

Introdução ao Astro Pi

Usar Raspberry Pi para compreender linguagem de programação

Bem-vindo(a) ao desafio europeu Astro Pi!

Astro Pi é o nome de um pequeno computador desenvolvido pela Fundação Raspberry Pi, em colaboração com a Agência Espacial do Reino Unido (UK Space Agency) e a Agência Espacial Europeia (ESA, European Space Agency).

Existem dois Astro Pi que são muitos especiais. São chamados Ed e Izzy e foram considerados aptos para voos espaciais. Estão agora a bordo da Estação Espacial Internacional (ISS, International Space Station) para que estudantes como tu os possam usar.

Com este primeiro conjunto de atividades, montarás o teu Raspberry Pi pela primeira vez e aprenderás a linguagem de programação essencial de que precisas para programar o teu Astro Pi. Ao fazê-lo, terás a oportunidade de executar o teu próprio código e experiências científicas na ISS.



Equipamento

- Kit Astro Pi
- Monitor
- Teclado USB
- Rato USB

Atividade 1 – Conhecer o Raspberry Pi

O Raspberry Pi é um computador, muito semelhante aos computadores que já conheces, exceto no tamanho e pacote. Podes usar o Raspberry Pi para navegar na Internet, enviar um e-mail ou ainda transformá-lo num leitor de música MP3 e ajudar os astronautas na ISS a divertirem-se nos seus tempos livres. Mas podes fazer muito mais! Podes fazer experiências científicas com esta ferramenta informática e analisar os dados recolhidos – tal como fazem os cientistas espaciais!

Nesta atividade, irás configurar o teu Raspberry Pi e explorar o que podes fazer com ele.

Exercício

1. Para começares a tua missão Astro Pi, precisarás de equipamento especial que poderás encontrar no teu kit Astro Pi. Precisarás também de um monitor, teclado e rato. Na Figura 1, podes ver todo o equipamento de que precisarás. Estes componentes são o hardware. Marca as diferentes partes do computador Raspberry Pi, ilustradas na Figura 1 (cartão Micro SD, adaptador de cartões SD, Raspberry Pi 3, fonte de alimentação com pinos de ficha UE, cabo HDMI).

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

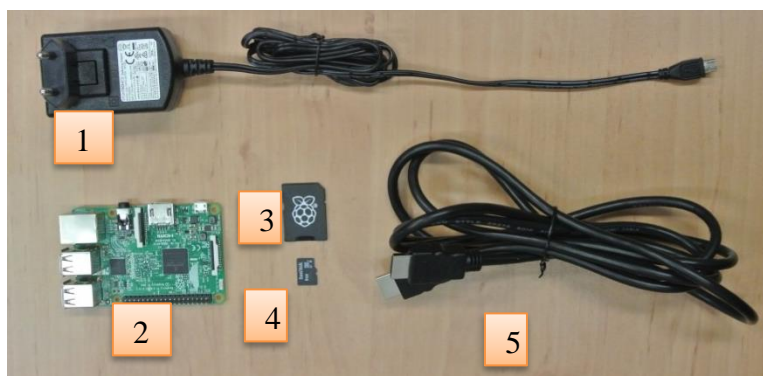


Figura 1 – Componentes do Raspberry Pi

2. Agora, vais montar o teu Raspberry Pi! Para fazê-lo, segue as instruções I a IV e vê a Figura 2 a seguir.

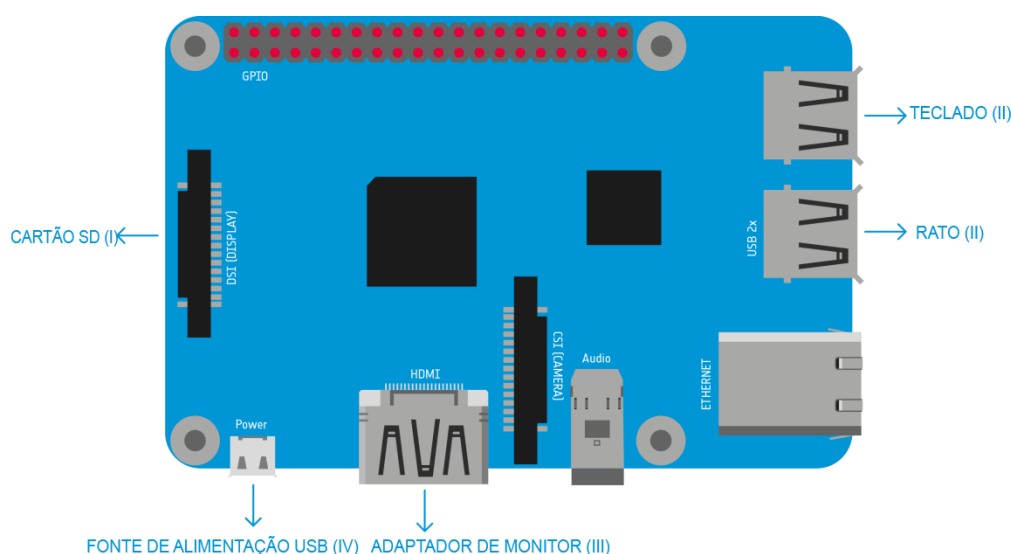


Figura 2 – Como montar o teu Raspberry Pi

- I. Retira o cartão SD do adaptador de cartões SD. Coloca-o na ranhura de cartão SD no Raspberry Pi. Só pode ser introduzido de uma forma.
- II. Liga o teu teclado e rato às portas USB.
- III. Liga o cabo HDMI do monitor (ou televisão) à porta HDMI no Raspberry Pi e liga o teu monitor.

CONSELHO: este pode ser o passo mais complicado ao montar o teu Raspberry Pi. Certifica-te de que seleccionaste a entrada correta (por ex., HDMI, DVI, etc). Caso o teu monitor tenha apenas VGA (vê a Figura 3), terás de usar o adaptador de HDMI para VGA (no KIT).



Figura 3 – Portas VGA e HDMI

- IV. Liga fonte de alimentação micro USB. Esta ação ligará e acionará o arranque do teu Raspberry Pi. É possível que demore até 5 minutos. Deverás ver o Raspberry Pi piscar e deverá surgir texto no monitor. Quando o Raspberry Pi terminar o arranque e o texto parar de deslizar no ecrã, deverás ver o ambiente gráfico, conforme mostrado na Figura 4.



Figura 4 – O ambiente de trabalho ou a Interface Gráfica de Utilizador

3. Podes agora começar a usar o software Raspberry Pi. Poderás, por exemplo, abrir um navegador Web e navegar na Internet. Para fazê-lo, clica no menu principal na parte superior do ecrã (o logótipo Raspberry) e escolhe *Internet > Chromium Web Browser* (deverás ter acesso a Wi-Fi para poderes usar a Internet). Usa a Internet para tentar encontrar a localização atual da ISS. Anota-a.

4. Explora o software Raspberry escolhendo diferentes opções a partir do Menu de Programação. Após teres feito isto, consegues explicar as diferenças entre hardware e software?

Nota: para desligares o teu Raspberry Pi, não retires a ficha da tomada. Sempre que terminares de trabalhar com ele, encerra-o clicando no botão Shutdown no menu principal.

Extensão

Nem tudo é o que parece ser! Faz uma lista de dispositivos digitais que consideres serem computadores e que usas no dia-a-dia. Compara a tua lista com a dos teus colegas e discute o que faz um computador ser um computador!

Atividade 2 – Conhecer Python

Para comunicares com os astronautas na ISS usando o Raspberry Pi deverás dar-lhe as instruções necessárias. Tal como qualquer outro computador, só fará aquilo que lhe ensinares ou aquilo para o qual o programares. Para isso, precisas de aprender uma linguagem de programação (da mesma forma que aprendeste a tua própria língua para comunicar com o mundo à tua volta!). Com esta atividade, aprenderás o vocabulário básico da linguagem Python e a escrever o teu primeiro programa.

Sabias que...?

Os astronautas também têm de aprender linguagens diferentes antes de poderem ir para o espaço. Por exemplo, todos os astronautas precisam de aprender russo, porque os controlos da nave espacial Soyuz estão em russo. Atualmente, a Soyuz é a única nave espacial disponível para transportar pessoas de e para a ISS. Soyuz significa “união” em russo. Embora tenha sido concebida pela União Soviética no início da era de exploração espacial, a nave espacial Soyuz é utilizada ainda hoje, mas com importantes modificações.



Exercício

1. Para começares a usar Python no teu Raspberry Pi, precisas de clicar no menu principal na parte superior do ecrã (logótipo Raspberry) e selecionar *Programming > Open Python 3(IDLE)*. Esta ação abrirá uma janela de resultados chamada Python Shell (Figura 5 à esquerda). Poderás digitar o teu programa diretamente nesta shell, embora seja mais fácil trabalhar numa janela de editor. Para abrir uma nova janela, seleciona *File> New File*. Deverás ter agora abertas duas janelas. Uma para escrever o teu programa (a janela de editor), e outra para mostrar os resultados de saída (o Python Shell), tais como efeitos, mensagens de erro, etc. Certifica-te de que ambas as janelas estão abertas.

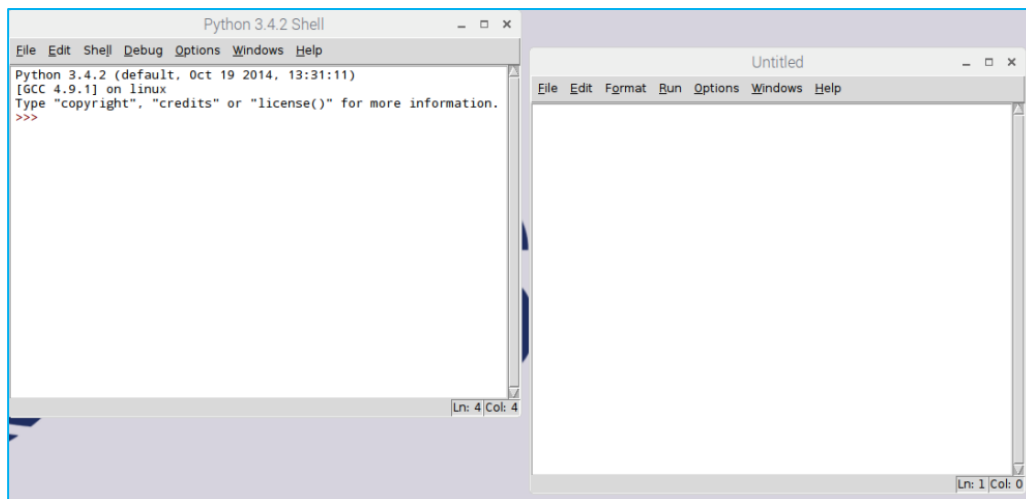


Figura 5 – Python Shell (esquerda) e janela de editor (direita)

2. Estás agora pronto para começar a criar códigos em Python. Introduz o seguinte código na janela de editor.

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

3. Para veres os resultados dos teus códigos, precisas de escolher *Run > Run Module*. O programa pedir-te-á para gravares o teu código. Dá um nome ao teu ficheiro e executa-o. O que fez?

Nota: lembra-te de guardares o teu trabalho regularmente! Para guardares o ficheiro em que estás a trabalhar, seleciona "File"->"Save".

4. Tenta apresentar-te aos astronautas que vivem na ISS. Copia o código abaixo para o teu ficheiro e completa-o com a mensagem que gostarias de partilhar com os astronautas.

Parabéns! Acabaste de criar o teu primeiro programa em linguagem Python.

```
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.")
```

Nota: o computador tentará compreender tudo aquilo que escreveres na janela. Se quiseres escrever um comentário apenas para ti, usa o símbolo # à frente da tua nota curta. A isto chama-se "comentar" e um comentário não será traduzido para a linguagem da máquina. O computador ignorará simplesmente o texto escrito após #.

Atividade 3 – Fazer um questionário com Python

Até agora, deste instruções simples ao computador. Agora, farás com que o computador te faça perguntas e reaja às tuas respostas.

Exercício

1. Abre uma nova janela Python 3 e digita o código a seguir.

```
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. Selecciona *Run>Run Module* para executares o teu código. O que aconteceu?

3. Python usa a função **input ()** para te fazer perguntas. Sabes quanto tempo demora a ISS a fazer a sua órbita à volta da Terra? Digita-o no Python Shell.

4. Podes também usar o Python Shell como calculadora. Repara nos próximos exemplos.

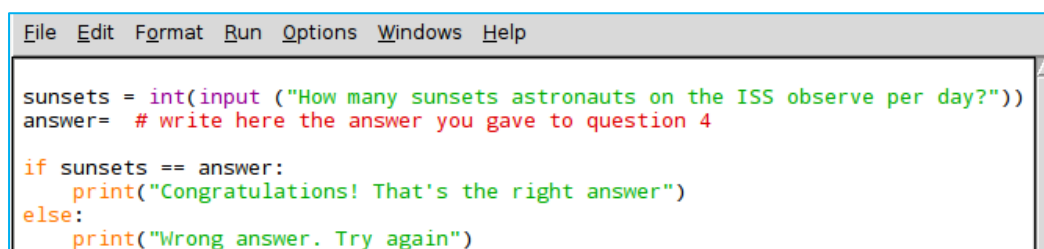
```
File Edit Shell Debug Options Windows Help
Python 3.4.2 (default, Oct 19 2014, 13:31:11)
[GCC 4.9.1] on linux
Type "copyright", "credits" or "license()" for more information.
>>> 12+12
24
>>> 12*2
24
>>> 12/2
6.0
>>> 12*2-12
12
```

Tenta responder à pergunta a seguir usando o Python Sheel para efetuar os cálculos necessários. Explica o cálculo utilizado.

Quantos pores-do-sol é que os astronautas observam na ISS todos os dias?

5. Farás agora um questionário científico usando o Raspberry Pi. Para isso, precisarás de seleccionar as respostas certas e erradas. Na linguagem Python, podemos fazer isto usando **instruções de seleção**, algumas vezes designadas **“instruções if”**. Abre uma nova janela e digita o código a seguir. Deves completar o código escrevendo a resposta. O que achas que este programa fará? Executa o módulo para ver.

Nota: o código **“int”** está à frente de **“input”** para dizer a Python que a tua resposta será um número inteiro. Tal como o nome indica, este é um número inteiro, como 5 ou 10. Em Python, se quiseses tratar a informação como um tipo de número, terás de dar essa instrução ao computador. Para além de números inteiros, poderás ter também um número flutuante (float). Um número flutuante é um número com casas decimais, tal como 5,1 ou 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. Poderás dar uma ajuda extra durante o questionário. Mesmo antes da linha **“else”**, adiciona o comando a seguir e executa o teu módulo novamente. Escreve um comentário a explicar o que significa a palavra-chave **“elif”**.

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

Extensão

“Instruções if” podem verificar uma lista de elementos em vez de 1 só. Para isto, só é preciso usares a quantidade de expressões **“elif”** de que necessitas. Escreve um programa que peça ao utilizador para introduzir um número entre 45 e 60. Dá uma resposta que diga ao utilizador se o número está **“dentro do intervalo”**, **“demasiado alto”** ou **“demasiado baixo”**.

Atividade 4 – Desenhar formas geométricas com Python

Existem muitos módulos em Python que podes usar nos teus programas. Pensa num módulo como um livro de instruções que explica como fazer algo. Estes módulos são armazenados por Python numa biblioteca. Nesta atividade, aprenderás a usar o “módulo Turtle” de Python para desenhares formas geométricas enquanto aprendes mais linguagem Python que te será útil para a tua missão.

Exercício

1. Abre uma nova janela Python 3 e digita o código a seguir. Guarda e executa-o.

```
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 wich 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. Acabaste de criar a tartaruga que te ajudará a criar bonitos desenhos e formas! Agora, darás algumas instruções à tua tartaruga.

2.1 Adiciona a seguinte linha ao teu código:

`turtle.forward(100)`

O que aconteceu à tartaruga?

2.2 Adiciona outra instrução:

`turtle.right(90)`

Muda os números entre parêntesis. Tenta alterar “right” (direita) para “left” (esquerda). O que está a tartaruga a fazer com estas instruções?

Nota: o valor 90 entre parêntesis em `turtle.right(90)` representa graus.

2.3 Adiciona outra instrução por baixo de forma a mover a tua tartaruga novamente:

`turtle.forward(100)`

O que aconteceu à tartaruga?

2.4 O que precisas de adicionar ao teu código para completar um quadrado? Na caixa a seguir, escreve o código final necessário para a tua tartaruga desenhar um quadrado.

3. Poderás dar muitas instruções diferentes à tua tartaruga. Verifica o seguinte código.

```
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 Qual é a forma geométrica que achas que a tua tartaruga desenhará? Escreve a tua resposta.

3.2 Guarda e executa o teu código. Acertaste?

3.3 No código acima, repetiste algumas linhas de código. Em informática, esta não é a forma mais eficiente de escrever código. Em vez de digitares muitas linhas de código, podes escrevê-las uma vez e adicionar uma instrução para as repetir. Em Python, há duas formas de repetir uma instrução: um *loop (ciclo) while* e um *loop (ciclo) for*.

3.3.1 Abre uma nova janela Python 3 e digita o código a seguir. O código dentro do loop (ciclo) (ou seja, o código com avanço ou empurrado para a direita) é o conjunto de instruções que o programa repetirá. Guarda e executa o teu programa. O que fez?

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

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

while True:
    turtle.forward(100)
    turtle.right(120)
```

3.3.2 Neste exemplo, foi usado um *loop (ciclo) for*. Copia o código a seguir e executa-o para veres o que acontece.

```
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)
```

Nota: reparaste no aninhamento de instruções por baixo dos loops (ciclos) while/for? A isto chamamos avanço. Python usa o avanço para delimitar blocos de código ou usar loops (ciclos) ou instruções "if". Cada nível de avanço tem quatro espaços, mas podes premir a tecla Tab e introduzir mais espaços. Lembra-te do avanço quando copiares e colares código.

3.3.3 Que loop (ciclo) escolherias para fazer o desenho? Justifica a tua escolha e explica as diferenças entre os dois tipos de loop (ciclo).

Extensão

Podes ir para a frente, direita e esquerda, podendo efetuar repetições usando loops (ciclos). Agora, chegou a altura de seres criativo! Consegues fazer o desenho da mascote Paxi da ESA? Quais são as formas geométricas que consegues reconhecer na imagem de Paxi?



PASSOS SEGUINTES

Acabaste de montar o teu Raspberry Pi e agora conheces a estrutura básica de programação de Python! Avança para a atividade 2 para aprender a usar a matriz LED do Sense HAT de forma a comunicares com os astronautas na ISS.