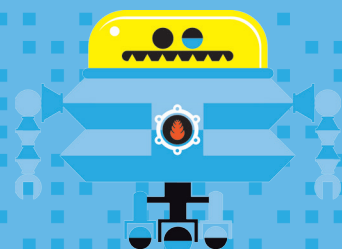
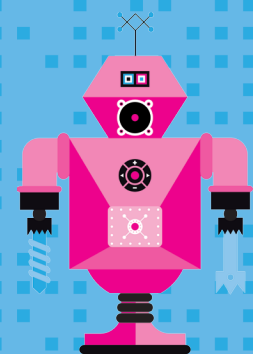
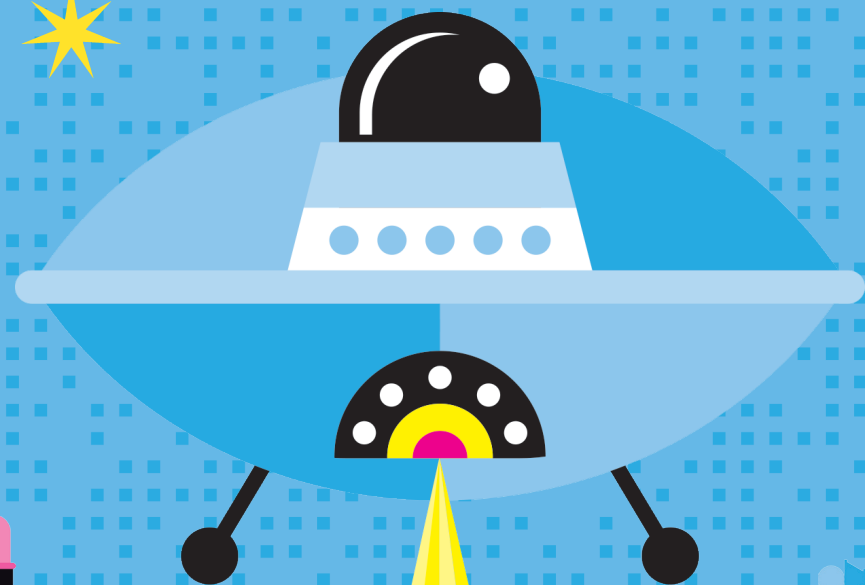


MARC SCOTT

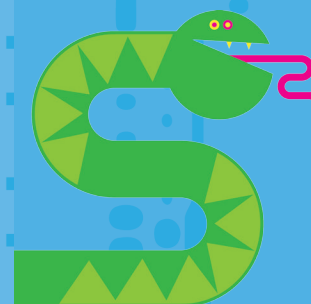
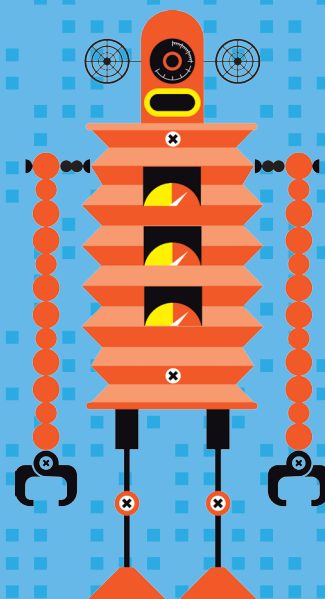
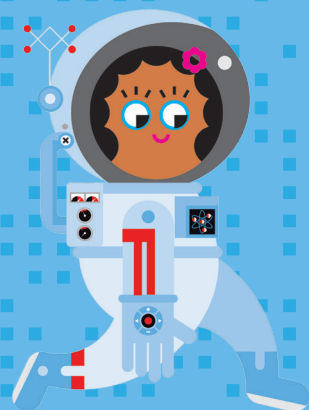


# Kodowanie dla dzieci

ilustracje  
MICK MARSTON

przełożył KRZYSZTOF WOŁCZYK

NASZA KSIĘGARNIA



Mojemu synowi Jamesowi,  
najlepszemu recenzentowi, jakiego można mieć.



Tytuł oryginału *A Beginner's Guide to Coding*

Copyright © 2016 A & C Black

Text copyright © 2016 Marc Scott

Illustration copyright © 2016 Mick Marston

This translation of *A Beginner's Guide to Coding* [first edition] is published by  
Wydawnictwo „Nasza Księgarnia”, Sp. z o.o. by arrangement with Bloomsbury  
Publishing Plc and Macadamia Literary Agency, Warsaw.

Polish edition copyright © by Wydawnictwo „Nasza Księgarnia”, Warszawa 2018

Zrzuty ekranu wykonano na podstawie wersji v437 programu Scratch.  
W innych wersjach niektóre elementy mogą się różnić.

ISBN: 978-83-10-13258-1

Redaktor prowadzący *Anna Garbal*  
Opieka redakcyjna *Magdalena Korobkiewicz*  
Redakcja *Ewa Mościcka*

Redakcja techniczna *Paweł Nowicki*  
Konsultacja merytoryczna *Przemysław Kasprzak*  
Korekta *Malwina Łozińska, Jolanta Gomółka*

Wydawnictwo NASZA KSIĘGARNIA Sp. z o.o.  
02-868 Warszawa, ul. Sarabandy 24c  
tel.: 22 643 93 89, 22 331 91 49  
faks 022 643 70 28  
e-mail: [naszaksięgarnia@nk.com.pl](mailto:naszaksięgarnia@nk.com.pl)

Dział Handlowy tel. 22 331 91 55, tel./faks 22 643 64 42  
Sprzedaż wysyłkowa tel. 22 641 56 32  
e-mail: [sklep.wysylkowy@nk.com.pl](mailto:sklep.wysylkowy@nk.com.pl)  
[www.nk.com.pl](http://www.nk.com.pl)

PRINTED IN POLAND

Wydawnictwo „Nasza Księgarnia”, Warszawa 2018 r.

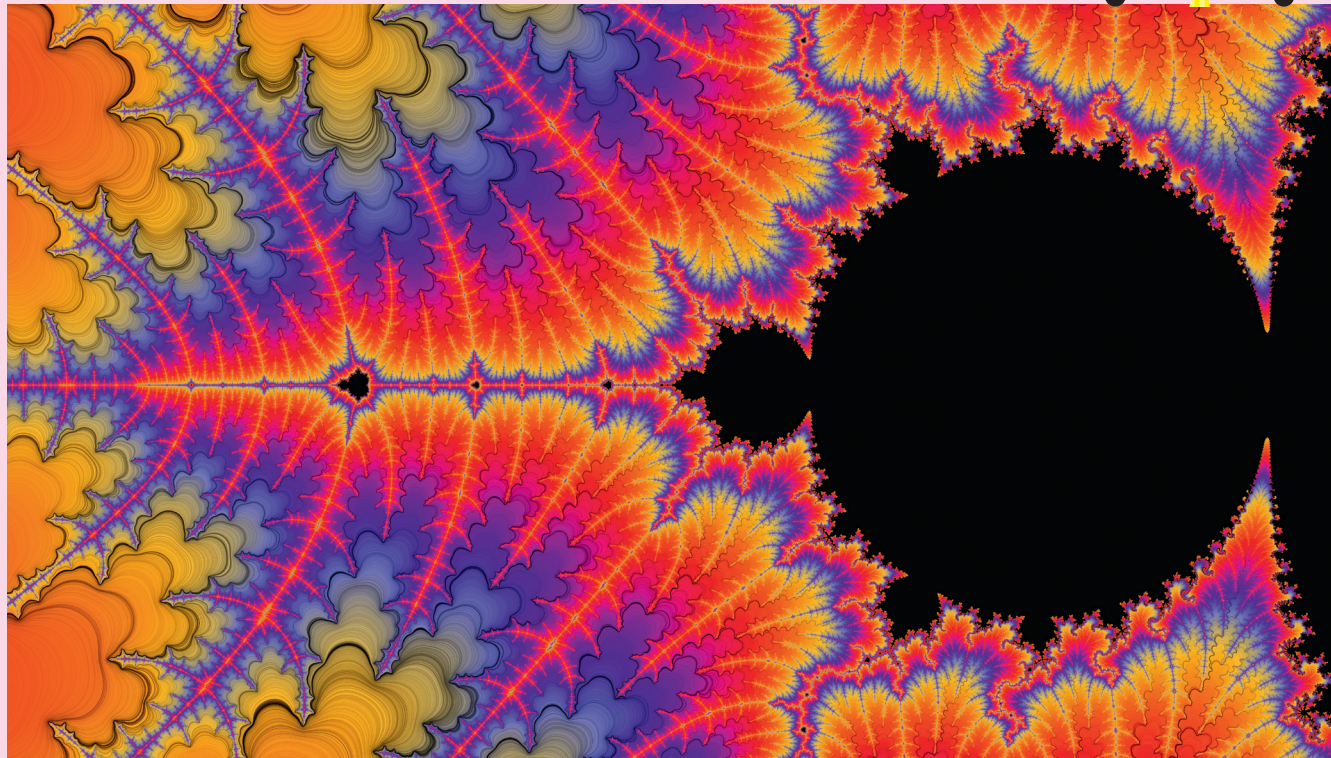
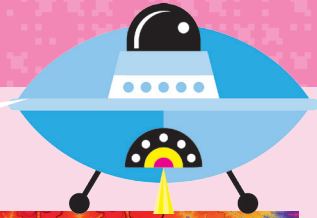
Wydanie pierwsze

Druk: Zakład Graficzny COLONEL, Kraków

# Rysowanie w Scratchu



Czy domyślasz się, jak powstał ten obrazek?



Został stworzony za pomocą komputera i matematyki. Nazywany jest **fraktalem**. W internecie znajdziesz wiele przykładów fraktali. To wspaniałe obrazki tworzone za pomocą powtarzających się wzorów.

W ramach tego projektu nauczysz się samodzielnie rysować w Scratchu i zauważysz, że powtarzając wielokrotnie rysunki prostych kształtów, da się skomponować ciekawy obraz.

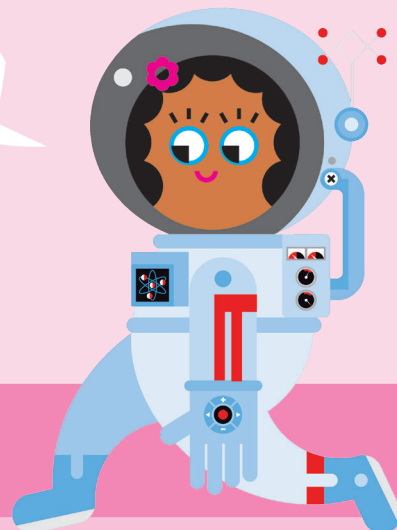
## Podstawowe rysunki

1. Na początku dodaj nowy projekt i usuń duszka kota, podobnie jak w poprzednim projekcie.
2. Następnie wybierz nowego duszka. Tym razem niech to będzie strzałka. Łatwo będzie zauważyć, którą stronę wskazuje.

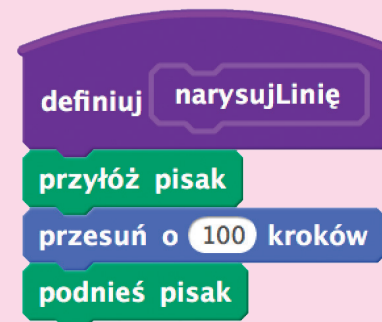


### Fraktal

- nigdy niekończące się, powtarzające się wzory.



3. Utwórz teraz kolejny własny blok (pamiętaj, że możesz to zrobić, korzystając z sekcji **Więcej bloków**). Nazwij go **narysujLinie**.
4. W palecie bloków wybierz sekcję **Pisak**.
5. Przeciągnij blok **przyłóż pisak** na obszar skryptów i dołącz do bloku „narysujLinie”.
6. Chcemy mieć możliwość przemieszczania duszka, by narysować linię. W sekcji **Ruch** wybierz blok **przesuń o 10 kroków** i umieść u dołu bloku **przyłóż pisak**. Zmień 10 na 100. Następnie do skryptu dodaj blok **podnieś pisak**.
7. Twój blok powinien wyglądać tak:

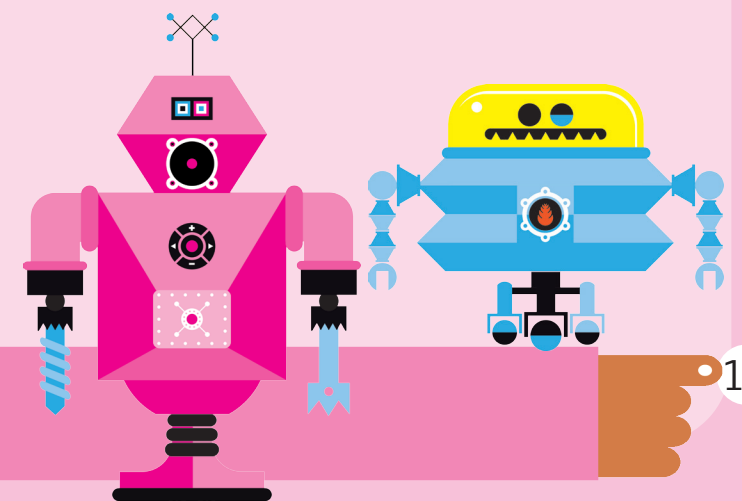


8. Kiedy klikniesz na zdefiniowany właśnie blok, zauważysz, że strzałka na scenie się przemieszcza i rysuje linię.
9. Teraz należy wyczyścić scenę i ponownie ustawić strzałkę w początkowym położeniu, pośrodku sceny. Utwórz nowy blok własny i nazwij go **resetuj**.

10. Do nowego bloku **definiuj resetuj**, korzystając z sekcji **Ruch**, dołącz blok **idź do x: 0 y: 0**, a następnie **wyczyść** z sekcji **Pisak**. Po kliknięciu na nowo zdefiniowany blok strzałka powinna wrócić na swoje początkowe miejsce, a ekran powinien zostać wyczyszczony.
11. Obszar skryptów powinien zawierać teraz dwa bloki:



12. Zawsze gdy klikniesz **definiuj resetuj**, linia zostanie wyczyszczona, a strzałka wróci na swoje początkowe położenie – na środek sceny.
- A kiedy klikniesz na **definiuj narysujLinie**, linia zostanie narysowana.



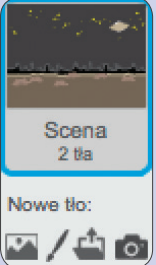


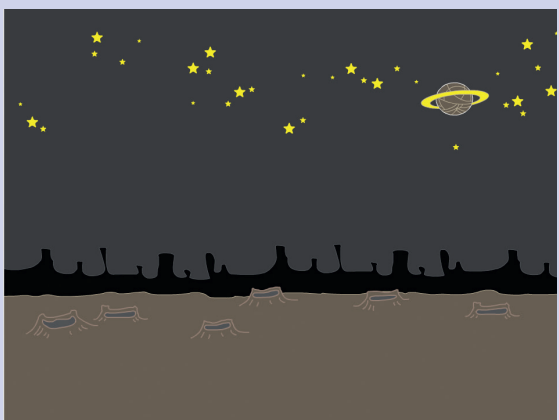


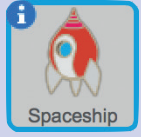
# Atak klonów

Już wiesz, jak się poruszać w środowisku Scratcha. Najwyższy czas stworzyć pierwszą grę. Zaczniemy od takiej, w której gracz porusza się statkiem, próbując uniknąć zderzenia z latającymi robotami.

## Manewrowanie statkiem kosmicznym

1. Utwórz w Scratchu nowy projekt.
2. Usuń duszka kota i kliknij ikonę , by dodać nowe tło do sceny. Tło o nazwie „space” wydaje się najbardziej odpowiednie do naszej gry.



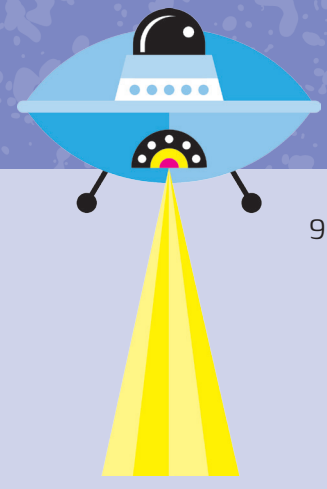
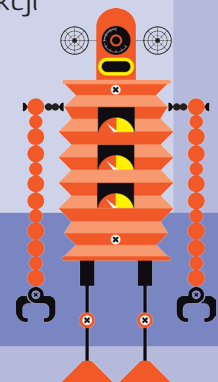
3. Następnie wybierz nowego duszka. Niech to będzie „Spaceship”. 
4. Zacznijmy od tego, by upewnić się, że przed rozpoczęciem gry statek jest odpowiedniej wielkości, znajduje się w miejscu przewidzianym na start oraz że jest skierowany we właściwą stronę. By mieć taką pewność, należy przygotować zestaw bloków jak na rysunku:

```
kiedy kliknięto
ustaw kierunek na 180
ustaw rozmiar na 40 %
idź do x: 0 y: 0
```

5. Teraz dopiero zacznie się zabawa... Sprawmy, by statek się poruszał! W tym celu potrzebujemy pętli

```
zawsze
  [blok], która będzie działała,
  dopóki gra będzie trwać. Blok z pętlą
  „zawsze” jest bardzo przydatny do
  wykrywania zdarzeń zachodzących
  podczas gry, takich jak np. wciśnięcie
  klawisza czy zderzenie z innym
  statkiem.
```

6. Przenieś blok **Kontrola** na dół twojego skryptu. Chcemy, aby nasz program zareagował na przyciśnięty klawisz, zatem do środka pętli przenieś blok **Czujniki** pobierz blok **klawisz spacja naciśnięty?**.



```
kiedy kliknięto
ustaw kierunek na 180
ustaw rozmiar na 40 %
idź do x: 0 y: 0
zawsze
  jeżeli klawisz spacja naciśnięty? to
```

7. Jaki klawisz wybierzemy i co ma się wydarzyć, kiedy zostanie on przyciśnięty? Zacznijmy od „strzałki w górę”. Chcemy, aby pozycja „y” rosła o 10 za każdym razem, gdy zostanie naciśnięty klawisz „strzałka w górę”.


```
zawsze
  jeżeli klawisz strzałka w górę naciśnięty? to
    zmień y o 10
```

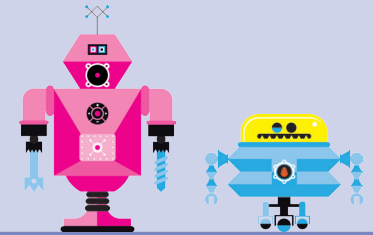
8. Klikając prawym przyciskiem myszy (lub lewym, trzymając wciśnięty klawisz „Shift”) na dany fragment skryptu, możesz go zduplikować, używając opcji „duplikuj” z menu, które wtedy zostanie wyświetlone. W ten sposób nie musisz za każdym razem przenosić tych samych bloków kodu czy całych fragmentów programu z palety bloków. My chcemy powielić cały blok „jeżeli”.

```
kiedy kliknięto
ustaw kierunek na 180
ustaw rozmiar na 40 %
idź do x: 0 y: 0
zawsze
  jeżeli klawisz strzałka w górę naciśnięty? to
    duplikuj
    usuń
    dodaj komentarz
    help
```

9. Zduplicuj go trzykrotnie i zmień odpowiednio klawisz, na który program ma reagować, oraz kierunek ruchu statek. W ostatnich dwóch powielonych fragmentach zamień blok **zmień y o 10** na **zmień x o 10**, pamiętając również o zmianie kierunku ruchu w zależności od klawisza używanego przez gracza. Ostatecznie skrypt powinien wyglądać tak:

```
kiedy kliknięto
ustaw kierunek na 180
ustaw rozmiar na 40 %
idź do x: 0 y: 0
zawsze
  jeżeli klawisz strzałka w górę naciśnięty? to
    zmień y o 10
  jeżeli klawisz strzałka w dół naciśnięty? to
    zmień y o -10
  jeżeli klawisz strzałka w prawo naciśnięty? to
    zmień x o 10
  jeżeli klawisz strzałka w lewo naciśnięty? to
    zmień x o -10
```

10. Teraz możesz już kliknąć ikonę  i sprawdzić, czy masz możliwość manewrowania statkiem zgodnie z oczekiwaniem.



# Zaczynamy z modulem turtle

## Kilka ciekawych komend w module turtle

1. Otwórz IDLE.
2. Zaczniemy naukę od części Pythona nazywanej „turtle”. To moduł używany w Pythonie do rysowania.

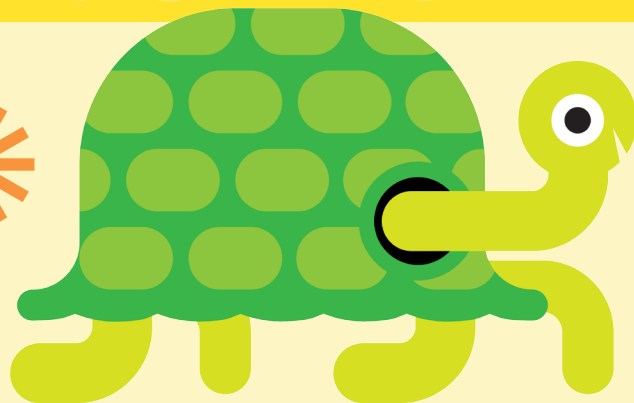
**Moduł** – kod napisany przez innych ludzi, który możesz wykorzystać we własnych programach. Programiści nie lubią pisać ponownie kodu, który już przez kogoś został napisany, dlatego wykorzystują istniejące moduły, co pozwala szybciej tworzyć inne, bardziej złożone programy.

3. W shellu wprowadź poniższą komendę i wciśnij „Enter”. Nie obawiaj się, że nie przyniesie to na ekranie IDLE żadnego efektu. To poprawne zachowanie.

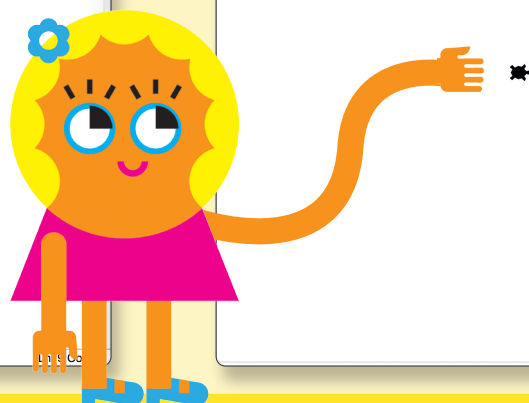
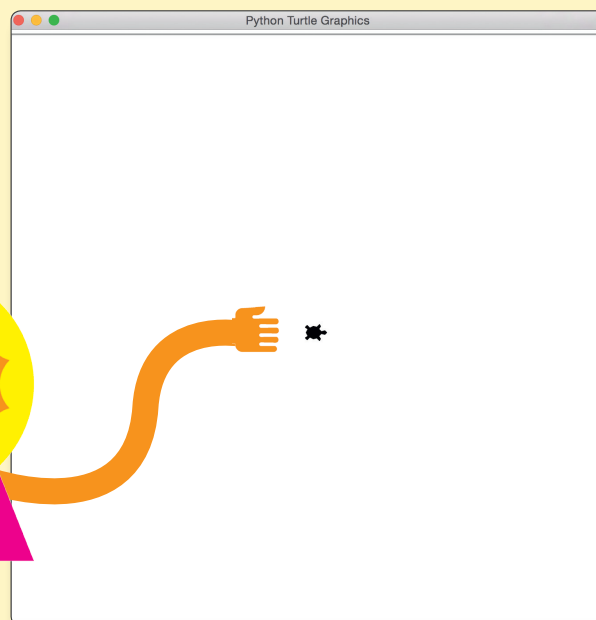
```
from turtle import *
```

```
Python 3.4.3 (default, Jul 3 2015, 01:28:31)
[GCC 4.2.1 Compatible Apple LLVM 6.1.0 (clang-602.0.53)] on darwin
Type "copyright", "credits" or "license()" for more information.

>>> from turtle import *
>>> shape("turtle")
>>>
```



4. Pamiętaj o składni. Upewnij się, że na końcu komendy został wpisany znak \*.
5. W kolejnym kroku wyświetlisz kształt żółwia w nowym oknie. W tym celu wpisz poniższą komendę w shellu i naciśnij „Enter”:  
**shape("turtle")**
6. Wyświetlone zostanie okno podobne do poniższego:



7. Ruszmy teraz żółwia. Wpisz następujące komendy do shella:

```
fd(100)
rt(90)
fd(100)
```

Być może już się domyślasz, że „fd” oznacza „naprzód” (ang. *forward*), „rt” oznacza „prawo” (ang. *right*). Możemy również używać komend „bk” – „wstecz” (ang. *backward*) i „lt” dla „w lewo” (ang. *left*). Komenda „lt(90)” oznacza, że żółw zmieni kierunek w lewo o 90 stopni:

```
bk(100)
lt(90)
```

8. Spróbuj teraz samodzielnie przemieszczać żółwia po ekranie, zmieniając liczby w nawiasach przy ustalaniu długości ruchu czy wielkości kąta zmiany kierunku.
9. Spróbuj narysować inne kształty. Jeśli się pomylisz, zawsze możesz wrócić do pozycji początkowej żółwia, wpisując komendę:

```
home()
```

## Kontrola nad pisakiem żółwia

1. Przenieś teraz żółwia do położenia początkowego i wyczyść ekran.

```
home()
clear()
```

Te dwie komendy możesz wywoływać zawsze, gdy chcesz rozpocząć pracę nad nowym rysunkiem.

2. Możesz też wpływać na kolor i grubość pisaka. Spróbuj następujących poleceń:

```
fd(50)
color("red")
fd(50)
color("green")
width(10)
fd(50)
```

3. Możemy również podnieść pisak (komenda „pu”, ang. *put up*) i ponownie go opuścić (komenda „pd”, ang. *put down*), jeśli chcemy skończyć pisanie bądź zacząć je od nowa.

```
home()
clear()
fd(100)
pu()
fd(100)
pd()
fd(100)
```

4. Jeśli popełnisz błąd, możesz cofnąć ostatnio wydaną komendę:

```
undo()
```

5. Spróbuj jeszcze jednej komendy przed podjęciem poważniejszych wyzwań. Stosując to polecenie, sprawisz, że żółw narysuje okrąg:

```
circle(100)
```

