Activity 2.1.1

Transitioning from Blocks to Text

Distance Learning Support

- Check with your teacher about:
- Materials or resources you need for this activity
- What work you need to turn in and how to submit it
- Collaboration strategies

GOALS

- Compare and contrast lower-level programming languages with higher-level programming languages
- Learn basic rules related to programming with syntax
- · Get started with the interactive code editor
- Develop programs independently that uncover what is abstracted in block-based programming languages



TASK DESCRIPTION

Welcoming you to the start-up, your mentor, a developer named Salma, describes the programming environment you will work in. "We use a programming language called Python. It's pretty flexible and used for many data science and robotics applications. We want to make sure that you have an easy transition to Python, so we're using a tool where you can develop in blocks and then see corresponding Python code. This should help you get familiar with the language." You sit down with Salma to write a few test programs using coding concepts that you are already familiar with in this new language.

ESSENTIAL QUESTIONS

- 1. What are some advantages to programming in a text-based language compared to a blockbased programming language?
- What are some of the ways concepts in blocks are represented in languages like *Python®*? (Example: What do loops look like in a text-based language compared to blocks?)
- 3. How realistic is it to expect coding professionals to be experts in all programming languages? What are they *really* experts at?

ESSENTIAL CONCEPTS

- Programming Language Abstraction
- Algorithms, Variables, Arguments, Procedures, Strings and Concatenation, Data Types, and Logic
- Arithmetic Operators, Relational Operators, and Logical Operators
- Conditionals and Event-driven Programming

Interactive Code Editor (Blocks)

Loops in Text-based Languages

While highly abstracted languages make coding simple, sometimes you lose the ability to do exactly what you want to do in the program. Also, some aspects of coding, like managing mathematical operators or using loops, might be easier in a text-based language than using predefined blocks. The power of programming in a text-based language is that you have flexibility to manage certain aspects of a program outside of a limited selection of blocks.

In this activity you will identify what coding fundamentals look like in a textbased language. You will begin to uncover what details are being abstracted or hidden by a programming language. The code blocks used in MIT App Inventor and text used in *Python* are not that dissimilar. The coding concepts remain the same; they just look a little different. Later in this course, you will program in the text-based language *Python*. in this activity, you get a preview to see how *Python* has the same essential computer science concepts.

Run the *Hello World* code below and view the *Python* code. For help running the code, review the **Interactive Code Editor Introduction** (Blocks).

Accessibility Tip: To view the code in a new tab, click the Trinket icon



Distance Learning Support

You cannot save your work directly in the interactive code editor. In order to keep a copy of your work, you must download your program or copy the code to a file on your computer.



Complete the loop review:

Loop Parts

Review the parts of the loop.

Get Started

3 Questions

PLTW COMPUTER SCIENCE NOTEBOOK - Compare Code Describe what the sample program *Hello World* is doing in natural language or pseudocode. Identify some similarities and differences of the same code in block-based code versus text-based code.

> The language that people use in daily conversation with each other. Example:

Say Hello world 3 times, count each time you say it to know when to stop.

A way to work out the logic without worrying about the specifics of the programming language. Example:

Say hello world and increase the count by 1.

Loop.

When the count = 3, stop.

Procedures, Functions, Parameters, and Arguments in Text-based Languages

Not all text-based languages have the same rules. Later in this course, you will program in the text-based language *Python*. As you explore what block code looks like as text, you'll be introduced to new terms and some of the syntax rules specific to *Python*.

The Countdown Loop

Have you noticed anything different about the block choices in the interactive code editor compared to MIT App Inventor? What is different about the purple blocks?



MIT App Inventor

What was called a "procedure" in MIT App Inventor is called a "function" in the interactive code editor. This is because the term "function" is slightly more descriptive for text-based languages.

A function is a type of procedure or routine. Even though *Python* does not use graphical blocks, chunks of reusable code are still retained as a "function" to provide modularity in a text format.

A function "block" in *Python* begins with the keyword def followed by the function name and parentheses. Inside the parentheses are special types of variables called "parameters" or "arguments".

```
def function_name(parameter):
```

Important: The interchangeable use of computer science terms

The terms "procedure" and "function" are often used by coding professionals interchangeably because they are so closely related. The terms "parameter" and "argument" are another example of interchangeable words professionals use when communicating about code.

The interchangeable use of words in computer science can make it confusing sometimes, and that's okay. In later lessons, you'll discover the subtle differences between these concepts, but for now, you may use them interchangeably. In this unit, you will begin to practice using text-based programming. To practice a function definition, define a function named drive with no parameters.

| 0 of 1 Answered | Submit |
|-----------------------|--------|
| | |
| | |
| Question 1 | |
| Fill in the blank. | |
| | |
| Stuck? Show me a hint | |

Check your answer

Functions, Variables, Loops, Strings, Print Statements, and Incrementing

You're going to create a simple algorithm that will display a countdown from 10 to 0 each time the user clicks the **Run** button. At the end of the countdown, a string will display to tell the user they have reached the end.



Review the program below to identify the function, the function call, the loop, and the variable.



Check your Answer

Δ

Arrange and modify the blocks to complete the function in step 3 so that it will count down from 10 to 0. You can use the *Python* code below as a reference.

```
def count_down():
    for number in range(x, x, x):
        print(variable?)
    return 'Message'
print(count_down())
```

Important The variable number will store the number that is being decremented. If you read the count_down procedure in natural language, you might say, "Count down with number starting at 10 and ending at 0, **decrementing** by 1, and printing the number in each iteration."

PLTW COMPUTER SCIENCE NOTEBOOK - Index Variable Naming

Previously, when you incremented, you typically gave the variable the name Count. Does it make a difference whether the variable name is Count or number?

Variable Naming

Review the Python version of your completed code.

PLTW COMPUTER SCIENCE NOTEBOOK - Functions

Add notes about:

5

- a. Using functions.
- b. What functions look like in Python.
- c. Nested functions in Python, such as:

The *count_down()* function is inside the *print()*. This is called a **Nested Function**. The *print()* execution calls the *count_down()* to execute. After the *count_down()* completes, it returns the value from the *count_down* procedure to print the returned value in place of the inner *count_down()*.

Commenting Code

Any text within triple quotation marks or following a # sign, also called a hash tag, in *Python* is treated like a comment and will not be executed by a program. This allows you to write messages in the code for others (and yourself) to understand what specific functions do, when you revisit your code in the future.

Click the ? (Question Mark) and add a comment describing the *CountDown* function you created. Notice, you only get one option for
commenting code right now. In a full-text environment, you can use the
symbol to comment line by line if you wish. You will learn more about
commenting code in text-based languages later.

| ≡ 🥗trin | ket ⊘) ∕ Code | ▶ <u>R</u> un | < Share | • | |
|--|----------------------|--------------------|-------------------|----|--|
| ■ Colors Variables Lists Loops Numbers | ket ♪ ♪ Code | ▶_Run from () 0 | Share | by | |
| | | | | | |
| | | | | | |



6

Take a moment to look at the *Python* code. Compare what you see with what you know from coding in blocks.

```
def count_down():
    for number in range(x, x, x):
        print(variable?)
    return 'Message'
print(count_down())
```

PLTW COMPUTER SCIENCE NOTEBOOK - Countdown Function

- a. What kind of data is stored in *number* in your blocks? What is being done to the variable?
- b. What is the name of your procedure? How does this now look in the *Python* code?
- c. Does it matter that procedures are now called "functions"?
- d. What predefined function do you need to use to display text?
- e. How are strings of text identified as opposed to parts of the code?
- f. How are comments identified in Python?
- g. How are loops represented in Python?
- h. When creating your own new function in *Python*, what keyword do you start with?

Check your answers

Passing and Returning Values (Parameters)

A **parameter** is a special kind of **variable** in a function to refer to one of the pieces of data provided as input to the function. These pieces of data are also called **arguments** in .

| Argument | The values that a program provides to a function or function. |
|-----------|---|
| Parameter | A variable defined in the function to receive specific information. |

Important: Sometimes coding professionals use the terms "argument" and "parameter" interchangeably. In this course, "argument" is used.

When you click the play button to execute your count_down:

• The print function is called.

- The print function calls the count_down function.
- The count down function initializes the variable to "10".
- Each time the variable is decremented in the for loop, that the value is passed to the print function and is displayed to the user.
- Each time the variable comes back to the count_down function, it checks to see whether it has reached the end of the range.
 - If no, the loop iterates again and passes the argument to the print function as before.
 - If yes, the count_down() function returns the string "Blast Off!".

You are now going to adjust your blocks to see how the program execution changes.

Warning: As you view blocks of code in the interactive code editor, you may see a >_View Code button. This feature can be used to view a text version of the code, but this code is generated automatically from the blocks with no intervention and will likely include awkward or unfamiliar programming constructs.

Only use >_View Code if directed by your teacher.

| ≡ 🥬tri | nket 🤌 🖉 Code 🕨 Run < Share 👻 |
|-----------|---|
| Turtle | |
| Colors | |
| Variables | (c) to count down |
| Lists | count with number - from (0 to (1) by (1) |
| Loops | |
| Numbers | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

- What would happen if you remove the print statement in the loop? Discuss with your partner and justify what you think, but do not test your theory yet.
- 9)

8

After you have an educated guess, delete the print statement from the loop and run the program. Is it still counting down?

Code without print()



Save a screen capture of the block code and the *Python* code where you can access it later.

Sequencing in Text-based Languages

In text-based languages, code is executed in the order, or *sequence*, of the functions that are called as the program moves from the first line to the last line of code. The need to pay attention to *sequencing* was abstracted in many places when you worked in block-based programming.

What's Your Name?

You will now be directed to create a program that asks for the user's name and then displays a message back to the user with the input block. As you work through the next several steps, refer to the *Python* code below for guidance.



| 😑 🥝 trinket 🥝 | ℰ _Code | ▶_Run | < Share | • | |
|---------------|----------------|-------|-------------------|---|--|
| Turtle | | | | | |
| Colors | | | | | |
| Variables | | | | | |
| Lists | | | | | |
| Loops | | | | | |
| Numbers | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

Create the variable Name.

11

12

Set the variable Name to a string entered by a user. In this programming environment, you can do this with a pre-built text function called prompt for text with message "". This function is designed to display whatever string you type into the quotation marks and return whatever value the user types.

Try it in text below; in *Python* user input can be retrieved with the input function. Set name to have the value entered by the user, without including a prompt. This is just a chance to practice what you have seen in the reading above.

| Question 1 | |
|--------------------|------|
| Fill in the blank. | |
| | |
| , | |
| Stuck? Show me a h | lint |



Type "What is your name?" into the string in the prompt for text with message "" block. Recreate this step below in text. Check the hint if you get stuck.

| 0 of 1 Answered | Submit |
|--------------------|--------|
| | |
| | |
| | |
| Question 1 | |
| Fill in the blank. | |
| | |
| | |

Stuck? Show me a hint



Enter code that will output Name to the embedded page. Try it in text below by typing the *Python* code for the print statement that you just added to your program.

| Question 1 |
|-----------------------|
| Fill in the blank. |
| () |
| |
| Stuck? Show me a hint |



Run your program and note the outcome.

Check your code

When programming in blocks, you defined and initialized variables anywhere there was space in the App Inventor window. Your procedures were also placed anywhere you wanted.



17

Try recreating your *Python* code here; then run the program to test it for correctness.



PLTW COMPUTER SCIENCE NOTEBOOK - Printing a Name Visit with your elbow partner and explain the output.

Check your answer

(18

Modify the code to concatenate a string message to the user with the name the user enters when the program prompts them. An example of concatenation in this platform and in *Python* is shown in Figure 1.



Figure 1. Concatenation in App Inventor and Python

Output Example

Here is what the text version of concatenation looks like:

```
print("You are awesome " + name)
```



Try adding this concatenation in your own code here.



Lists in Text-based Languages

In this section, you will see a helpful list function in a text-based language. len() is a new function that finds the length of a list. The function counts how many items are in the list.

You will be directed on how to create a basic program that:

- Prompts a user for their name using the prompt for text with message""
- Stores their name to a variable Name using a function you create
- Prints the user's name using a print (Name) function
- Prints how many letters are in their name using the len (Name) function

| ≡ | 곙 trinket 곙 | ∦ Code | ▶_Run | ∢ Share | • | |
|----------------------|---------------------|------------------|--------------|----------------|----------------------------|---|
| Turt Colo Vari | ile ors ables | | | | | |
| List | S | | | | | |
| Nun | nbers | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| (20) | Examine the lengt | th of blc | ock and its | text repres | sentation in <i>Python</i> | • |
|) | | longth | | | | |
| | | lengtn | | L | .en() | |
| | Figure 2 | . Usina the le | nath functio | n in App Inver | ntor and in | |
| | Python | | | | | |
| | NI | | | | | |
| | Next, examine the l | DIOCKS and | text for cr | eating a lis | JT. | |
| | set my_list - | to (c | reate em | pty list | my_list = [|] |

Figure 3. Creating a List in App Inventor and in Python

PLTW COMPUTER SCIENCE NOTEBOOK - List Syntax If functions are represented with parentheses (), how are lists represented in a text-based *Python*?

Concatenate the output into one string that makes sense. Example: Hello Mylo. Your name has 4 letters in it.

Error Alert: Look for language and block shapes in the text drawer that would be able to concatenate. Different programming languages look different, but you can look for trends based on what you know.

Need help with code blocks?

21

22) Save a screen capture of your code, and remove all the blocks and variables from the interactive code editor.

Importing Libraries in Text-based Languages

In this section, you will explore text representations of **conditionals** ;, **variables** ;, **Python imports** ;, **functions** ;, and **operators** ;.



(24)

Put the blocks together to create code that generates a random number from 1 to 10 and allows the user to try to guess the number.

- a. You will need a conditional to compare whether the number is correct.
- b. Provide a message to the user if they guessed correctly or incorrectly.
- c. You may also want to adjust the range of the random number to help you test when a guess is correct or incorrect.
- d. Use the *Python* code below to help.



Import Random: At the top in the *Python* code view is a new statement, an import statement: import random. *Python* has built-in libraries of common functions, such as generating a random number. Using search engines, you can find all sorts of built-in functions that you may call and import without having to write functions of your own.

Identify where the import is occurring in the code.

PLTW COMPUTER SCIENCE NOTEBOOK - Imports
Why do text-based languages put the import code in its location?

Check your answer

26

25

Identify where the random function is actually being called. (Look for the word random in the code on the text side. *Python* has the same structure of the blocks, just without the words you see on the blocks, and with some more visible details than what you see in the blocks. PLTW COMPUTER SCIENCE NOTEBOOK - Random

What do the values in the () represent after the random function is called?

Check your answer

27

) Save a screen capture of your code and clear all blocks and variables.

Operators and Modulo in Text-based Languages

You are going to develop **dynamic** formulas that will take user input and return the outputs to a user so they can program a self-driving vehicle. The program will:

- Take a user-inputted number of seconds and convert it into minutes and seconds
- Calculate the tire circumference based on a user-inputted radius.

To know when your program is working properly, use these inputs and look for these outputs:

| Inputs | Outputs |
|------------------------|---|
| tire radius = 1.5 | tire circumference = 9.42477 |
| seconds traveled = 333 | traveled for = 5 minutes and 33 seconds |

| = 🥝 trinket 🥝 | ℯ∕ _Code | ▶_Run | < Share | • | |
|---------------|-----------------|-------|-------------------|---|--|
| Turtle | | | | | |
| Colors | | | | | |
| Variables | | | | | |
| Lists | | | | | |
| Loops | | | | | |
| Numbers | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

Based on the *Python* code below, create a block program in the program space above that gives you the correct outputs with the inputs being a 1.5 radius and 333 seconds traveled:



Circumference = $2\pi r$ π may be substituted with 3.14 (if π is not a built in function)

What blocks are needed?

- 29 After you have the code behaving as you like, modify it to allow inputs for the radius and time, making your program interactive.
- 30

When the code is **dynamic**, capture a screenshot of your completed code in both the block view and in the *Python* view. You will submit these to your teacher and save them to review later.



Complete the review below.

From Blocks to Text

Practice identifying the parts of the text that you have seen in blocks.

Get Started

4 Questions

PLTW COMPUTER SCIENCE NOTEBOOK - TEMP

After completing the review above, record what terms you will use interchangeably as you move from block-based to text-based coding.

The Human Computer

Katherine Johnson was an American mathematician, who was hired to work for the National Advisory Committee for Aeronautics (NACA) in 1953 and retired in 1986 from what became NASA. She began her work as a "human computer," doing calculations of trajectories, launch windows, and emergency return paths essential to the success of many early space missions, such as Project Mercury



Katherine Johnson Source

and Apollo 11. She was the first African-American woman to work as a NASA scientist and was awarded the Presidential Medal of Freedom by President Barack Obama in 2015. Source

CONCLUSION

- 1 What patterns in the essential coding concepts did you see in blocks and in text?
- 2 How did you interpret and respond to the **Essential Questions**? Capture your thoughts for future conversations.

Proceed to next activity