

SESSION 2

Introduction to Python Programming

AOLME Curriculum Level 1

GOALS

1. Apply basics of Python Programming.
2. Program basic operations and variables in Python.
3. Solve and create own operations, and arithmetic-algebraic expressions.
4. Program a number guessing game using Python.



Activities:

1. Exploring Python Programming (print, strings, basic operations)
2. Programming Operations with Python (linear equations, order of operations, and complex expressions)
3. Creating Your Guessing Game w/ Python (design, program, and share game by altering initial, base codes)

1	2
3	4

Each activity includes 1 or 2 cards. One side of the card is in Spanish and on the back, the same information is in English. Each card has 4 quadrants, each quadrant includes a task related to the main goal of the activity. The numbers in the square on the left describe the order to perform the tasks. The card must be at the center of the table. Students need to have access to it and take turns reading it. They can read it in the language they feel more comfortable.

2.1. EXPLORING PROGRAMMING WITH PYTHON

Activity 1 Goal:

Activity 1 Goal: Apply basics of Python Programming

Resources for the Activity

1. Activity Card
2. Folder: /home/pi/AOLME/Session 2/
3. the “Jupyter” notebook accessed via Terminal
4. Raspberry Pi and Monitor
5. Student journal

Interactions

In this activity students are challenged to think about computer programming. It is important that students get to think about the idea of how programming is like a set of steps to accomplish something. Then, use everyday life examples and have students talk and see programming related to their own lives. The game in this session aims at helping student think about programming this way instead of seeing programming as something too complicated. Always use student experiences as a starting point and then build on ideas. While in the guidelines for this activity there are four tasks, these tasks only describe ideas and how they could be processed. Throughout the implementation of the activity, provide a friendly environment, supporting participation from everyone. Notice who participates more or less and pay attention to why it might be and act on it, so participation can be more even from everyone. Support at all times the use of the language (Spanish or English) that the students want to use.

Activity 2.1. Card:



2.1. Exploring Programming with Python

Note: The numbers in the Card match the Tasks numbers in the Jupyter Notebook.

1. **Practice Python programming using Jupyter.** Open the terminal: . Go to AOLME directory & type in: `jupyter notebook`. Hit Enter & in Jupyter open 'Session 2' folder. Click New and then click Python 3.

2. **To play the Guessing Game:** (a) open Session 2 file in Jupyter,

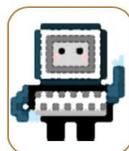
(b) In Task 2, select the cell named: EXAMPLE with game code,

(c) click 'run'

(d) at bottom, enter data it asks for

(e) take turns

(f) clear cell by clicking on 'run' again and enter new data.



In file you created, navigate Jupyter by exploring what each icon on the the taskbar does. What's a cell?

SAVE your work every time by clicking on this icon.

Copy & Paste

1. Select

2. Copy

3. Paste

A cell looks like a space to enter information. You can copy and erase cells..

3. Run & play CELLS A-C in Task 3. What do you notice about the print command?

4. **Discuss & Create:** Based on what you did and learned in task 3, in TASK 4 write your own code. Take turns typing.

Be creative and have fun!



Why does the cell color change?

These boxes are (Python or Text) cells. Some have: 'In []' and others don't. Why?

TASK 3: Assignments and 'print' Commands

The examples in the 3 cells (A, B, C) below will help us learn the print command. When you Computers process code in order, line by line. See below:

```
In [4]: # CELL A: What will the 'print' command do?
# Run CELL A and think about it and discuss with your friends.
x = 5
print(x)
5

In [ ]: # CELL B: The red text below in programming is called String.
# What do you think the print command does?
# What happens when you change the String?
name = input("What is your name ??? ")
hi = "Hello " + name + "!" # What do the plus signs do?
print(hi)

In [ ]: # CELL C: Python can process complex expressions.
# Run and think about what the code does.
x = 100
print(10000*x + 100000)
```

What do you notice about the red words and what appears below the cells? What if you changed the red words?

Have you played the guessing game before?

AOLME PROJECT - LEVEL 1- SESSION 2- 2019

MATERIALS DEVELOPED BY THE AOLME PROJECT AT THE UNIVERSITY OF NEW MEXICO, PLEASE DO NOT COPY OR DISTRIBUTE ANY OF THESE COPYRIGHTED TASKS WITHOUT PROPER AUTHORIZATION

Recommended Steps for the Activity

Task 1: Make sure that for the first task students open the 'Untitled' file in Session 2 folder. Have them type and try all icons in taskbar. Have them explore and then talk about what they discover. They might make connections to what they've done in other computers. Have students use related vocabulary when they talk about the tasks.

Task 2: When playing the guessing game, let students take turns playing game. Maybe by pairs? Taking turns typing? Use colors to think about functions of commands, texts, strings, and comments.

Task 3: Have students realize the different types of cells (text only and programming). it's important to notice 'print' and what if they didn't typed print? What happens? Let them experiment and especially notice that the red text or strings can be changed to any language even, but green words can't be changed. Highlight the idea of the 'print' command and ask students to think about other ways of using the command besides the examples they are doing. Have students take turns entering the data and running the cells. Use talk moves when needed, especially when someone says something important and so to highlight that either rephrase yourself or ask someone to do that, but always, make it fun! Have students make frequent decisions.

Task 4: Have students debrief what they learned by experimenting and have them write notes in journal.

Content: What is Python?

People use languages to communicate with each other (e.g., sign language, Chinese, Spanish, Native languages), programmers also use languages -or “programming languages”- to communicate with computers. Python is one of those programming languages.

Python Shell as a Calculator

Python Shell can also be used as a calculator to perform the four basic mathematical operations such as addition (+), subtraction (-), multiplication (*) or division (/).

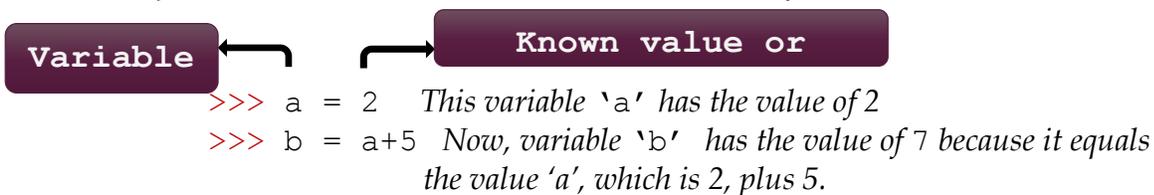
Type the operations suggested in the table below. And, **how do you run each operation command to get the answer from the computer? ...YES! pressing enter.**

Variables in Python

Have you studied Algebra? Algebra expresses values and relations through symbols such as letters and numbers. For example: $x + 4 = 10$

What is the value of “x”?

- ⊕ The use of variables is fundamental to mathematics and programming.
- ⊕ **In Python, a variable can be represented by a letter or a word which may hold a known or unknown value. For example:**



Different types of variables

Look at the codes below, run each operation in Python Shell, and write the answers.

The diagram shows two columns of code execution. On the left, two arrow-shaped boxes labeled 'Integer' and 'Float' point to the code blocks. Each code block has a title bar (INTEGER or FLOAT) and a terminal window showing the code and the prompt for the answer.

Type	Code	Prompt
Integer	<pre>>>> a = 5 >>> b = int(a)</pre>	_____
Integer	<pre>>>> a = 5.3 >>> b = int(a)</pre>	_____
Float	<pre>>>> a = 7.1 >>> b = float(a)</pre>	_____
Float	<pre>>>> a = 6 >>> b = float(a)</pre>	_____

String

```

STRING
>>> a = '10'
>>> print a

```

```

STRING
>>> a =
'Hello!'
>>> print a

```

>>> Game = "guessing number" Here the variable 'Game' is represented by a word and has a value of guessing number

Principal variables in Python	
Integer	<ul style="list-style-type: none"> ⊕ Referred to in programming as int ⊕ It is used to represent positive and negatives integer values. ⊕ The function <code>int()</code> converts a value into a integer value.
Float	<ul style="list-style-type: none"> ⊕ Referred to in programming as float ⊕ It is used to represent real values. ⊕ The function <code>float()</code> converts a value into a real value.
String	<ul style="list-style-type: none"> ⊕ It is represented by surrounding data with quotes "" ⊕ It is used to represent text, numbers and symbols.

2.2. PROGRAMMING NUMBER OPERATIONS W/ PYTHON

Activity 2 Goals:

- (a) Program basic operations and variables in Python.
- (b) Solve and create own operations, and arithmetic-algebraic expressions.

Resources for the Activity

1. Activity Card
2. the “Jupyter” notebook accessed via Terminal
3. Folder: /home/pi/AOLME/Session 2/
4. Raspberry Pi and Monitor
5. Student journal

Interactions:

Throughout the activity provide a friendly environment, supporting the participation of everyone. The goal is not to see how math someone knows but to think about how to use programming for doing math. If needed, have team members talk about the math that they know and ask why or how they think the results are coming about/from. For example, if one enters a 3 and then 1.5 comes, explore how that change happened should be explored regarding the math ideas and the programming code. Also, notice who participates more or less and pay attention to why it might be and act on it, so participation can be more even from everyone. Support at all times the use of the language (Spanish or English) that the students want to use.

Activity 2.2. Card:

2.2. Programming Number Operations w/ Python

Note: The numbers in the Card match the Tasks numbers in the Jupyter Notebook.

1. Let's use Algebra and Mathematics to program the Guessing Game with Python.



In the Guessing Game, we don't know the number the person is thinking. All we know is the operations we ask them to do. Next, focus on how the operations relate to the number to be guessed.

3. In CELL 2B in **jupyter**, there are variables (x, y and z). *What is a variable?* Run this Cell multiple times and change the numbers, operations and re-name the variables.

How do you think we can use variables and operations to guess a number we don't know?

2. In CELL 2A in **jupyter**, run the code multiple times & discuss what happens. Then make & predict changes.

```
# CELL 2A: Run this cell and enter a number.  
# How does your number change? Why?  
# Can you explain what the lines below do?  
# After playing a couple of times, take turns making changes  
# and predict what will happen.  
x = int(input("Enter a number: "))  
print("Multiplying x by 2 gives: 2*x =", 2*x)  
print("Adding 3 to the result gives 2*x+3 =", 2*x+3)  
Enter a number: x = 3  
Multiplying x by 2 gives: 2*x = 6  
Adding 3 to the result gives 2*x+3 = 9
```

The number entered for 'x' is 3. How are the 6 and 9 related to 3?

enter other numbers, explore how they relate.

4. In CELL 2C, create your own game by modifying the operations and replacing the '?' with numbers. Re-name variables, if wanted.



Talk: How do you think Python helps us do math? How are programming and thinking alike?

AOLME PROJECT - LEVEL 1- SESSION 2- 2019

MATERIALS DEVELOPED BY THE AOLME PROJECT AT THE UNIVERSITY OF NEW MEXICO, PLEASE DO NOT COPY OR DISTRIBUTE ANY OF THESE COPYRIGHTED TASKS WITHOUT PROPER AUTHORIZATION

Recommended Steps for the Activity

Task 1: Talk about what is Algebra in math and have students made connections between algebraic processes, variables, and programming, use the code in Jupyter to think about these ideas. Use the code as evidence to support the argument/ ideas.

Task 2: Motivate students to use related vocabulary as they refer to the red, or green text, like strings, comments, commands as they operate the variables. Also make sure the idea of variable is understood by everyone. Use prediction as way to learn how students are thinking about math and programming. Make sure all students participate in the process.

Task 3: Make sure that students understand that variables can be determined in many ways. Have them identify and talk about them every time they define them. When running each cell, make sure to have them predict and then try to identify what each code lines does. Perhaps changing numbers can help figure out better what the codes do. Discuss how the order of operations, grouping and definitions of variables might help solve a problem.

Task 4: As you create a game, make sure to include everyone's ideas. Have each other check on mathematical and programming accuracy. If something goes wrong, have students themselves find the issue. Interject only when needed or if the students cannot realize what the issue is.

Functions	Meaning
print	print displays the expression you want to type. Just place your text in quotation marks after the word print. For example, <code>print "Think of a number between 1 and 10"</code>
input ()	input is a function that will save your text and store it as a string in a variable. For example, <code>answer = input("What's the result?")</code>

answer is a string variable

input () saves the number the user inputs

Content: Data Flow in a Computer System.

Let's look at the function integer used in the example below.

Function	Meaning
int ()	The int () function will convert a variable (in the example the variable is the word "answer") into an integer (into a positive or negative number). Initially, the variable answer is a string (text or symbol) when it is taken in using raw_input() function. However, we need to convert the variable answer into a number, so it can be divided by another number, in this case, the number 3. <code>answer = input("What's the result?")</code> <code>print ("Your initial number was:")</code> <code>print (int(answer)/3)</code>

We have learned about functions, variables, and basic mathematical operations and that is all you need to start programming the Math game in Python!

2.3. CREATING YOUR NUMBER GUESSING GAME W/ PYTHON

Activity 3 Goal:

Program a number guessing game using Python.

Resources for the Activity

1. Activity Card
2. Folder: /home/pi/AOLME/Session2
3. the "Jupyter" notebook accessed via Terminal
4. Raspberry Pi kit
5. Student journal

Activity Card 3:

2.3. Creating Your Number Guessing Game w/ Python

Note: The numbers in the Card match the Tasks numbers in the Jupyter Notebook.

What makes this game work?

1. The Number Guessing Game.
To make the game work, we need to think about inverse operations.
Why do we need inverse operations for the number guessing game to work?

2. Inverse Operations: Multiplication is inverse to division
Consider the following for a game:
Give me a number x
Multiply by 2. $2*x$
Divide by 2. $(2*x)/2=x$
After these steps, we are back to x . We "divide by 2" because it is the inverse of "multiply by 2". So it helps get rid of what we multiply by 2. *Why do we want to get rid of values we enter? What's the inverse operation of addition? Why?*

3. Think: What would this game do?
Give me a number. x
Add 3. $x+3$
Subtract 3. $(x+3)-3=x$

What would be some values or numbers to combine in the guessing game?

4. Design & program in your team a number guessing game. Take turns typing in the codes. Save your work. Use the code in cell 2.3. to start your game. Then, ask someone from another team to play your game.

Use paper to check and predict whether your game will work.



Discuss & write in your journal:

- What did you learn about programming variables and equations?
- How do variables and operations make the game work?

Recommended Steps for the Activity

Task 1: Have students reflect on how numbers cancel each other through inverse operations. Have students practice or make little games with inverse operations. Write the equations when doing it. Ask the students links they see between equations, inverse equations and Have

Task 2: Have them first practice on paper getting rid of numbers to 'guess' the number. Compare with them how the terminal is similar of different from GUI

Task 3: Promote collaboration by listening to and helping each other. An error is just a step to get better. Try not to tell them, instead ask questions to prompt their thinking. Invite members from other teams to play the game of your team.

Task 4: Let students collaborate and have fun and encourage creativity and communication. Have students debrief what they learned at the end of the session and write in their journal at least 3 thoughts. Motivate students to say more than, it was fun!