Lekce 5

Funkce

Funkce, argumenty funkcí, návratové hodnoty, rekurze a anonymní funkce lambda.

C2184 Úvod do programování v Pythonu
podzim 2015

Mgr. Stanislav Geidl
Národní centrum pro výzkum biomolekul
Masarykova univerzita

Funkce

- funkce je **část programu** (je podprogram, angl. subroutine), která lze volat opakovaně z různých části programu, příkladem může být funkce print()
- každá funkce má svůj **identifikátor** (podobně jako proměnná), pomocí kterého ji můžeme volat
  `moje_funkce()`
- požadavky na funkci můžeme specifikovat pomocí **parametrů**
  `moje_funkce(parametr1, parametr2)`
- výsledek funkce může být předán jako **návratová hodnota**
  `navratova_hodnota = moje_funkce()`

Definice funkce

- pomocí klíčového slova `def` a bloku (: a odsazení)

```
def moje_funkce(parametr):
    pass
```

- návratovou hodnotu definujeme klíčovým slovem `return`, POZOR: po jejím zavolání se již v provádění funkce nepokračuje

```
def moje_funkce(parametr):
    print()
    return True
    print() # nedosažitelný kód
```

- pomocí `return` můžeme navracet hodnotu, hodnotu proměnné nebo také nic (`None`)

```
def moje_funkce(parametr):
    return
```

- defaultní hodnoty a názvy parametrů (`a, b = 15`)
- více parametrů předaných pomocí jedné N-tice (`*parametr`)

Proměnné a funkce

- **globální proměnné** - existují mimo funkci, ale můžeme ji v rámci funkce používat, upravovat pouze za použití **global**

```
a = 1
b = 2
def moje_funkce(parametr):
    print(a)
    print(b) # zde je b lokální, spadne to
    b = 20
print(b)
```

- **parametry**

```
a = 1
def moje_funkce(parametr):
    print(a)
    print(parametr)
```

- **lokální proměnné** - existují pouze v rámci konkretního volání funkce a má přednost před globální proměnnou

```
a = 1
def moje_funkce(parametr):
    a = 10
    b = 20
print(b)
```

Rekurze a rekurzivní volání funkce

- volání sama sebe

```
def moje_funkce():
    moje_funkce()
```

- ukázka výpočtu faktoriálu

```
def faktorial(a):
    if a < 2:
        return 1
    else:
        return a * faktorial(a - 1)
```


Anonymní lambda funkce

- v případě, kdy potřebuje funkci volat "jednorázově"

```
square = lambda x: x**2
square(5) # 25
```

- příklad vlastního porovnávače ve funkci sort:

```
list.sort(cmp=lambda x, y: cmp(abs(x), abs(y)))
```