

قصد کشیدن یک مثلث متساوی الاضلاع را به کمک کتابخانه‌ی Turtle داریم.

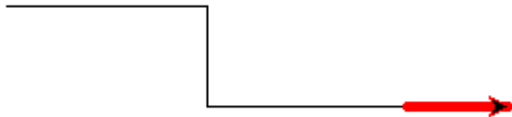
کتابخانه‌ی Turtle شامل یک **pointer** هست که شبیه لاکپشت هست و این لاکپشت روی صفحه‌ی **screen** شما راه می‌رود و یک تصویر از حرکت این لاکپشت به وجود می‌آید. برای مثال کشیدن یک خط صاف به شکل زیر تعریف می‌شود و خروجی زیر را به همراه خواهد داشت.

<pre># import turtle import turtle # makes the turtle to move forward turtle.forward(100) # pause screen turtle.done()</pre>	
--	---

خط اول کتابخانه‌ی **turtle** یا لاکپشت رو وارد میکنیم. نکته جالب در رابطه با این کتابخانه آن است که کتابخانه **built-in** شده است و نیازی به نصب ندارد. سپس در دستور دوم به **turtle** گفته میشه که ۱۰۰ پیکسل به جلو حرکت کنه و یک خط سیاه ۱۰۰ پیکسلی پشت سرش ایجاد شده و بعد از آن هم برای اینکه بتونیم صفحه را به حالت **pause** ببریم و شکل را با چشم ببینیم دستور **turtle.done** را اضافه میکنیم والا به سرعت صفحه پس از اجرای دستورات بسته میشود.

کتابخانه‌ی **turtle** یک ابزار بسیار جالب برای خلق ایده‌های هندسی هست. که امروز در انتهای این پست یکی از آن‌ها را تجربه خواهیم کرد.

حال به جای خط **turtle.forward(100)** مجموعه دستور زیر را در نظر بگیرید. این مجموعه دستورات بخش از ویژگی و صفاتی را نشان می‌دهد که شما می‌توانید به شکل ترسیمی خود بدهید.

<pre># import turtle import turtle turtle.forward(100) turtle.right(90) turtle.forward(50) turtle.left(90) turtle.forward(100) turtle.pencolor("red") turtle.pensize(5) turtle.forward(50) # pause screen turtle.done()</pre>	
---	--

میتوانید سباز خط ترسیمی را با دستور **pensize** بزرگ کنید، یا با دستور **pencolor** رنگ را تغییر دهید. و یا با دستورات **left** سر پیکان را به چپ یا درجه‌ی دلخواه و یا به راست به درجه‌ی دلخواه بچرخانید. برای مثال در شکل فوق پیکان ابتدا به مقدار ۱۰۰ پیکسل

جلو آمده ۹۰ درجه به سمت راست چرخیده ۵۰ تا جلو آمده و اینبار ۹۰ درجه به چپ چرخیده و ۱۰۰ پیکسل دیگر جلو رفته رنگ پیکان قرمز شده و اندازه‌اش ۵ پیکسل شده و ۵۰ پیکسل دیگر جلو رفته است.

بسیار عالی! پس فهمیدیم چطور مربع، مستطیل، مثلث و یا هر شکل هندسی ساده‌ای را بکشیم. و الان می‌خواهیم مثلث متساوی‌الاضلاع را بکشیم.

کشیدن متساوی‌الاضلاع:

مثلث متساوی‌الاضلاع مثلثی است که هر سه ضلع آن با هم برابر و زاویه‌های داخلی آن ۶۰ درجه است. برای ترسیم کافایت پیکان را ۱۲۰ درجه به چپ بچرخانیم (بمقدار زاویه‌ی خارجی یک راس مثلث)، ۱۰۰ پیکسل به جلو برویم، دوباره ۱۲۰ درجه به چپ بچرخیم، ضلع دوم را ۱۰۰ پیکسل ترسیم کنیم و دوباره ۱۲۰ درجه پیکان را به چپ بچرخانیم و ضلع سوم را ترسیم کنیم.

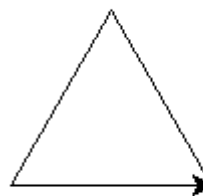
```
# import turtle
import turtle

# first edge
turtle.left(120)
turtle.forward(100)

# second edge
turtle.left(120)
turtle.forward(100)

# third edge
turtle.left(120)
turtle.forward(100)

# pause screen
turtle.done()
```



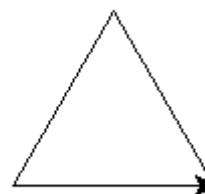
در سورس کد بالا مشخص است دو دستور `turtle.left(120)` و `turtle.forward(100)` سه بار تکرار شده است و به سادگی میتوان از ساختارهای حلقه برای رسم آن استفاده کرد.

```
# import turtle
import turtle

flag = 3
while(flag):

    turtle.left(120)
    turtle.forward(100)

    flag -=1
# pause screen
turtle.done()
```



یه نکته‌ی جالب؛ به کمک کتابخانه‌ی turtle ما میتوانیم چند turtle روی صفحه داشته باشیم و هر کدام تصویری خاص را رسم کنند. برای اینکار کافیه از کلاس turtle اشیای مختلفی را بسازیم. اینکار را با دستور turtle.Turtle() انجام می‌دهیم. دستور زیر دقیقاً مثلث را می‌کشد اما اینبار پیکانی را بعنوان نمونه ساختیم تا بر ایمان اینکار را کند.

```
# import turtle
import turtle

# initialising variables
dist = 100
flag = 3

# initialising turtle
triangle = turtle.Turtle()

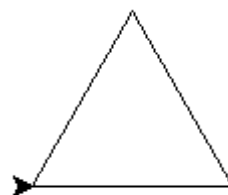
# changing speed of turtle
triangle.speed(10)

# making pattern
while flag:

    # makes the turtle to move forward
    triangle.forward(dist)

    triangle.left(120)
    flag -= 1

turtle.done()
```



بیایید بکبار ساده به کد نگاهی بیاندازیم،

1. پیکان ۱۰۰ پیکسل جلو می‌رود
2. ۱۲۰ درجه به چپ می‌چرخد
3. گام ۱ و ۲ را سه بار تکرار میکند.

جالب آنست اگر به جای ۱۲۰ درجه ۱۴۴ درجه به چپ بچرخد و به جای سه بار گام ۱ و ۲ را پنج بار تکرار کند شکلی کاملاً جدید بدست می‌آید و آن هم ستاره‌ی ۵ پر خواهد بود.

1. پیکان ۱۰۰ پیکسل جلو برود.
2. ۱۴۴ درجه به چپ بچرخد
3. گام ۱ و ۲ را پنج بار تکرار کن.

```

# import turtle
import turtle

# initialising variables
dist = 100
flag = 5

# initialising turtle
star = turtle.Turtle()

# changing speed of turtle
star.speed(10)

# making pattern
while flag:

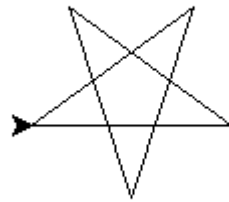
    # makes the turtle to move forward
    star.forward(dist)

    # makes the turtle to move left
    star.left(144)

    flag -= 1

turtle.done()

```



به شبه کد یکبار دیگر نگاه بیاندازیم؛

1. پیکان ۱۰۰ پیکسل جلو برود.
2. ۱۴۴ درجه به چپ بچرخ
3. گام ۱ و ۲ را پنج بار تکرار کن.

هر سه خط آن قابل دستکاری هست و میتوانیم بجای ۱۰۰ پیکسل بیشتر و یا کمتر جلو برویم و یا درجه چرخش را تغییر دهیم و گام ۱ و ۲ را به جای پنج بار ده بار تکرار کنیم. پس بگذارید چند بار دیگر این متغیرها را تغییر دهیم. برای اینکار شی با نام pattern میسازیم. به کد مثلث متساوی الاضلاع خود بر میگردیم و یک خط کد (1) pattern.left(1) به حلقه while اضافه می‌کنیم. تعداد حلقه را هم روی 120 بالا می‌بریم. خروجی بسیار جالب و به شکل دونات خواهد بود. آیا میتوانید تحلیل کنید چرا این اتفاق افتاد؟

```

# import turtle
import turtle

# initialising variables
dist = 100
flag = 120

# initialising turtle
pattern = turtle.Turtle()

# changing speed of turtle
pattern.speed(10)

# making pattern
while flag:

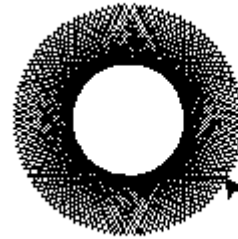
    # makes the turtle to move forward
    pattern.forward(dist)

    # makes the turtle to move left
    pattern.left(120)
    pattern.left(1)

    flag -= 1

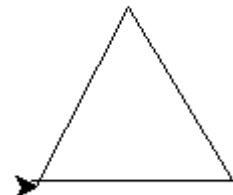
turtle.done()

```



جواب. فرض کنید `pattern.left(1)` را اضافه نمی‌کردید. اما تعداد حلقه را ۱۲۰ می‌گذاشتید. چه رخ می‌داد؟ لاکپشت بروی مثلثی که ساخته بود ۶۰ دور میزد و در سر جای خود قرار می‌گرفت. حال اگر به جای ۱۲۰، اینبار ۱۲۱ درجه بدهید. در واقع ساختار مثلث متساوی‌الاضلاع را بهم زدید و اضلاع این مثلث روی هم می‌افتد.

بعبارتی لاکپشت پس از ترسیم سه ضلع به جای خود بر نمی‌گردد و کمی پایین‌تر می‌ایستد.



حال از جای جدید مثلث دوم را می‌سازد. و این اتفاق برای کشیدن مثلث بعدی هم می‌افتد ...



حال اگر مقدار حرکت forward هم در هر حلقه افزایش میدادیم چه رخ میداد؟ میتوان حدس زد، درسته؟

```
# import turtle
import turtle

# initialising variables
dist = 100
flag = 120

# initialising turtle
pattern = turtle.Turtle()

# changing speed of turtle
pattern.speed(10)

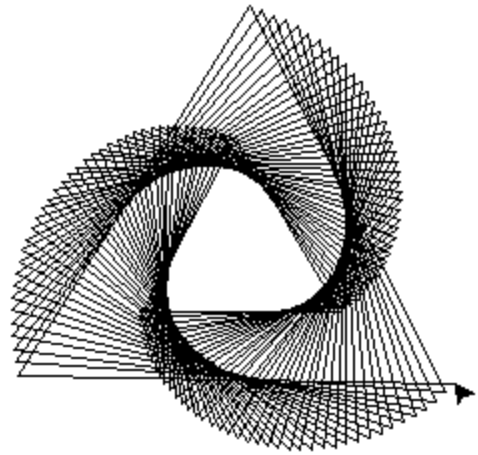
# making pattern
while flag:

    # makes the turtle to move forward
    pattern.forward(dist)

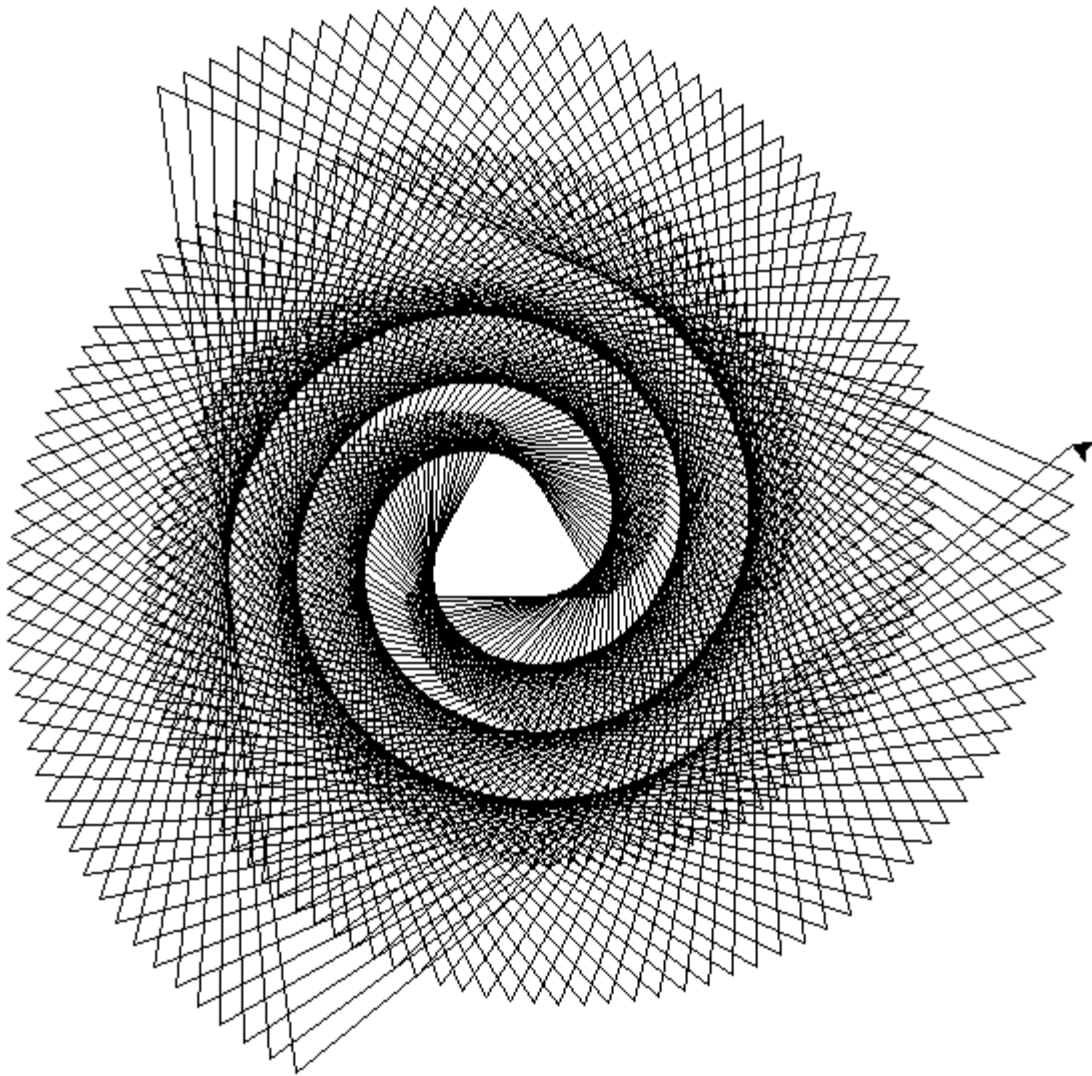
    # makes the turtle to move left
    pattern.left(120)
    pattern.left(1)
    dist += 1
    flag -= 1

turtle.done()
```

و خروجی به شکل زیر خواهد بود:



گویا یک مثلث دائم میچرخد و به شما نزدیک می‌شود. حال قرار است به جای ۱۲۰ بار ۴۰۰ بار اینکار را بکنیم. خروجی چیزی خواهد بود که به آن spiral doodle pattern گفته می‌شود. و شکل آن در زیر آمده است.



سورس کد به همراه خروجی در گیت‌هاب اینجانب به آدرس زیر به همراه همین فایل متنی قابل دسترسی است.

https://github.com/sohrabkhanbadr/computerworkshop/tree/main/CW_Python/turtle