# Lab 1 Hello and Greetings

## Instructor Guide

## Overview

This lab is designed to be an introduction to Teknowledge along with the first exposure to computer programming for students in the Teknowledge curriculum. After getting to know and understand the purpose of Teknowledge, students will have a general discussion on what computer science is, for the purpose of demystifying the subject. Followed by learning how to manage and save files in a uniform way.

After the general introduction and talk, the **Hello Name** activity will begin. Students will learn the basic idea of what a *variable* is, along with what the *print* and *input* functions do. Additionally, they will learn about their first *type*, a *string*. Lastly students will learn about error messages, setting students on a path for debugging on their own. It should be emphasized that they can easily see what line the problem is on, and they can use critical thinking to solve the problems. Students will then complete the **Syntax Practice** activity on their own with help from the mentors and the Syntax Guide.

There should be a general feel of comfort and excitement during this lesson. If students have never programmed before they may be uncomfortable approaching programming, so thoroughly going over simpler ideas can greatly improve the confidence and interest in these young coders!

## Learning Goals

- Comments are used to describe code and are the symbolized in Python with the '#'
- Variables hold values
- Variables can be different types, we are just learning about the string type today
- Strings are a type
- A string is represented as any word, or combination of letters and numbers, surrounded by quotation marks

- Print(string) displays text to the console
- Print() prints an empty line
- Input(string) asks for the player to put in text then press enter, with the prompt text as a parameter
- Input returns a string of all the information typed by the user before s/he hits 'enter'
- Simple debugging skills, such as understanding the error message and knowing what line to go to debug the code
  - Understanding common syntax errors that could occur, such as not closing a string, or not closing a print or input statement

## Personal Growth Goals

- <u>Excitement</u>: Programming is fun and I can do it!
- <u>Confidence</u>: Focusing a lot on the underlying ideas behind computer science, and simple programming can help students feel like they have the ability to do this.
- <u>Responsibility</u>: If students run into any bugs, they should first try to debug them on their own. If that concept can be instilled early on, they can feel more ownership over their code, which will help them learn the material better and feel more attached to the activity they are working on and the Teknowledge program.

## Skills Required

- None

## Resources Required

- Computers for either every student or every pair of students
- Python 3 and a text editor needs to be installed on all the computers
- One mentor per 2-3 students
- A projector to project the central instructor's computer

## Instructor Preparation

1. Make sure all the computers students will use have Python and a text editor (right now, we use Pyzo) installed (check to see that students have a way to save/access files)
2. Load the following programming files onto each computer:
   a. 01_01_hello_name.py
   b. 01_02_syntax_practice.py

# In Depth Description of Lab Activities

## Phase 1: Setup

1. Before the students arrive, open the following files in a text editor on each computer:
    a. 01_01_hello_name.py
    b. 01_02_syntax_practice.py
2. Have the [Teknowledge video](#) open on a laptop that can be projected to the entire class.

## Phase 2: Introduction to Teknowledge

1. Introduce yourself as the program instructor, then play the [Teknowledge video](#) so students can gain a better understanding of what they are here to do.
2. Next, every mentor and student need to have time to introduce themselves with their name, favorite hobby, favorite color etc. Personalizing the experience can be hard, but it is the most effective means of teaching. If each mentor knows what their students are interested in, it will help the mentors show the students how they can apply the skills they've already learned to cool projects or ideas in that field.
3. Then have a quick follow up discussion with the class about the video asking questions like:
    a. What is computer science?
    b. What is the difference between coding/programming and computer science? (Mention that this program focuses mostly on coding/programming, and some on computer science.)
4. Students should now learn how to create a folder and save files in a uniform way:
    a. Saving to a Flash Drive:
        i. Students should create a new folder in your flash drive and name it Teknowledge.
        ii. Next, create a new folder within Teknowledge and label it Lab 1.1.
        iii. Next save the programs opened in the text editor to Lab 1.1
        iv. Additionally, students can save the Syntax Guide in the main Teknowledge folder.
    b. Saving to Computer:
        i. Same steps as above, but save it to the desktop or the documents depending on what the instructor wants.

## Phase 3: Hello Name Activity

1. From the Syntax Guide or from a plan, teach the following topics. Additionally, the students should be following the code in 01_01_hello_name.py.
    a. Run the Code

       i.     Have the students run the code with Ctrl+Shift+E, or Run → Run File As Script if Pyzo is being used as the text editor.
      ii.     Have them look at the console, to see what it printed out.
     iii.     Students should gain an understanding of what 'running' code means.

b. Variable
       i.     Relate to math, that a variable is some letter like 'x' that can store a value, like x = 2. In programming, you can have whole words that are variables, and they can have multiple types.
      ii.     Question: Can anyone point out where the variables in this program are?

c. Strings
       i.     One of the types a variable can take is a string.
      ii.     Strings are represented by any value enclosed by quotation marks (single or double quotation marks).
     iii.     Strings are kind of like speech in a book.
     iv.     In the file, show how the variable 'question' is equal to the question, "What is your name?"
      v.     To put multiple strings together you add them with the plus (+)

d. Comments
       i.     A comment is a specific tool for explaining code, anything in the same line after the symbol '#' is a comment.
      ii.     Comments don't tell the computer to do anything, they are for people reading the code to help understand the code better.
     iii.     You can have comments on the same line as the code or on its own line.
     iv.     In the line of the question = .. there is a comment that the program does not run.
      v.     Have the students change the comment to something that could explain the code better.

e. Printing
       i.     Have the students run the code, if they have not done so already, and see if they can find where the lines in the console came from.
      ii.     Question: Why are elements within the print statement in quotation marks and others not?
           1.    Currently we are printing strings, and to do so, we need quotation marks.
           2.    If no quotation marks were used it would assume that everything inside if a variable, and it isn't.
     iii.     Question:: "Why do you think the word 'question' didn't need quotation marks?"
           1.    Answer: Because 'question' is already a string.
     iv.     Question: "What do you think the print() is doing?"
           1.    Answer: Printing an empty line.

f. Input

i. Explain: Input is how you can interact with your code. Anything you type into the console after input is called will be saved to whatever variable you set it equal to.
  ii. Input returns a string.
g. With their mentors have the students work through each of the challenges, then move into the Syntax Practice.

## Phase 4: Syntax Practice Activity

1. After the students are done with all the challenges in Greetings, mentors should review with their group what variables and strings are, and what print and input do.
2. Transition then, into having the students working individually or with a group to complete the challenges in Syntax Practice. The students should be using the Syntax Guide for any questions, and continue to discuss with their mentors how to solve the challenges.

## Phase 5: Pack up | Review

1. Mentors should lead a discussion with their students based on the question: What do you think that you can do with these tools now?
2. This question may be useful to use this as a form of review, and can also be used to increase interest in the subject.

# Lesson Plan

(:10) means that this part should be done by the tenth minute of the lesson
1. Setup (:0)
2. Introduction to Teknowledge(:20)
3. Hello Name Activity (:40)
4. Syntax Practice Activity (:55)
5. Pack up | Review (:End)

# Take Away

After completing this lab, students should feel comfortable with the concept of variables, strings, print statements and input, and be able to apply some of this information to a simple unique idea.