

PYTHON

5. forduló

Starschema
Part of **HCLTech**

A kategória támogatója: Starschema

Ismertető a feladatlaphoz

Kezdj neki minél hamarabb, mert a feladatot a forduló záró időpontjáig lehet beküldeni, nem addig lehet elkezdni!

Sok sikert!



Ez a hét egy kis lazítást kínál: Az első, megszokott feleletválasztós kérdésünk most némi - tanulságos - kutatómunkát igényel. Ezt követően egy izgalmas programozási feladattal segíthetsz Flóriánnak a tervezésben. A feladatok sorát egy kis gondolkodtató kérdés zárja.

Jó munkát!

1. feladat 4 pont

Kakukktojás

Flórián sörkóstolón látja vendégül barátait, s poharazás közben egyiküktől (aki tapasztalt programozó) tájékozik a Python képességeiről, azt tudakolva, egyes feladatokra milyen Python-csomagok valók. A barát, kicsit kábán a sok megízlelt újdonságtól, a válaszában véletlenül hibákat is ejt.

Az alábbi felsorolásban csoportonként egy-egy kakukktojást találsz.

Melyek a kakukktojások? (A választ a felsorolás sorrendjében, csupa kisbetűvel, vesszővel elválasztva, szóköz nélkül kérjük, pl: matplotlib,streamlit,tensorflow,nltk,flask

1. Matplotlib, ggplot, Seaborn, Bokeh
2. Streamlit, Dash, Numpy, Voilà
3. Tensorflow, Requests, Pytorch, MXNet
4. NLTK, SpaCy, Gensim, OpenCV
5. Flask, SciPy, Django, Pyramid

Válaszok

2. feladat 5 pont

Bevásárlólista

Flóriának fogytán van az otthoni sörkészlete, ezért írt egy Python függvényt, amivel bevásárlólistákat szeretne létrehozni és bővíteni. Úgy tervezte, hogy hétközben az éppen elfogyott söröket felírja a listára, majd minden vasárnap elmegy bevásárolni. Ezután egy új, kezdetben üres bevásárlólistát hoz létre. Az alábbi függvényt írta:

```
def append_to(element, to=[]):  
    to.append(element)  
    return to
```

Az első héten Soproniból és Borsodiból volt hiány, ezért felírta ezeket a lista_1 nevű bevásárlólistára:

```
lista_1 = append_to("soproni")  
lista_1 = append_to("borsodi", lista_1)
```

Vasárnap megvette az említett söröket. Következő héten Heinekenből és Staropramenből fogyott ki, ezért ezeket

```
lista_2 = append_to("heineken")  
lista_2 = append_to("staropramen", lista_2)
```

írta fel az új listájára (lista_2):

Végül lista_3 néven szeretne volna elmenteni a lista_1 tartalmát, kiegészítve az Arany Ászokkal:

```
lista_3 = append_to("arany ászok", lista_1)
```

Vajon mi lett az eredménye Flórián listáinak? Válaszd ki a helyes válaszokat!

Válaszok

☐ lista_1 tartalma: ['soproni', 'borsodi']

- ☐ lista_1 tartalma: ['soproni', 'borsodi', 'heinekein', 'staropramen']
- ☐ lista_1 tartalma: ['soproni', 'borsodi', 'heinekein', 'staropramen', 'arany ászok']
- ☐ lista_2 tartalma: ['soproni', 'borsodi']
- ☐ lista_2 tartalma: ['heinekein', 'staropramen']
- ☐ lista_2 tartalma: ['soproni', 'borsodi', 'heinekein', 'staropramen']
- ☐ lista_2 tartalma: ['soproni', 'borsodi', 'heinekein', 'staropramen', 'arany ászok']
- ☐ lista_3 tartalma: ['arany ászok']
- ☐ lista_3 tartalma: ['soproni', 'borsodi', 'arany ászok']
- ☐ lista_3 tartalma: ['heinekein', 'staropramen', 'arany ászok']
- ☐ lista_3 tartalma: ['soproni', 'borsodi', 'heinekein', 'staropramen', 'arany ászok']

3. feladat 5 pont

Pandas index

Adott egy pandas DataFrame. Melyik indexeléssel lehet kiválasztani a két, hazai márka nevét úgy, hogy a végeredmény DataFrame maradjon?

| | A | B | C |
|---|--------|------------|-----|
| 1 | import | Heineken | 300 |
| 5 | import | Hoegaarden | 400 |
| 6 | hazai | Soproni | 300 |
| 4 | hazai | Borsodi | 400 |
| 2 | import | Budweiser | 500 |

Válaszok

- ☐ df.loc[6:4,'B':'B']
- ☐ df.loc[2:4,'B':'C']
- ☐ df.loc[6:4,'B':'C']
- ☐ df.loc[df['A'] == 'hazai', 'B':'B']
- ☐ df.loc[df['A'] == 'hazai', 1:2]
- ☐ df.iloc[6:4,'B':'B']
- ☐ df.iloc[2:4,1:2]
- ☐ df.iloc[2:3,1:2]

☐ `df.iloc[df['A'] == 'hazai', 1:2]`

Megoldások beküldése