

# Práce se soubory - motivace

- Doted' jsme pracovali jen interaktivně
  - Vstup z klávesnice, výstup na obrazovku
- Co když chceme
  - Číst vstup ze souboru a/nebo
  - Zapisovat výstup do souboru?
- Python obsahuje mnoho vestavěných funkcí pro I/O (= input/output; vstup/výstup):
  - <https://docs.python.org/3.5/tutorial/inputoutput.html>
- Mějme soubor input.txt s tímto obsahem:

*I am a little testfile.*

*One day, I will be awesome.*

# Vstup (čtení) ze souboru po řádcích

```
with open("file", "mode") as variable:  
    # Read (or write to) the file  
    # using the file handle object
```

- Příklad

```
with open("input.txt", "r") as textfile:  
    for line in textfile:  
        print(line, end="")
```

- Výstup našeho skriptu

```
I am a little testfile.  
One day, I will be awesome.
```

# Vstup (čtení) ze souboru – znak \n

- Znak \n představuje konec řádku
- Mějme stejný vstupní soubor a skript

```
lst = []  
with open("input.txt", "r") as textfile:  
    for line in textfile:  
        lst.append(line)  
print(lst)
```

- Výstup

```
['I am a little testfile.\n', 'One day, I  
will be awesome.']
```

# Vstup (čtení) ze souboru – znak \n

- Ekvivalent: metoda `readlines()`
  - Vrací seznam řádků dokud nenarazí na EOF
- Pokud se chceme zbavit znaku \n při čtení souboru
  - Metoda `rstrip()`
  - Např. `lst.append(line.rstrip())`

# Výstup (zápis) do souboru

- Režim w (write): Předešlý obsah se přepíše novým!
- Režim a (append): Nový obsah se zapíše na konec souboru

```
with open("output.txt", "mode") as f:  
    f.write("Data to be written.")
```

- Funkce write(s) zapíše proměnnou typu řetězec do souboru (tak jak je, nepřidává znak nového řádku)

# Nutnost zavírat soubory

- Pokud nepoužíváte příkaz `with`, musíte zavírat soubory
- Při zápisu se data zapisují nejprve do vyrovnávací paměti, až potom do souboru na disku (tzv. buffering)
- Může se stát, že data zůstanou v bufferu, ale nestihnou se přenést do souboru
- Zavření souboru je dobrý způsob, jak vyprázdnit (spláchnout do souboru) obsah bufferu

