Tartalomjegyzék

- 1 1. gyakorlat - Ismétlés
  - 1.1 CloudCoder használata
  - 1.2 Feladatok megoldása
    - 1.2.1 nagybetu
    - 1.2.2 udvarias
    - 1.2.3 leltar_frissites
    - 1.2.4 leltar_sorszam
    - 1.2.5 leltar_egyben
  - 1.3 Házi feladatok megoldása
    - 1.3.1 szorzo
    - 1.3.2 atment

1. gyakorlat - Ismétlés

Ezen a gyakorlaton az el?z? féléves Informatika 1 anyagát ismételjük, az ott tanultakat kell használni. Amire biztosan szükség lesz ma:

- Python alapjai, futtatása; karakterláncok; elágazások, ciklusok; függvények definiálása: Informatika1-2014/eloadas3
- Listák, listaértelmezések, szótárak: Informatika1-2014/eloadas4

Ha valaki nem hallgatta ezt a tárgyat, és nincs tisztában a python alapjaival, nyugodtan kérdezzen laboron, és igyekezzen egy-két héten belül behozni a lemaradást.

CloudCoder használata

A legtöbb python feladathoz gyakorlaton egy CloudCodernek nevezett rendszert fogunk használni. Ennek elõnye, hogy helyben ki is javítja a feladatot. Elérés:

- https://ccweb.math.bme.hu/cloudcoder/

Részletes leírás a tárgylapon.

Feladatok megoldása

nagybetu

```
def nagybetu(s):
    return str.isupper(s[0])
```

udvarias

```
def udvarias(l):
    return ["Ms. " + name for name in l]
```

leltar_frissites

Megoldás listaértelmezéssel:

```
def leltar_frissites(leltar, ujszam):
    return [ujszam[elem] if elem in ujszam else elem for elem in leltar]
```

Megoldás listaértelmezés nélkül:

```
def leltar_frissites(leltar, ujszam):
    uj_leltar = []
    for elem in leltar:
        if elem in ujszam:
            uj_leltar.append(ujszam[elem])
        else:
            uj_leltar.append(elem)
    return uj_leltar
```

leltar_sorszam

```
def leltar_sorszam(szamok):
    ujszam = {}
    i = 100
    for szam in szamok:
        ujszam[szam] = i
        i = i+1
    return ujszam
```

leltar_egyben

```
def leltar_frissites(leltar, ujszam):
    return [ujszam[elem] if elem in ujszam else elem for elem in leltar]

def leltar_sorszam(szamok):
    ujszam = {}
    i = 100
    for szam in szamok:
        ujszam[szam] = i
        i = i+1
    return ujszam

def leltar_egyben(leltar):
    szamok = []
    for elem in leltar:
        if elem not in szamok:
            szamok.append(elem)

    szamok.sort()
```

```
    ujszam = leltar_sorszam(szamok)

    return leltar_frissites(leltar, ujszam)
```

Házi feladatok megoldása

szorzo

```
def szorzo(l, k):
   return [i * k for i in l]
```

atment

Megoldás listaértelmezéssel:

```
def atment(pontok, hatar):
    nevek = [nev for nev in pontok if pontok[nev] >= hatar ]
    nevek.sort()
    return nevek
```

Megoldás listaértelmezés nélkül:

```
def atment(pontok, hatar):
    nevek = []
    for nev in pontok:
        if pontok[nev] >= hatar:
            nevek.append(nev)
    nevek.sort()
    return nevek
```