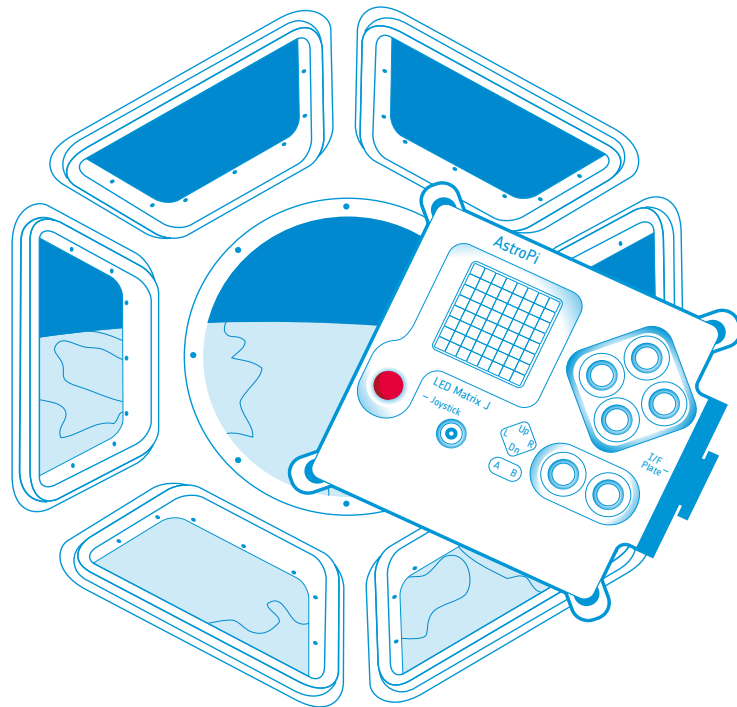
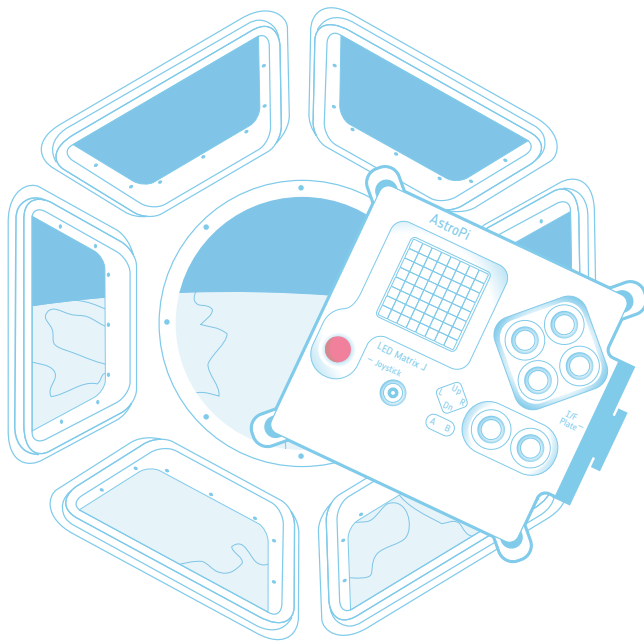


teach with space

→ GETTING STARTED WITH ASTRO PI

Using Raspberry Pi to understand programming language





European Astro Pi Challenge

European Astro Pi Challenge	page 3
Activity 1 – Meet Raspberry Pi	page 4
Activity 2 – Meet Python	page 6
Activity 3 – Do a quiz with Python	page 8
Activity 4 – Draw geometrical shapes with Python	page 10

Welcome to the European Astro Pi challenge!

Astro Pi is the name of a small computer developed by the Raspberry Pi Foundation, in collaboration with the UK Space Agency and the European Space Agency (ESA).

There are two very special Astro Pi's. Their names are Ed and Izzy, and they have been qualified for spaceflight. They are now onboard the International Space Station (ISS) for students like you to use.

With this first set of activities you are going to assemble your Raspberry Pi for the first time and learn the essential coding language that you need to program your Astro Pi. By doing so, you will get the chance to have your own code and scientific experiment run on the ISS.

Figure A1



↑ Astro Pi logo

Equipment

- Astro Pi Kit
- Monitor
- USB keyboard
- USB mouse

Fast facts

Age range: 10-14 years old

Complexity: easy

Location: indoor room

Includes use of: Astro Pi Kit; monitor;
USB keyboard and USB mouse

Outline

In this set of activities, students will set up and program a Raspberry Pi computer. The objective is to learn basic programming skills.

Students will learn

- What a Raspberry Pi is, what its main features are, and how to set up and use one
- The difference between hardware and software
- What a programming language is
- How to program using Python
- To use input and output messages, variables, data types, 'if' statements and loops
- How to use the Turtle module from Python to draw geometrical shapes

→ ACTIVITY 1: MEET RASPBERRY PI

Raspberry Pi is a computer, much like the computers with which you're already familiar, except for its size and package. You can use Raspberry Pi to surf the internet or send an email, or you can turn it into an MP3 music player and help the astronauts on the ISS have some fun during their free time. But you can also do much more! You can carry out scientific experiments using this computer tool and analyse the collected data - just like space scientists do!

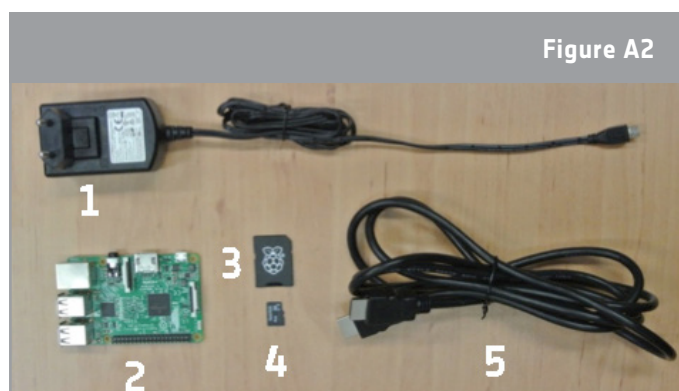
In this activity you are going to set up your Raspberry Pi and explore what you can do with it.

Exercise

1. To start your Astro Pi mission you will need special equipment that you can find in your Astro Pi kit. You will also need a monitor, a keyboard, and a mouse. In Figure A2 you can see all the equipment that you will need.

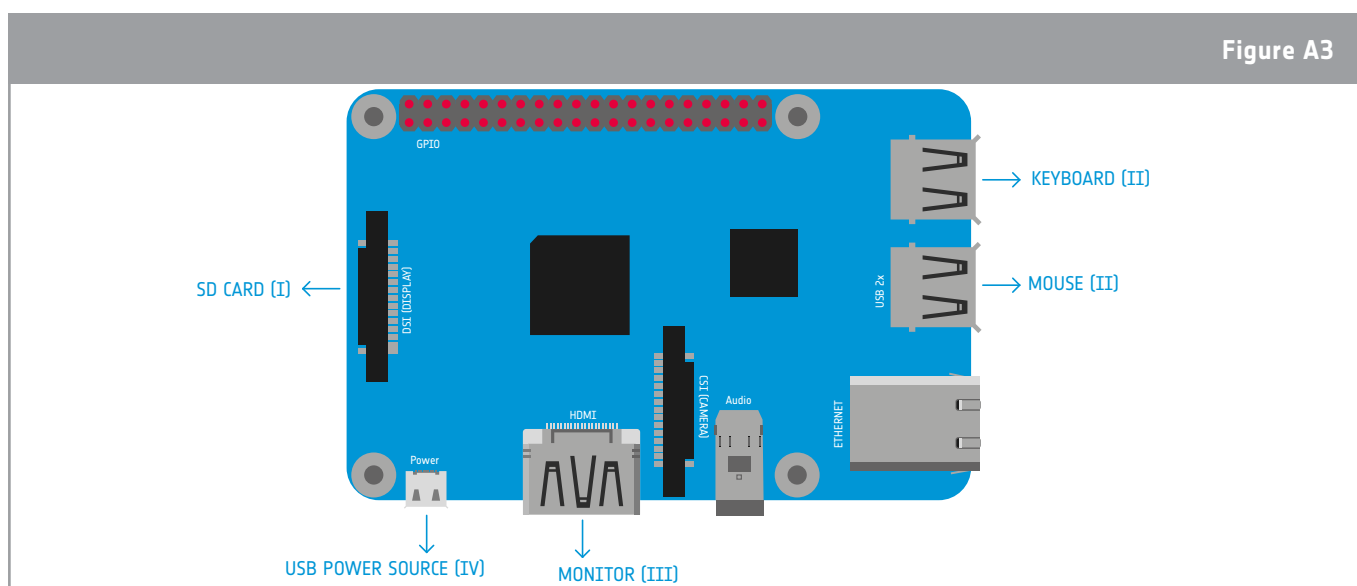
These components are the hardware. Label the different parts of the Raspberry Pi computer seen in Figure 2 (Micro SD card, SD card adaptor, Raspberry Pi 3, power supply, HDMI Cable).

1. _____
2. _____
3. _____
4. _____
5. _____



↑ Components of Raspberry Pi

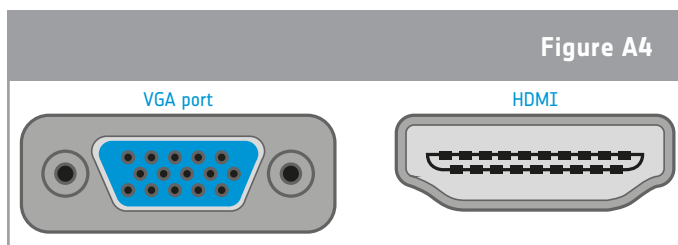
2. Now you are going to assemble your Raspberry Pi! In order to do that, follow instructions I to IV, and see Figure A3 below.



↑ How to assemble your Raspberry Pi

- I. Remove the SD card from the SD card adapter. Place it into the SD card slot on Raspberry Pi. It will only fit one way.
- II. Plug your keyboard and mouse into the USB ports.
- III. Connect the HDMI cable from the monitor (or TV) to the HDMI port on Raspberry Pi and turn on your monitor.

TIP: This can be the trickiest step when assembling your Raspberry Pi. Make sure that you have selected the right input (e.g. HDMI, DVI, etc). In case your monitor only has VGA (see Figure A4), you have to use the HDMI to VGA adapter (in the KIT).



- IV. Plug in the micro USB power supply. This action will turn on and boot your Raspberry Pi. This can take up to 5 minutes. You should see the Raspberry Pi flash, and text should appear on the monitor. Once the Raspberry Pi has completed booting and the text has stopped whizzing past on the screen, you should see the graphical environment as shown in Figure A5.



↑ The desktop environment or Graphical User Interface

3. You can now start using the Raspberry Pi software. You can, for example, open a web browser and surf the Internet. To do so, click on the main menu at the top of the screen (the Raspberry logo) and choose Internet > Chromium Web Browser (you must have access to wifi to be able to use Internet). Use the web to try to find the current location of the ISS. Write it down.

4. Explore the Raspberry Pi software by choosing different options from the Programming Menu. After doing this, can you explain the differences between hardware and software?

Note: To turn off your Raspberry Pi, don't just pull the plug. Whenever you have finished working with it, shut it down by clicking on the Shutdown button in the main menu.

Extension

Not everything is what it seems to be! Make a list of digital devices that you consider to be computers and that you use in your everyday life. Compare your list with your peers and discuss what makes a computer a computer!

→ ACTIVITY 2: MEET PYTHON

For you to communicate with the astronauts on the ISS using Raspberry Pi, you have to give it the necessary instructions. Like any other computer, it will only do what you teach it or what you program it to do. For that you need to learn a programming language (the same way that you've had to learn your own language to communicate with the world around you!). With this activity you will learn the basic vocabulary of the Python language and write your first program.

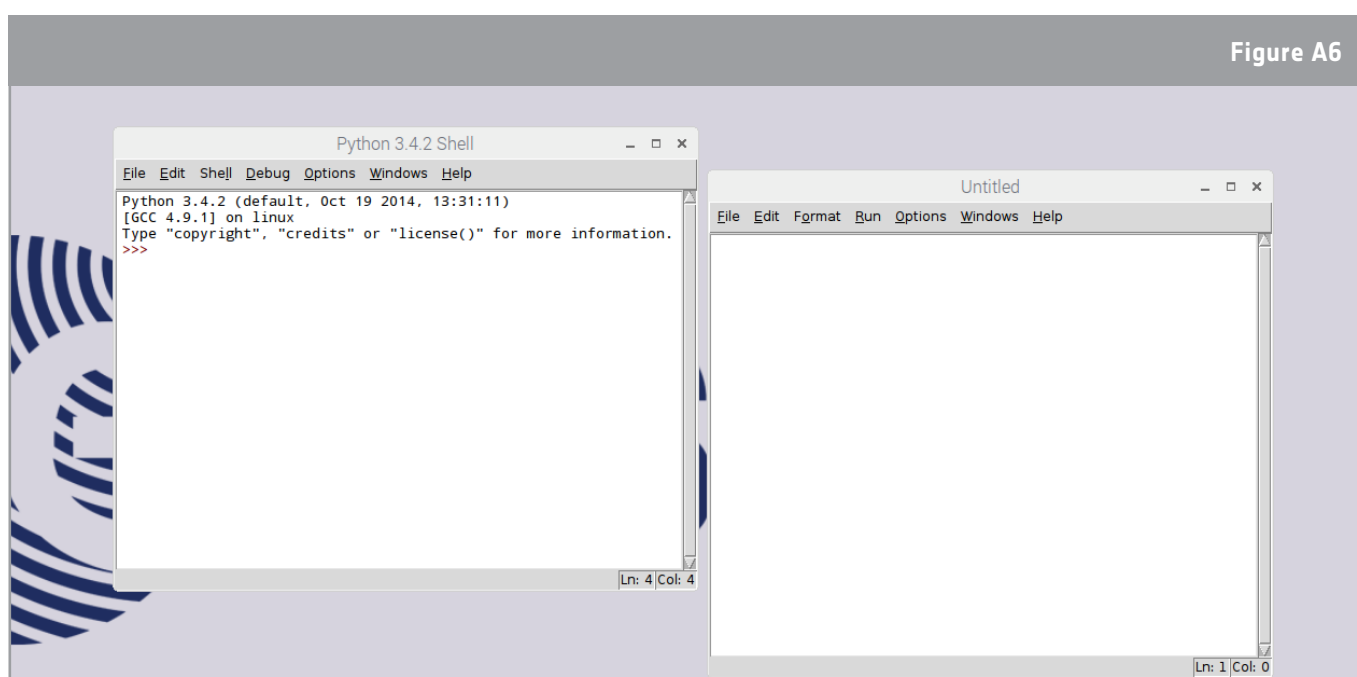
Did you know?

Astronauts also have to learn different languages before they are allowed to go to space. For example, all astronauts need to learn Russian because the controls of the Soyuz spacecraft are in Russian. Currently the Soyuz is the only spacecraft available to transport people to and from the ISS. Soyuz means 'union' in Russian. Although it was conceived by the Soviet Union at the start of the space exploration era, the Soyuz spacecraft is still used today, but with important modifications.



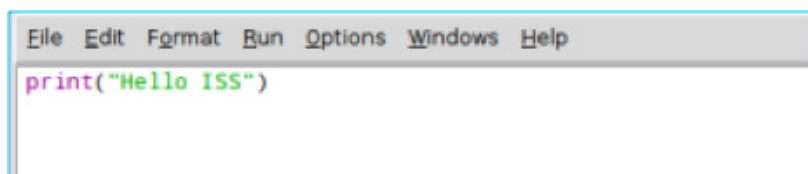
Exercise

1. To start using Python in your Raspberry Pi, you have to click on the main menu at the top of the screen (Raspberry logo) and select Programming > Open Python 3(IDLE). This will open an output window called Python Shell (Figure A6 below). You can type your program directly into this shell, although it's easier to work in an editor window. To open a new window, select File> New File. You should have two windows open now. One for writing your program (the editor window), and another for showing output (the Python Shell) like results, error messages, etc. Make sure both windows are visible.



↑ Python Shell (left) and editor window (right)

2. You are now ready to start coding in Python. Enter the following code into the editor window.

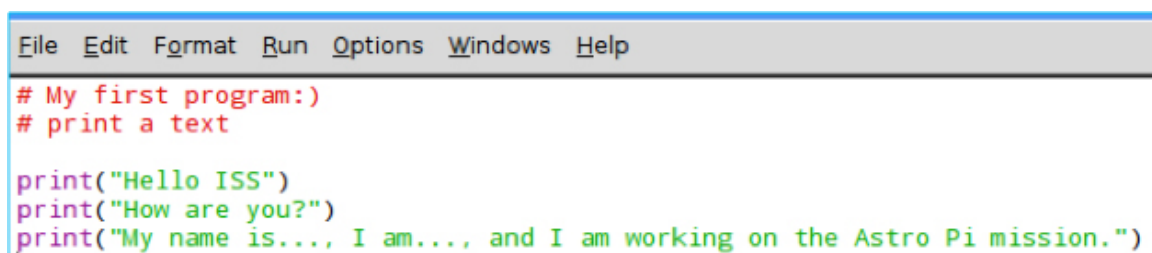


```
File Edit Format Run Options Windows Help
print("Hello ISS")
```

3. In order to see the results of your coding, you need to choose Run > Run Module. The program will ask you to save your code. Write a name for your file and run it. What did it do?

Note: Remember to save your work on a regular basis! In order to save the file you are working on, choose File->Save.

4. It's time for you to introduce yourself to the astronauts living on the ISS! Copy the code below to your file and complete it with the message that you would like to share with the astronauts.



```
File Edit Format Run Options Windows Help
# My first program:)
# print a text

print("Hello ISS")
print("How are you?")
print("My name is..., I am..., and I am working on the Astro Pi mission.")
```

Congratulations! You have just created your first program in Python language.

Note: The computer will try to understand everything you write in the window. If you want to write a short comment only for yourself, use the # sign in front of your short note. This is called 'commenting', and a comment will not be translated into machine language. The computer will simply ignore text written after #.

→ ACTIVITY 3: DO A QUIZ WITH PYTHON

So far you have given simple instructions to the computer. Now you will make the computer ask you questions and react to your answers.

Exercise

1. Open a new Python 3 window and type the code below.

```
File Edit Format Run Options Windows Help
# whenever a word or number follows the '=' symbol, it is called a variable.
# variables are used to store information that the computer must remember.

orbits = input (" How long does it take the ISS to orbith the Earth?")
```

2. Select Run>Run Module to execute your code. What happened?

3. Python uses the input () function to ask you questions. Do you know how long it takes the ISS to orbit the Earth? Type it in the Python shell.
4. You can also use the Python Shell as a calculator. See the following examples.

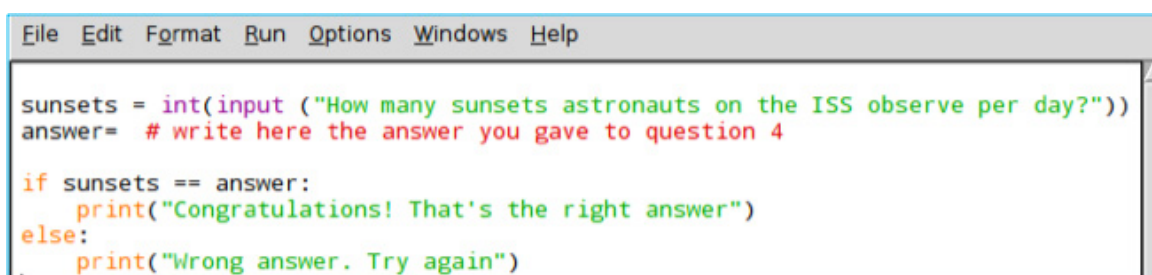
```
File Edit Shell Debug Options Windows Help
Python 3.4.2 (default, Oct 19 2014, 13:31:11)
[GCC 4.9.1] on linux
Type "copyright", "credits" or "license()" for more information.
>>> 12+12
24
>>> 12*2
24
>>> 12/2
6.0
>>> 12*2-12
12
```

Try to answer the question below by using the Python Shell to do the necessary maths. Explain the calculation used.

How many sunsets do astronauts on the ISS observe every day?

5. You will now do a scientific quiz using Raspberry Pi. For that you will need to select the right and wrong answers. In Python language we can do that using **selection** statements, sometimes also referred to as '**if statements**'. Open a new file and write the code below. You have to complete the code writing the answer. What do you think this program is going to do? Run the module to see.

Note: The code '**int**' is in front of '**input**' in order to tell Python that your answer will be an integer number. An integer number is a whole number, such as 5 or 10. In Python, if you want to treat the information as a number type, you will need to give that instruction to the computer. Besides integer numbers, you can also have a float number. A float number is a number with decimals, such as 5.1 or 10.99.



```
File Edit Format Run Options Windows Help
sunsets = int(input ("How many sunsets astronauts on the ISS observe per day?"))
answer= # write here the answer you gave to question 4

if sunsets == answer:
    print("Congratulations! That's the right answer")
else:
    print("Wrong answer. Try again")
```

6. You can give some extra help during the quiz. Just before the '**else**' line, add the command below, and run your module again. Write down a comment to explain what the '**elif**' keyword does.

#

```
elif sunsets >13 and sunsets <16:
    print("Almost there! Try again")
```

Extension

'If statements' can check for a list of things instead of just one thing. For that you only have to use the number of '**elif**' statements you need. Write down a program that will ask the user to input a number between 45 and 60. Give a response which tells the user if the number is 'within range', 'too high', or 'too low'.

→ ACTIVITY 4: DRAW GEOMETRICAL SHAPES WITH PYTHON

There are many modules in Python that you can use in your programs. Think of a module as an instruction book that explains how to do something. These modules are stored by Python in a library. In this activity you will learn how to use the 'Turtle module' from Python to draw geometrical shapes while learning more Python language that will be useful for your mission.

Exercise

1. Open a new Python 3 window and type the code below. Save and run it.

```
File Edit Format Run Options Windows Help
# to begin using Turtle you need to import the Turtle from the Python library
from turtle import Turtle, Screen

screen = Screen() # creates the window which will display the turtle drawings
turtle = Turtle() # creates a turtle. The variable (ou noun) turtle is made to refer to it.
turtle.shape("turtle") # gives the turtle shape
```

2. You have just created the turtle that will help you make nice drawings and shapes! Now you will give some instructions to your turtle.

- 2.1. Add the following line to your code:

```
turtle.forward(100)
```

What happened to the turtle?

- 2.2. Add another instruction:

```
turtle.right(90)
```

Change the numbers in the brackets. Try to change 'right' to 'left'. What is the turtle doing with these instructions?

Note: The value 90 inside the brackets in `turtle.right(90)` is in degrees.

- 2.3. Add another instruction underneath to move your turtle again:

```
turtle.forward(100)
```

What happened to the turtle?

2.4. What do you need to add to your code in order to complete a square? In the box below, write the final code needed for your turtle to draw a square.

3. You can give many different instructions to your turtle. Inspect the following code.

```
File Edit Format Run Options Windows Help
from turtle import Turtle, Screen

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

turtle.forward(100)
turtle.right(120)
turtle.forward(100)
turtle.right(120)
turtle.forward(100)
```

3.1. Which geometrical shape do you think your turtle is going to draw? Write down your answer.

3.2. Save and run your code. Were you correct?

3.3 In the code above, you repeated a few lines of code. In computer science, this is not the most efficient way of doing it. Instead of typing many lines of code, you can write them once and add an instruction to repeat them. In Python there are two ways for you to repeat an instruction: a while loop and a for loop.

3.3.1 Open a new Python 3 window and write the code below. The code inside the loop (i.e. the code which is indented or pushed to the right) is the set of instructions that the program is going to repeat. Save and run your program. What did it do?

```
File Edit Format Run Options Windows Help
from turtle import Turtle, Screen

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

turtle.forward(100)
turtle.right(120)
turtle.forward(100)
turtle.right(120)
turtle.forward(100)
```

3.3.2 In this example, a for loop has been used. Copy the code below and run it to see what happens.

```
File Edit Format Run Options Windows Help
from turtle import Turtle, Screen

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

for i in range(3):
    turtle.forward(100)
    turtle.right(120)
```

Note: Did you notice the nesting of statements underneath the while/for loops? This is called indentation. Python uses indentation to delimit blocks of code when using loops or 'if' statements. Each level of indentation is of four spaces, but you can press Tab and it will insert spaces for you. Remember indentation when copying and pasting code.

3.3.3 Which loop would you choose to make the drawing? Explain your choice and explain the differences between the two types of loops.

Extension

You can go forward, right, and left, and you can do repetitions using loops. Now it's time to be creative! Can you make a drawing of ESA's mascot Paxi? Which geometrical shapes can you recognise in Paxi's image?



NEXT STEPS

You have just assembled your Raspberry Pi and now know Python's basic programming structure! Go to Activity 2 to learn how to use the Sense HAT LED matrix to communicate with the astronauts on the ISS.

teach with space – Getting started with Astro Pi | T05.1
www.esa.int/education

**An ESA Education production in collaboration with Raspberry Pi
Foundation, ESERO Poland and ESERO UK**

Copyright 2017 © European Space Agency