

Programování 1.9: Lambda funkce

Petr Čermák

`cermak@mag.mff.cuni.cz`

Katedra Fyziky Kondenzovaných Látek
MFF UK Praha

2020

forked from <https://gitlab.kam.mff.cuni.cz/mj/prm1>

Jak python chápe funkce

- Funkce je objekt, stejně jako int, str, list
- Dají se
 - definovat,
 - mazat,
 - předávat jiným funkcím jako parametr,
 - vracet pomocí slova return

Funkce jako parametr

```
1 # Spočítá z[i] = f(x[i], y[i])
2 def po_slozkach(f, a, b):
3     return [ f(x, y) for x, y in zip(a, b) ]
4
5 def secti(x, y):
6     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

Vestavěná funkce `map` má dva argumenty, funkci a iterable (=list, generátor...). Aplikuje funkci postupně na všechny prvky a yielduje výsledky.

```
>>> 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

```
1 def f():
2     n = 0
3
4     def krok():
5         nonlocal n
6         n += 1
7         return n
8
9     return [krok(), krok(), krok()]
```

```
>>> f()
[1, 2, 3]
```

Vnořené funkce: vracíme vnitřní funkci

```
1 def f():
2
3     n = 0
4
5     def krok():
6         nonlocal n
7         n += 1
8         return n
9
10    return krok
```

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

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

```
1
```

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

```
2
```

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

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

```
1
```

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

```
3
```

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

```
2
```


Úkoly na hodinu

- Napište funkci `compose(f,g)`, která pro dvě funkce `f` a `g` (obě s jedním parametrem) vrátí funkci, jež je jejich složením ($f(g(x))$).
- Napište funkci `red(s,f)`, která dostane seznam `s` a funkci `f(x,y)` a spočítá $f(f(f(s[0], s[1]), s[3]), s[4])$.
- Zapište pomocí redukce součet prvků seznamu.
- Zapište pomocí redukce nalezení maxima seznamu.
- Zapište pomocí redukce nalezení prvního nenulového prvku (není-li, vraťte 0).
- Co redukce udělá pro operátor `-`?
- Co udělá `red(s, lambda x, y: (x,y))`?