

# Code Uncommon Summer Immersion 2023

## Session 1: Programming Basics in Python



Session Overview	
<b>Session Objectives:</b>	<b>Session Agenda:</b>
<ul style="list-style-type: none"> <li>▪ Teachers will write a basic Python program that accepts user input and produces output using variables and operations on both Strings and numerical values.</li> <li>▪ Teachers will write a Python program that uses a variety of conditional statements.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Welcome &amp; Do Now</li> <li>▪ Python Basics: Hello, world!</li> <li>▪ Variables and Concatenation</li> <li>▪ Mathematical Operators</li> <li>▪ Challenge Set #1: Variables and Mathematical Operators</li> <li>▪ Unplugged Activity: Binary Numbers</li> <li>▪ Conditionals</li> <li>▪ Challenge Set #2: Conditionals</li> <li>▪ Survey &amp; Closing</li> </ul>
<b>Session Key Points:</b>	
<ul style="list-style-type: none"> <li>▪ A <b>variable</b> stores one piece of information at a time. If the value of the variable is <b>updated</b>, the old value is erased.</li> <li>▪ Python can “operate” on both <b>numerical values</b> and <b>strings</b>. Each operation results in the output that is reasonable for the type of input.</li> <li>▪ <b>Conditional statements</b> allow us to write more realistic programs whose output is not fixed, but rather depends on user input or other information.</li> </ul>	

### Vision Writing

*Please answer the questions below as thoroughly as possible. It will be interesting to refer back to these and refine them throughout the week!*

1. In your own words, what is computer science?

---

---

---

---

---

---

2. Why do you want to learn to code? Is there something specific you'd like to create?

---

---

---

---

---

---

3. How do you imagine bringing coding and computer science to the students we serve?

---

---

---

---

---

---

---

---

**Do Now**

Turn on your computer and sign in to **replit.com**. Then, put your screen at 45 degrees and answer the following questions.

*If you are having technical difficulties, please raise your hand while you answer the questions, and someone will help you troubleshoot!*

1. Consider a program that most of us use every day: Google! One neat feature of Google is auto-complete, which appears to “guess” your search query based just on the first couple of letters. **How do you think auto-complete works? How does Google “know” what you want to search for?**

---



---



---



---



---



---

2. The following code is from a drag ‘n drop programming language called Scratch. Programming in Scratch allows you to control the motion of this little cat<sup>1</sup>.



Take a look at the code<sup>2</sup> below. What do you think will happen when we send the code to the cat?

```

when green flag clicked
  forever loop
    if key space pressed? then
      change x by 1
    else
      change x by -1
  
```

---



---



---



---



---



---

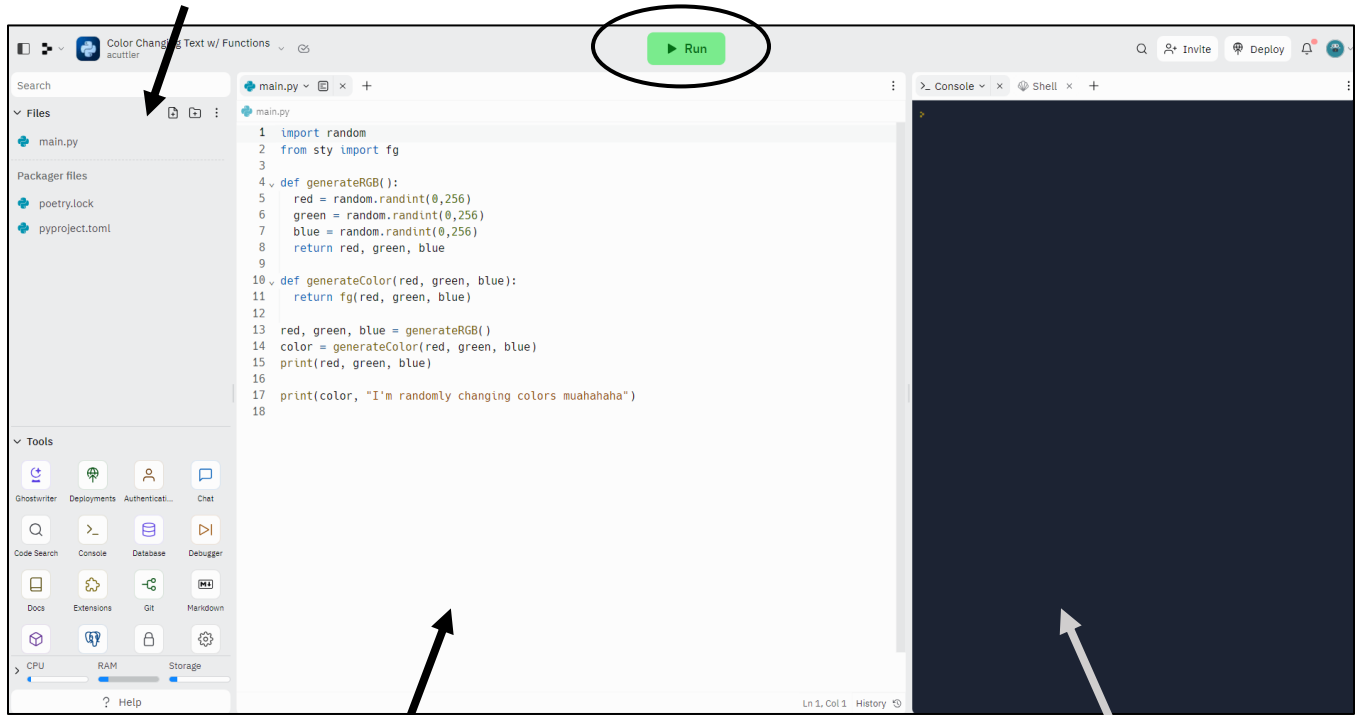
<sup>1</sup> MIT Media Lab. (n.d.) Sprite1 [Online image]. Retrieved from [https://scratch.mit.edu/projects/editor/?tip\\_bar=getStarted](https://scratch.mit.edu/projects/editor/?tip_bar=getStarted). Available under a Creative Commons Attribution-ShareAlike 2.0 Generic license. Image has not been altered in any way.

<sup>2</sup> MIT Media Lab. (n.d.) Scripts code [Online image]. Retrieved from [https://scratch.mit.edu/projects/editor/?tip\\_bar=getStarted](https://scratch.mit.edu/projects/editor/?tip_bar=getStarted). Available under a Creative Commons Attribution-ShareAlike 2.0 Generic license. Image was created using Scratch, but none of the features have been altered in any way.

## The Replit IDE

Label the components of the Replit IDE below.

IDE: I \_\_\_\_\_ D \_\_\_\_\_ E \_\_\_\_\_



## You Try: Hello, world!

Create a new project in your **Code Uncommon** folder called **01\_HelloWorld\_CFU**.

Write a program that greets the user and waits for them to press a key in order to close the program by displaying something like “Press enter to exit.”

**Bonus!** Personalize your program by asking the user to enter their name and then responding with a personalized greeting (like “Hey there, Anna!” or “How’s it going, Isaiah?”).

### Check for Understanding #1: Error Analysis

Explain why each Python code segment below will result in an **error**. Then, fix it so that it will run.

Code Segment	Explanation of Error	Corrected Code Segment
<code>print (Hello World)</code>		
This program greets a user. <code>print ("Hello world!")</code>		

### Code Demo: Variables & Concatenation

Annotate the program below and record the output as we go through the code line-by-line.

```

1  name = input("What is your name? ")
2
3  print("Hello,", name)
4
5  name += " the Great"
6
7  print("Hello,", name)
8
9  name = "Marcus"
10
11 print("Hello,", name)
12
13 print("Hello", "name")
14
15 name += name
16
17 print("Hello,", name)
18
19 print("Hello,", name*3)

```

**Output/Notes**

## You Try: Food Smooshers

Create a new project in your **Code Uncommon** folder called **02\_Variables\_CFU**.

Write a program that asks the user for their name and their two favorite foods. Respond with a suggestion that they try a “smooshed” food based on their favorites!

**Sample program run:**

```
>_ Console x Shell x +  
Hi there! What is your name? Allison  
Nice to see you, Allison  
I'm feeling hungry. Please enter one of your favorite foods: ice cream  
Please enter another favorite food: tacos  
If you like ice cream and tacos you really need to try ice cream tacos!  
>_
```

**Bonus!** Also ask the user how old they are, and then tell them in what year they were born.

## Code Demo: Mathematical Operators

Annotate the program below and record the output as we go through the code line-by-line.

```
1 print(11 + 5)  
2  
3 print("11 + 5")  
4  
5 print(11 / 5)  
6  
7 print(11 % 5)  
8  
9 print(11 + " cups of coffee")  
10  
11 print(11, "cups of coffee")  
12  
13 a = 6  
14 a *= 2  
15  
16 print(a)  
17  
18 print(a - 20)
```

**Output/Notes**

**Check for Understanding #2: Predict the Output**

Predict the output of each code segment below. Try doing this *WITHOUT* entering the code into Python at first.

Code Segment #1	Code Segment #2
<pre>x = 3 y = 7 print(x - y)  #BONUS print(abs(x - y))</pre>	<pre>num_one = 7 num_two = 8 num_one += 5 print(num_one*4, num_two)  #BONUS print(num_two + num_one * 2)</pre>
Output	Output
Code Segment #3	Code Segment #4
<pre>print(25 / 4) print(20 % 7)  #BONUS print(4 ** 2)</pre>	<pre>inches = 40 INCHES_IN_FEET = 12 feet = int(inches / INCHES_IN_FEET) inches = inches % INCHES_IN_FEET print(f“{feet} feet &amp; {inches} inches”)</pre>
Output	Output

## Challenge Set #1: Variables and Mathematical Operators

As you work through the programs below, pay close attention to the following:

- (1) Create a folder called **Challenge Set #1** inside your Code Uncommon folder.
- (2) Create a **new** Python project for each program. The name of the project is indicated next to the prompt.
- (3) Sample input and output for a program is sometimes given. Text that is **boldfaced** indicates that the user would be typing that in. You should test your program to ensure that your output is formatted similarly to the sample output.
- (4) **Ask three before me!**

### Level 1 Challenge (Mandatory)

**01\_Useless Trivia:** Write a program that asks the user to input some basic numerical information about themselves (i.e., their age, height in inches, etc.). Respond with at least two pieces of “useless trivia” based on what they enter.

#### Sample Input

Hello, dear user! How old are you? (Round to the nearest year and please don't include units) **26**

Thank you! Also, how many pounds do you weigh? (Again, no units) **152**

#### Sample Output

Did you know that you have been alive for 227760 hours, and that you would only weigh 25.08 pounds on the moon?

### Level 2 Challenge (Optional Bonus)

**02\_Cash Register:** Write a program that asks the user to enter the price of an item, along with how much the customer paid. Then, tell the user how to return the correct change to the customer using dollar bills, quarters, dimes, nickels, and pennies.

#### Sample Input

Please enter the price of the item: **\$5.53**

How much did the customer pay? **\$20.00**

#### Sample Output

Give the customer the following change:

14 one dollar bill(s)

1 quarter(s)

2 dime(s)

0 nickel(s)

2 penny/pennies



### Exploration: Conditional Statements

Consider the following code segments in your groups.

Code Segment #1	Does this represent a valid traffic light? <i>Explain your reasoning.</i>
<pre>if(color == red):     stop() else:     go()</pre>	
Code Segment #2	Does this represent a valid traffic light? <i>Explain your reasoning.</i>
<pre>if(color == "red"):     stop() else:     if(color == "green"):         go()     else:         slow_down()</pre>	
Code Segment #3	Does this represent a valid traffic light? <i>Explain your reasoning.</i>
<pre>if(color == "green"):     go() if(color == "yellow"):     slow_down() else:     stop()</pre>	
Code Segment #4	Does this represent a valid traffic light? <i>Explain your reasoning.</i>
<pre>if(color == "green"):     go() elif(color == "yellow"):     slow_down() else:     stop()</pre>	

**Explanation: Conditional Statements**

The code segments on the previous page are all examples of **conditional statements** because their output depends on whether certain **conditions** are true or false. Answer the questions below to check your understanding of conditional statements.

1. What is the purpose of a conditional statement?

---

---

---

2. What is an example of a computer program (other than a traffic light) that might use conditional statements?

---

---

---

3. In your own words, clearly explain the difference between **Code Segment #3** and **Code Segment #4** above.

---

---

---

**Live Coding: Conditional Statements**

Create a new project in your **Code Uncommon** folder called **04\_Conditionals\_Demo**.

Write a program that checks whether a student should earn a scholarship based on their GPA and whether they met the application deadline:

- If the student met the deadline and their GPA is at least 3.5, they get the scholarship.
- If they didn't meet the deadline but their GPA is at least 3.8, they also get the scholarship.

Your program should start by **asking the student** for their GPA, and whether they met the application deadline.

## Challenge Set #2: Conditional Statements

As you work through the programs below, pay close attention to the following:

- (1) Create a folder called **Challenge Set #2** inside your Code Uncommon folder.
- (2) Create a **new** Python project for each program. The name of the project is indicated next to the prompt.
- (3) Sample input and output for a program is sometimes given. Text that is **boldfaced** indicates that the user would be typing that in. You should test your program to ensure that your output is formatted similarly to the sample output.
- (4) **Ask three before me!**

### Level 1 Challenges (Mandatory)

**01\_Hi Lo:** Have the computer choose a random number between 1 and 20. Then, ask the user to guess the number. Respond with a customized message based on whether their guess is correct, too high, or too low!

**Bonus:** If the user is incorrect, let them guess again!

**Hint:** To have your program generate a random number, you must first import Python's random library BEFORE all of your other code, with this line of code:

```
import random
```

Then, to generate a random number between 1 and 20, you use the command:

```
random.randrange(1, 20)
```

#### Sample Input

Hello, dear user! Please guess a number between 1 and 20: **11**

#### Sample Output

That's too bad - you were way too low! The number was 13.

**02\_Grade Calculator:** Write a program that prints out a student's letter grade based on their percentage (assuming the percentage is an integer). Use the following scale:

Percentage	Letter grade
90 or above	A
80 – 89	B
70 – 79	C
60 – 69	D
Below 60	F

## Level 2 Challenge (Optional Bonus)

**03\_Login Screen:** Ask the user to enter their username and password. If they enter the correct password (you will have to decide what that is), respond with a welcome message such as “Hi, jsmith! Welcome to the Super Exclusive Program!” If they enter an incorrect password, respond appropriately. Here’s the catch, though: **the user should only have 3 chances to enter the correct password before they get “locked out.”** Inform them once they’ve been locked out.