

Programování 1: List comprehensions

Martin Mareš

`mj@ucw.cz`

Katedra Aplikované Matematiky
MFF UK Praha

2024

Odbočka: n-tice (tuples)

```
>>> t = (1, 2, 3)    (vytváříme trojici)
>>> t[1]            (fungují běžné operace se seznamy)
2

>>> t[0] = 0
<chyba>            (n-tice nelze měnit)

>>> list(t)
[1, 2, 3]          (převedení n-tice na seznam)

>>> tuple([4, 5])
(4, 5)            (nebo seznamu na n-tici)

>>> s = 11, 22, 33
>>> s              (více hodnot oddělených čárkou utvoří n-tici)
(11, 22, 33)

>>> a, b, c = s
>>> b              (přiřazení každého prvku n-tice/seznamu zvlášť)
22
```

Příklad: Funkce s více výsledky

```
def deleni(x, y):  
    return x//y, x%y
```

```
podil, zbytek = deleni(1000, 7)
```

Příklad: Zipování seznamů

```
>>> x = [1, 2, 3]
>>> y = ["a", "b", "c"]
>>> list(zip(x, y))
[(1, 'a'), (2, 'b'), (3, 'c')]

>>> for i, j in zip(x, y):
>>>     print(i, j)
1 a
2 b
3 c
```

List comprehension

```
>>> seznam=[11, 12, 13, 14, 15]
>>> [a**2 for a in seznam]
[121, 144, 169, 196, 225]

>>> [a**2 for a in seznam if a%2 == 1]
[121, 169, 225]

>>> [(a,b) for a in [1,2,3] for b in [5,6]]
[(1,5), (1,6), (2,5), (2,6), (3,5), (3,6)]

>>> [(a,b) for a in range(4) for b in range(4)
if a<b]
[(0,1), (0,2), (0,3), (1,2), (1,3), (2,3)]

>>> věta="Kobyla má malý bok"
>>> [len(slovo) for slovo in věta.split()]
[6, 2, 4, 3]
```

List comprehension: Další triky

```
>>> [int(s) for s in input().split()]
>>> 1 2 3 4 5    (načtení seznamu čísel na jednom řádku)
[1, 2, 3, 4, 5]

>>> s=[[1, 2, 3], [4, 5]]
>>> [x for t in s for x in t]
[1, 2, 3, 4, 5]    („zploštění“ seznamu)

>>> m=[[0]*5 for _ in range(3)]
>>> m    (výroba matice: seznam 3 různých seznamů nul)
[[0,0,0,0,0], [0,0,0,0,0], [0,0,0,0,0]]

>>> n=[[1, 2, 3], [4, 5, 6]]
>>> [[r[i] for r in n] for i in range(3)]
[[1,4], [2,5], [3,6]]    (transpozice matice)
```

Agregační funkce

```
>>> s = [2**x % 37 for x in range(10)]
>>> s
[1, 2, 4, 8, 16, 32, 27, 17, 34, 31]
>>> min(s)
1
>>> max(s)
34
>>> sum(s)
172
>>> all([x < 37 for x in s])
True    (and přes všechny prvky seznamu)
>>> any([x > 33 for x in s])
True    (or přes všechny prvky seznamu)
>>> sorted(s)
[1, 2, 4, 8, 16, 17, 27, 31, 32, 34]
```