

SESSION 3

Loops, Conditionals, & Sequential Thinking

AOLME Curriculum Level 1

GOALS

1. Introduce students to the notion and relevance of loops, conditionals and sequential thinking in programming.
2. Understand the link between algorithms and mathematics.
3. Practice pseudocodes through sequential thinking.



Activities:

3.1. Using Loops to Count

3.2. Guessing Numbers: Using Loops & Conditionals

3.3. Moving AOLME Sprite w/ Python IDLE

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.

3.1 USING FOR LOOPS TO COUNT

Activity 1 Goal:

Introduce students to the notion and relevance of loops, conditionals and sequential thinking in programming.

Resources for the Activity

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

Interactions

In this activity students are to think how loops can help them create sequences of numbers as if they were counting. It is important that they recall what they about counting and use it in what they are learning about programming. Have students get into the ideas of the commands and notice how the commands have a thought, an idea that is embedded in the code. Have them explain with own words, and maybe use diagrams, how they understand what loops do. Throughout the activity provide a friendly environment, supporting the participation of 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 1 Card:

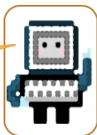
3.1. Using for Loops to Count

Note: The numbers in the Card match the Tasks numbers in the [Jupyter Notebook](#).

1. Go to the terminal `>` to open **Jupyter**:

- type `jupyter notebook`
- click enter
- go to the directory `/home/pi/AOLME/Session 3/`
- open the file : **Session 3-Loops**

Once in this file, note that under title of activity 3.1, there are 3 CELLS: A, B, C



2. **Run & Analyze:** In CELLS A, B, and C there is some Python code. Pay attention to the code in each cell and run them. What happens when you run `▶` each CELL? When you change some of the numbers in the code and run it, what happens?

What differences do you notice between CELLS C and A & B?

```
start_i = 1
step_i = 1
i = start_i
print(i)
```

In CELL B, what if you changed `step_i` value to 5?

Use comments in the CELLS as a guide.

3. **Think & Create:** Now it's your time to show what you've learned. In PRACTICE CELLS A, B, and C type in the codes and run them to see how you are completing the challenge. A little typo can be an issue for the code not to run.

When a code is entered in the right order and without typos or errors, then your code **syntax** is good and the code will run. **What's syntax?**

Talk about the syntax of this code: `for i in range 4):`

4. **Discuss & Write:** Based on the tasks you've done, discuss in your team: *What is a for loop?* Analyze the *for loop* statement below that we used. What parts of the statement could be changed, what words?



`for i in range(start_i, stop_i, step_i):`

Write down in your journal what you've learned.

AOLME PROJECT - LEVEL 1- SESSION 3- 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: Have students use laptop to access Jupyter, review ideas of how to navigate directories and using Linux commands to do so. Use cards from session 1 if needed. Move between using the card and the CELLS in Jupyter

Task 2: Have students take turns running each CELL, but also have them take turns analyzing the code line by line, sometimes they can predict without trying, sometimes they can start explaining more when they make small changes to the code, like changing numbers. After they see the difference between the previous results and the new ones they can infer what the code does. Have them write down what they discover. If they don't see it, ask them questions that may help them realize the links between results and the code.

Task 3: Highlight what syntax is by noticing the errors of the code in green letters. Have students make connections to how syntax is included in language like plurals and singular of a noun, like computer and computers.

Task 4: Have students discuss about what they learned about what a loop is. Have them share the ideas and combine what they think. Add some ideas only when needed. Support their understanding of the parts of the loop and what makes it work or not. Use the examples on Jupyter to have them write what they learned.

3.2 GUESSING NUMBERS: USING LOOPS & CONDITIONALS

Activity 2 Goals:

Understand the link between algorithms and mathematics.

Resources for the Activity

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

Activity 2 Card:

3.2. Guessing Numbers: Using Loops & Conditionals

Note: The numbers in the Card match the Tasks numbers in the [Jupyter Notebook](#).

1. To program the game, we need **conditional statements**. Computers understand conditions. For example, if $x = 5$, then stop counting. *How do we know it is a condition? Do your math teachers give you conditions? How are the conditions in CELLS similar or different from each other?*

Run: CELLS A & B to explore conditions. Try out different numbers.

```
number = int(input(" Input your number "))
secret_number = 5
if (number == secret_number):
    # Which number(s) will execute the code below?
    # For number=???, the following code works:
    print("You guessed it!")
    print("You win.")
```

In the code, where's the 'condition'?

What do strings do here?

3. **Run & Compare:** What happens in CELLS 3.A and 3.B? Why?

3.A. `while(0):` 3.B. `while(1!=2):`
`print(1)` `print(1)`

Change 2 to 1. What happens? Why?

It's called an infinite loop. *What's the condition in the loop?*

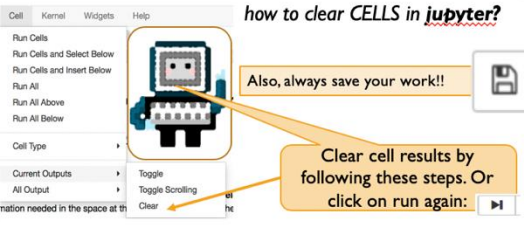
Run and change values in CELLS C and D.

Which of these ideas can you use in your game?

`!=` means **not equal** `while (computer_guess_number != secret_number):`
Why is it useful here?

If your code runs forever, press stop

2. **Clear CELLS** to enter new values or numbers. Remember *how to clear CELLS in Jupyter?*



Also, always save your work!!

Clear cell results by following these steps. Or click on run again:

4. **Design:** In CELL E create your game. Change the code as you need. Plan the steps to play the game.



Think & Compare:

How are these statement different? What do they do?

-`for i in range(start_i, stop_i, step_i):`
-`if (number == secret_number):`
-`while (number != secret_number):`

Think: what code will help you create your game?

AOLME PROJECT - LEVEL 1- SESSION 3- 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: Have students think about conditions in everyday life and how conditions are presented or look in the computer. Perhaps have them think about a condition a teacher would give them and write that in a code. Help students see the colors of the code and not how strings and code combine in the programming of conditionals. Have students predict and rethink what the code does before and after running it. Have them point to the lines of the code they are referring to.

Task 2: Motivate students to take turns saving their work and using the keyboard.

Task 3: have students describe their ideas about what codes 3a and 3b do and how they connect to not equal. Encourage them to use own words and ideas of how the ! not equal sign could be used in other context and what it does the While code.

Task 4: Create the game by referring back to examples they came up with. Make sure that as they plan their program, they know and name what type of codes they are using like, this is a for loop, a conditional, a while, and describe why those codes would go at certain parts of the program or what they do.

3.3 MOVING THE AOLME SPRITE WITH PYTHON IDLE

Activity 3 Goal:

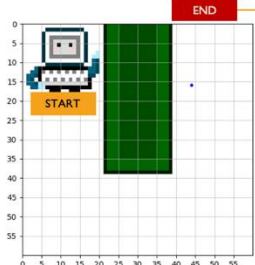
Practice pseudocodes through sequential thinking.

Resources for the Activity

1. Activity Card
2. Folder: `/home/pi/AOLME/Session3/` and file: `game.py`
3. the “Jupyter” notebook accessed via Terminal
4. Raspberry Pi kit
5. Student journal

Activity 3 Card:

3.3. Moving the AOLME Sprite with Python IDLE




. Think & Plan: AOLME wants to go around the park. Can you help him? Think: *How much and in what directions will AOLME need to move to go around the park?*

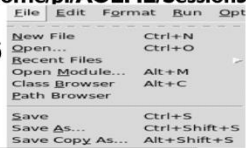
Think & make a plan.

2. To access AOLME game:

- Click on **Python IDLE** icon.
- In IDLE top bar, click on **File**.
- Select **Open** and navigate to `/home/pi/AOLME/Session3/`
- Open the file: `game.py`
- Run the code by pressing **F5**

Also run code by pressing 

Always save your work.



3. Think & Program: Below there are codes for you to use and help AOLME go around the park.

```

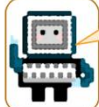
from AOLME game import *
scene.block_move_sprite down('aolme')
scene.block_move_sprite right('aolme')
scene.block_move_sprite up('aolme')
scene.block_move_sprite left('aolme')
show = scene.play_game()
```

Last line shows the scene.

Why do we need this line?

These commands help AOLME move. Which ones will you use?

4. Write: In your team discuss what you learned in session 3 and write main ideas of what you learned about programming loops, conditional statements, etc.



1 <3 walking!

Here are some ideas to write about:

- for i in range(start, i, stop, i, step, i):
- if (number == secret_number):
- while (number != secret_number):

Use cards 3.1. & 3.2 to remember.

AOLME PROJECT - LEVEL 1- SESSION 3- 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: Have students think about how they can move AOLME without programming and then use those ideas with the codes.

Task 2: Use IDLE, click on IDLE icon on raspberry PI to see aolme

Task 3: Motivate students to take turns typing in or copying and pasting the codes to help AOLME move or walk around the park. Highlight the meaning of each line of code.

Task 4: Have students debrief what they learned at the end of the session and write in their journal at least 3 thoughts.