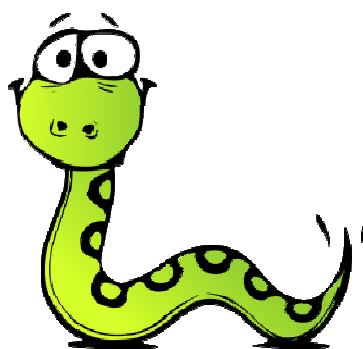


Základy programování v jazyce Python pro střední školy

Lekce 8

Podprogramy



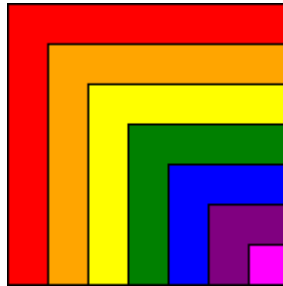
Andrej Blaho

Ľubomír Salanci

Václav Šimandl

beta verze - 2019

1. Vytvoř program `duha.py`, který nakreslí kostičkovou duhu. Do proměnných `x`, `y` přiřaď souřadnice pravého dolního rohu kostičkové duhy a použij je při kreslení barevných čtverců. Nejmenší čtverec má rozměry 20 x 20 a každý další je o 20 větší:



2. Doposud jsi mohl psát jen takové příkazy, které počítač znal. Teď ho naučíš nové, své vlastní příkazy – tzv. **podprogramy**. Postupuj následovně:

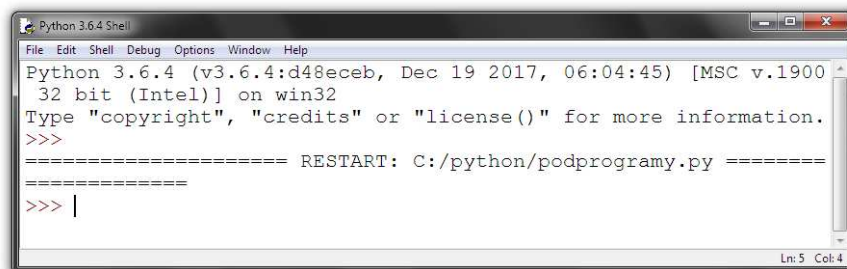
A) Vytvoř nový program `vypis.py`, ve kterém bude napsaný jen následující kód:

prázdné závorky i dvojtečka jsou velmi důležité

```
def vypis_text():
    print('*****')
    print('** Python **')
    print('*****')
```

↑
Příkazy nech odsazené od kraje (Python tam automaticky vložil 4 mezery)

B) Program spust' – jestli je všechno v pořádku, uvidíš:



C) Do příkazového řádku napiš:

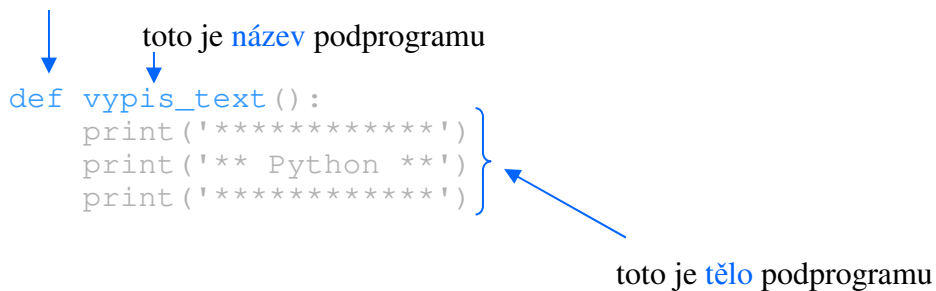
```
>>> vypis_text()
```

D) Jestli jsi postupoval správně, Python zobrazí text:

```
*****
** Python **
*****
```

Co se stalo?

slovem `def` začíná **definice** tvého nového příkazu – podprogramu



```
def vypis_text():
    print('*****')
    print('** Python **')
    print('*****')
```

toto je **tělo** podprogramu

Po spuštění programu se počítač naučil nový příkaz `vypis_text`. Počítač ho zatím nevykonával, jen se ho naučil. Skupinu příkazů `print` – tedy tělo podprogramu `vypis_text` – počítač vykoná až tehdy, když do příkazového řádku napíšeš:

```
>>> vypis_text()
```

prázdné závorky jsou velmi důležité

Takovýto zápis se nazývá **volání** podprogramu.

3. Přidej do svého programu další příkazy (jsou zvýrazněny žlutě) – pozor, tyto příkazy nesmí mít odsazení:

```
def vypis_text():
    print('*****')
    print('** Python **')
    print('*****')

print('Vítej!')
vypis_text()
print()
vypis_text()
print('to je konec')
```

Když program spustíš, uvidíš takovýto výsledek:

```
Vítej!
*****
** Python **
*****

*****
** Python **
*****
to je konec
```

V tomto programu se nejdříve definoval podprogram `vypis_text`. Za ním následují příkazy `print` a příkazy pro volání podprogramu `vypis_text`. Python zobrazil svoji vizitku dvakrát, protože v programu jsou dvě volání podprogramu `vypis_text`.

4. Změň předcházející program tak, aby počítač vypsál:

```

Hello!
*****
** I am Python **
*****
How are you?
*****
** I am Python **
*****
I am fine.
*****
** I am Python **
*****
The end

```

5. Před několika týdny jsi vytvářel program, který zobrazil tvoji vizitku, které byla podobná následující:

```

+-----+
|               |
| Petr          | ~~~ |
| RUMCAJS       | / o o \ |
|               | \  ~  / |
|               | " " " |
|               | IT loupežník |
|               |
+-----+

```

Vytvoř nový program `vizitka.py` a v něm definuj podprogram `vizitka`, který takovou vizitku zobrazí. Nakonec tento podprogram zavolej, abys ověřil, že funguje správně.

6. Dopln do programu `vizitka.py` volání podprogramu `vizitka` tak, aby se pod sebe zobrazilo 10 tvých vizitek.

7. Ve svém programu můžeš definovat i více podprogramů. Vytvoř nový program `obrazce.py` a definuj v něm tři podprogramy. Každý z nich zobrazí jeden z následujících obrázků:

- podprogram `noha` nakreslí takovouto nohu (dole jsou dvě podtržítka vlevo i vpravo):

```

  |
—|—

```

- podprogram `obdelnik` nakreslí takovýto obdélník:

```

####
#   #
####

```

- podprogram `trojuhelnik` nakreslí takovýto trojúhelník:

```

 *
 ***
 *****

```

Na konec programu vlož volání podprogramů, abys každý z nich otestoval. Potom zkus pomocí vytvořených podprogramů zobrazit následující obrázky:

<pre> * *** ***** * *** ***** _ _ </pre>	<pre> * *** ***** ##### ##### # # ##### _ _ </pre>	<pre> ##### # # ##### _ _ ##### # # ##### </pre>	<pre> toto je noha: _ _ toto je obdélník: ##### # # # # ##### toto je trojúhelník: * *** ***** </pre>
--	---	---	---

8. Teď budeš vytvářet podprogramy, které kreslí do grafického okna. Vytvoř nový program `kresba_podprogram.py` a vyzkoušej:

```

import tkinter

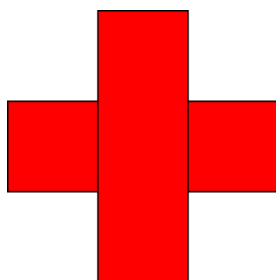
canvas = tkinter.Canvas()
canvas.pack()

def kresli():
    canvas.create_rectangle(10, 20, 30, 40, fill='red')

kresli()

```

9. Vytvoř nový program `kriz.py` a v něm definuj podprogram `kriz`, který po zavolání nakreslí červený kříž:



10* Vytvoř nový program `robot.py`, který bude schopen nakreslit robota. V programu budou čtyři podprogramy – `hlava`, `ruce`, `nohy`, `telo` – a každý z nich bude schopen nakreslit část robota. Když je zavoláš v následujícím pořadí:

```
hlava()  
ruce()  
nohy()  
telo()
```

nakreslí se celý robot jako na obrázku vpravo:

