Programování 1: Funkce

Martin Mareš
mj@ucw.cz

Katedra Aplikované Matematiky
MFF UK Praha

2024

Definice funkce

```
def stekej():
    print("Haf!")

stekej()
stekej()
```

- Opakující se část programu stačí napsat jednou a spustit vícekrát.
- Také jsme tím část programu pojmenovali, takže je při čtení jasné, co má dělat.

Funkce s parametrem

```
def stekej(n):
    for i in range(n):
        print("Haf!")

stekej(5)
```

- Funkci můžeme předat parametr.
- Když zavoláme `stekej(5)`, funkce se spustí s proměnnou `n` nastavenou na `5`.
- Proměnné `n` a `i` existují jen lokálně uvnitř funkce, zvenčí nejsou vidět.

Funkce s výsledkem

```
def minimum(a, b):
    if a < b:
        return a
    else:
        return b

print(minimum(3, 5))
```

- Příkaz **return** ukončí vykonávání funkce a vrátí výsledek.
- Můžeme použít i **return** bez výsledku (vrátí **None**).
- Výsledek funkce nemusíme použít.

Rekurzivní funkce

```
def faktorial(n):
    if n == 0:
        return 1
    else:
        return n*faktorial(n-1)
```

- Funkce volá sama sebe.
- Každé volání má své vlastní lokální proměnné (včetně parametrů).
- Nezapomeneme rekurzi zastavit :)

Ještě o parametrech

Nepovinné parametry:

```
def stekej(n=1, zvuk="Haf!"):
    for i in range(n):
        print(zvuk)
```

Různé způsoby volání:

```
stekej()
stekej(5)
stekej(5, "HAF!")
stekej(n=5, zvuk="HAF!")
stekej(zvuk="HAF!")
```

Viditelnost proměnných: chyták

```
zvuk = "Kuku!"
kolik_hodin = 0

def zakukej():
    print(zvuk)
    kolik_hodin += 1
```

- Funkce vidí globální proměnnou **zvuk**
- Proměnná **kolik_hodin** je ale lokální, protože funkce do ní přiřazuje.

Viditelnost proměnných: řešení

```
zvuk = "Kuku!"
kolik_hodin = 0

def zakukej():
    global kolik_hodin
    print(zvuk)
    kolik_hodin += 1
```