CSC101 Assignment

Python Programming

Overview

In this assignment, you'll deepen your understanding of programming concepts by taking your first steps into text-based programming in Python. You'll practice debugging error-prone code and create your own personal profile using variables, lists, and loops. By the end, you'll see how these same programming concepts power the apps and games you use every day.

Assignment Preparation

Make a copy of this document to record your answers. Rename your copy of the document to **lastname_python**, where **lastname** is your last name. You will **place all answers and screenshots** in this Google Document. The steps below will tell you what to put in your answer document - use the spaces provided to insert your answers accordingly.

Part 1: Text-Based Programming

Block-based programming allows users to build powerful programs without knowing all of the details involved in text-based code. If you look behind the scenes of the Music Lab programming environment, you'd see JavaScript code, which is a popular programming language used in many applications. We saw a bit of JavaScript code in the previous Dance Party assignment. Now let's take a look at another popular programming language called Python. Python has been growing in popularity and is commonly used to process, analyze, and visualize data.

Printing Data Types

📖 Read: What are Data Types?

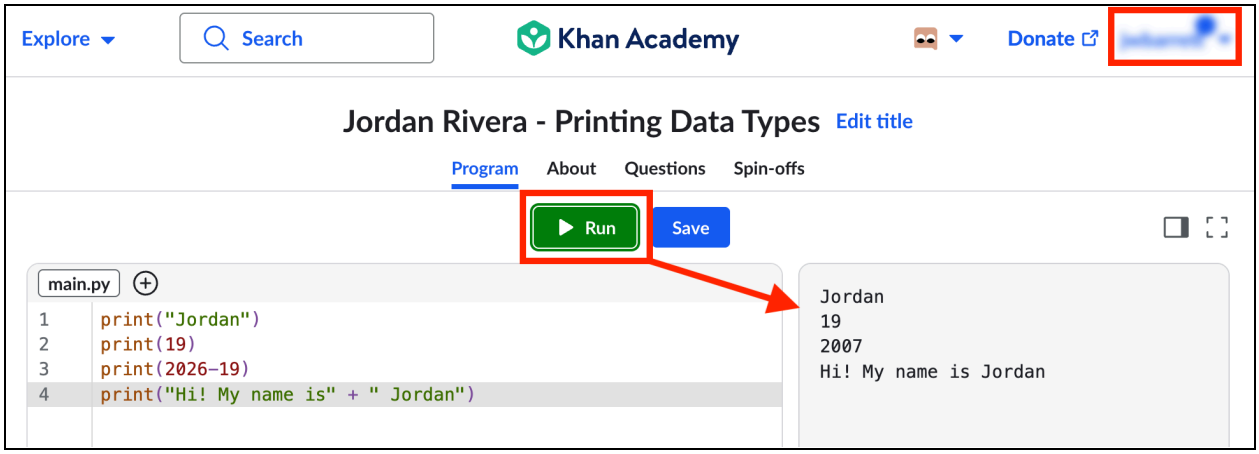
▶️ Watch Print Statements and expressions [4:49 min]

DO THE FOLLOWING:

- ➔ Log into Khan Academy using your school provided Google Account.
- ➔ Go to the Intro to Computer Science - Python page
- ➔ Click the "Create a New Python Program" button and name your program "[yourname] Printing Data

Types” where [yourname] is your own name, then click Save.

- ➔ Use print statements to print the following data types (each on a separate line):
    - Print your name as a string
    - Print your age as an int
    - Evaluate an expression by subtracting your age from the current year
    - Concatenate a string of text: “Hi! My name is” and your own name.
- ➔ Run your program, then take a screenshot of your fully executed program and provide it below (see example)



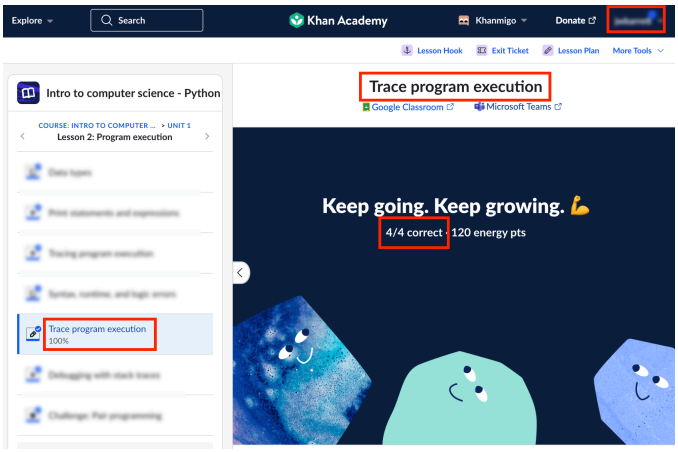
	[Insert screenshot here]

Program Tracing

▶️ Watch: Tracing Program Execution [5:08 min]

DO THE FOLLOWING:

- ➔ Complete the practice set: Trace Program Execution
- ➔ Take a screenshot of the results summary page including your username, the practice set title, and your score showing at least 3/4 to earn proficiency (see example).



Part 2: Variables

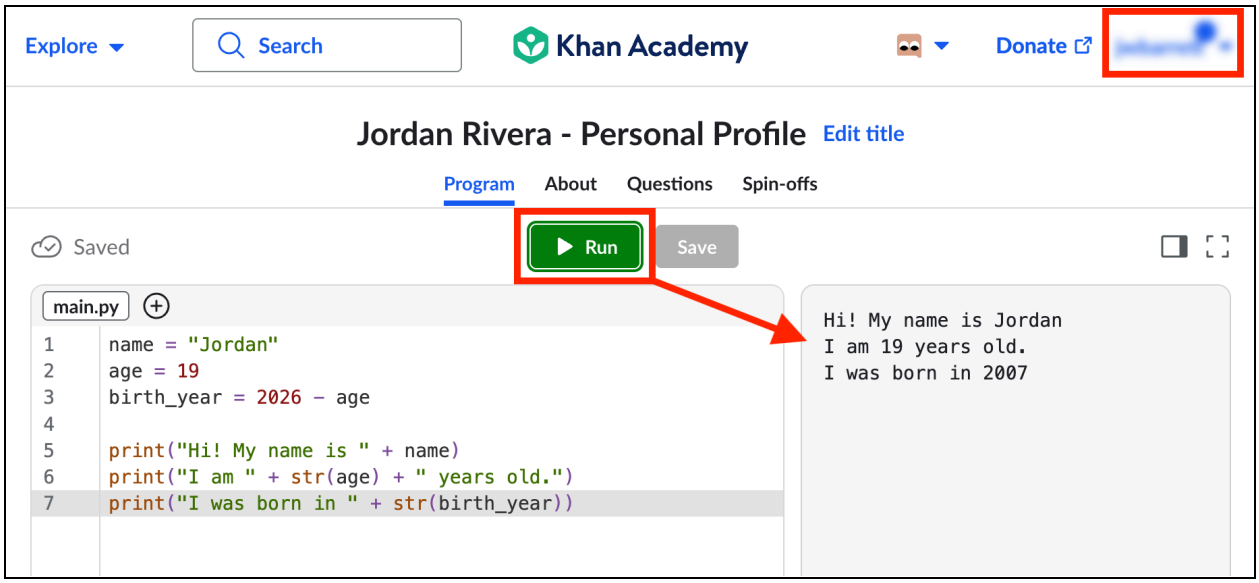
Assigning Variables

▶️ Watch: Variables and Assignments [4:44 min]

DO THE FOLLOWING:

- ➔ While still logged into Khan Academy, go to the Intro to Computer Science - Python page and create a new program called: “[yourname] Personal Profile” where [yourname] is your own name, then click Save.
- ➔ Use variables to print your personal profile information:
    - Create a variable for your name
    - Create a variable for your age
    - Create a variable for your birth_year that evaluates the expression of subtracting your age variable from the current year
    - Print the following statements using variables where appropriate:
        - Introduce yourself using your name variable
        - State how old you are using the age variable
        - State the year you were born using the birth_year variable

      ⚠️ *Note that you need to make the age and birth_year variables strings using str() in order to concatenate them with other strings in your print statements*
- ➔ Run your program, then take a screenshot of your fully executed program and provide it below (see example)



[Insert screenshot here]

➔ Reflect on your personal profile program. Explain how you could use a variable for the current year. Why might this be useful?

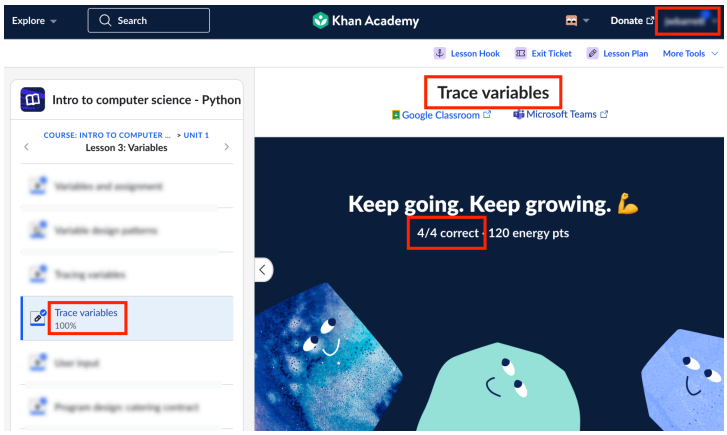
Variable Tracing

▶️ Watch: Tracing Variables [5:46 min]

DO THE FOLLOWING:

➔ Complete the practice set: Trace Variables

➔ Take a screenshot of the results summary page including your username, the practice set title, and your score showing at least 3/4 to earn proficiency (see example).



[Insert screenshot here]

Part 3: Comments & Debugging

In the previous steps, you created a simple program using variables and print statements. Reflect on the process - did you run into any issues or errors in your code? Would someone else understand what the program is meant to do? Using comments and debugging strategies will help you when you get stuck!

📖 Read: What are Comments in Python?

▶️ Watch: Debugging with Stack Traces [5:50 min]

DO THE FOLLOWING:

- ➔ Log into Khan Academy and open the Fix My Code! program.
- ➔ Make a copy of the program by clicking the "Spin-Off" button at the bottom right

- ➔ Name your project "[yourname] Profile v2"
- ➔ The program has 3 errors that prevent it from running correctly - identify and fix the errors in the code.

  Hint: Use quotes for string data types, use parentheses for print statements, and check the variable names for spelling errors

- ➔ Update the student profile information to use your own name, age, favorite subject, and gpa
- ➔ Save the changes to your program. When you run your program, the output should look similar to the example shown here.

```
Student Profile:
Name: Jordan
Age: 19
Favorite Subject: Computer Science
GPA: 3.5
```

- ➔ Run your program and take a screenshot showing the working code and the output

[Insert screenshot here]

Part 4: Lists & Loops

Enhance your Personal Profile program from the previous step by adding a few more programming concepts including lists and loops.

Loops

📖 Read: What are Lists?

📖 Read: What is a For Loop in Python?

DO THE FOLLOWING:

- ➔ Open your “Profile v2” program from the previous step (if you closed it, you can find it on your saved Programs page)
- ➔ Review this sample code that adds a new section of courses to your profile. You will add similar code to your program in the following steps.

```
# Display all courses
print("Current Courses:")

courses = ["Intro to Computing", "English Composition", "Psychology"]

for course in courses:
    print("- " + course)
```

- ➔ Add a comment that starts a new section where you will add your courses, for example:

```
# Display all courses
```

- ➔ Add a print statement that prints the text “Current Courses:”, for example:

```
print("Current Courses:")
```

- ➔ Add a list that contains 3-5 courses you are currently taking (or have taken in the past), for example:

```
courses = ["Intro to Computing", "College Algebra", "English
Composition", "Psychology"]
```

- Add a for loop that goes through the list of courses, printing each one, for example:

```
for course in courses:
    print("- " + course)
```

- Be sure to fix any bugs in your program and save it as you make changes. Your output should look similar to the example shown here.
- Run your program and take a screenshot showing the working code and the output

```
Student Profile:
Name: Jordan
Age: 19
Favorite Subject: Computer Science
GPA: 3.5
Current Courses:
- Intro to Computing
- English Composition
- Psychology
```

	[Insert screenshot here]

Part 5: Programming Reflection

Reflect on what you have learned so far (from the previous assignment and this one) about the core programming concepts of *variables*, *loops*, *conditionals*, and *functions* and how these might be used in your favorite program.

- Describe an app that you use (e.g. a game, a social media app - any software application) and provide a link to its description (e.g. a link to the game's website).

Name of app/program:	
Describe the program (in your own words):	
Link to program description:	

- Describe in 1-3 sentences each where the programmers of the software might have used each of the following terms.

Term	**Description**

Variable	
Loop	
Conditional	
Function	

—---------- Example Response —----------

App Name:	*Frogger*
Describe the program (in your own words):	*In the game, the user tries to get a frog across a busy street by moving it around the screen while avoiding cars. The user gets points for getting the frog across the street and loses points for being hit by a car.*
Link to program description:	*https://en.wikipedia.org/wiki/Frogger*

Term	**Description**
Variable	*The user's score is probably a variable where the program adds to the value of the variable when the frog gets across the street and deducts from the value of the variable when the frog gets hit by a car.*
Loop	*The cars keep moving regularly across the screen over and over again. That is probably a loop in the code that moves the cars.*
Conditional	*The programmer probably has a conditional statement like "if the frog is hit by a car, deduct 100 points from the score variable"*
Function	*The programmer probably wrote a function move that moves the frog.*