

Unidade 8: micro:bit com Python

Lição 1: O display

Nesta lição, vai escrever os seus primeiros programas em python para controlar o display do micro:bit de diferentes formas. Esta lição está dividida em duas partes:

Parte 1: encontro alienígena

Parte 2: exibir imagens

Objetivos:

- Controlar a exibição na placa do micro:bit usando `.show()`, `.scroll()` e `.show(imagem)`
- Controlar a velocidade de exibição usando `sleep(ms)`

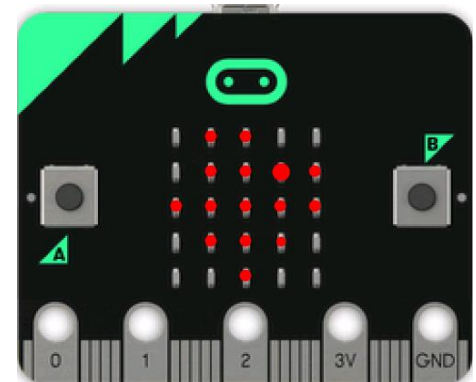
1. Antes de começar, certifique-se que:

- a sua TI-Nspire CX II tem Sistema Operativo **5.3** ou mais recente
- se sente confortável a programar em python e/ou completou as unidades 1 a 5
- o seu **micro:bit** está conectado com a tua TI-Nspire CX II
- seguiu as instruções de configuração e transferência de ficheiros no **Guia de iniciação** do micro:bit : <https://education.ti.com/pt/teachers/microbit>

Este processo de configuração apenas deverá ter de se feito uma vez, mas mantém-te informado periodicamente acerca de atualizações.

2. Se está tudo bem e a configuração foi realizada corretamente, o seu micro:bit deve apresentar-se assim quando tem energia da TI-Nspire:

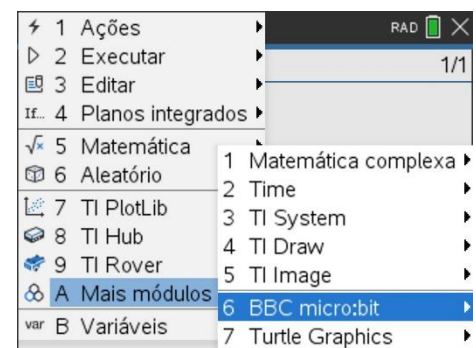
O display do **micro:bit** mostra o logótipo da TI, um ícone do estado do Texas com um ponto brilhante perto de Dallas, a casa da **Texas Instruments, Inc.**



3. E finalmente...

O módulo do micro:bit está instalado na tua Biblioteca Python. Num Editor Python, clica **[menu] > Mais módulos** e repara que **BBC micro:bit** se encontra listado na parte de baixo do módulos TI.

*Nota: No sistema operativo 5.3 e mais recentes, os módulos de Pyton que estão armazenados na sua pasta **Pylib** do seu dispositivo aparecem neste menu em adição aos **ti_modules**. A sua lista pode diferir da aqui mostrada. Os módulos estão listados alfabeticamente de acordo com o nome do ficheiro, logo **BBC micro:bit** encontra-se entre os “m’s” da lista, não entre os “B’s”.*



Dica para o professor: Iniciar com o micro:bit

Esta unidade parte do princípio de que o micro:bit e a unidade portátil TI-Nspire (ou o software de computador) estão prontos a usar. Certifique-se de que foram completadas (por si ou pelos alunos) as etapas de configuração nas calculadoras e micro:bits de cada aluno. Certifique-se de ler e executar as instruções de configuração que constam no 'Guia de Iniciação' que está incluído na pasta de download a partir do website TI Education. <https://education.ti.com/pt/teachers/microbit>. Como em todos os softwares e hardwares novos, mantenha-se informado acerca de atualizações.

Existe também um documento TI-Nspire de introdução, **my first program.tns**, que ajuda a garantir que o micro:bit se encontra conectado e a funcionar corretamente.

O **módulo BBC micro:bit (microbit.tns)** consiste num módulo especial fornecido pela Texas Instruments que fica instalado na pasta Pylib de cada unidade portátil e no software de computador TI-Nspire CX II e assim, aparece em **[menu] > Mais módulos > como BBC micro:bit**.

As primeiras cinco unidades de python: 10 minutos de código devem ser completadas antes desta unidade.

O Código aqui apresentado não é complicado, mas deve existir um certo nível de conforto na programação com código python na TI-Nspire CX II.

Esta unidade destina-se a trabalhar com as versões micro:bit 1 e 2 mas não aborda recursos exclusivos da versão 2.

O módulo BBC micro:bit suporta esses recursos.

Todos os projetos micro:bit encontrados online (incluindo projetos da versão 2 do micro:bit) podem ser desenvolvidos na TI-Nspire CX II, utilizando python e um micro:bit associado. Mas existem duas distinções importantes:

- O micro:bit não pode ser desconectado da calculadora enquanto o programa estiver a correr mesmo que exista uma bateria ligada ao micro:bit. O tempo de execução do micro:bit (.hex file) está configurado para obter as suas instruções a partir da calculadora. A calculadora está a fornecer energia e a controlar o micro:bit.
- Muitos dos programas encontrados online utilizam `While True:` para criar um ciclo infinito. Este ciclo corre diretamente na placa do micro:bit até o programa ser substituído por outro utilizando um .hex file do qual foi feito download. Quando usada a TI-Nspire, a calculadora está a controlar o micro:bit, pelo que o ciclo comumente utilizado é

While get_key() != 'esc':

o qual permite ao utilizador premir a tecla [esc] para terminar o programa.

Importante: Se a mensagem 'micro:bit não conectado' aparecer, desligue e volte a ligar o micro:bit (reset).

4. **Parte 1: encontro alienígena:** Como seria de esperar, tal como em qualquer outra primeira experiência de programação, irá começar por exibir uma expressão no display do micro:bit.

Comece um novo documento TI-Nspire e selecione **Adicionar Python > Novo** para iniciar um novo programa, com o nome 'saudacoes'. No Editor Python use **[menu] > Mais módulos > BBC micro:bit** para seleccionar o comando **import** no topo dos itens do menu:

from microbit import *

Dica: Se a mensagem 'micro:bit não conectado' alguma vez aparecer, desligue e volte a ligar o micro:bit (reset).

Dica para o professor: Se o micro:bit não estiver conectado, o próprio comando `import` irá reportar um erro quando o programa estiver a correr. Como o python é uma linguagem 'modular', recursos suplementares são adicionados à medida que forem necessários. Este comando `import` contém funções/métodos específicos necessários para operar com o micro:bit, utilizando a TI-Nspire CX II. Ele também contém alguns métodos úteis de outros módulos padrão e módulos ti.

5. Para exibir uma expressão no display do micro:bit, use o comando:

display.show()

Este comando é encontrado em:

[menu] > Mais módulos > BBC micro:bit > Display > Methods

O comando é inserido como **display.show(valor)**, mas (valor) é apenas um marcador de posição que terá de ser substituído por algo. Dentro dos parênteses, substitui `valor` escrevendo a sua expressão entre aspas:

“saudações, terráqueos”

Quando puser este programa a correr premindo **[ctrl] [R]** irá ver as letras da sua expressão aparecerem uma de cada vez, no display.

Se cometer um erro...volte à página 1.1 para editar o seu programa, e depois volte a correr o programa. Esqueceu-se de colocar o texto que queria exibir entre aspas?



```
1.1 *Doc RAD 3/3
*saudacoes.py
from microbit import *
```



```
1.1 *Doc RAD 3/5
saudacoes.py
from microbit import *
display.show("Saudações, terráqueos")
```



6. Um melhor método para exibir a expressão é:

display.scroll("saudações, terráqueos")

que também pode ser encontrado em

[menu] > Mais módulos > BBC micro:bit > Display > Methods

Para completar o comando `.scroll()` pode copiar/colar a expressão a partir do comando `.show`.

Transforme o anterior comando `.show()` num comentário (coloque o cursor nessa linha e prime **[ctrl] [T]**) para desativá-lo e depois coloque novamente o programa a correr.

Sim, também pode simplesmente alterar de `.show` para `.scroll` digitando.

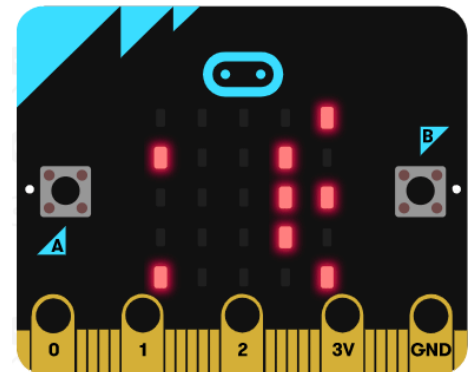
7. O método `.scroll()` faz com que a expressão se mova da direita para a esquerda como uma faixa através do display tornando mais fácil a leitura. Pode controlar a velocidade a que se move adicionando o parâmetro **delay=** :

display.scroll("saudações, terráqueos", delay = 200)

que faz com que haja um atraso de 200 milissegundos (0.2 segundos) na movimentação. Tente também com outros valores de delay.



```
1.1 *Doc RAD 5/6
saudacoes.py
from microbit import *
#display.show("Saudações, terráquios")
display.scroll("Saudações, terráquios")
```



<greetings, earthlings.gif>

8. **Parte 2: Seja.Ainda.Minha.Batida.Coração** prima **[ctrl] [doc]** para inserir uma página e selecione **Adicionar Python > Novo** para adicionar um novo programa Python ao seu documento (o nosso tem o nome 'batida').

No Editor Python, vá a **[menu] > Mais módulos > BBC micro:bit** e selecione o comando **import** no topo da lista:

from microbit import *



```
1.1 1.2 *Doc RAD 2/1
*batida.py
from microbit import *
```



9. Para exibir o símbolo do coração no display do micro:bit, utilize o comando:

```
display.show(valor)
```

Este comando encontra-se em:

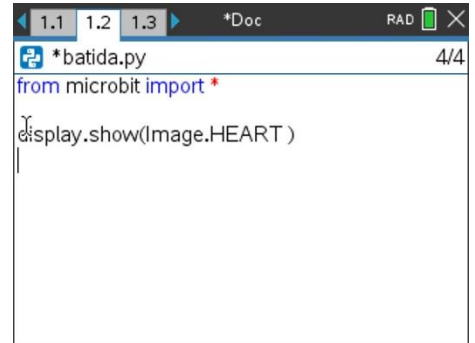
[menu] > Mais módulos > BBC micro:bit > Display > Methods

Entre os parênteses substitui “valor” selecionando:

Image.HEART

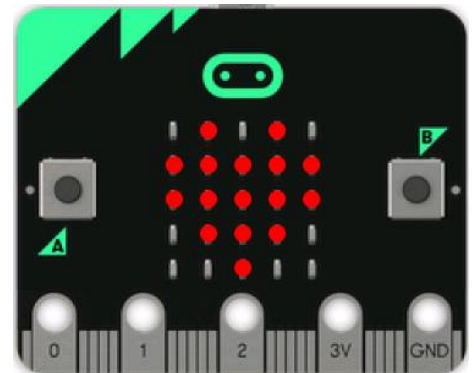
A partir de

[menu] > Mais módulos > BBC micro:bit > Display > Images > Set 1 > Heart



```
*Doc RAD 4/4
*batida.py
from microbit import *
display.show(Image.HEART)
```

10. Corra o programa (prime [ctrl] [R]) para ver o ícone do coração exibido na grelha 5x5 LED do micro:bit. Esta exibição mantém-se até que outra ocupe o seu lugar, mesmo depois do programa terminar.



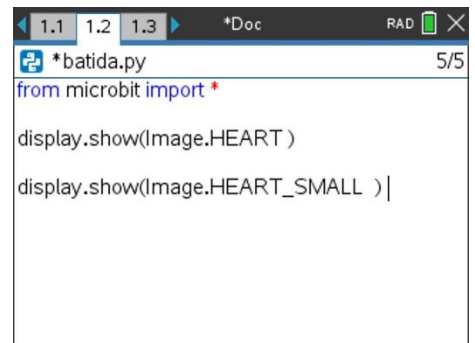
11. Volte ao editor do programa na página anterior e adicione outro comando de exibição para mostrar o coração pequeno:

```
display.show(Image.HEART_SMALL)
```

Pode encontrar esta imagem no mesmo menu de **Imagens**:

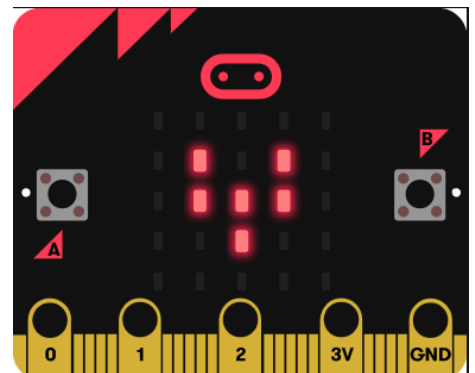
[menu] > Mais módulos > BBC micro:bit > Display > Images > Set 1

*Dica: Também pode copiar/colar o primeiro comando de exibição e editar (digita **_SMALL**). É necessário que tenha o underscore **_** e que esteja em maiúsculas.*



```
*Doc RAD 5/5
*batida.py
from microbit import *
display.show(Image.HEART)
display.show(Image.HEART_SMALL )
```

12. Corra o programa outra vez. Irá rapidamente ser exibido o coração maior seguido do mais pequeno, que tem este aspeto.



13. **Criar um ciclo:** Para fazer com que os dois corações pisquem repetidamente ('coração a bater'), inclui os dois comandos de exibição num ciclo. Antes dos dois comandos de exibição insira:

while get_key() != "esc":

encontrado em [menu] > Mais módulos > BBC micro:bit > Commands e faça a indentação dos dois comandos de exibição de modo a que estes formem o corpo do ciclo.

Dica importante: A Indentação é crucial em programação Python. É assim que o python interpreta blocos de ciclo e blocos condicionais. Se os dois comandos de exibição não tiverem uma indentação com o mesmo número de espaços irás ver um erro de sintaxe. Usa a tecla [espaço] ou a tecla [tab] para fazer a indentação das duas linhas na mesma quantidade. Neste Editor, os espaços de indentação estão indicados com símbolos em forma de losango e cor cinzento claro (◆◆), para ajudar a uma indentação apropriada.

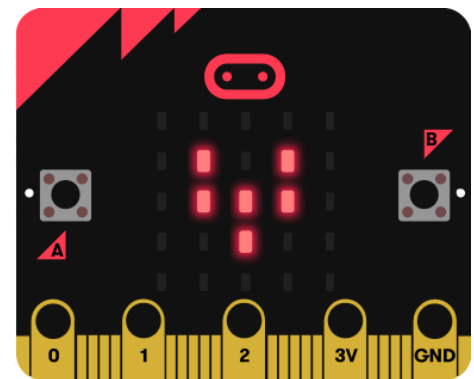
```

1.1 1.2 1.3 *Doc RAD X
*batida.py 2/5
from microbit import *

while get_key() != "esc":
    display.show(Image.HEART)
    display.show(Image.HEART_SMALL )
    
```

14. Corra o programa novamente e observe o coração a bater! Prima a tecla [esc] para terminar o programa.

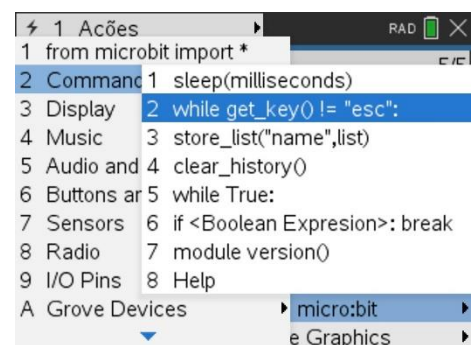
Dica: Se alguma vez lhe parecer que o seu programa está encravado num ciclo infinito pressione por alguns segundos a tecla [home/on] da sua TI-Nspire para 'interromper' o programa. Isto pode acontecer se utilizar while True: a partir do menu de comandos de forma não apropriada. Nestas lições evita-se esse tipo de estrutura.



<beating_heart.gif>

15. O menu de **Comandos** do micro:bit contém alguns comandos python úteis que também podem ser encontrados noutros menus. O módulo do micro:bit importa estes comandos python para si.

Pode ver estes e quaisquer outros comandos python a partir de outros menus. Não está limitado a utilizar apenas o menu BBC micro:bit mas pode ter de fornecer os comandos import apropriados.




16. Para controlar a velocidade com que o coração bate, adicione dois comandos **sleep()**, um a seguir a cada comando de exibição:

sleep(1000) significa um atraso de 1000 milissegundos (1 segundo).

Também pode ser encontrado em [menu] > **Mais módulos** > **BBC micro:bit** > **Commands**

Dica: Atenção à indentação!

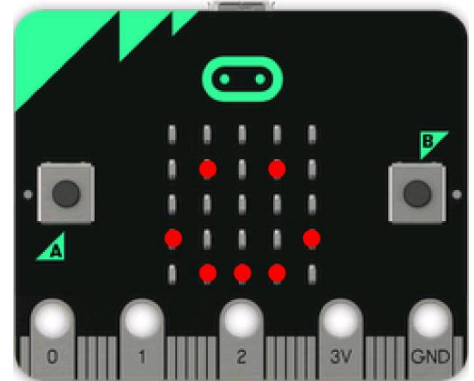


```

1.1 1.2 1.3 *Doc RAD
batida.py 7/7
from microbit import *

while get_key() != "esc":
    display.show(Image.HEART)
    sleep(1000)
    display.show(Image.HEART_SMALL)
    sleep(1000)
    
```

17. **Extensão:** Tente 'Criar Caras'. Para o programa use uma estrutura semelhante à do 'Coração a bater' mas utilize as imagens de caras encontradas no Set 1 em vez das imagens dos corações.



Dica para o professor: As lições online acerca do micro:bit usam tipicamente um **While True: ciclo em python** e um **forever: ciclo em MakeCode**. Estes 'ciclos infinitos' correm diretamente no micro:bit até este ser desligado ou até que o programa seja substituído por um outro (.hex file). Quando está a ser utilizada a TI-Nspire para controlar o micro:bit, o ciclo infinito não é necessário, uma vez que a calculadora tem o controlo total. **While True: encontra-se no menu de comandos mas deve ser utilizado em conjunto com o if (Expressão Booleana): interrompe para sair do ciclo infinito.**

Se o programa de um estudante ficar encravado num ciclo infinito na unidade portátil, de forma a 'interromper' o programa pressione por alguns segundos a tecla [on]. Consulte o 'Guia de Iniciação' (PDF) que veio com o software do micro:bit da TI para obter informação acerca de como interromper programas python no computador.

O menu de '**Comandos**' contém um conjunto de comandos python comuns e frequentemente usados que são úteis ao programar com o micro:bit. O módulo do micro:bit importa estes comandos a partir de outros módulos quando necessário. No entanto, todos os comandos python estão disponíveis a partir de outros menus e podem ser utilizados em projetos micro:bit. Utilizar os comandos import adequados é importante.

Opcional: Para que o display volte ao seu estado 'TI' (o logótipo TI do Texas) no final do programa, acrescente o comando

display.show(ti) *simplesmente digite as letras ti (em minúsculas)*

a seguir ao final do ciclo While (identado).

Isto não é necessário mas ajuda a indicar que o programa está terminado no display do micro:bit assim como no ecrã.

No início de um programa micro:bit (antes de começar o ciclo while) pode ajudar a imprimir algo no ecrã da calculadora como por exemplo ('A correr...').

Sobre o sleep(): A única função python que é significativamente modificada para o micro:bit é o comando sleep() que se encontra no módulo tempo. Normalmente, o valor do argumento representa 'segundos', mas o modulo do micro:bit revê o método sleep de modo a que o valor represente milissegundos (milionésimos de segundo). Depois de importar o modulo do micro:bit, sleep(1000) representa um atraso de um segundo. De notar que sleep() controla a velocidade do programa da calculadora e não do próprio micro:bit. Certifique-se de que from microbit import * ocorre após o tempo de importação (ou qualquer módulo que possa importar tempo) por esta razão.

The .show() method tem dois parâmetros adicionais: delay= e wait=
display.show(value, delay=xxx , wait=False/True)

delay value está em milissegundos e retarda o display.

wait=True irá bloquear o processamento até a animação estar terminada, caso contrário a animação acontecerá em segundo plano.

Aqui encontra-se a lista completa de imagens:

Set1	Set 2	Set 3	Set 4
1:HEART	1:YES	1:MUSIC_CROTCHE	
2:HEART_SMALL	2:NO	2:MUSIC_QUAVER	
3:HAPPY	3:TRIANGLE	3:MUSIC_QUAVERS	
4:SMILE	4:TRIANGLE_LEFT	4:PITCHFORK	
5:SAD	5:CHESSBOARD	5:XMAS	
6:CONFUSED	6:DIAMOND	6:PACMAN	
7:ANGRY	7:DIAMOND_SMALL	7:TARGET	
8:ASLEEP	8:SQUARE	8:TSHIRT	
9:SURPRISED	9:SQUARE_SMALL	9:ROLLERSKATE	
A:SILLY	A:RABBIT	A:DUCK	1:BUTTERFLY
B:FABULOUS	B:COW	B:HOUSE	2:STICKFIGURE
C:MEH	C:TI LOGO	C:TORTOISE	3:GHOST
			4:SWORD
			5:GIRAFFE
			6:SKULL
			7:UMBRELLA
			8:SNAKE

Um projeto semelhante é 'Criar Caras' usando a mesma estrutura do programa 'Coração a bater' mas utilizando os ícones de caras presentes no Set 1.