

Programování 1: Triky s funkcemi

Martin Mareš

mj@ucw.cz

Katedra Aplikované Matematiky
MFF UK Praha

2024

Funkce jako parametr

```
# Spočítá  $z[i] = f(x[i], y[i])$ 
def po_slozkach(f, a, b):
    return [ f(x, y) for x, y in zip(a, b) ]

def secti(x, y):
    return x + y

>>> po_slozkach(secti, [1, 2, 3], [10, 20, 30])
[11, 22, 33]    (secti je jméno funkce, secti() její volání)

>>> po_slozkach(max, [10, 2, 3], [1, 20, 30])
[10, 20, 30]

>>> import operator
>>> po_slozkach(operator.add, [1, 2, 3], [10,
20, 30])
[11, 22, 33]
```

Lambda-funkce (funkce bez jména)

```
>>> plus = lambda x, y: x+y
```

```
>>> plus(1, 5)
```

```
6
```

```
>>> po_slozkach(lambda x, y: x*y, [2,4], [3,7])
```

```
[6, 28]
```

Lambdy ve standardní knihovně: třídění

```
>>> k = ["kočka", "sedí", "na", "okně"]
>>> sorted(k, key=lambda x: len(x))
['na', 'sedí', 'okně', 'kočka']

>>> k = ["kočka", "sedí", "na", "okně"]
>>> sorted(k, key=lambda x: (len(x), x))
['na', 'okně', 'sedí', 'kočka']

>>> min(k, key=lambda x: len(x))
'na'

>>> p = [(1, 'leden'), (2, 'unor'), (4, 'duben')]
>>> sorted(p, key=lambda x: x[1])
[(4, 'duben'), (1, 'leden'), (2, 'unor')]
```

Lambdy ve standardní knihovně: map

```
>>> cisla = map(int, ["12", "34"])
>>> list(cisla)
[12, 34]    (nebo také: map(int, input().split()))
>>> suma = map(operator.add, [1,2], [10,20])
>>> list(suma)
[11, 22]
```

Vnořené funkce

```
def f():  
    n = 0  
  
    def krok():  
        nonlocal n  
        n += 1  
        return n  
  
    return [krok(), krok(), krok()]  
  
>>> f()  
[1, 2, 3]
```

Vnořené funkce: vracíme vnitřní funkci

```
def f():
```

```
    n = 0
```

```
    def krok():
```

```
        nonlocal n
```

```
        n += 1
```

```
        return n
```

```
    return krok
```

```
>>> a = f()
```

```
>>> a()
```

```
1
```

```
>>> a()
```

```
2
```

```
>>> b = f()
```

```
>>> b()
```

```
1
```

```
>>> a()
```

```
3
```

```
>>> b()
```

```
2
```

Generátor s yield (korutina):

```
def gen(n):  
    for i in range(n):  
        yield i**2
```

```
for x in gen(10):  
    print(x)
```

Generátorové výrazy: jako list comprehension, ale () místo []

```
>>> g = (i**2 for i in range(100))
```

```
>>> max(g)
```

9801 (seznam není nikde uložený)

```
>>> max(g)
```

Chyba (podruhé selže, generátor se vyčerpá)

```
>>> max(i**2 for i in range(100))
```

9801 (nejsou třeba dvojí závorky)