

## Chapter 4

# Python Graphics

നിറം നൽകാനും പൈത്തൺ കോഡുകൾ

Python code for giving colour

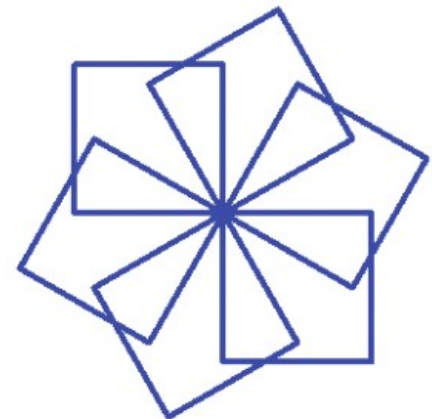
ഇന്നത്തെ ക്ലാസ്സ് കാണുന്നതിന് ഇവിടെ ക്ലിക്ക് ചെയ്യുക

പൈത്തൺ ഗ്രാഫിക് ജാലകത്തിൽ പ്രത്യക്ഷപ്പെടുന്ന രൂപങ്ങൾക്ക് കറുപ്പുനിറമാണ് സ്വാഭാവികമായും ലഭിക്കുക. എന്നാൽ, ജ്യാമിതീയരൂപങ്ങളും പാറ്റേണുകളും വിവിധ നിറങ്ങളിൽ നിർമ്മിക്കുന്നതിനുള്ള നിർദ്ദേശങ്ങളും പൈത്തൺ ടർട്ടിൽ ഗ്രാഫിക്സിൽ ഉൾപ്പെടുത്തിയിട്ടുണ്ട്.

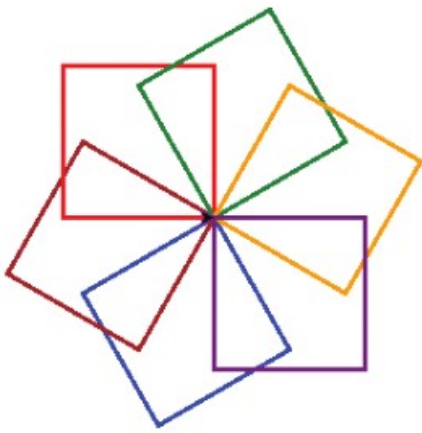
The shapes that appear in the Python Graphics window normally get black colour. But Python Turtle Graphics also has commands to create shapes and patterns in different colours.

```
from turtle import*
clear()
pensize(3)
for i in range(6):
    right(60)
    color("blue")
    for j in range(4):
        forward(100)
        right(90)
```

പാറ്റേൺ നീല നിറത്തിൽ പ്രിന്റ് ചെയ്യുന്നത് ഈ പുതിയ നിർദ്ദേശം പ്രോഗ്രാമിൽ ഉൾപ്പെടുത്തിയതിനാലാണ്.



പ്രോഗ്രാം 4.4



```

from turtle import*
clear()
pensize(3)
clr=["blue","brown","red","green","orange","purple"]
for i in clr:
    right(60)
    color(i)
    for j in range(4):
        forward(100)
        rt(90)

```

പാറ്റേണിന് വിവിധ കളറുകൾ ലഭിക്കുന്നതിനുള്ള നിർദ്ദേശം  
Instruction to receive Different colors for the pattern

പ്രോഗ്രാമിൽ clr=["blue", "brown", "red", "green", "orange", "purple"] എന്ന വരി പുതിയതായി കൂട്ടിച്ചേർത്തിരിക്കുന്നു. ആറു നിറങ്ങൾ clr എന്ന ചരത്തിൽ ശേഖരിക്കുന്നതിനുള്ള നിർദ്ദേശമാണിത്. അതിനുശേഷം for i in clr: എന്ന ആവർത്തന നിർദ്ദേശം പ്രവർത്തിക്കുമ്പോൾ ശേഖരിക്കപ്പെട്ട ഓരോ നിറവും i എന്ന ചരത്തിലെ വിലയായി ലഭിക്കുന്നു. (അതിനായി for നിർദ്ദേശത്തിൽ വരുത്തിയിരിക്കുന്ന മാറ്റവും ശ്രദ്ധിക്കുക). പിന്നീട് color(i) എന്ന നിർദ്ദേശം പ്രവർത്തിക്കുമ്പോൾ i യിലെ വിലയ്ക്കനുസരിച്ച് ഓരോ ചതുരത്തിന്റെയും നിറങ്ങളിൽ മാറ്റം വരും.

In the program, the line clr= ["blue", "brown", "red", "green", "orange", "purple"] is newly added. It is a command for collecting six colours in the variable clr. Thereafter, when the loop instruction for i in clr: is executed, it selects all the six colours one by one as the value of the variable i (note the change in the for statement made for this purpose). Later, when the command color(i) is given, the square gets a different colour depending on the value of i.

```

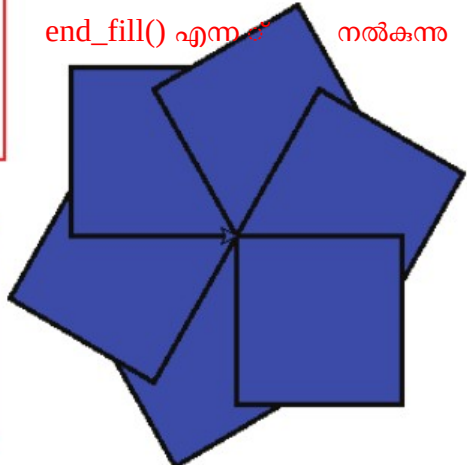
from turtle import*
clear()
pensize(3)
for i in range(6):
    right(60)
    color("black","blue")
    begin_fill()
    for j in range(4):
        forward(100)
        rt(90)
    end_fill()

```

പാറ്റേണിലെ കളങ്ങളിൽ നീലനിറം നിറയ്ക്കാനും ബോർഡർ കളർ കറുപ്പാക്കാനുമുള്ള നിർദ്ദേശം.

begin\_fill(), end\_fill() എന്നീ നിർദ്ദേശങ്ങൾ ഒരേ ടാബ് പൊസിഷനിലാണ് എഴുതേണ്ടത്.

കളങ്ങളിൽ മുഴുവനായി നിറം നൽകുന്നതിന് begin\_fill() end\_fill() എന്നിവ നൽകുന്നു



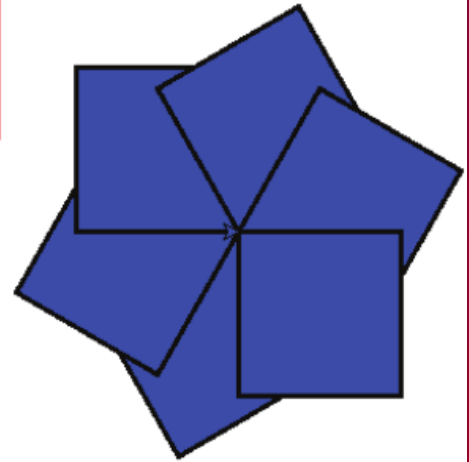
```

from turtle import*
clear()
pensize(3)
for i in range(6):
    right(60)
    color("black","blue")
    begin_fill()
    for j in range(4):
        forward(100)
        rt(90)
    end_fill()

```

The instruction is to fill the spaces in the pattern with blue colour and to draw the border in black

Begin\_fill () and end\_fill () to give full color to cells



color("black","blue")

begin\_fill()

The instructions begin\_fill() and end\_fill() should be written in the same tab position

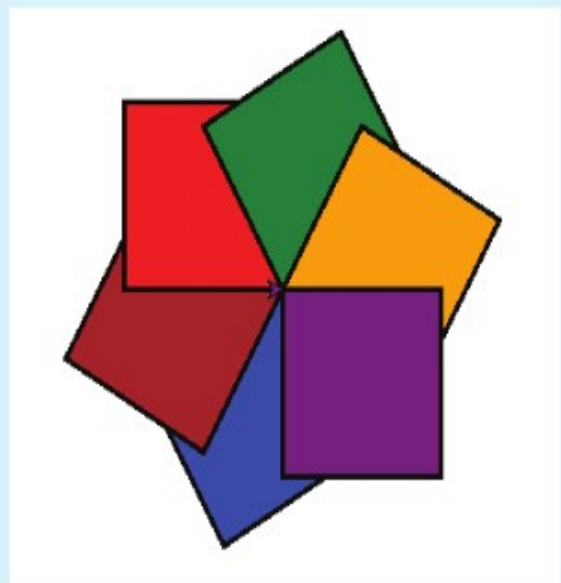
end\_fill()

```

from turtle import*
clear()
pensize(3)
clr=["blue","brown","red","green","orange","purple"]
for i in clr:
    right(60)
    color("black",i)
    begin_fill()
    for j in range(4):
        forward(100)
        rt(90)
    end_fill()

```

കളങ്ങളിൽ വ്യത്യസ്ത നിറങ്ങൾ ലഭിക്കുന്നതിന്



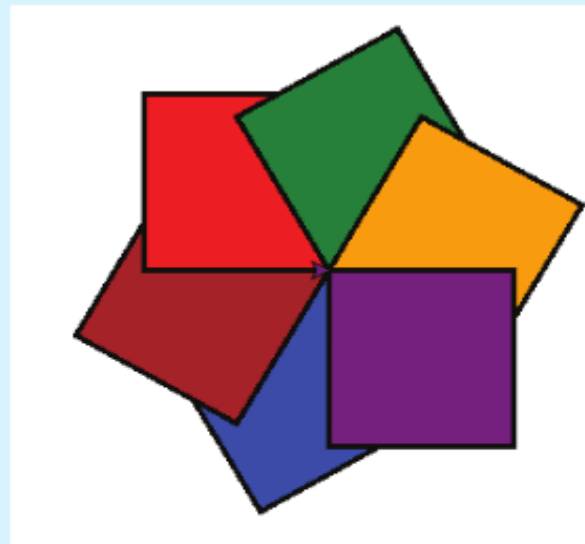
പ്രോഗ്രാമിലെ clr = ["blue","brown","red","green","orange","purple"] എന്ന നിർദ്ദേശം ആറു നിറങ്ങൾ clr എന്ന ചരത്തിൽ ശേഖരിക്കുന്നതിനുള്ളതാണ്. അതിനുശേഷം for i in clr: എന്ന ആവർത്തന നിർദ്ദേശം പ്രവർത്തിക്കുമ്പോൾ ശേഖരിക്കപ്പെട്ട ഓരോ നിറവും i എന്ന ചരത്തിലെ വിലയായി ലഭിക്കുന്നു. പിന്നീട് color("black", i) എന്ന നിർദ്ദേശം പ്രവർത്തിക്കുമ്പോൾ i യിലെ വിലയ്ക്കനുസരിച്ച് ഓരോ ചതുരത്തിലും നിറയ്ക്കുന്ന നിറങ്ങളുടെ കളറുകളിൽ മാറ്റം വരും. അതോടൊപ്പം ബോർഡർ നിറം കറുപ്പുമാകും.

```

from turtle import*
clear()
pensize(3)
clr=["blue","brown","red","green","orange","purple"]
for i in clr:
    right(60)
    color("black",i)
    begin_fill()
    for j in range(4):
        forward(100)
        rt(90)
    end_fill()

```

To get different colors in the cells



The line `clr=["blue", "brown", "red", "green", "orange", "purple"]` in the program is an instruction to collect six colours in the variable `clr`. When the loop instruction `for i in clr:` is executed after that, each collected colour is obtained as the value of the variable `i`. Afterwards, when the instruction `color("black", i)` is given, the colour to be filled in each space changes with the value of `i`. Along with that, the colour of the border becomes black.

പൈത്തൺ നിർദ്ദേശം ഉപയോഗിച്ച് കണ്ണിൻ കുളിർമയേകുന്ന ഗ്രാഫിക്സ് ആനിമേഷനുകൾ നിർമ്മിക്കാനാവും. വളരെ കുറച്ചു വരികൾ മാത്രം ടൈപ്പ് ചെയ്താൽ തന്നെ ഇതു സാധ്യമാകുന്നു എന്നതാണ് ഇതിന്റെ പ്രത്യേകത. വൃത്തങ്ങൾ ഉപയോഗിച്ചുള്ള ചില പാറ്റേണുകൾ നിർമ്മിക്കുന്നതിനുള്ള പൈത്തൺ പ്രോഗ്രാമുകളാണ് ചുവടെ നൽകിയിരിക്കുന്നത്.

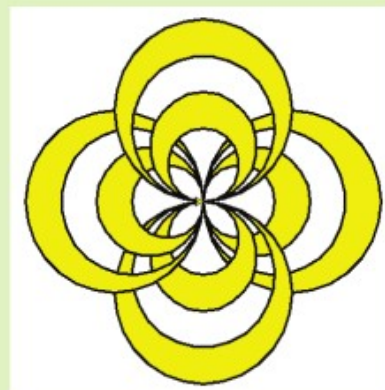
പ്രോഗ്രാം a

```

from turtle import*
pensize(3)
color("black","yellow")
for r in range(4):
    rt(90)
    begin_fill()
    for i in range(40,101,20):
        circle(i)
    end_fill()

```

right(90) എന്നതിന്റെ ചുരുക്കം



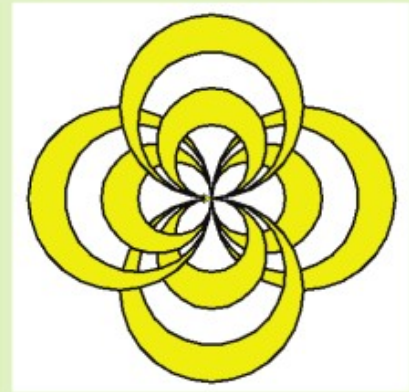
ഒന്നിൽ കൂടുതൽ രൂപങ്ങൾ നിർമ്മിച്ച ശേഷം ഒരൂമിച്ചു നിറം നൽകുമ്പോൾ ഒന്നിടവിട്ട കളങ്ങളിൽ കളർ നിറയുന്ന കാഴ്ചയാണ് ചിത്രത്തിൽ.

It is possible to create beautiful animations using python instructions. The advantage is that it can be done with a few lines of code. Given below are some python programs for creating certain patterns with circles.

```

from turtle import*
pensize(3)
color("black", "yellow")
for r in range(4): Short forms right(90)
    rt(90)
    begin_fill()
    for i in range(40,101,20):
        circle(i)
    end_fill()

```



You can see in the picture how alternate shapes get coloured, when more shapes are created and coloured together.



നാം ഇതുവരെ പരിചയപ്പെട്ട പൈത്തൺ നിർദ്ദേശങ്ങളും അവയുടെ ഉപയോഗവും ഉദാഹരണസഹിതം പട്ടികയിൽ നൽകിയിരിക്കുന്നു.

നിർദ്ദേശം	ഉപയോഗം	ഉദാഹരണം
forward()	ടർട്ടിൽ വലതുവശത്തേക്ക് ചലിപ്പിക്കാൻ.	forward(50) അഥവാ fd(50)
circle()	ടർട്ടിൽ വൃത്താകൃതിയിൽ ചലിപ്പിക്കാൻ.	circle(100)
dot()	ഗ്രാഫിക് സ്ക്രീനിൽ ഒരു ബിന്ദു രേഖപ്പെടുത്താൻ.	dot (20, "blue")
pencolor()	ഗ്രാഫിക് സ്ക്രീനിൽ തെളിയുന്ന വരകളുടെ നിറം നിർണ്ണയിക്കാൻ.	pencolor("blue")
pensize()	ഗ്രാഫിക് സ്ക്രീനിൽ തെളിയുന്ന വരകളുടെ വീതി നിർണ്ണയിക്കാൻ.	pensize(5)
right()	ടർട്ടിലിന്റെ ദിശ നിശ്ചിത ഡിഗ്രി വലത്തോട്ടു തിരിയാൻ. right(90) എന്നു നൽകിയാൽ, ടർട്ടിൽ 90 ഡിഗ്രി വലത്തോട്ടു തിരിയും.	rt(120) അഥവാ right(120)
left()	ടർട്ടിലിന്റെ ദിശ നിശ്ചിത ഡിഗ്രി ഇടത്തോട്ടു തിരിയാൻ. left(90) എന്നു നൽകിയാൽ, ടർട്ടിൽ 90 ഡിഗ്രി ഇടത്തോട്ടു തിരിയും.	lt(120) അഥവാ left(120)

color(a)	പൈത്തൺ ഗ്രാഫിക്സ് ജാലകത്തിൽ നിർമ്മിക്കപ്പെടുന്ന രൂപങ്ങളുടെ നിറം നിർണ്ണയിക്കാൻ.	color("blue")
color(a,b)	ഒരു ബഹുഭുജമോ വൃത്തമോ വരയ്ക്കുമ്പോൾ ഒട്ട് ലൈനിനും ഉൾഭാഗത്തും വ്യത്യസ്ത നിറങ്ങൾ നൽകുന്നതിനുള്ള നിർദ്ദേശം.	color("black","blue")
clear()	ഗ്രാഫിക് സ്ക്രീനിലെ വരകളും എഴുത്തുകളും പൂർണ്ണമായും ഇല്ലാതാക്കാൻ.	clear()
from turtle import*	ടർട്ടിൽ ഗ്രാഫിക്സ് കമ്പ്യൂട്ടറിൽ പ്രവർത്തന ക്ഷമമാക്കുന്നതിന് പ്രോഗ്രാമിന്റെ തുടക്കത്തിൽ ഈ നിർദ്ദേശം നൽകണം.	from turtle import*

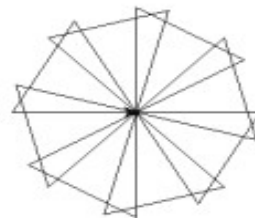
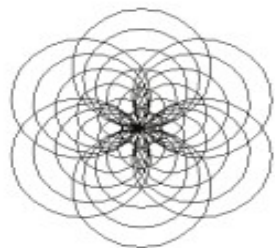
The Python instructions that we came across and their uses are tabulated below with examples.

Instruction	Used for	Example
forward()	Moving the turtle to the right	forward(50) or fd(50)
circle()	For moving the turtle in a circle	circle(100)
dot()	Marking a point on the graphic screen	dot(20, "blue")
pencolor()	Specifying the colour of the line to be drawn on the screen	pencolor("blue")
pensize()	Specifying the thickness of the line to be drawn on the screen	pensize(5)
right()	Turning the turtle to the right as per the angle specified. right(90) turns the turtle to the right by 90 degrees.	rt(120) or right(120)
left()	Turning the turtle to the left by the angle specified. right(90) turns the turtle to the left by 90 degrees.	lt(120) or left(120)
color(a)	Specifying the colour of the shapes to be drawn in the Python Graphics window	color("blue")
color(a,b)	Instructing python to give different colours to the border and the body of shapes drawn.	color("black","blue")
clear()	Removing all the lines and colours in the graphic screen	clear()
from turtle import*	This instruction is to be given at the beginning of the program to activate turtle graphics in the computer	from turtle import*

പ്രോഗ്രാം	ഔട്ട്പുട്ട് 1	ഔട്ട്പുട്ട് 2
<pre> from turtle import* color("yellow") dot(250,"black") pensize(25) for i in range(8):     right(45)     for i in range(3):         forward(100)         right(120)     dot(100,"purple") </pre>		

write program to give output 2 രണ്ടാമത്തെ ഔട്ട്പുട്ട് ലഭിക്കുന്നതിനുള്ള പ്രോഗ്രാം എഴുതുക

1. Draw patterns as given below using circles and triangles.



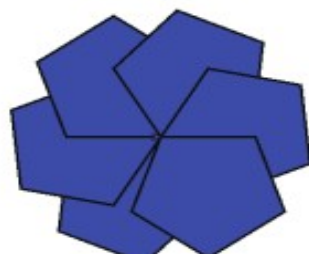
വീഡിയോ കണ്ടതിന്റെ അടിസ്ഥാനത്തിൽ ഇത്തരം പാറ്റേണുകൾ

2. Draw patterns as given below using pentagons and hexagons.



ലഭിക്കുന്നതിനുള്ള പ്രോഗ്രാം എഴുതുക

3. Create different patterns using colours as given below.



Write program such as reccei Ving patterns Based on video