3.2 Vlastnosti

Python je interpretovaný, interaktivní a objektově orientovaný programovací jazyk. Často je srovnáván s Tcl, Perl, Scheme nebo Javou.

Python se jednoduše učí a je to mocný programovací jazyk. Má výkonné vysokoúrovňové datové struktury a jednoduchý, přesto mocný, přístup k objektovému programování. Pythonovská elegantní syntaxe a dynamické typování, společně s jeho interpretovanou povahou, ho činí ideálním jazykem pro skriptování a rychlý vývoj aplikací v mnoha oblastech na většině platforem.

Pythonovský interpret a rozsáhlá standardní knihovna jsou zcela volně šiřitelné ve zdrojové nebo binární formě na všech hlavních platformách z pythonovské webové stránky Python.org. Na stejné stránce naleznete distribuce nebo odkazy na mnoho volně šiřitelných pythonovských modulů, programů a nástrojů třetích stran a i další dokumentaci.

Pythonovský interpret se dá snadno rozšířit pomocí nových funkcí a datových typů implementovaných v C nebo C++ (nebo jiného jazyka volatelného z C). Je také vhodný jako rozšiřující jazyk pro aplikace, které si uživatel přizpůsobuje svým potřebám.

3.3 Verze

16. 10. 2000 – verze 2.0 (komunitní vývoj, full garbage collector, prvotní podpora pro Unicode...)

3. 12. 2008 – verze 3.0, záměrně první zpětně nekompatibilní (skutečná podpora Unicode'u, sjednocení syntaxe a „vnitřností“, vyčištění systémové knihovny...)

4. Práce s Pythonem

4.1 Pohyb želvy

forward( )	fd( )	Posun želvy o x kroků dopředu	fd(25)
back( ), backward( )	bk( )	Posun želvy o x kroků dozadu	bk(25)
right( )	rt( )	Otočení želvy o x stupňů doprava	rt(90)
left( )	lt( )	Otočení želvy o x stupňů doleva	lt(90)
goto( ), setposition( )	setpos( )	Přesunutí želvy do určité pozice o souřadnicích x, y. V případě že je pero dole, nakreslí želva čáru.	goto(60,30)
setx( )		Nastavení souřadnice x	setx(10)
sety( )		Nastavení souřadnice y	sety(10)
setheading( )	seth( )	Nastavení směru želvy ve stupních: 0 - východ	seth(90)

		90- sever 180 - západ 270 - jih	
home( )		Želva se postaví doprostřed plátna (do své výchozí pozice) .	home( )
circle( )		Želva vytvoří kruh	circle(50)
dot( )		Želva vytvoří puntík	dot( )
undo( )		Smaže poslední krok želvy	Undo(
speed( )		Nastavení rychlosti želvy: “nejrychlejší” - 0 “rychlý” - 10 “normální” - 6 “pomalý” - 3 “nejpomalejší” - 1	sped(9) speed(“normal”)

4.2 Práce s perem

pendown( )	down( ), pd( )	Položení pera želvy - želva kreslí čáru když se pohybuje.	pd()
penup( )	up( ), pu( )	Zvednutí pera želvy - želva nekreslí čáru když se pohybuje.	pu()
pensize( ), widht( )		nastavení tloušťky čáry pera.	pensize(3)

4.3 Nastavení barvy čáry a výplně

color( )	Nastavení barvy čáry pera i výplně.	color(“yellow”, “red”)
pencolor( )	Nastavení barvy čáry pera.	pencolor(“yellow”)
fillcolor( )	Nastavení barvy výplně.	fillcolor(“red”)
bgcolor( )	Nastavení barvy celého plátna.	bgcolor(“black”)
begin_fill( )	Obraz se bude vyplňovat.	begin_fill( )
end_fill( )	Ukončení výplně.	end_fill( )

4.4 Smazání plátna

reset( )	Smazání všech obrazců nakreslených želvou a posun želvy do výchozí pozice.
clear	Smazání všech obrazců nakreslených želvou - želva se nepohne.

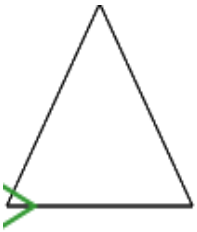
4.5 Objevení a skrytí želvy

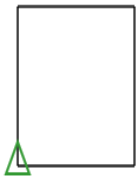
showturtle( )	st( )	Objevení želvy, želva je viditelná.
hideturtle( )	ht( )	Skrytí želvy, želva je neviditelná

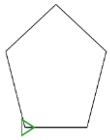
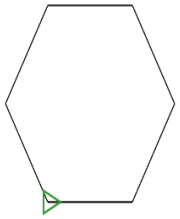
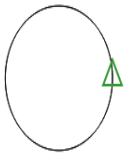
5. Příklady želví grafiky

5.1 Základní tvary

Editor	Python	Obrázek

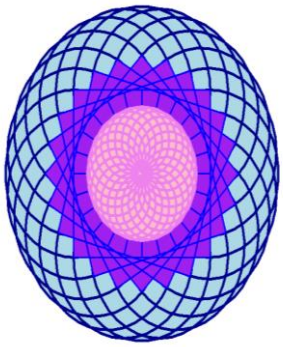
`rt 30`
`repeat 3 [`
`fd 100`
`rt 360/3`
`]`
	`from turtle import fd, lt`
	`for k in range(3):`
	`  fd(100)`
	`  lt(120)`
		

`repeat 4 [`
`fd 100`
`lt 360/4`
`]`
	`from turtle import fd, lt`
	`for i in range(4):`
	`  fd(100)`
	`  lt(360/4)`
		

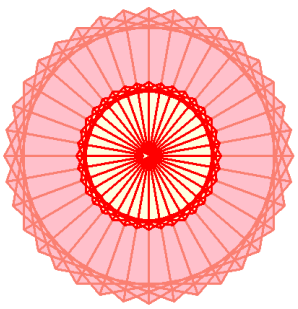
right 90 repeat 5 [forward 100 left 360/5]	from turtle import lt, fd for i in range(5): fd(100) lt(360/5)	
right 90 **repeat 6[** **forward 100** **left 360/6** **]**	from turtle import lt, fod for i in range(6): fd(100) lt(360/6)	
repeat 360[**forward 1** **left 1** **]**	from turtle import circle circle(50)	

5.2 Složitější tvary v Pythonu

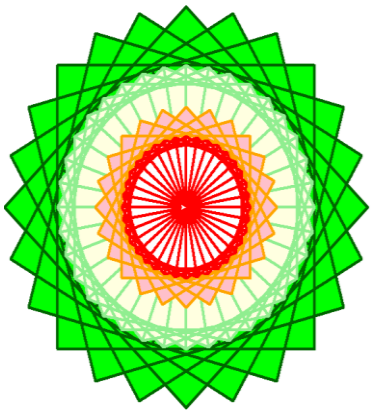
```
from turtle import *

speed(0)
pensize(5)
color("dark blue", "light blue")

begin_fill()
for k in range(24):
   circle(150)
   left(15)
end_fill()

pensize(3)
color("blue", "purple")

begin_fill()
for k in range(24):
   for k in range(4):
       forward(150)
       left(90)
   left(15)
end_fill()

pensize(5)
color("violet", "Pink")

begin_fill()
for k in range(24):
   circle(60)
   left(15)
end_fill()

done()
```



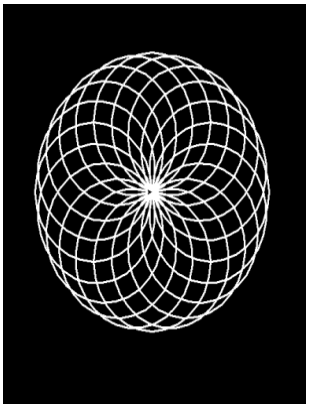
```
from turtle import *
speed(0)
pensize(3)
color("salmon", "pink")
begin_fill()
for i in range(36):
   for k in range(3):
       forward(200)
       left(360/3)
   left(10)
end_fill()

color("red", "light yellow")
begin_fill()
for i in range(36):
   for k in range(3):
       forward(150)
       left(360/3)
   left(10)
end_fill()

done()
```



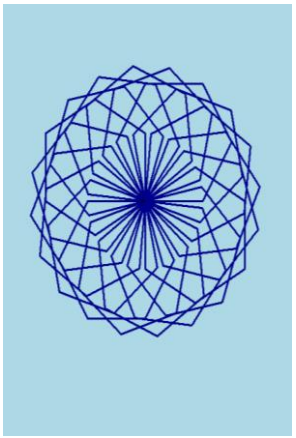
```
from turtle import *
pensize(4)
speed(0)
color("dark green", "green")
begin_fill()
for k in range(24):
    for k in range(4):
        forward(200)
        left(90)
    left(15)
end_fill()

color("light green", "light yellow")
begin_fill()
for i in range(36):
    for k in range(3):
        forward(200)
        left(360/3)
    left(10)
end_fill()

color("purple", "pink")
begin_fill()
for k in range(24):
    for k in range(4):
        forward(100)
        left(90)
    left(15)
end_fill()

color("red", "white")
begin_fill()
for i in range(36):
    for k in range(3):
        forward(100)
        left(360/3)
    left(10)
end_fill()

done()
```



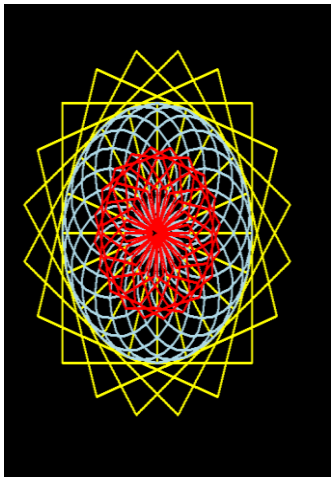
```
from turtle import *
speed(0)
bgcolor("black")
pensize(3)
pencolor("white")

for k in range(20):
    circle(100)
    left(18)

done()
```



```
from turtle import *
speed(0)
pensize(3)
bgcolor("light blue")
pencolor("dark blue")

for k in range(20):
    for i in range(6):
        forward(100)
        left(360/6)
    left(18)
done()
```

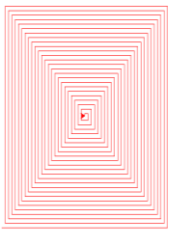


```
from turtle import *
speed(0)
pensize(3)
bgcolor("black")

pencolor("yellow")
for k in range(20):
    for i in range(4):
        forward(150)
        left(360/4)
    left(18)

pencolor("light blue")
for k in range(20):
    circle(75)
    left(18)

pencolor("red")
for k in range(20):
    for i in range(6):
        forward(50)
        left(360/6)
    left(18)

done()
```

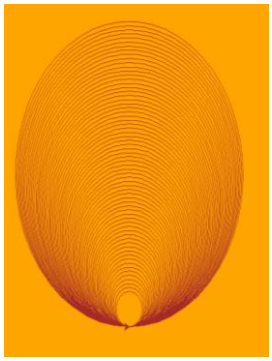


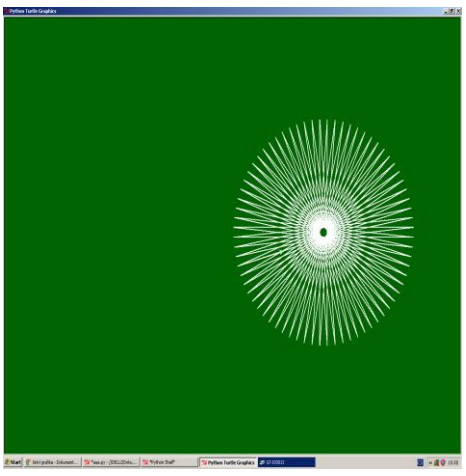
```
from turtle import *
speed(0)
bgcolor("light yellow")
color("green")

x = 1
while x < 400:
    fd(25 + x)
    rt(72)
```

`    x = x+1.5`	

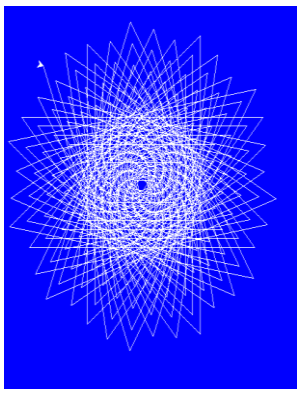
`from turtle import *`
 	
`speed(0)`
`color("red")`
 	
`side=400`
 	
`for i in range (100):`
`  fd(side)`
`  lt(90)`
`  side=side-4`
	

`from turtle import *`
 	
`speed(0)`
`color("light blue")`
 	
`side=400`
 	
`for i in range (100):`
`  fd(side)`
`  lt(120)`
`  side=side-4`	

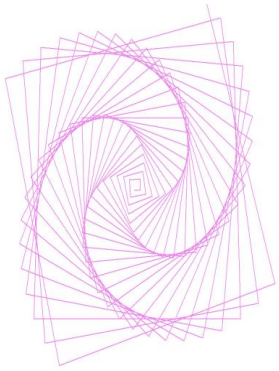
`from turtle import *`
`speed(0)`
`bgcolor("orange")`
`color("brown")`
 	
`x = 1`
`while x < 200:`
`    circle(25 + x)`
 	
`    x = x+3`
	

`from turtle import *`
`speed(0)`
`pensize(2)`
`bgcolor("dark green")`
`color("white")`
 	
`x = 1`
`while x < 400:`
`    fd(500 + x)`
`    rt(185)`
	

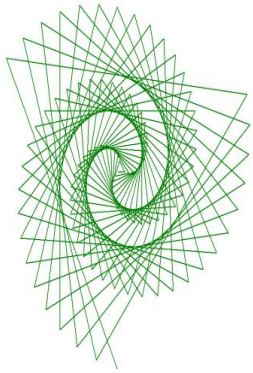
```
from turtle import *
speed(0)
bgcolor("blue")
color("white")

x = 1
while x < 400:
   fd(25 + x)
   rt(360/3)
   rt(10)

   x = x+1.5
```



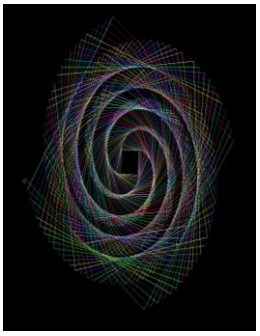
```
from turtle import *

speed(0)
pensize(2)
color("violet")

side=20

for i in range (100):
 fd(side)
 lt(92)
 side=side+7

penup()
goto(500,500)
```



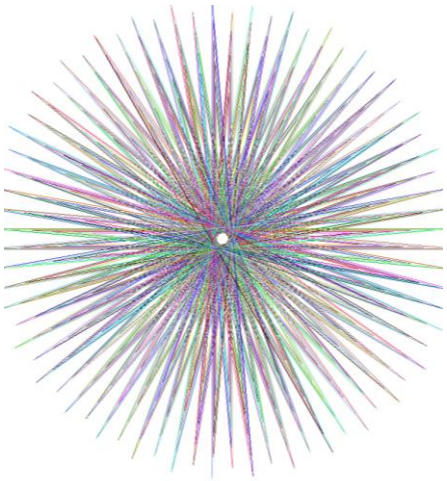
```
from turtle import *

speed(0)
pensize(2)
color("green")
side=20

for i in range (100):
 fd(side)
 lt((360/3)+3)
 side=side+7

penup()
goto(500,500)
```



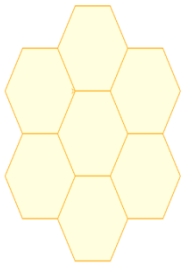
```
from turtle import *
from random import randint

speed(0)
bgcolor('black')

x = 1
while x < 400:

   r = randint(0,255)
   g = randint(0,255)
   b = randint(0,255)
   colormode(255)
   pencolor(r,g,b)

   fd(50 + x)
   rt(91)

   x = x+1
```



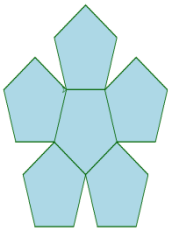
```
from turtle import *
from random import randint

speed(0)

x = 1
while x < 400:

   r = randint(0,255)
   g = randint(0,255)
   b = randint(0,255)
   colormode(255)
   pencolor(r,g,b)
   fd(450+x)
   lt(185)

   x = x+1
```



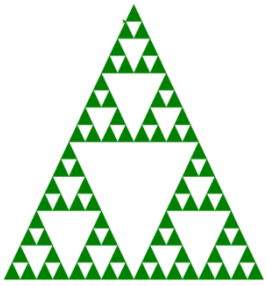
```
# žluté šestitiúhelníky (fexagony)
from turtle import *

pensize(2)
color("orange", "light yellow")

begin_fill()
def hexagon():
 for i in range(6):
     forward(100)
     left(60)

for i in range (6):
   hexagon()
   forward(100)
   right(60)
end_fill()

done()
```



```
# modré pětiúhelníky (pentagony)
from turtle import *
pensize(2)
color("dark green", "light blue")

begin_fill()
def pentagon():
  for i in range(5):
     fd(100)
     lt(360/5)

for i in range (5):
  pentagon()
  forward(100)
  right(72)
end_fill()

done()
```



```
from turtle import *

color("green")
size=800
min=64
pf=0.8660254
def S(l,x,y):
   if l>min:
       l=l/2
       S(l,x,y)
       S(l,x+l,y)
       S(l,x+l/2,y+l*pf)
   else:
       goto(x,y); pendown()
       begin_fill()
       fd(l); left(120)
       fd(l); left(120)
       fd(l)
       end_fill()
       setheading(0)
       penup(); goto(x,y)
penup()
speed(0)
S(size,-size/2,-size*pf/2.0)
done()
```



Závěr

Seminární práce pojednávala o Želví grafice v Pythonu a výukovém webovém editoru studenta Masarykovy univerzity Martina Korbela. Úvodní část byla věnována historii, zároveň bylo v této části uvedeno a vysvětleno množství pojmů. Následující kapitola je zaměřena na metodiku výuky programování s množstvím názorných příkladů od nejjednodušších po složitější.

Literatura

http://programujte.com/clanek/1970010106-python-popis-jazyka/

https://odevzdej.cz/auth/podob/2c57f8e8d9321d16/b73ef336a18bb122/

http://cs.wikipedia.org/wiki/Logo_(programovac%C3%AD_jazyk)

http://cs.wikipedia.org/wiki/Logo_programming_language

https://cs.m.wikipedia.org/wiki/Python_(programovac%C3%AD_jazyk)