

# Szkriptnyelvek - Python minta ZH

## 1. feladat: Disarium (12 pont)

Készíts egy `is_disarium` nevű függvényt, amely egy egész számot kap paraméterül. A függvény adja vissza, hogy a paraméterül kapott szám disarium szám-e.

Egy számra akkor mondjuk, hogy disarium szám, ha minden adott helyen lévő számjegy adott helyedik hatványát összeadva megkapjuk magát a számot. A 175-re ez a tulajdonság igaz, hiszen:

$$175 \Rightarrow 1_1 + 7_2 + 5_3 = 1 + 49 + 125 = 175$$

De például a 42-re nem, hiszen:

$$42 \Rightarrow 4_1 + 2_2 = 8$$

## Példa

```
Input: 175  
Return: True
```

```
Input: 42  
Return: False
```

## 2. feladat: Betűkombinációk (14 pont)

A régi, nyomógombos telefonok billentyűzete a képen látható módon néz ki. A számjegyeket tartalmazó billentyűkkel (a 0 és 1 billentyűk kivételével) különböző betűket lehet leírni (pl. a 3-as billentyű segítségével a `d`, `e` és `f` betűket írhatjuk le).



Írj egy `letter_combinations` nevű függvényt, amely egy olyan szöveget kap paraméterül, amely a telefonon megnyomott számjegyeket tartalmazza a megnyomás sorrendjében. A függvény adja vissza egy listában az adott gombok adott sorrendben történő lenyomásával kapható összes betűkombinációt. Feltételezhetjük, hogy az input mindig megfelelő lesz (tehát nem lesznek benne oda nem illő karakterek (0, 1, \*, #)).

## Példa

```
Input: "532"  
Return:  
[  
  "jda", "jdb", "jdc", "jea", "jeb", "jec",  
  "jfa", "jfb", "jfc", "kda", "kdb", "kdc",  
  "kea", "keb", "kec", "kfa", "kfb", "kfc",  
  "lda", "ldb", "ldc", "lea", "leb", "lec",  
  "lfa", "lfb", "lfc"  
]
```

## 3. feladat: Savanyúság (54 pont)

- Készíts egy `Savanyusag` nevű osztályt, amely egy savanyúságot reprezentál. Az osztály

a következő adattagokkal rendelkeznek: `minoseget_megorzi`, `nyitva`, `elemek` és `_tipus`.

- A konstruktorban inicializáljuk az adattagokat. A konstruktor paraméterei rendre a következők:
  - a savanyúság szavatossági dátuma - ez egy `(év, hónap, nap)` formátumú tuple lesz, ezzel inicializáljuk a `minoseget_megorzi` adattagot
  - egy logikai érték, ami azt mutatja meg, hogy nyitva van-e a savanyúság teteje, ezzel inicializáljuk a `nyitva` adattagot
  - ezután tetszőlegesen sok olyan alkotóelem következik, amelyet sorban belerakunk az `elemek` listába (feltehetjük, hogy mindig van legalább egy elem).

A `_tipus` értékét a konstruktor a következőképpen állítsa be: ha a savanyúságba csak egyféle alkotóelemet teszünk (pl. csak uborkát), akkor az értéke legyen az alkotóelem neve. Egyéb esetben, több különböző hozzávaló esetén legyen az adattag értéke a `csalamade` szöveg.

- Írj `get` és `set` property-t a `_tipus` adattaghoz, `tipus` néven. A setterben az adattag értékének beállítása előtt ellenőrizd, hogy a beállítani kívánt érték szerepel-e már az `elemek` listában. Ha nem szerepel, a setter ne módosítsa az adattag értékét.
- Írj egy `szavatos` metódust, amely három paramétert vár: rendre év, hónap, nap. A metódus adja vissza, hogy a paraméterben kapott napon szavatos lesz-e a savanyúság (`minoseget_megorzi` adattag).
- Írj egy paraméter nélküli `fedel_csavar` metódust, amely a `nyitva` logikai adattag értékét állítsa be az ellenkezőjére (ha nyitva volt a savanyúság fedele, zárja le, ha pedig zárva volt, akkor nyissa ki).
- Definiáld felül az osztályban a `+=` operátor működését (`__iadd__`). A metódus egy másik savanyúságot vár paraméterül (ezt ellenőrizd), és megpróbálja az aktuális savanyúságba áttölteni azt, amennyiben mindkét savanyúság fedele nyitva van.
  - Ha az aktuális savanyúság fedele zárva van, dobj egy Exception típusú kivételt, amit az `A savanyusag fedele zarva van!` szöveggel inicializálj.
  - Ha a paraméterben érkező savanyúság fedele zárva van, dobj egy Exception típusú kivételt, amit az `A másik savanyusag fedele zarva van!` szöveggel inicializálj.
  - Ha mindkét savanyúság fedele nyitva van, fűzzük hozzá az aktuális savanyúság alkotóelemeihez a másik savanyúság alkotóelemeit. Az így kapott savanyúságunk szavatossági ideje a két savanyúság szavatossági idejének a minimuma lesz. A típust a megszokott módon állítsuk be: ha csak azonos elemeket tartalmaz a savanyúság, akkor az elem neve, ellenkező esetben pedig a `csalamade` szöveg

legyen a típus neve.

- Definiáld felül a szöveggé alakítást megvalósító metódust, amely a következő szöveggel térjen vissza: `Savanyított {típus}, aminek a fedele {nyitva/zarva}`. (a `nyitva` vagy `zarva` szövegek valamelyikét értelemszerűen a megfelelő adattag értékének függvényében választjuk).
- Valósítsd meg a `*=` operátort ( `__imul__` ), amely egy számot kap paraméterül. Az operátor többszörözze az aktuális savanyúság minden alkotóelemét annyiszor, mint amennyi a kapott szám, és ennek megfelelően változtassa az `elemek` listát.
- Definiáld felül az egyenlőséget vizsgáló operátort. Az operátor pontosan akkor adjon vissza igazat, ha az összehasonlítani kívánt két savanyúság összes adattagja rendre megegyezik. Az elemek listát sorrendtől függetlenül vizsgáljuk!

Jó munkát!