

České vysoké učení technické v Praze FIT

# Programování v Pythonu

Jiří Znamenáček

*Příprava studijního programu Informatika je podporována projektem financovaným z Evropského sociálního fondu a rozpočtu hlavního města Prahy.*

*Praha & EU: Investujeme do vaší budoucnosti*



# Příkazová řádka

**1.** Napište program, který vypíše řádku po řádce soubor, cestu k němuž bude brát jako svůj vstupní argument. Přitom na začátek každé řádky přidá její pořadové číslo v souboru plus počet slov a počet znaků v příslušné řádce.

**[-] řešení ([cmd/01.py](#))**

```
import sys

if len(sys.argv) != 2:
    #print( "Usage: python3 %s SOUBOR " % sys.argv[0] )
    print( "Usage: python3 {} SOUBOR ".format( sys.argv[0] ) )
    sys.exit()
soubor = sys.argv[1]

with open(soubor, mode='r', encoding='utf-8') as f:
    for i, line in enumerate(f):
        slov = len( line.split() )
        znaku = len( line )
        print(i, slov, znaku, line, end='')
```

**2.** Upravte předchozí program tak, aby přebíral i druhý řetězcový argument. Program pak bude tisknout ze vstupního souboru pouze ty řádky, které budou jako podřetězec obsahovat hodnotu tohoto druhého argumentu.

**[-] řešení ([cmd/02.py](#))**

```
import sys

if len(sys.argv) != 3:
    #print( "Usage: python3 %s SOUBOR TEXT " % sys.argv[0] )
    print( "Usage: python3 {} SOUBOR TEXT ".format(
sys.argv[0] ) )
    sys.exit()
soubor = sys.argv[1]
text   = sys.argv[2]

with open(soubor, mode='r', encoding='utf-8') as f:
    for i, line in enumerate(f):
        if text in line:
            slov = len( line.split() )
            znaku = len( line )
            print(i, slov, znaku, line, end='')
```

**3.** Upravte předchozí program tak, že na výstupu zapíše všechny vyhovující řádky seřazené podle počtu slov do souboru, jehož jméno bude brát jako třetí vstupní parametr. Počet slov v příslušné řádce bude ve výstupu vytištěn před každou řádkou.

**[-] řešení ([cmd/03.py](#))**

```
import sys

if len(sys.argv) != 4:
    #print( "Usage: python3 %s SouborIN TEXT SouborOUT " %
    sys.argv[0] )
    print( "Usage: python3 {} SouborIN TEXT SouborOUT
    ".format( sys.argv[0] ) )
    sys.exit()
soubor_in  = sys.argv[1]
text       = sys.argv[2]
soubor_out = sys.argv[3]

lines = []
with open(soubor_in, mode='r', encoding='utf-8') as f:
    for i, line in enumerate(f):
        if text in line:
            slov  = len( line.split() )
            znaku = len( line )
            lines.append( (i, slov, znaku, line) )

with open(soubor_out, mode='w', encoding='utf-8') as f:
    for line in lines:
        f.write( str(line[1]) + '\n' )
        f.write( ' '.join( [str(i) for i in line] ) )
```

**4.** Napište program, který za pomoci `sys.argv` vyhodnotí své vstupní argumenty následujícím způsobem:

- samostatné přepínače `-x` uloží v podobě n-tice (`x, ...`)
- přepínače `--xarg=hodnota` uloží v podobě slovníku `{ 'xarg': hodnota, ... }`
- vlastní argumenty uloží do další n-tice

(Tj. zjednodušeně předpokládáme, že zkrácené přepínače nemají žádnou hodnotu a dlouhé přepínače vždy obsahují rovnítko, abychom od sebe všechny tři kategorie argumentů mohli snadno rozlišit.)

**[-] řešení ([cmd/arguments.py](#))**

```
import sys

argumenty = []
prepinace_kratke = []
prepinace_dlouhe = []

for arg in sys.argv[1:]:
    if arg.startswith('--'):
        k, v = arg[2:].split('=')
        prepinace_dlouhe.append( (k, v) )
    elif arg.startswith('-'):
        k = arg[1:]
        prepinace_kratke.append(k)
    else:
        argumenty.append(arg)

argumenty = tuple(argumenty)
prepinace_kratke = tuple(prepinace_kratke)
prepinace_dlouhe = dict(prepinace_dlouhe)

print(argumenty)
print(prepinace_kratke)
print(prepinace_dlouhe)
```