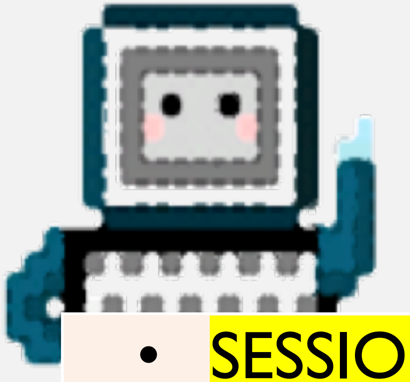
THE UNIVERSITY *of* NEW MEXICO

AOLME CURRICULUM

SESSION 2



LEVEL 1

- SESSION 1: ***Basic of Raspberry PI and Linux-****(motivational overview of projects-images, ls, cd)*
- SESSION 2: ***Introduction to Python Programming*** *(print, algebra, strings, binary images)*
- SESSION 3: ***Algorithms*** *(for and while loops, range commands, if statements, inequalities)*
- SESSION 4: ***The Coordinate Plane and Black & White Images in Python***
- SESSION 5: ***Binary and Hexadecimal number systems***
- SESSION 6: ***Images and Their Components (histograms)***
- SESSION 7: ***Creation of Images and Video***

INTRODUCTION TO PYTHON PROGRAMMING

OBJECTIVES:

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.

2.1. Exploring Programming with Python

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

Resources for the Activity

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

Everyone in the team gets to play a role:
Discussion Expert: Leads the team discussion asking questions about what the session is about.
Fair Participation Expert: makes sure of fair participation of everyone.
Hardware Setup/Teardown Expert: in charge of setting up & putting away materials and computer equipment.
Summary Expert: summarizes and records team questions and what the teams has learned.

Recommended Steps for the Activity

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 the discover. They might make connections to what they've done in other computers.
2. Have students realize different types of cells.
3. When playing guessing game, let students take turns playing game, maybe by pairs? Taking turns typing?
4. Task 3 it's important to notice 'print' and what if they didn't typed print? What happens? Let them experiment.
5. Have students debrief what they learned by experimenting and have them write notes in journal.

Activity 1 Goal: Apply basics of Python Programming.

2.2. Programming Number Operations w/ Python

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

Resources for the Activity

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

Recommended steps for the Activity

1. Make sure that students understand that variables can be determined in many way. Have them identify and talk about them every time they define them
2. 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.
3. Discuss how the order of operations, grouping and definitions of variables might help solving a problem.

Activity 2 Goal 1: Program basic operations and variables in Python. **Goal 2**: Solve and create own operations, and arithmetic-algebraic expressions.

2.3. Creating Your # Guessing Game w/ Python

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

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

Evaluate how did the team roles work?

Discussion Expert: Leads the team discussion asking questions about what the session is about.

Fair Participation Expert: makes sure of fair participation of everyone.

Hardware Setup/Teardown Expert: in charge of setting up & putting away materials and computer equipment.

Summary Expert: summarizes and records team questions and what the teams has learned.

Recommended steps for the Activity

1. Have students reflect on how numbers cancel each other through inverse operations.
2. Have them practice on paper getting rid of numbers to 'guess' the number.
3. Let students collaborate and have fun and encourage creativity and communication
4. Try not to tell them, instead ask questions to prompt their thinking.
5. Invite members from other teams to play the game of your team.

Activity 3 Goal: Program a number guessing game using Python.