JupyterVox Teaching Script   
Lesson 2, 02/25/23

1. General Notes to Instructors
   1. Take notes on the difficulties/issues encountered by the students
   2. Be patient and wait for the student to finish, don’t jump in to assist immediately. Tell the student to ask for help when the student feels help is needed. We were intrusive in the first two sessions. If you feel you will need to type on the student’s keyboard, please ask permission from the student first. Also narrate what it is you are doing/looking for/trying to understand.
   3. When you move from one task to another (or to a different subtask), communicate this to the student. They are highly aural. As you communicate which task or subtask you are starting, it will help them to understand where they are at in terms of their present location relative to the entirety of what they have to accomplish for the morning.
   4. At the end of the session, we will need to ask the student for feedback for teaching improvements.
      1. We may also ask the parents for feedback
   5. We will separate the students to keep them focused on coding
   6. Need to make sure students don’t stray too much from teaching materials
   7. Ask Dr. Ewoldt and Dr. Neely for help, if we can’t manage it
   8. Give the students an overview of today’s tasks at the beginning.
   9. Need to disable auto-complete, auto-close brackets, and diagnose (and all editing assistants) in Colab settings.
2. Task 0: Set-up & Overview of today’s tasks
   1. Explain new layout as students enter
   2. Have students set up their laptops & braille readers on individual table then go to community table for snack/drink/fellowship
   3. Dr. Ewoldt will explain overview of tasks
   4. Need to disable auto-complete, auto-close brackets, and diagnose (and all editing assistants) in Colab settings.
3. Task 1: Recap of last week (20min)
   1. Notes to instructor:
      1. Tell the student we will not assist, and they will need to work on this themselves
      2. Take notes on the difficulties/issues encountered by the students
   2. Teaching/Lecture:
      1. Create a new Colab notebook
      2. Create a new code cell, code “3+5” (sum), execute, and exam results
      3. Create a new code cell, code “7-4” (subtract), and execute
      4. Create a new code cell, code “6\*7” (multiplication), and execute
      5. Create a new code cell, code “10/3” (division), and execute
      6. Create a new code cell, code “2\*\*10” (exponential), and execute
      7. Move the cursor back to, the first “sum” cell, and change the sum to “3-6”.
4. Task 2: Decimal numbers (floating point numbers) (20min)
   1. Notes to instructor: N/A
   2. Teaching/Lecture
      1. Tell the student we can also do decimal number math. They are called floating point numbers in Computer Science
      2. Do math equations for basic operations,
         1. Add: 3.4 + 5.1
         2. Subtract: 1223.498 – 895.3372
         3. Multiply: 45.77 \* 96.82
         4. Divide: 11.1/33.3
         5. Exponent: 10.1\*\*12.3
5. Task 3: Complex Expressions and Parentheses (20 to 30 min)
   1. Notes to instructor:
      1. Need to assist students
      2. When describe equations, please use plain English as the description. See (b).ii.1)
      3. Parentheses matching and auto-complete could be an issue
      4. Decimal point may be an issue
      5. Takes notes on the difficulties/issues encountered by the students
   2. Teaching Lecture:
      1. Complex equation:
         1. Create a new code cell
         2. Write a slightly complex equation, 3+5.4\*6, and execute
         3. Let the student read the equation, process it in mind and ask them if they know the order of the operations.
      2. Single pair of parentheses:
         1. The instructor writes the equation (1+2.2)\*4 in a colab code cell, and ask the student to read and explain the order of operation. Tell the student we will write similar equations.
         2. Ask the students what should the equation be for “add 3+5.4 first, then multiply the result by 6”. i.e., (3+5)\*6. The instructor can tell the students this is the same with PEMDAS.
         3. Create a new code cell, and ask the student to type the above equations by themselves
         4. Execute
         5. Do complex equations:

(5-1)/2

(9+3)\* 3

(3\*2)+1

3.0\*(1.0+2.0)

* + 1. Two pairs of parentheses
       1. The instructor writes the equation (1+2.2)\*(2+2) in a colab code cell, and ask the student to read and explain it. Tell the student we will write similar equations.
       2. Ask the student to come up with this equation first, (4+5) \* (7+8). For example, tell the student we want to do two sums 4.2+5 and 7+8.3, then multiply their results. We will need to add parentheses, and ask the student how to or where to add the parentheses.
       3. Ask the student to type and execute this equation with a new code cell
       4. Ask the student to remove and parentheses and compare the results to understand the impact of parentheses
       5. Do these exercises:

(5\*2) + (8/2)

(4/2) - (1\*1)

(5.1- 1.0)/( 2 \* 1.0)

(3\*2)+(4.0/2)

(7+1)\* (2-1)

* + 1. Nested parentheses
       1. The instructor writes the equation (5-(3-2))\*10 in a Colab code cell, and ask the student to read and explain it. Tell the student we will write similar equations.
       2. If the student has trouble understanding nested parentheses, then we can skip this task.
       3. Ask the student to come up with expression, 8 - (6-5). Tell the student we want to do 6-5, then use 8 to subtract the result. Ask the student how and where to add parentheses.
       4. Ask the student to type and execute this equation in a new code cell
       5. Ask the student the expression that multiply 10 to this equation, i.e., (8-(6-5)) \*10. The student needs to understand that there needs another pair of parentheses
       6. Ask the student to modify the code to generate the new expression, and execute
       7. Do these:

(9-(30-12)) \*2

(2+(70/2)) - 1

(4 + (2\*2))/2

* + 1. ~~Parentheses practices. Ask the students to execute the following equations. First three with instructor assistant, second three without instructor assistant~~  
       ~~Note: need to describe the equation to the student first,~~
       1. 5+ 5/2
       2. (5+5)/2
       3. (2 + 4.1) / (3-1)
       4. 3 \* (2+1)
       5. (9 + 0.2) \* (1 + 2)
       6. (5-(100-12)) \*8
  1. Task 4: Playing Sound (30 minutes)
     1. Overview and Goal
        1. Goal:
           1. To give the students a chance to use their code to do something fun
           2. As a complex example to test their independent coding skills
        2. Overview:
           1. write some string,
           2. do some math, assign the result to a variable, generate a string to describe the result, let use Speech-to-Text to convert the
     2. Steps
        1. Part 1
           1. Install gTTS in a new code cell:  
               !pip install gTTs
           2. In a new code cell, write and execute the following. The instructor needs to explain the code.  
              Before executing, let the student review the code.  
                
              from gtts import gTTS

from IPython.display import Audio   
s = “Hello, my name is Neo” # or any name

t = gTTS(s, slow=True)

t.save(“hello.wav”)

sound\_file = “hello.wav”

Audio(sound\_file, autoplay=True)

* + - * 1. Go back and check the string to something else
      1. Part 2
         1. In a new code cell, write an equation, e.g.,   
            51\*36\*37
         2. Change the line to assign the result to a variable, e.g.,   
            a = 51\*36\*37
         3. Use the following code to speak the result in a new code cell.  
            Explain the code, and let the student review the code before execution.  
              
            t = gTTS(str(a), slow=True)

t.save(“math.wav”)

sound\_file = “math.wav”

Audio(sound\_file, autoplay=True)

* + - * 1. Go back the let the student change the math equation, and data type
  1. Wrap-up
     1. Let the student share the Colab file with us. We need to send the files back to the parents
     2. Ask for feedback from the students and parents
     3. Additional questions for the students:
        1. Can you screen read pictures
        2. How do you read math plots
        3. Any math problem you want to learn how to solve with a computer?