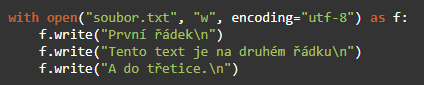
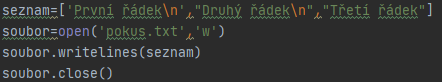
**16. PRG – Soubory**

Otevření a zavření souboru:

* otevření soboru se provádí pomocí funkce open()
* soubor se může otevřít dvěma způsoby:
  + pomocí bloku **with**

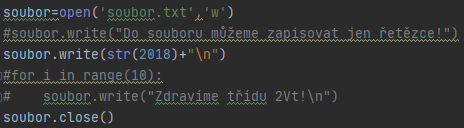


* + - při použití with není potřeba soubor manuálně zavírat
    - určitou drobnou nevýhodou je, že celý blok, kde se souborem pracujeme, musí být odsazen (akorát prostě se ti to třeba nelíbí na funkčnost to nemá vliv xd)
  + pouze pomocí funkce **open()**
    - f
    - v takovém případě je potřeba soubor po použití manuálně zavřít pomocí funkce **close()**, jinak hrozí překročení limitu na počet otevřených souborů a jiné další nepříjemnosti, nedá se smazat, data nejsou uložena na disku ale pouze v RAM
    - **flush()** – vynucené uložení souboru, systém ho v případě with provádí automaticky, ale je možné jej vyvolat ručně

Módy pro otevírání souborů:

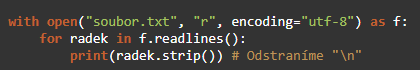
* v pythonu můžeme soubor otevřít ve třech základních módech (tyto se týkají textových souborů, existují ještě binární soubory ale to jsme neprobírali):
  + w (write) – otevře pro zápis, neexistující soubor bude  
     vytvořen, existující soubor bude přepsán
  + r+ - otevře existující soubor jak pro čtení, tak pro zápis
  + r (read) – otevře daný soubor pro čtení, soubor musí  
     existovat
  + a (append) – pro zápis na konec souboru, neexistující   
     bude vytvořen, pokud soubor už existuje tak se nepřepíše při uložení

Zapisování do souboru:

* provádí se pomocí módů write, r+, append
* Funkce pro zápis:
  + *write()* – zapíše do souboru daný řetězec na aktuální pozici na jeden řádek, pokud samotný řetězec neobsahuje \n
    - 
    - může zapisovat pouze řetězce, čísla atd. se musí převést na řetězec
    - můžeme zapsat pouze jeden řetězec najednou, na rozdíl od *print()*
  + *writelines()* – zapisuje seznamy, viz obrázek u open() výše
    - každý prvek seznamu se zapíše na samostatný řádek
* soubor se v těchto všech příkladech uloží do stejného adresáře jako program

Čtení ze souboru:

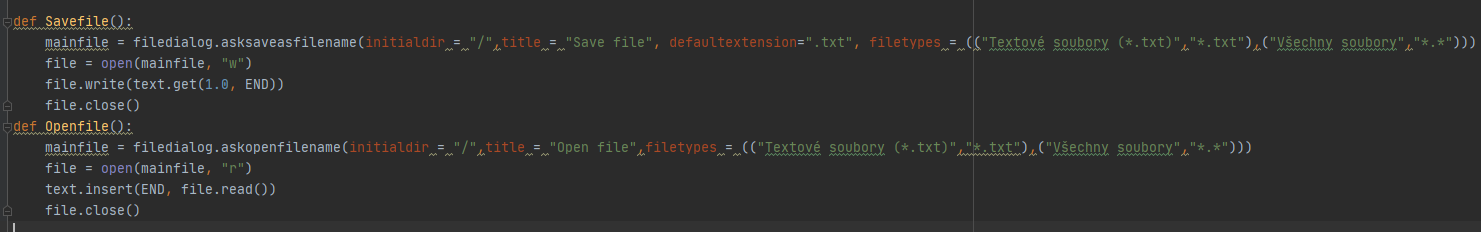
* čtení se provádí pomocí módu read, r+
* při čtení nemůžeme zapisovat
* Funkce pro čtení:
  + *read()* – přečte řetězec o zadané délce, když neuvedeme nic tak přečte celý soubor, vrací řetězec, pokud provádíme několikrát *read()* za sebou a u každého uvedeme počet znaků tak vždy přečte ten uvedený počet znaků od aktuální pozice
  + *readline()* – přečte řetězec od aktuální pozice do konce řádku
  + *readlines()* – vrátí seznam řádků souboru od aktuální pozice, každý řádek je jedna položka seznamu, přečte celý soubor
* pokud jsme již na konci souboru a zkusíme něco přečíst, tak se vrátí prázdný řetězec, podle toho poznáme že jsme na konci souboru
* při čtení je občas problémem \n, které je na konci každého řádku, to se dá odstranit pomocí funkce strip()
  + v tomto případě se do proměnné **radek** v cyklu uloží jeden řádek souboru a odstraní se z něj \n



Pohyb v souboru:

* *tell(x)* – vrací aktuální pozici v souboru
* *seek(pocet)* –posune ukazatel o daný počet bytů od začátku souboru
* Pozice je počet znaků od začátku souboru

Souborové dialogy:

* tato komponenta umožňuje otevírání a vyhledávání souborů, tak jak to funguje všude normálně, výsledkem tohoto dialogu je cesta k souboru ve formě řetězce, to se dosadí do standardního otvírání/zavírání souborů v pythonu
* musí se, podobně jako messagebox samostatně importovat jako *from tkinter import filedialog*
* dvě základní funkciality jsou *asksaveasfilename()* (uložení souboru)a *askopenasfilename()* (otevření souboru)
* *asksaveasfilename()* – stará se o ukládání, lze v něm nastavit základní příponu ukládaného souboru, titulek a nebo třeba výchozí directory
* *askopenasfilename()* – stará se o otevření souboru, znovu se v něm dá nastavit výchozí přípony, které soubory chceme otevírat