

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 sraze ostatních, a řešte společně!

0. Udělej si seznam domácích zvířat. Budeš ho potřebovat v dalších úlohách.
Domácí zvířata známe tyto: "pes", "kočka", "králík", "had".
1. Napiš funkci, která vypíše jména domácích zvířat, která jsou kratší než 5 písmen.
2. Napiš funkci, která vypíše jména domácích zvířat, 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
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, zjistíš 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).

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. Obráť 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, ...)

Projekty 13-19 závisí jeden na druhém, řeš je postupně. Až to uděláš, můžeš si zahrát hru! Tahle sekce není jednoduchá (a poslední dva projekty jsou obzvláště náročné). Kdyžtak můžeš zkusit spojit síly s ostatními účastnicemi kurzu :)

13. Napiš funkci, která dostane seznam souřadnic (párů čísel menších než 10), a vypíše je jako mapu. Například:

```
nakresli_mapu([(0, 0), (1, 0), (2, 2), (4, 3), (8, 9)])
X X . . . . .
. . . . .
. . X . . . . .
. . . . X . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . . X .
```

Jak na to?

- (a) Udělej tabulku (seznam seznamů) se samými tečkami, něco jako:
[['.', '.', '.', '.', '.', '.', '.', '.', '.', '.',],
['.', '.', '.', '.', '.', '.', '.', '.', '.', '.',],
['.', '.', '.', '.', '.', '.', '.', '.', '.', '.',]].
- (b) Na příslušných místech nahraď tečky X-ky.
- (c) Tabulku vypiš pomocí dvou cyklů for zanořených do sebe.

14. Napiš funkci pohyb, která dostane seznam souřadnic a světovou stranu ("s", "j", "v" nebo "z"), a přidá k seznamu poslední bod „posunutý“ v daném směru. Např.:

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

Funkce by neměla nic vracet.

15. Napiš cyklus, který se bude ptát uživatele na světovou stranu, podle ní zavolá pohyb, a následně vykreslí seznam jako mapu. Pak se opět sezeptá na stranu, atd.
Začinej se seznamem [(0, 0), (1, 0), (2, 0)]
16. Doplň funkci pohyb tak, aby při pohybu umazala první bod ze seznamu souřadnic. Výsledný seznam tak bude mít stejnou délku, jako před voláním.
17. Doplň funkci pohyb tak, aby zamezila
- pohybu ven z mapy
 - pohybu na políčko, které už v seznamu je
18. Přidej do hry hadí potravu. Tady jsou pravidla pro vegetariánského hada, ale můžeš si je změnit podle chuti:
Seznam ovoce obsahuje na začátku jedno ovoce, na políčku na kterém není had. (například: [(2, 3)] znamená jedno ovoce na pozici (2, 3).) Když had sežere ovoce, vyrostе („nesmaže“ se mu ocas, tedy neprovede se to, cos přidala v projektu 16), a pokud na mapě zrovna není další ovoce, na náhodném místě (kde není had) vyrostе ovoce nové.
Každých 30 tahů vyrostе nové ovoce samo od sebe.
Na mapě se toto tajemné ovoce zobrazuje jako otazník (?).
19. Hadí hřiště může mít libovolné rozměry větší než 4×1. Třeba 20×20 nebo 10×30.

A nakonec projekt na přemýšlení.

20. Může seznam obsahovat sám sebe? Zkus co nejjednodušeji udělat takový seznam, aby platilo:
seznam[5][5][5][5][5][5][5][5][5][5][5][5][5][5][5][5][5][5][0] == 5.