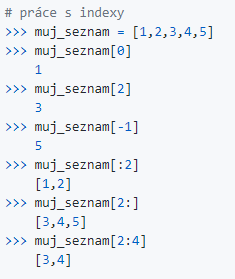
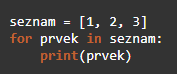
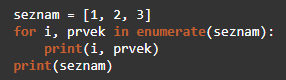
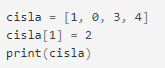
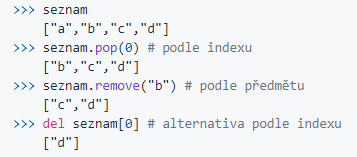
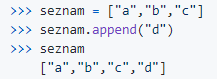
**12. PRG – Seznamy, n-tice, slovníky**

Seznamy:

* asi nejvyužívanější z těchto tří
* poskytuje způsob, jak ukládat větší množství informací stejného druhu přehledně a na jedno místo
* velice podobné vlastnosti jako n-tice, rozdíl je v tom že seznam lze za běhu programu upravit
* značí se [] a funkce pro jeho tvorbu se jmenuje *list(),* prvky se oddělují čárkou
* Operace se seznamy:
  + Vybírání ze seznamů – jedná se o nejzákladnější operaci
    - seznamy se dají také „ořezávat“ – (muj\_seznam[2:4])
  + Procházení seznamů – existují tři způsoby:
    - for … in
    - funkce *enumerate()* – ta vrací jak hodnotu prvku, tak i jeho index

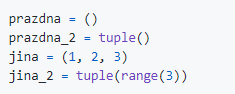


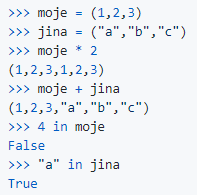
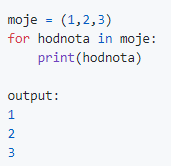


* + Měnění prvků – v seznamu se dají jednotlivé prvky měnit pomocí indexu
  + Mazání prvků – používají se tři metody:
    - *pop()* – podle indexu
    - *remove()* – podle prvku, odebere pouze první výskyt daného prvku
    - *del* – podle indexu, alternativa k pop
    - metoda *clear()* – vyčistí celý seznam
  + Aritmetrické operace – používá se sčítání – spojují se seznamy; násobení – seznam se dá násobit (seznam se třeba ztrojnásobí)
  + Přidávání prvků – metoda append – přidá prvek na konec seznamu
  + Další funkce a operace se seznamy:
    - *len()* – vrátí počet prvků seznamu
    - *min(), max()* – vrátí prvek s nejmenší/největší hodnotou
    - *insert(position, prvek)* – vloží prvek na dané místo v seznamu
    - *count()* – vrátí počet výskytů daného prvku v seznamu
    - *sort()* – seřadí seznam podle velikosti vzestupně; variací je sorted() – seřadí seznam jednou (jen pro print), ale samotný seznam zůstane nezměněn
    - *reverse()* – obrátí seznam, první bude poslední prvek a naopak

N-tice:

* typově velmi podobné seznamům, mají stejnou i práci s indexy
* hlavním rozdílem je nemožnost n-tici upravit jakmile je vytvořena
* značí se () a funkce pro jejich tvorbu je *tuple(),* tuple umí převést jiné datové typy na n-tice



* Operace s n-ticemi:
  + n-tice lze spojovat (+), replikovat (\*) a testovat, jestli se daná hodnota nachází uvnitř
  + n-tici lze procházet cyklem for, stejně jako seznam

Slovníky:

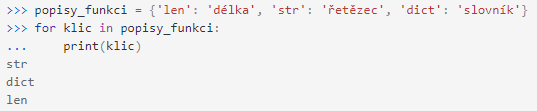
* ve slovnících se nachází dva druhy prvků – klíč (key) a hodnota (value), každému klíči je přiřazena nějaká hodnota
* Slovníky se dají také využít, když máme různé informace, ale chceme s nimi pracovat jako s jedním celkem (v některých jazycích, jako je například javascript se proto slovníkům říká objekt)
* značí se {} a jeho funkcí pro tvorbu je *dict()*
* Operace se slovníky:
  + 
  + získání hodnoty – hodnota se získává pomocí klíče, pokud klíč neexistuje, python vyhodí error
  + změna hodnoty

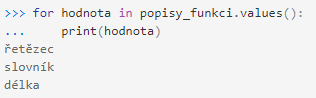
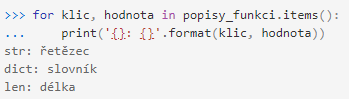


* + přidání hodnoty
  + odebrání hodnoty – odebírá se podobně jako u seznamů příkazem *del*, pomocí klíče



* slovníky se dají procházet pomocí cyklu for, stejně jako seznamy nebo n-tice; pouhým for cyklem ale získáme pouze klíče



* pro získání hodnot je potřeba přidat metodu *values()*
* pokud chceme jak klíče, tak i hodnoty, použijeme metodu *items()*