

Istnieje kilka sposobów definiowania kolorów.

Predefiniowane jest wiele kolorów, na przykład:

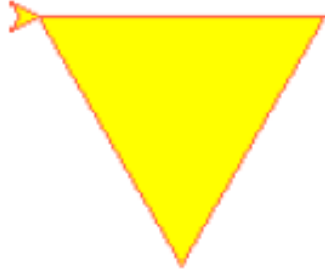

black		darkred		darkgray		red	
darkblue		darkmagenta		blue		magenta	
darkgreen		olivedrab		green		yellow	
darkcyan		gray		cyan		white	

Spis wszystkich nazw kolorów znajduje się na stronie <http://www.tcl.tk/man/tcl8.5/TkCmd/colors.htm>.

```
from turtle import *
```



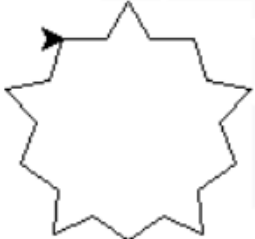
Żółw może rysować pisakiem o dowolnej grubości:

Zapis w Pythonie	Efekt na ekranie
<pre>pensize(5) fillcolor("#a0c8f0") begin_fill() for i in range(2): fd(50); rt(90) fd(100); rt(90) end_fill()</pre>	 A blue rectangle with a thick black border, drawn by a turtle. The rectangle is oriented vertically and is centered in the right-hand cell of the table. A small blue arrow (the turtle's head) is visible at the top-left corner of the rectangle.

Zapis w Pythonie	Efekt na ekranie
<pre>pencolor("tomato") fillcolor("yellow") begin_fill() for i in range(3): fd(100); rt(120) end_fill()</pre>	
<pre>fillcolor(0, 0.5, 0) begin_fill() for i in range(5): fd(100); rt(72) end_fill()</pre>	

Instrukcja iteracji

Do powtarzania poleceń można użyć instrukcji **for**. Przeanalizuj poniższe przykłady:

Zapis w Pythonie	Efekt na ekranie
<pre>for i in range(10): #poniższe polecenia będą wykonane 10 razy fd(10); pu(); fd(10); pd()</pre>	
<pre>for i in range(5): fd(10); lt(90) fd(20); rt(90) fd(10); rt(90) fd(20); lt(90)</pre>	
<pre>for i in range(5): fd(20); lt(60) fd(20); rt(120) fd(20); lt(60) fd(20); rt(360/5)</pre>	

Korzyści ze stosowania pętli:

- skrócenie kodu programu;
- kod staje się czytelniejszy.