

## **Unit 1: Repetition and Loops**

### **1.1 Loops**

#### Introduction

In this lesson, students learn about loops.

The students will:

- Be able to define and code while loops and loop control variables

Components:

- Video Lesson
- Lesson Practice
- Code Practice

What to Emphasize:

- The loop control variable must have an opportunity to be changed in a while loop in order for the while loop to end. If the control variable never changes, the loop will be infinite.
- The formatting for a while loop is the same as an if statement, where there needs to be a colon after the condition and the code to be executed should be indented underneath.

Additional Notes:

The Viz Mode can be beneficial to help students see how a variable changes throughout a program. This helps them trace through the program and identify where an error may have occurred.

If you'd like to customize the lesson slides, click [here](#).

#### Curricular Standards

This lesson adheres to the following CSTA Standards: 3B-AP-14

### **1.2 Count Variables**

#### Introduction

This lesson builds on students' knowledge of loops, and introduces count variables, which help keep track of how many times a loop has run.

The students will:

- Be able to define initializing
- Be able to define and code the count and sum variables

Components:

- Video Lesson
- Lesson Practice
- Code Practice (2)

What to Emphasize:

- A count variable must be initialized before using it throughout the program. Most other variables are initialized or set when we get the input from the user, but a count variable is not a user input, so we must initialize it ourselves.
- Whenever you are altering a variable, like sum or count, by building off of its current value, the variable appears on both sides of the assignment operator. The alteration to the variable is done on the right side.
- **Examples:**
  - `count = count - 1`
  - `sum = sum + number`
  - `answer = answer / 4`
  - `duplicate = duplicate * 2`

Additional Notes:

Counting and summing are very common in programs, so instead of having to use the syntax mentioned above to edit a variables current value there is something called shorthand notation. We can instead use something like this: `count += 1` or `sum += number`. Whenever a math symbol appears right before the assignment operator, Python knows to take the variable on the left and perform the operation with the variable on the right and then assigns it back to the variable on the left.

If you'd like to customize the lesson slides, click [here](#).

Curricular Standards

This lesson adheres to the following CSTA Standards: 3A-AP-18

### 1.3 Two Ways to End a Loop

Introduction

This lesson introduces two ways to end a loop: using a count variable to run a loop a specific number of times, and using user input.

The students will:

- Be able to end loops using count variables and user input
- Be able to state when to use them

Components:

- Video Lesson
- Lesson Practice
- Code Practice (2)

What to Emphasize:

- Just like when using strings in my control expression, or a user input variable, any variable, including the count variable must change inside the loop in order for the loop to eventually end.
- A while loop can run an indefinite amount of times by gathering user input or it can run a set amount of times by controlling it with a count variable.

Additional Comments:

If you'd like to customize the lesson slides, click [here](#).

Curricular Standards

This lesson adheres to the following CSTA Standards: 3A-DA-12

## **1.4 Data Revisited**

Introduction

This lesson uses recently learned techniques to harness various forms of data.

The students will:

- Be able to understand and create the algorithms behind the max and min function

Components:

- Video Lesson

What to Emphasize:

- Whenever you get unexpected outputs for your program, feel free to use the Python Tutor Visualization Mode to help step through the program. [Click here to access Python Tutor.](#)

Additional Notes:

Notice in the video how the variable “max” appears in a dark blue color. Just like with other keywords in Python, print, while, int, input, etc. max is also a keyword. A keyword is a word that is reserved by a program because the word has a special meaning. Python anticipates when you use the word “max” that you will be using the max function - max() - so it changes the color of the word to help it stand out to the programmer. Normally in these instances, you would want to use another variable. What happens if you try to use the words print or if as variables?

If you'd like to customize the lesson slides, click [here](#).

Curricular Standards

This lesson adheres to the following CSTA Standards: 3A-AP-18

## 1.5 Review Loops

As we learned earlier in this unit, a loop is a programming command used to repeat code. There are two basic types of loops: a user-controlled loop and a count loop.

A **user-controlled loop** asks the user to input information until a test condition is met. The user may enter one piece of information; they might enter thousands of pieces of information. The number of times a user-controlled loop will repeat is variable and unknown when the loop begins running.

A count loop uses a count variable to count the number of times the loop will repeat. This count variable is also known as the **loop control variable**. The number of times a count loop will repeat is known before the loop even begins running.

## 1.6 Range Function

Introduction

This lesson introduces a new function, the range function, which returns a set of numbers based on the parameters provided.

The students will:

- Be able to define and utilize the range function using 1, 2, or 3 parameters

Components:

- Video Lesson
- Lesson Practice

What to Emphasize:

- The range function can be used in a for loop in order to determine not only how many times a program will run, but which values will be stored to the i variable. Ex: for i in range(1, 10, 2) would run 5 times, and would store the values of 1, 3, 5, 7, and 9 to i each time through the loop.
- Range can be used to count up, count down and to count by different amounts besides 1 at a time.

Additional Notes:

Although you will frequently see the range function used with a for loop, there are other applications of the range function as well.

If you'd like to customize the lesson slides, click [here](#).

Curricular Standards

This lesson adheres to the following CSTA Standards: 3A-AP-21

## **1.7 For Loops**

Introduction

This lesson introduces the for loop, which is a type of count loop that uses the range function to set the value of the loop control variable.

The students will:

- Be able to define and code for loops
- Learn when to use a for loop rather than a while loop

Components:

- Video Lesson
- Lesson Practice
- Code Practice (2)

What to Emphasize:

- For loops are used only when you know exactly how many times you want the loop to run.

Additional Notes:

Having students create a while AND a for loop that do the same thing can be helpful. Seeing them side-by-side helps to show how each can use a count variable, but in slightly different ways.

If you'd like to customize the lesson slides, click [here](#).

Curricular Standards

This lesson adheres to the following CSTA Standards: 3A-AP-18

## **1.8 Counting by Other Than 1**

Introduction

This lesson explores how a for loop uses parameters to count by numbers other than 1.

The students will:

- Code for loops with control variables that increase or decrease by amounts other than 1

Components:

- Video Lesson
- Lesson Practice
- Code Practice (3)

What to Emphasize:

- Although a for loop will run a set amount of times, you are still allowed to get user input inside of the loop.
- Typically a for loop looks neater in a program because it requires a few less lines of code.

Additional Notes:

A for loop in other languages like C++ or Javascript look slightly different: `for( i = 0; i < 10; i + 2)`. Even though this looks different, the parts of the for loop are the same. There is a starting point (`i = 0`), and ending point (`i < 10`) and how we count (`i+2`).

If you'd like to customize the lesson slides, click [here](#).

## Curricular Standards

This lesson adheres to the following CSTA Standards: 3A-AP-18

### 1.9 Summing

#### Introduction

In this lesson, students will learn about summing using a for loop.

The students will:

- Be able to use a for loop with a sum variable to sum a set of variables

Components:

- Video Lesson
- Lesson Practice
- Code Practice (4)

What to Emphasize:

- The order of your code is very important inside of a loop. If you find that the output is off a little, but the code looks correct, trace through the code and you may see that some lines of code just need to be switched or moved around.
- When using a sum variable, it normally is initialized to 0 before the loop begins. If you do not first initialize it as something, how can you add to it?

Additional Notes:

A variable is initialized whenever it is first assigned a value. However, you cannot use a variable on the right side of an assignment without it first being assigned a value. Ex: `x = input("Name: ")` is fine to use even if `x` is not yet defined, because it is on the left side of the assignment. Ex: `x = 4 * r` is not okay if `r` has not yet been initialized - how can I multiply by something I do not know the value of?

If you'd like to customize the lesson slides, click [here](#).

## Curricular Standards

This lesson adheres to the following CSTA Standards: 3A-AP-18

### 1.10 Review of Algorithms and Tracing

## Introduction

This lesson recalls the lesson on algorithms and code tracing from Unit 3.

The students will:

- Be able to define algorithms and code tracing
- Be able to explain techniques for tracing code, and successfully trace code to determine behavior and outputs

Components:

- Video Lesson
- Lesson Practice

What to Emphasize:

- In order to understand what is happening in a specific algorithm or program, we trace the code and follow each variable and track how that variable is changing and what it should be doing.
- Adding comments into the code to help explain what each step is doing can be helpful during code tracing

Additional Notes:

Even though we are trying to “break” our code with different input values, at this point in time, students do not know how to properly make code work if they enter a string when an integer is expected or a float when an integer is expected. However, if we are expecting a string, checking to make sure a single word works, two words, long and short words, no words, etc.

If you'd like to customize the lesson slides, click [here](#).

## Curricular Standards

This lesson adheres to the following CSTA Standards: 3A-CS-03

### **1.11 Modeling and Simulation**

## Introduction

In this lesson, students learn how computer models and computer simulations can help predict and test real-world situations.

The students will:

- Be able to define computer models and simulations and explain why these are used



- Be able to explain the two key features used to create simulations

Components:

- Video Lesson
- Lesson Practice

What to Emphasize:

- A computer model is used to replicate a phenomenon in the world and a simulation is running the model repeatedly while changing variables to predict what will happen in the future.
- Most simulations will use random numbers and large scale repetition

Additional Notes:

Modeling and simulation is a staple in all areas. Some reasons to use simulations and modeling is when the real testing is not feasible because of money, safety, time or technology.

If you'd like to customize the lesson slides, click [here](#).

Curricular Standards

This lesson adheres to the following CSTA Standards: 3A-AP-17

**Test**

Test Review

[Unit 4 Test Review \(pdf\)](#)

[Unit 4 Test Review Solutions \(pdf\)](#)

Online Test

[Unit 4 Online Test \(pdf\)](#)

[Unit 4 Online Test Solutions \(pdf\)](#)

Alternative Test

For each online test, there is an alternative, paper-based test that you can use as additional prep, a makeup test, or as the official test that you proctor offline. If you do use this alternative as the official test, just be sure to manually input your students' scores into the Gradebook.

## Performance Task

### Examine Sample Solution in Python Tutor

Before helping students with this unit assignment, you may find it valuable to examine the sample solution below and visualize it for yourself. Python Tutor is an extremely helpful website that provides a step-by-step walkthrough of each line of code that you enter. Try entering and running the solution below in [Python Tutor's Visualization Mode](#) to see how each line of code produces output and affects the program.

### Sample Solutions

#### Question 1

```
n = int(input("How many numbers do you need to check? "))
```

```
even = 0
```

```
odd = 0
```

```
for i in range(1, n + 1):
```

```
    n = int(input("Enter number: "))
```

```
    if (n % 2 == 0):
```

```
        print(str(n) + " is an even number.")
```

```
        even = even + 1
```

```
    else:
```

```
        print(str(n) + " is an odd number.")
```

```
        odd = odd + 1
```

```
print("You entered " + str(even) + " even number(s).")
```

```
print("You entered " + str(odd) + " odd number(s).")
```