("Be sure to utilize the trail maps.")
    print("Ride safe!\n")

def Triathalon():
    print("You will have to complete a 1 mile swim, \n" +
          "then a 12 mile bike, then a 3.2 mile run.")
    print("Best of luck!\n")

def Hike():
    print("Here's a trail map and some suggested trip plans.")
    print("Please complete our survey concluding your hike. \n" +
          "We would love to hear about the quality of our planned"
          + " trips!")
    print("Have fun!\n")

##Initialization

choice = ''

##While loop

print("Welcome to the Recreational Center! Here's our weekend line-up:")

print("\n1. Bike Ride (Enter 1).")
print("2. Triathalon (Enter 2).")
print("3. Hike (Enter 3).")
print("Enter q to quit.\n")

choice = input("What would you like to do?\n")
```

```
while choice != 'q':

    if choice == '1':
        BikeRide()
        choice = input("What else would you like to do?\n")
    elif choice == '2':
        Triathalon()
        choice = input("What else would you like to do?\n")
    elif choice == '3':
        Hike()
        choice = input("What else would you like to do?\n")
    elif choice == 'q':
        print("Enjoy the rest of your day!")
    else:
        print("Enter a valid response.")
        choice = input("What would you like to do?\n")

print("Thank you, good bye!")
```

Annotated Bibliography

- [1] Downey, A. (2012). *Think Python: How to Think Like a Computer Scientist* (Vol. 2.0.17). Needham, MA: Green Tea Press. From <http://www.greenteapress.com/thinkpython/thinkpython.pdf>

This online textbook was used for excerption within the Python Loops lesson plan as part of the Computer Science module. This reference aided in the completion of providing definitions for the key concepts and definitions sections and for the associated worksheets. This book was useful because of its layout and completeness. The lesson expands upon the material used from this book as it uses the material in the creation of worksheets and activities that are not provided in the textbook.

- [2] Standards Aligned System. (n.d.). Retrieved from <https://www.pdesas.org/>

This website was used in each lesson in the Computer Science module to select proper Pennsylvania State standards, which are based in Common Core, that each lesson is centered around.

- [3] Nsta. (n.d.). Science and Engineering Practices. Retrieved January 18, 2019, from <https://ngss.nsta.org/PracticesFull.aspx>

This website used in every lesson in the Computer Science module to find Standards for Scientific Practices that are applicable in each lesson.

- [4] Nsta. (n.d.). Disciplinary Core Ideas. Retrieved from <https://ngss.nsta.org/DisciplinaryCoreIdeasTop.aspx>

This website was used in each lesson in the Computer Science module to select appropriate disciplinary core ideas set forth by the NSTA that are at the center of each lesson.

- [5] Nsta. (n.d.). Crosscutting Concepts. Retrieved from <https://ngss.nsta.org/CrosscuttingConceptsFull.aspx>

This website was used in each lesson in the Computer Science module to selecting appropriate crosscutting concepts set forth by the NSTA that apply to each Python lesson.

- [6] Python - For and While Loops. (n.d.). Retrieved from <https://www.youtube.com/watch?v=4gg0JekxsYI>

This video is used as an engagement in the Python Loops lesson plan as part of the Computer Science module. Questions were developed based on this video for students to answer.

- [7] Python for loop. (n.d.) Retrieved from <https://www.programiz.com/python-programming/for-loop>

Python by Programiz is an online tool that is useful for understanding the underlying concepts of computer programming in Python. The discussion notes for the *Loops* answer key were excerpted to be utilized by the instructor during the open class discussion that covers the utilization and syntax of for loops in Python programming.

- [8] Python while loop. (n.d.) Retrieved from <https://www.programiz.com/python-programming/while-loop>

Python by Programiz is an online tool that is useful for understanding the underlying concepts of computer programming in Python. The discussion notes for the *Loops* answer key were excerpted to be utilized by the instructor during the open class discussion that covers the utilization and syntax of while loops in Python programming.

- [9] What is initialization?. (n.d.) Retrieved from <https://www.quora.com/What-is-initialization-Why-is-it-important-in-C-programming>

Quora is reputable forum that is useful (in this case) for understanding the underlying concepts of computer programming in Python in reference to initialization. The discussion notes for the *Loops*

answer key were excerpted to be utilized by the instructor during the open class discussion that covers the understanding of variable initialization in Python programming.