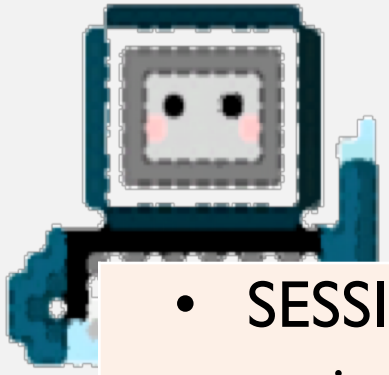
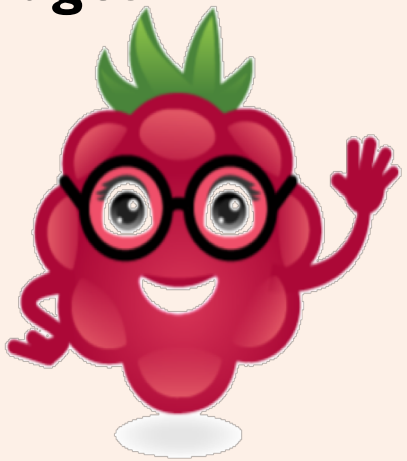
AOLME CURRICULUM

SESSION 3



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)*
- SESSION 3: ***Algorithms: Loops, Conditionals, and Sequential Thinking*** *(for and while loops, range commands, if statements, inequalities, sprite movement)*
- 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***
- FINAL PROJECT: VIDEO



LOOPS, CONDITIONALS, & SEQUENTIAL THINKING

OBJECTIVES:

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.

SELECTED ACTIVITIES

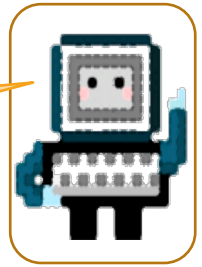
1. Using Loops to Create Number Sequences
2. Guessing Numbers: Using Loops & Conditionals
3. Moving AOLME Sprite w/ Python IDLE

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)
```

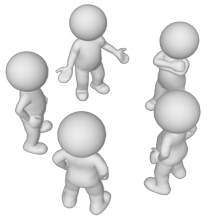
In CELL B, what if you changed step_i value to 5?

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: **fox** 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.

3.1. Using for Loops to Create Number Sequences

Resources for the Activity

1. Activity Card
2. Folder: /home/pi/AOLME/Session 3/
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 students use own words to define loop and how they see it working in computer programming.
2. Have students work with different cells and compare the code across.
3. When playing with numbers have students come up with own examples too.
4. Have students debrief what they learned by experimenting and have them write notes in journal.

Activity 1 Goal: Introduce students to the notion and relevance of loops, conditionals and sequential thinking in programming.

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 these conditions similar or different from each other?*

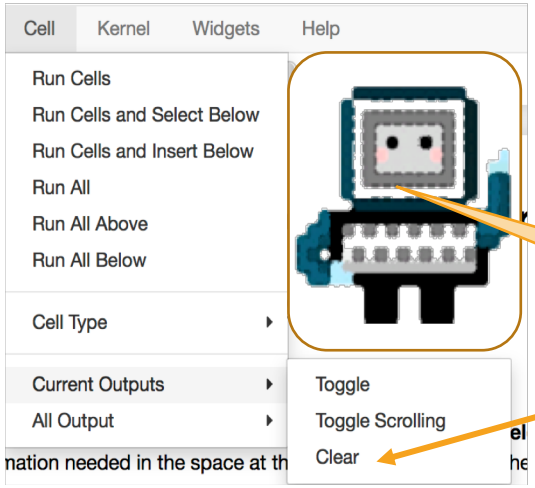
In the code, where's the 'condition'?

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.")
```

What do strings do here?

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:

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

3.A. `while(0):`
`    print(1)`

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

If your code runs forever, press stop

Change 2 to 1. What happens? Why?

People call it an infinite loop. *What's the condition in the loop?*

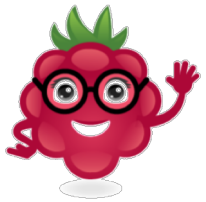
Run and change values in CELLs C and D.

What ideas can you use in your game?

!= means **not equal**
Why is it useful here?

```
while (computer_guess_number != secret_number):
```

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



Think: what code will help you create your 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):`

3.2. Guessing Numbers: Using Loops & Conditionals

Resources for the Activity

1. Activity Card
2. the **"Jupyter"** notebook accessed via Terminal
3. Folder: /pi/AOLME/Session 3/
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.

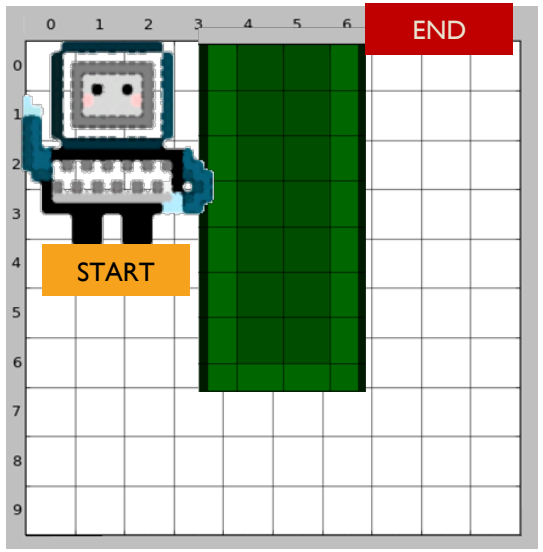
Summary Expert: summarizes and records team questions and what the teams has learned.

Recommended steps for the Activity

1. Make sure that students understand the conceptual understanding of loops and conditionals (if) as they 'play' with the code.
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. Contrast what codes do across cells.
4. Discuss how the order of operations, grouping and definitions of variables might help solving a problem.

Activity 2 Goal: Understand the link between algorithms and mathematics.

3.3. Moving the AOLME Sprite with Python IDLE

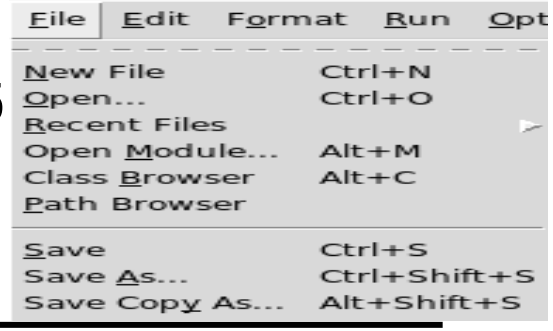


1. **Think & Plan:** AOLME wants to go around the park. Can you help him? Think: *How much and in what directions will AOLME need to move to go around the park?*

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.

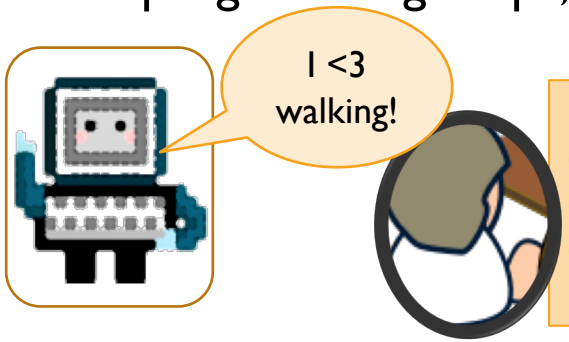
```
from AOLME_game import *

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()
```

Why do we need this line?

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

Last line shows the scene.

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.



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.

3.3. Moving AOLME Sprite with Python IDLE

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

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 much and in what direction AOLME needs to move.
2. Make sure students 'play' with the code
3. Let students copy and paste codes for moves. The goal is to think with the codes.
4. Try not to tell them, instead ask questions to prompt their thinking.
5. Have students debrief and write about what they learned in session 3

Activity 3 Goal: Practice pseudocodes through sequential thinking.