

INTRODUZIONE A LIBRELOGO

PREPARAZIONE

Un linguaggio di programmazione per giocare con il disegno e la grafica.
Si programma all'interno di un editor di testi.

LibreOffice è una raccolta di programmi per l'elaborazione di testi, fogli di calcolo, presentazioni e altro, un po' come Office di Microsoft. È opensource, gratuitamente scaricabile e sempre aggiornata.

Nell'editor di testi è possibile abilitare con un semplice clic le funzionalità di LibreLogo. LibreLogo è un linguaggio di programmazione semplice che permette di creare disegni e tanto altro.

Due i passi per arrivare ad utilizzare LibreLogo:

1. Installare LibreOffice
2. Abilitare LibreLogo all'interno di Writer

1. Installare Libreoffice

Il sito di riferimento è raggiungibile al seguente indirizzo:

it.libreoffice.org/


La pagina per il download della suite LibreOffice

it.libreoffice.org/download

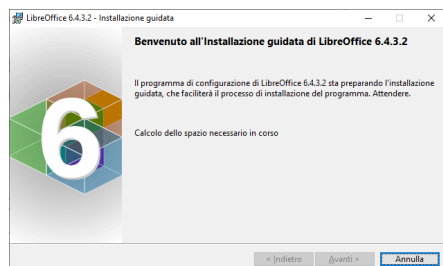


The screenshot shows the LibreOffice website's download page. The header is green with the LibreOffice logo and navigation links. The main content area is white with a green sidebar on the right containing links like 'DOCUMENTATION', 'PROVIDE FEEDBACK', 'JOIN THE PROJECT!', 'MEET US AT EVENTS', 'DISCOVER LIBREOFFICE', and 'GOOD LOOKING DOCUMENTS'. The main text area includes a 'Download / LibreOffice Fresh' section with a 'SCARICATE LA VERSIONE 6.4.3' button and a link to 'Aiuto integrato di LibreOffice in italiano'.

Scaricare l'ultima versione disponibile seguendo le istruzioni.

 LibreOffice_6.4.3_Win_x64.msi

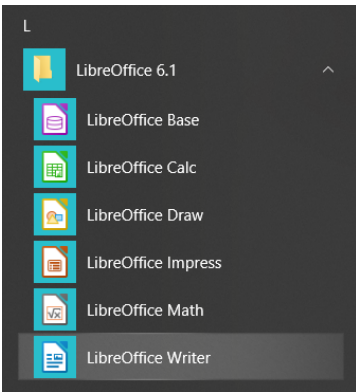
Per installare il file scaricato è sufficiente fare doppio clic sul nome del file.



Cliccare su Avanti e seguire le istruzioni.

Attenzione, l'immagine potrebbe essere differente perché dipende dalla versione disponibile.

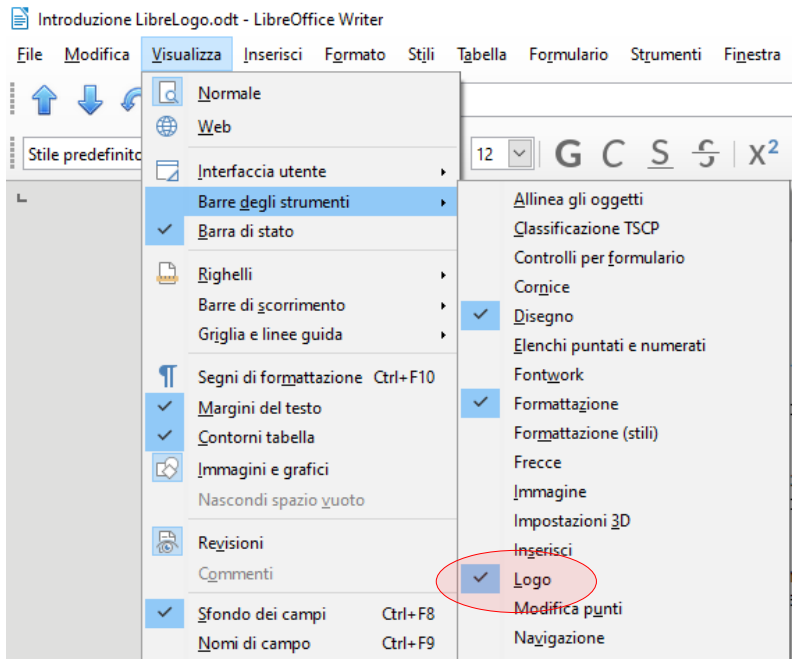
2. Abilitare LibreLogo



Terminata l'installazione LibreOffice compare nell'elenco dei programmi disponibili.

Aprire LibreOffice Writer.
Writer è un normale editor di testo. La sua interfaccia risulterà, quindi, familiare.

Occorre abilitare la *toolbar* (barra degli strumenti) per le funzionalità di LibreLogo.

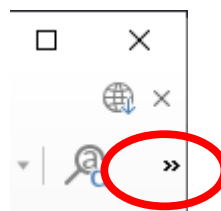


È sufficiente cliccare sulla voce di menù *Visualizza* → *Barre degli strumenti* (in inglese, *View* → *Toolbars*) e mettere il segno di spunta sulla voce *Logo*.

Comparirà la seguente barra (chiudere e riaprire il programma se non compare immediatamente)







A volte la barra sembra lo stesso non comparire, la si trova cliccando sulla doppia freccia a destra della barra degli strumenti







È possibile disporre gli strumenti sulla barra come meglio si preferisce, trascinandoli e spostandoli su di essa.

Le prime quattro icone degli strumenti di Logo ne permettono le istruzioni base:

-  • **FORWARD 10** (avanti di 10 punti)
-  • **BACK 10** (indietro di 10 punti)
-  • **LEFT 15°** (ruota a sinistra di 15°)
-  • **RIGHT 15°** (ruota a destra di 15°)

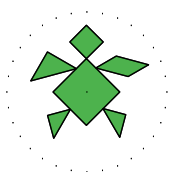
Seguono quattro icone per l'esecuzione dei programmi:

-  • **AVVIA** il programma
-  • **STOP** ferma l'esecuzione
-  • **HOME** riporta al centro del foglio la *tartaruga* inizializzando il programma
-  • **CLEARSCREEN** pulisce lo schermo



La bacchetta magica estende e mette in maiuscolo i comandi nel testo (ad esempio: **fd** viene trasformato in **FORWARD**).

Il campo di immissione testo permette di scrivere ed eseguire una o più istruzioni *al volo*.



La *tartaruga* è il cursore che si presenta quando si eseguono le istruzioni di LibreLogo, è disegnata da una combinazione del Tangram e ricorda la tartaruga di Logo di Papert, un vero e proprio robot che si muoveva sul pavimento grazie alla programmazione fatta con istruzioni simili a quelle che vengono presentate in questo testo.

PRIMI PASSI

Il centro del foglio è la posizione di partenza della tartaruga, vedremo in seguito come dire alla tartaruga di iniziare da una posizione scelta da noi.

La prima istruzione, quella più semplice, è far muovere in avanti la tartaruga di un po'.

Per fare questo si deve scrivere

FORWARD

```
FORWARD 100
```

Le istruzioni sono in inglese, ma non sono poche le parole da imparare e sono semplici, in più è possibile utilizzare le abbreviazioni, **FORWARD** diventa **FD** e significa *avanti* (in ultima pagina sono riportate le istruzioni in ordine alfabetico).

100 è la quantità che è stata scelta come esempio, è possibile utilizzare un numero a scelta, in base al disegno che si vuole realizzare e tenendo presente le dimensioni del foglio.

Dimensioni
foglio

La larghezza del foglio è di circa 595, l'altezza 841, l'unità di misura sono i punti tipografici, vedremo che è possibile anche i centimetri, i millimetri e i pollici nel seguente modo:

Le possibili unità di misura sono le seguenti e devono essere impostate nel seguente modo:

FORWARD 10 o 10pt, punti tipografici (1 punto = 1/72 di pollice)

FORWARD 10in o FORWARD 10", pollici (1 pollice = 2,54 centimetri)

FORWARD 10cm, centimetri (1 punto = 0,035 centimetri)

FORWARD 10mm, millimetri (1 punto = 0,35 millimetri)

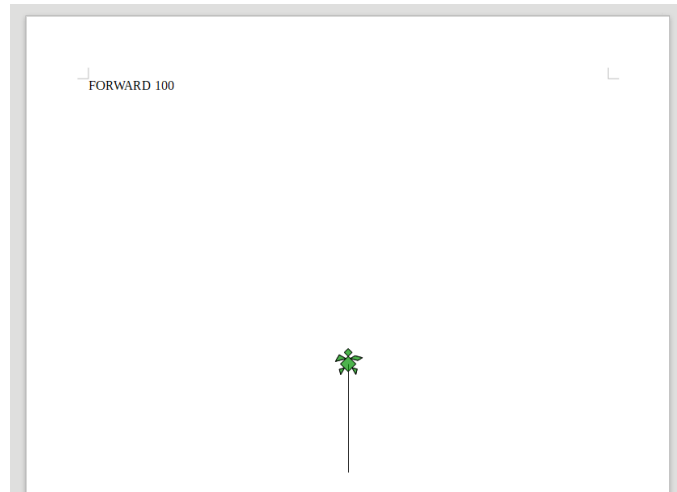
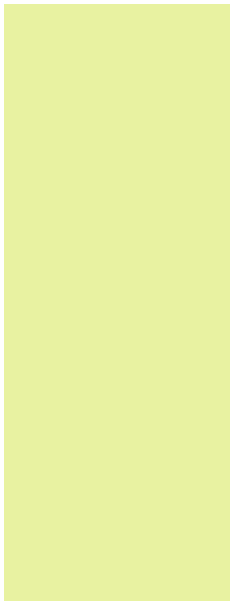
È importante che tra il valore e l'unità di misura non ci siano spazi, si scrive 10cm e non 10 cm.

Origine

Il punto di origine si trova nell'angolo in alto a sinistra, questo significa che la tartaruga, quando è al centro del foglio si trova in una posizione che vale circa 297 sul lato corto (in orizzontale) e 421 sul lato lungo (in verticale). La coppia di valori per indicare una posizione viene sempre indicata in questo modo; posizione lungo l'asse orizzontale e posizione lungo l'asse verticale.

Dopo aver scritto l'istruzione cliccare sull'icona di avvio 

Comparirà la tartaruga rivolta verso l'alto e tratterà una linea lunga 100 punti tipografici.

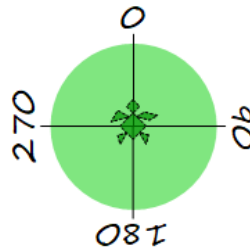


Questo il risultato

IL DISEGNO DEL QUADRATO [introduzione_01.odt - introduzione_02.odt]

RIGHT
LEFT




Aggiungiamo un'istruzione: la rotazione.
Per ruotare a destra si utilizza l'istruzione **RIGHT (RT)**, per ruotare a sinistra **LEFT (LT)**. L'istruzione è completa solo indicando di quanti gradi ruotare.




La rotazione destra e sinistra sono sempre riferiti alla direzione della tartaruga, la sua testa indica 0°, la sua destra è 90°, la sua coda 180° e la sua sinistra 270°.

Aggiungiamo **RIGHT 90** sotto all'istruzione precedente

```
FORWARD 100
RIGHT 90
```

Prima di cliccare su  cliccare su clearscreen (pulisci schermo) 
in questo modo lo schermo viene ripulito dalla linea precedente, cliccare  (home) per riportare la tartaruga al centro del foglio.

Cliccare su  per provare il nuovo codice.



La tartaruga si muove in avanti come prima e poi ruota verso la sua destra.
È l'inizio di un quadrato!

Le istruzioni **CLEARSCREEN** (**CS**) e **HOME** si possono mettere prima delle istruzioni per non dover cliccare sui pulsanti della barra, in questo modo:

CLEARSCREEN
HOME

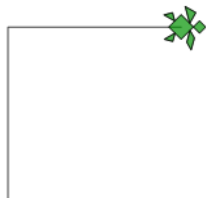
```
CLEARSCREEN  
HOME  
FORWARD 100  
RIGHT 90
```

Così nel codice ci sono già le istruzioni che ripuliscono lo schermo e riportano la tartaruga al centro e ogni volta che si clicca su **avvio** la pagina risulterà pulita.

Ma come si arriva al quadrato?

Proviamo a far camminare la tartaruga dopo la rotazione, bisogna usare ancora **FORWARD 100**, in questo modo:

```
CLEARSCREEN  
HOME  
FORWARD 100  
RIGHT 90  
FORWARD 100
```



Poi occorre di nuovo aggiungere la rotazione verso destra per prepararsi al disegno del nuovo lato.

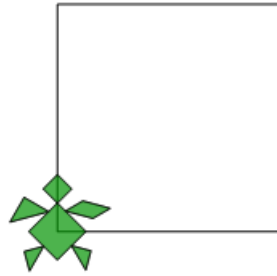
Così:

```
CLEARSCREEN  
HOME  
FORWARD 100  
RIGHT 90  
FORWARD 100  
RIGHT 90
```

Occorre ripetere le istruzioni per disegnare un lato tante volte quanti sono i lati del quadrato.

```
CLEARSCREEN  
HOME
```

```
FORWARD 100
RIGHT 90
FORWARD 100
RIGHT 90
FORWARD 100
RIGHT 90
FORWARD 100
RIGHT 90
```



L'ultima istruzione è ancora una rotazione che riporta la tartaruga alla posizione di partenza e nella direzione di partenza.

Si può fare meglio di così?

Perché bisogna riscrivere le stesse righe così tante volte?

C'è un'istruzione che permette di ripetere le sole due righe che servono per tracciare una riga e ruotare per prepararsi per la riga successiva?

E poi i programmatori sono pigri e non amano riscrivere le stesse cose tante volte.

Ciclo - REPEAT

L'istruzione si chiama proprio **REPEAT** (ripeti) e ha bisogno di qualcosa, dei segni, che indichino dove iniziano e dove finiscono le istruzioni che devono essere ripetute.

Per fare questo si utilizzano le parentesi quadre [], all'interno si mettono le istruzioni da ripetere.

Dopo **REPEAT** e prima dell'apertura della parentesi bisogna indicare quante volte occorre ripetere le istruzioni, nel caso del quadrato 4 volte. Cancelliamo tutto e riscriviamo così le istruzioni:

```
HOME
CLEARSCREEN

REPEAT 4 [
  FORWARD 100
  RIGHT 90
]
```

Cliccando su **avvia** vedremo che il risultato è sempre il nostro quadrato.

RIPETIZIONI E SPIRALI [introduzione_03.odt – introduzione_04.odt - introduzione_05.odt]

Vediamo come si può giocare con le istruzioni che abbiamo visto finora e, in particolare, come utilizzare il ciclo **REPEAT**.

Questa volta creeremo una spirale. Partendo da una linea cortissima, una linea che avrà la lunghezza pari al ciclo che sta facendo.

Il ciclo **REPEAT**, come abbiamo visto, permette di ripetere le istruzioni racchiuse tra le parentesi quadre il numero di volte indicato. Per fare questo il nostro computer deve ricordarsi quante volte ha già eseguito le istruzioni. In altre parole deve tenere il conto del numero di ciclo che sta eseguendo.

Contatore -
REPCOUNT

Il numero del ciclo che il sistema sta eseguendo lo possiamo avere grazie all'istruzione **REPCOUNT**. È a tutti gli effetti un *contatore*, conta i cicli e ci dice in che ciclo ci troviamo.

Variabile

Possiamo utilizzare **REPCOUNT** come se fosse un numero, in effetti si tratta di una *variabile*, ovvero un contenitore di valori che possono, per l'appunto, variare nel corso dell'esecuzione del programma, in questo caso il contenuto di **REPCOUNT** varia ad ogni ciclo incrementandosi di 1.

Il valore contenuto in **REPCOUNT** lo utilizzeremo per indicare di quanto si deve spostare in avanti la tartaruga.

Vediamo cosa succede se a **FORWARD** assegniamo il valore di **REPCOUNT** e se facciamo qualche ciclo in più di 4, ad esempio proviamo con **100**, così:

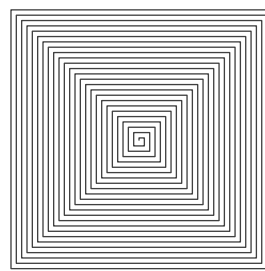
```
HOME
CS

REPEAT 100 [

  FD REPCOUNT
  RT 90

]
```

Cosa otteniamo?
Una spirale come questa



Proviamo ora a movimentare un po' le cose, ruotiamo di poco più di 90°. In questo modo:

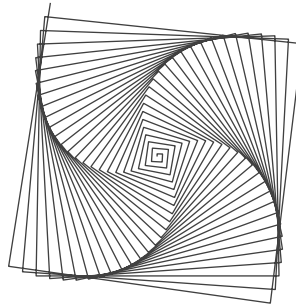
```
HOME
CS

REPEAT 100 [

    FD REPCOUNT
    RT 91

]
```

E ora osserviamo che differenza fa un solo grado in più nella rotazione:



Aumentiamo ancora i gradi di rotazione, 105° ad esempio:

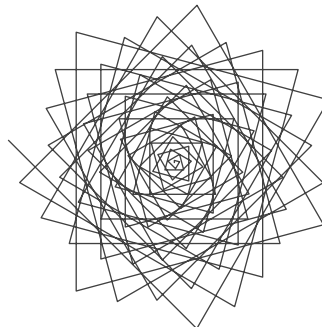
```
HOME
CS

REPEAT 100 [

    FD REPCOUNT
    RT 105

]
```

La spirale si perde, compare un fiore



Provare con valori di rotazione diversi come: 162, 185, 215, 222, 225.

Da notare e riflettere su come anche un solo grado dia effetti completamente differenti.

IL COLORE [introduzione_06.odt – introduzione_07.odt]

FILLCOLOR

FILL

Aggiungiamo il colore, diamo vita ai nostri disegni. Certamente possiamo sempre stamparli e colorarli con varie tecniche ma possiamo anche iniziare a farli colorare da LibreLogo. Ci servono almeno un paio di istruzioni:

FILLCOLOR (colore di riempimento) permette di definire quale colore utilizzare come colore di riempimento

FILL (riempi) chiude e colora i nostri disegni

Nella tabella che segue sono indicati i colori che possiamo utilizzare chiamandoli per nome.

Ad esempio, se vogliamo utilizzare il colore "oro" possiamo scrivere **FILLCOLOR "gold"**

black	silver	gray grey	white	maroon
red	purple	magenta fuchsia	green	lime
olive	yellow	navy	blue	teal
aqua cyan	pink	tomato	orange	gold
violet	skyblue	chocolate	brown	invisible

Abbiamo a disposizione ben più dei 25 (anzi 24) colori qui indicati, ma vedremo in seguito come questo sia possibile.

Ho scritto 24 colori e non 25 anche se la tabella presenta 25 rettangoli. **"white"** e **"invisible"** non sono la stessa cosa, **"invisible"** è un non colore, **"invisible"** sfrutta il valore di *trasparenza*, in questo caso massima trasparenza, ovvero, il colore non è visibile.

Come utilizzare il colore nel nostro disegno?
Ecco un esempio con lo script delle spirali:

SHOWTURTLE

```
HOME
CS
SHOWTURTLE

FILLCOLOR "gold"
```

HIDETURTLE

```
REPEAT 400 [
    FD REPCOUNT
    RT 222
]
FILL
HIDETURTLE
```

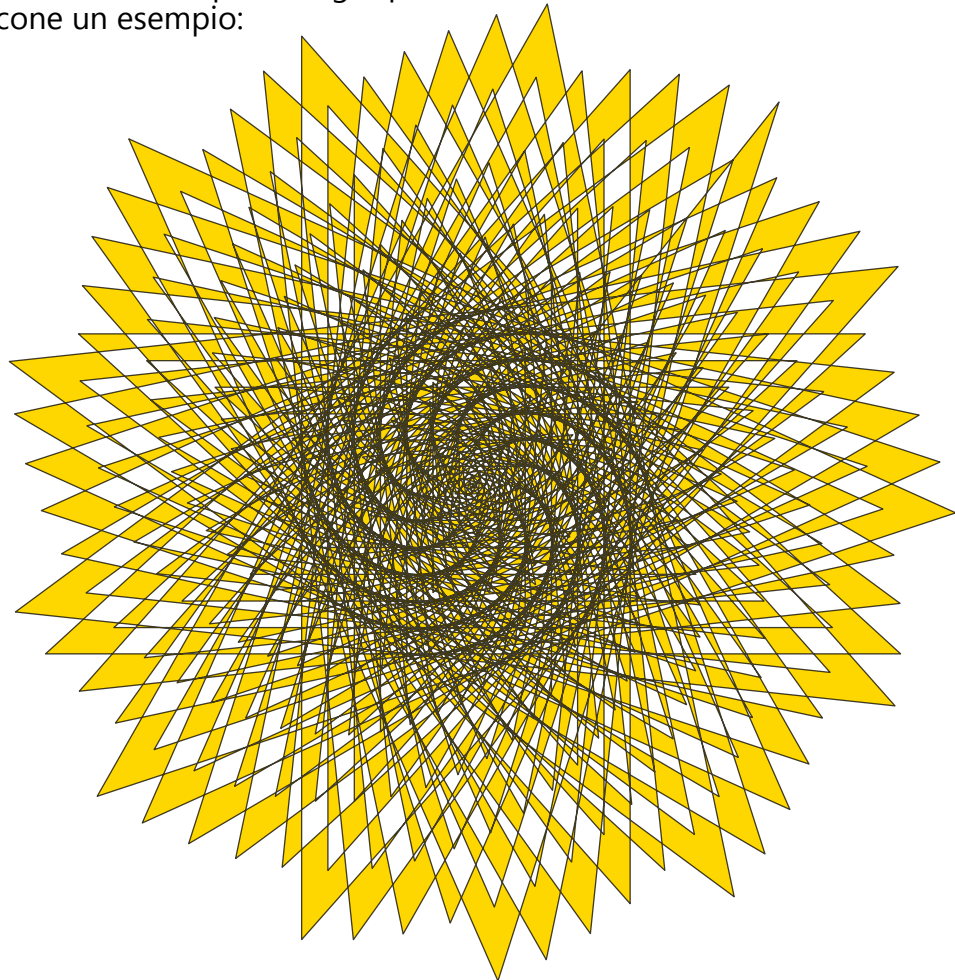
In questo caso è stato aumentato il numero di ripetizioni, è stato portato a **400**, per avere un'immagine più grande.

Prima delle ripetizioni è stato impostato il colore con **FILLCOLOR** "gold", la scelta del colore è ricaduta sul colore oro per un effetto girasole.

Dopo le ripetizioni è stata messa l'istruzione **FILL**, come abbiamo detto chiude la figura e riempie di colore, in questo caso vedremo una linea che parte dall'ultima linea disegnata e arriva al centro, il punto di partenza.

Il colore non riempie tutti gli spazi.

Eccone un esempio:



PENCOLOR

Il colore può essere utilizzato anche per le linee, i colori a disposizione sono i medesimi ma l'istruzione necessaria è **PENCOLOR**, il colore della penna.

Si utilizza nello stesso modo **PENCOLOR "navy"**, ad esempio, imposta il colore della penna a blu navy. L'istruzione va impostata all'inizio del codice.

Eccone un esempio:

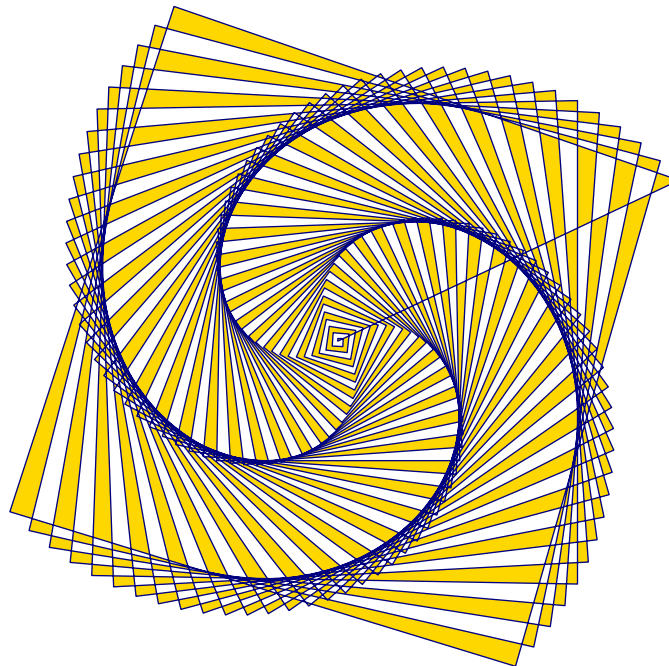
```
HOME
CS
SHOWTURTLE

PENCOLOR "navy"
FILLCOLOR "gold"
REPEAT 200 [

    FD REPCOUNT
    RT 91

]
FILL
HIDETURTLE
```

In questo caso le ripetizioni sono state portate a **200** e l'angolo di rotazione a **91°**. **PENCOLOR** e **FILLCOLOR** sono all'inizio. E questo è il risultato



I colori si basano sul sistema **RGB**, ciascun colore è definito dalla componente di **R**ed (rosso), **G**reen (verde) e **B**lue (blu) che costituiscono il colore.

Ciascuna componente può avere un valore che va da **0** (assenza) a **255** (pieno). Ogni colore è, quindi, composto da una terna di valori che vanno da **[0, 0, 0]**, nero, a **[255, 255, 255]**, bianco.

Per avere un colore di riempimento rosso si scrive:

```
FILLCOLOR [255, 0, 0]
```

Per avere un colore di riempimento verde si scrive:

```
FILLCOLOR [0, 255, 0]
```

Per avere un colore di riempimento blu si scrive:

```
FILLCOLOR [0, 0, 255]
```

I colori si possono indicare anche con i relativi valori esadecimali preceduti dai caratteri "0x".

Per avere un colore di riempimento rosso si scrive:

```
FILLCOLOR 0xFF0000
```

Per avere un colore di riempimento verde si scrive:

```
FILLCOLOR 0x00FF00
```

Per avere un colore di riempimento blu si scrive:

```
FILLCOLOR 0x0000FF
```

[INTRODUZIONE A LIBRELOGO]

ISTRUZIONI

In corsivo il parametro (valore) necessario all'istruzione:

n = numero

BACK *n* [BK] - *Indietro di quanto*

CLEARSCREEN [CS] - *Pulisci schermo*

FILL - *Riempi*

FILLCOLOR *quale colore* - *Imposta colore di riempimento*

FORWARD *n* [FD] - *Avanti di quanto*

HIDETURTLE - *Nascondi tartaruga*

HOME - *Riporta la tartaruga al centro del foglio*

LEFT *n* [LT] - *Sinistra di quanto*

PENCOLOR *quale colore* - *Imposta il colore della penna, le linee*

REPCOUNT - *Contatore di ciclo*

REPEAT *n* [*istruzioni*] - *Ripeti quante volte [istruzioni da ripetere]*

RIGHT *n* [RT] - *Destra di quanto*

SHOWTURTLE - *Mostra tartaruga*