

Domáci projekty 8

Na začátek trochu práce se seznamy. Některé z projektů – a zvláště ten poslední – potřebují trochu přemýšlení; nebudeš-li vědět jak dál, zeptej se na internetu nebo na srazu ostatních a řešte společně!

0. Udělej si seznam domácích zvířat. Budeš ho potřebovat v dalších úlohách.
Domácí zvířata známe tato: "pes", "kočka", "králík", "had".
1. Napiš funkci, která vrátí jména domácích zvířat (zadaných argumentem), která jsou kratší než 5 písmen.
2. Napiš funkci, která vrátí jména domácích zvířat (zadaných argumentem), která začínají na k.
3. Napiš funkci, která dostane slovo a zjistí, jestli je v seznamu domácích zvířat.
„Zjistí“ znamená, že funkce vrátí *True* nebo *False*.
4. Napiš funkci, která dostane dva seznamy jmen zvířat a vrátí tři seznamy:
 - (a) zvířata, která jsou v obou seznamech,
 - (b) zvířata, která jsou v jen prvním seznamu,
 - (c) zvířata, která jsou v jen druhém seznamu.Napiš (a pusť) k této funkci testy, aby sis ověřila, že funguje správně.
5. Napiš program, který seřadí seznam domácích zvířat podle abecedy.
6. Had byl pyšný na to, že je v abecedě první. Dokud nepřiletěla "andulka".
Abys hada uklidnila, vytvoř funkci, která zvířata seřadí podle abecedy, ale bude ignorovat první písmeno (t.j. vrátí ["had", "pes", "andulka", "kočka", "králík]).

Postup:

- Máš seznam hodnot, které chceš seřadit podle nějakého klíče. Klíč se dá z každé hodnoty vypočítat.
- Vytvoř seznam dvojic (klíč, hodnota).
- Seřaď tento seznam dvojic – dvojice se řadí nejdříve podle prvního prvku, pak druhého atd.
- Nakonec vytvoř ze seznamu dvojic opět jen seznam hodnot.

Proč má zrovna had takovéhle výsadní postavení, zjistiš později.

Jedna klasická programovací úloha, která nejspíš pořádně potrápí tvé logické myšlení. Je nepovinná, nemáš-li na ni aspoň pár hodin, tak ji přeskoč.

7. Napiš funkci, která převede římské číslice na číslo (*int*).
Tip: Nejdříve napiš k této funkci nějaké testy, aby sis ověřila, že (a co) (ne)funguje správně.

Dadaistický koutek: procvičení práce se seznamy, řetězci a soubory. Opět: nebudeš-li vědět jak dál, zeptej se!

8. Vyber si básničku, která má aspoň tři sloky po aspoň třech verších. Ulož ji do souboru *basnicka.txt*,
9. Napiš program, který vypíše básničku ze souboru *basnicka.txt*, ale obrátí pořadí veršů (t.j. jako první vypíše poslední řádek atd.).
Nápověda: Každý seznam má metodu *reverse*, která ho „obráť“.
10. Napiš program, který obrátí pořadí slov v jednotlivých verších.
11. Obrát pořadí slov (ty by měly být oddělené jedním prázdným řádkem).
12. Vypiš slova básně v náhodném pořadí.
Bonusový projekt: Snaž se přitom co nejlépe zachovat strukturu básně (sloky, verše, interpunkci, velká písmena, ...).

-
- A 10x10 grid of dots. The dots are arranged in 10 rows and 10 columns. The following positions contain an 'X': (1,1), (1,2), (2,3), (3,4), and (9,9). All other positions contain a single dot.

```
souradnice = [(0, 0)]
polyb(souradnice, 'v')
print(souradnice)           # → [(0, 0), (0, 1)]
polyb(souradnice, 'v')
print(souradnice)           # → [(0, 0), (0, 1), (0, 2)]
polyb(souradnice, 'j')
print(souradnice)           # → [(0, 0), (0, 1), (0, 2), (1, 2)]
polyb(souradnice, 's')
print(souradnice)           # → [(0, 0), (0, 1), (0, 2), (1, 2), (1, 1)]
```

```
seznam[5][5][5][5][5][5][5][5][5][5][5][5][5][5][5][5][5][5][0] == 5.
```