

# Disegnare con turtle in Python: riepilogo comandi

Questa guida contiene un riepilogo di comandi utili per disegnare con la libreria `turtle` di Python.

Trovi questo riepilogo online all'indirizzo:

[coderdojotrento.it/pythontr](https://coderdojotrento.it/pythontr)

Per una lista di figure e giochi più avanzati che puoi fare con `turtle`, guarda qui:

[coderdojotrento.it/python](https://coderdojotrento.it/python)



- `forward(100)`
- `left(90)`
- `pencolor(0,255,0)`
- ...



## PRIMA DI TUTTO:

1. Carica la *libreria* `turtle`
2. Crea le tartarughe di nome `ada` e `bob`

Per le tartarughe puoi usare i nomi che vuoi, possibilmente in minuscolo

Attenzione: se vuoi spazi usa invece le linee basse `_` (dette *underscore*), es. `maria_grazia`

**Muovi avanti:**

**Muovi indietro:**

```
from turtle import *
```

```
ada = Turtle()
```

```
bob = Turtle()
```

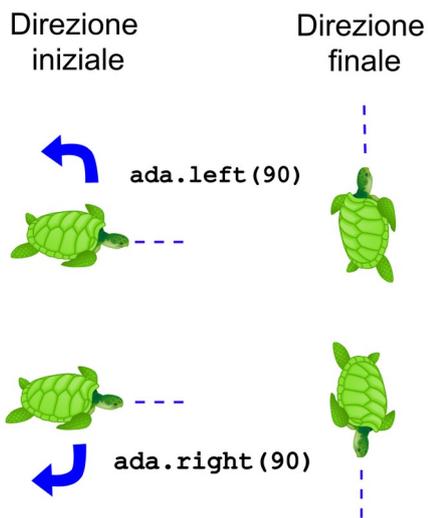
```
ada.forward(100)
```

```
ada.backward(100)
```

**Gira a sinistra / destra:**

```
ada.left(90)  
ada.right(90)
```

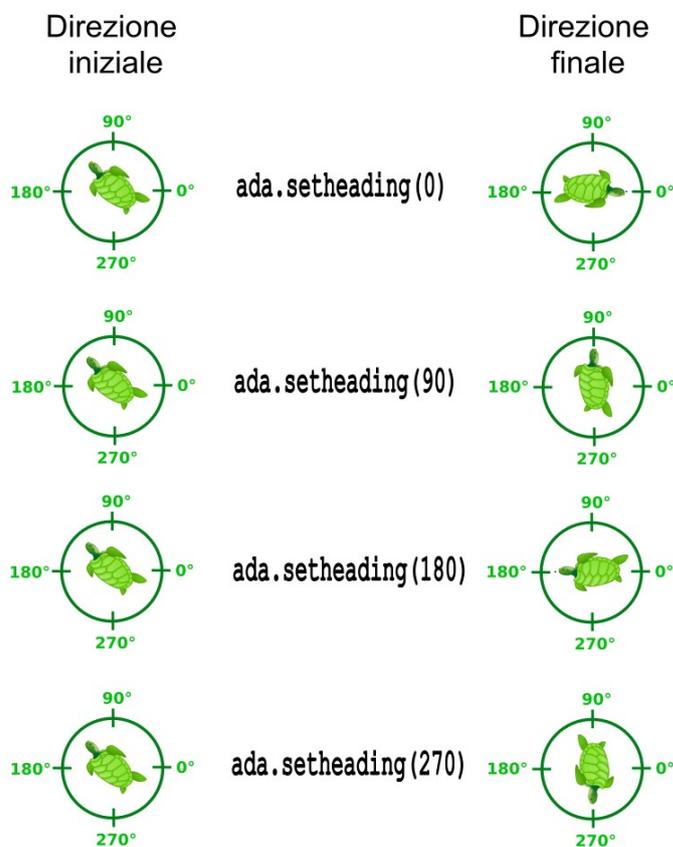
La direzione finale è *relativa*  
rispetto alla tartaruga



**Guarda verso (in gradi):**

```
ada.setheading(90)
```

La direzione finale è *assoluta*  
rispetto all'osservatore



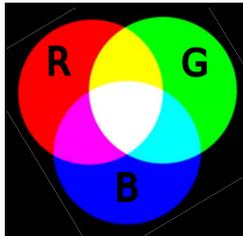
## Cambiare il colore della penna

Puoi miscelare le componenti

**RGB** (Red, Green, Blue)

per ottenere il colore che vuoi

Ogni componente è un numero che può andare da 0 a 255.



### SE VENGONO FUORI ERRORI STRANI:

Certe implementazioni di turtle preferiscono numeri da 0.0 a 1.0. Per forzare il riconoscimento dei numeri da 0 a 255 prova a dare prima questo comando: `colormode(255)`

### Verde:

```
ada.pencolor(0, 255, 0)
```

### Viola:

```
ada.pencolor(255, 0, 255)
```

### Alternativa 1 - con variabile *tupla*:

```
rosso = (255, 0, 0)
```

```
ada.pencolor(rosso)
```

### Alternativa 2 - per nome (in inglese):

```
ada.pencolor("green")
```

## Cambia spessore penna:

1: 10:

```
ada.pensize(10)
```

## Cambia la forma, scegliendo tra:

"arrow", "turtle", "circle",  
"square", "triangle", "classic"

(Altre se ne possono aggiungere coi comandi `screen.register_shape()`)

```
ada.shape("turtle")
```



## Manda la tartaruga in un punto

Servono gli *assi cartesiani* e le *coordinate x, y*:



```
ada.goto(-100, 100)
```



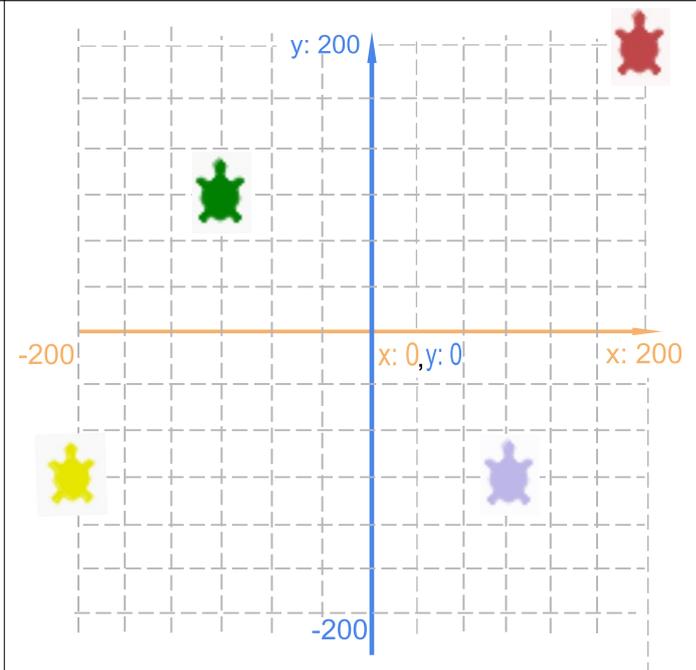
```
bob.goto(200, 200)
```



```
ron.goto(-200, -100)
```



```
eva.goto(100, -100)
```

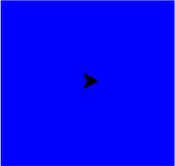
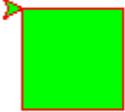


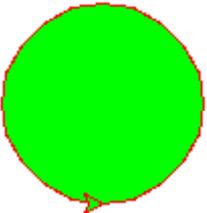
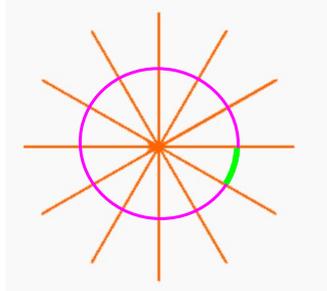
## Penna su:

```
ada.penup()
```

## Penna giù:

```
ada.pendown()
```

<b>Nascondi tartaruga:</b>	<code>ada.hideturtle()</code>
<b>Mostra tartaruga:</b>	<code>ada.showturtle()</code>
<b>Pulisci i disegni della tartaruga</b>	<code>ada.clear()</code>
<b>Stampa la tartaruga sullo sfondo</b> 	<pre>ada.shape("turtle") ada.stamp() ada.forward(50) ada.stamp() ada.forward(50)</pre>
<b>Cambia colore di sfondo:</b> Crea una variabile di tipo <code>Screen</code> (S maiuscola!) e chiamala <code>screen</code> (s minuscola!)	<pre>screen = Screen() screen.bgcolor(0, 0, 255)</pre> 
<b>Cambia la dimensione dello sfondo:</b> Nei nostri esempi assumiamo larghezza 400 e altezza 400	<pre>screen.setup(800, 800)</pre> <p style="text-align: right;">larghezza      altezza (width)          (height)</p>
<b>FIGURE</b>	
<b>Per iniziare riempimento:</b>	<code>ada.begin_fill()</code>
<b>Per finire riempimento:</b>	<code>ada.end_fill()</code>
<b>Imposta colore del <i>solo</i> riempimento:</b>	<code>ada.fillcolor(0, 255, 0)</code>
<b>Quadrato pieno</b>   pippo è una <i>variabile</i> , possiamo anche usare un nome inventato  <b>ATTENZIONE</b> ai due spazi prima dei comandi <code>forward</code> e <code>right</code>	<pre>from turtle import * ada = Turtle() ada.pencolor(255, 0, 0) ada.fillcolor(0, 255, 0) ada.begin_fill()  for pippo in range(4):     ada.forward(50)     ada.right(90)  ada.end_fill()</pre>

<p><b>Cerchio</b></p> 	<pre>ada.pencolor(255, 0, 0) ada.fillcolor(0, 255, 0) ada.begin_fill() ada.circle(50) ada.end_fill()</pre>
<p><b>Arco</b></p> <p>per ottenerlo basta aggiungere un angolo come secondo parametro di <code>circle</code></p> 	<pre>ada.circle(50, 180)</pre>
<p><b>Cerchio pieno monocromatico</b></p> 	<pre>ada.pencolor(255, 0, 255) ada.dot(50)</pre>
<p><b>Fiocco a tante punte con variabile</b></p>  <p><b>Creiamo una variabile</b> per il numero di punte: è come una scatola con un numero dentro</p> <p><b>ATTENZIONE</b> ai due spazi prima dei comandi dentro il <code>for</code></p> <p><b>SFIDA:</b> prova ad aggiungere una variabile per la lunghezza del raggio</p>	<pre>from turtle import * ada = Turtle()  ada.color(255, 100, 0) punte = 12 for pippo in range(punte):     ada.forward(100)     ada.forward(-100)     ada.right(360 / punte)</pre>

<p><b>Mostrare scritte</b></p> <p>Il primo parametro è il testo (ricordati di chiudere le doppie virgolette)</p> <p>Il secondo si chiama <code>align</code> e stabilisce come la tartaruga sarà rispetto al testo, nell'esempio è a sinistra. Opzioni: "left", "center", "right"</p> <p>Il terzo si chiama <code>font</code> e indica, in ordine:</p> <ul style="list-style-type: none"> <li>- quale carattere vogliamo usare (es. "Arial")</li> <li>- la dimensione in punti dello schermo (es. 14)</li> <li>- come vogliamo mostrare il carattere.</li> </ul> <p>Opzioni: "normal", "bold" o "italic"</p>	<pre>ada.pencolor("orange") ada.write("Ciao sono una tartaruga",          align="left",          font=("Arial", 14,"normal"))</pre> 
---	--

**GESTIRE IL TEMPO** 

<p><b>Cambia la velocità:</b></p> <ul style="list-style-type: none"> <li>• da 1 (lento) a 10 (veloce)</li> <li>• 0 : <i>immediato</i></li> </ul>	<pre>ada.speed(3)</pre>
--	-------------------------

<p><b>Come aspettare?</b></p> <p><b>PRIMA DI TUTTO, in cima:</b></p> <p>Aspetta due secondi</p> <p>Aspetta mezzo secondo (occhio al punto ! )</p>	<pre>import time  time.sleep(2)  time.sleep(0.5)</pre>
---	--

**ATTENZIONE:** Questo è il modo più semplice di gestire attese, utile se stai facendo semplici disegni. Se invece vuoi fare un vero videogioco che deve rispondere ai comandi di un giocatore con più cose che accadono contemporaneamente, dovrai usare altre tecniche, per es. con il modulo `asyncio`.



**Piaciuto ? Trovi questo e altri tutorial gratuiti alla sezione *Guide* del sito [coderdojotrento.it](http://coderdojotrento.it)**

Questa guida è stato realizzata grazie al sostegno dell'Associazione CoderDolomiti e sperimentato in classe nei laboratori dell'Associazione Glow.



[www.coderdolomiti.it](http://www.coderdolomiti.it)



[www.glow.earth](http://www.glow.earth)