

Programování 1: Seznamy, řezy a řetězce

Martin Mareš

mj@ucw.cz

Katedra Aplikované Matematiky
MFF UK Praha

2019

Řez seznamem

```
>>> x = [11, 22, 33, 44, 55, 66, 77]
```

```
>>> x[2:5]  
[33, 44, 55]    (podseznam)
```

```
>>> x[:3]  
[11, 22, 33]   (prefix)
```

```
>>> x[5:]  
[66, 77]       (suffix)
```

```
>>> x[:]  
[11, 22, 33, 44, 55, 66, 77]   (kopie seznamu)
```

```
>>> x[::2]  
[11, 33, 55, 77]   (každý druhý prvek)
```

```
>>> x[::-1]  
[77, 66, 55, 44, 33, 22, 11]   (pozpátku)
```

Operace se seznamy

```
>>> 3 in [1, 2, 3, 4, 5]
True    (projde seznam a zjistí, zda v něm je daný prvek)

>>> x = [1, 2, 3, 4, 5]
>>> x.pop()    (odebere z konce)
5

>>> x.pop(0)   (odebere na zadané pozici)
1

>>> x    (co zbylo)
[2, 3, 4]

>>> x.insert(1, 42)    (vloží na zadanou pozici)
>>> x
[2, 42, 3, 4]

>>> help(x)    (vypíše dostupné operace)
>>> help([])   (totéž)
```

Použití řezů

```
>>> x = [11, 22, 33, 44, 55, 66, 77]
>>> x[:3] + x[4:]
[11, 22, 33, 55, 66, 77]    (vypuštění prvku)
>>> x[:3] + [0] + x[3:]    (vložení prvku)
[11, 22, 33, 0, 44, 55, 66, 77]
>>> x[2:4] = [1, 2, 3, 4]
>>> x
[11, 22, 1, 2, 3, 4, 55, 66, 77]
(přiřazení do řezu)
>>> y=[1,2,3]
>>> y[0:0] = [-1]
>>> y
[-1, 1, 2, 3]    (jiný způsob, jak vložit prvek)
```

Řetězce se chovají jako seznamy

```
>>> x="Sedmikráska"  
>>> len(x)  
11  
  
>>> x[0]  
'S'  
  
>>> x[:4] + x[10]  
'Sedma'  
  
>>> x[0]="s"  
<chyba> (řetězce nelze měnit)  
  
>>> for a in x:  
...     print(a)  
S  
e  
...
```

Operace s řetězci

```
>>> "velká".upper()
```

```
'VELKÁ'
```

```
>>> "banana".find("na")
```

```
2
```

```
>>> "banana".find("baba")
```

```
-1
```

```
>>> "Na počátku bylo slovo.".split()
```

```
['Na', 'počátku', 'bylo', 'slovo.']
```

```
>>> "+".join(["Alice", "Bob", "Cyril"])
```

```
'Alice+Bob+Cyril'
```

```
>>> "1+2+3".split(sep="+")
```

```
['1', '2', '3']
```

```
>>> help("")
```

Lexikografické porovnávání

```
>>> [1, 2, 3] < [1, 2, 5]
```

```
True
```

```
>>> [1, 2, 3] < [1, 2, 3, 4]
```

```
True
```

```
>>> [1, 2, 3] == [1, 2, 3]
```

```
True
```

```
>>> "červ" > "čert"
```

```
True
```

```
>>> "červ" > "datel"
```

```
True (pozor, nerespektuje česká pravidla)
```