

# ASTROPI MISSION ZERO

Per prima cosa mettiamoci a gruppi di due e scegliamo un nome per il gruppo!



Questo tutorial e altri approfondimenti sono disponibili alla pagina

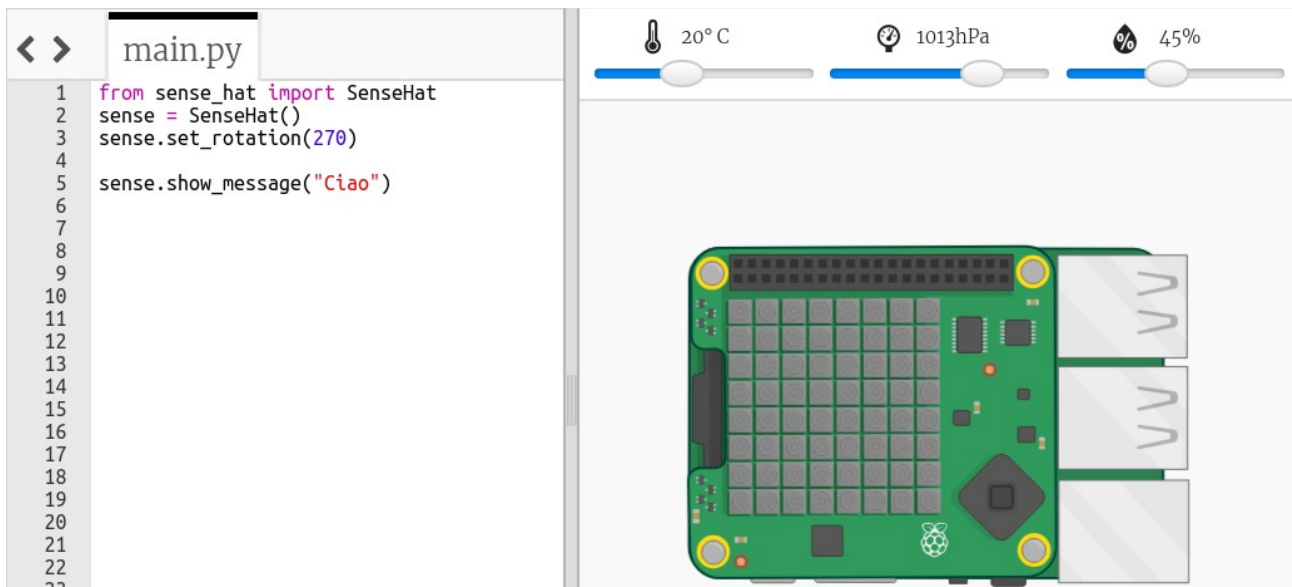
[coderdojotrento.it/astropi1](http://coderdojotrento.it/astropi1)

## Scriviamo un messaggio

Utilizzando Google Chrome (o Firefox), colleghiamoci alla pagina: [trinket.io/mission-zero](http://trinket.io/mission-zero)

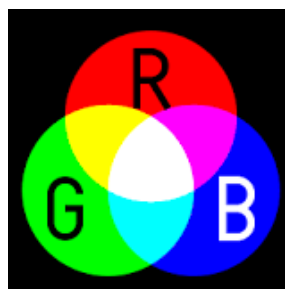
Per prima cosa facciamo dice "Ciao". Utilizziamo per questa prima parte di codice il comando:

```
sense.show_message("Ciao")
```



Tra le doppie virgolette "" è possibile inserire anche altri testi. Con il pulsante ► facciamo partire l'emulazione.

Vogliamo mettere del colore di sfondo? Possiamo fare delle prove definendo i colori (prima di utilizzarli!). Spesso i colori nei linguaggi di programmazione si definiscono nell'ordine con le componenti rossa (Red), verde (Green) e blue (Blue):



```
rosso = (255, 0, 0)
verde = (0, 255, 0)
blu = (0, 0, 255)
bianco = (255, 255, 255)

sense.show_message("Ciao", back_colour=rosso)
```

**Sfida:** prova a definire e usare il viola. E come si fa il giallo ? E il nero ?

Mostriamo il testo di colore verde, usando `text_colour` :

```
sense.show_message("Ciao", text_colour=verde)
```

**Sfida:** prova a scrivere testo verde su sfondo rosso. Come si potrà fare?

**Proviamo a utilizzare l'emulatore come se fosse una calcolatrice!!!**

Oltre ad una frase, possiamo anche scrivere dei valori, ma prima dobbiamo convertirli a testo con la funzione `str()`:

```
a=5

sense.show_message(str(a), back_colour=blu)
```

Possiamo quindi usarlo come una calcolatrice:

```
a=5
b=6
c=a+b
sense.show_message(str(c), back_colour=blu)
```

## Ricaviamo i dati dai sensori

Possiamo andare a prendere i dati dai sensori utilizzando le seguenti funzioni:

per la temperatura:

```
sense.get_temperature()
```

per l'umidità:

```
sense.get_humidity()
```

per la pressione:

```
sense.get_pressure()
```

Quindi se vogliamo ricavare la temperatura:

```
temperatura = sense.get_temperature()
```

Dal momento che la temperatura è un valore numerico, se la vogliamo visualizzare:

```
sense.show_message( str(temperatura) )
```

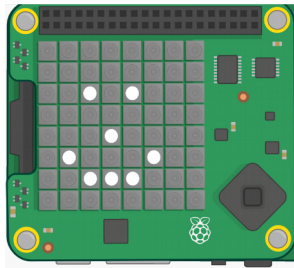
Vedrete che verrà fuori un numero con tante cifre decimali. Possiamo diminuirli scrivendo

```
temperatura = round( sense.get_temperature() ,1)
```

dove 1 è il numero di cifre decimali da mostrare.

## Creiamo un'immagine

In modo simile agli sfondi colorati di prima, possiamo anche creare delle immagini.



Definiamo prima i colori:

```
bianco = (255, 255, 255)
nero = (0, 0, 0)
```

e l'immagine, in una tabella 8x8 (tanti quanti sono i led in orizzontale e verticale):

```
immagine = [
    nero, nero, nero, nero, nero, nero, nero, nero,
    nero, nero, nero, nero, nero, nero, nero, nero,
    nero, nero, bianco, nero, bianco, nero, nero, nero,
    nero, nero, nero, nero, bianco, nero, nero, nero,
    nero, bianco, nero, nero, nero, bianco, nero, nero,
    nero, nero, bianco, bianco, bianco, nero, nero, nero,
    nero, nero, nero, nero, nero, nero, nero, nero,
]
```

Visualizziamo l'immagine con il comando:

```
sense.set_pixels(immagine)
time.sleep(2)
```

**NOTA:** per poter usare `time.sleep` devi aver scritto nella prima riga del programma:  
`import time`

**Sfida:** Crea la tua immagine ! Cosa manderai nello spazio? Un missile, un fiore, un cuore ....

Se hai difficoltà, disegna prima l'immagine su carta quadrettata.

## Mandiamo il nostro codice nello spazio

Quando tutto funzionerà, nella parte bassa:

## Classroom Code

Once your program is ready, enter your classroom code here to continue to the official submission form.

inserisci il codice che ti verrà dato dai mentor.

Nella scheda successiva, inserite il nome del gruppo (SE USATE NOMI SGRADREVOLI VI ESCLUDERANNO DALLA MISSIONE!), i partecipanti, accettate la norma sulla privacy e aspettate che un mentor venga a controllare che i dati siano corretti.



**Piaciuto? Trovi questo e tanti altri tutorial gratuiti alla  
sezione Risorse su [coderdojotrento.it](https://coderdojotrento.it)**