Elmélet

1. *Függvények*
   1. **ord() függvény:** visszaadja a karakter UTF-8-as értéket
   2. **seek() függvény:** a beolvasott fájlban a megadott indextől folytatja a beolvasást
   3. **chr() függvény:** visszaadja az UTF8-as értékből a karaktert
   4. **replace(mit,mire):** felülírja az elemet(listában nem lehet)
   5. **.close():** –bezárja a fájlt a program lefutása után
   6. **.find(mit,tól,ig):** visszaadja hogy hányadik tag a megadott elem a listában vagy sztringben, előlről kezdi el
   7. **.join(„x”):** sztringet készít egy listából vagy tuple-ből, amit x-el köt össze

pl.: almaxbanánxatimeleg

* 1. **sorted() függvény:** abc sorrendbe írja ki minden megadott karaktert listát ad vissza

pl: sorted(„hi mom”) =[’ ’,’h’,’i’,’m’,’m’,’o’]

* 1. **open(fájl,”r”):** kapcsolatot teremt a fájl és a futó program közt, lehet vele írni vagy olvasni, vagy mindkettő,de nyitva hagyja a programot
     + 1. **Műveletei :** **’r’** – csak olvasás, kiakad ha nincs fájl

**’w’** – csak írás, csinál egy új fájlt ha nincs a megadott

**’a’** – csak írás, hozzátűzi a végéhez a cuccokat, új fájlt csinál ha nincs a megadott

**’r+’** – írás és olvasás, kiakad ha nincs fájl

**’w+**’- írás és olvasás, új fájlt csinál ha még nincs olyan

**’a+’** –írás és olvasás, a végére ír, ha kell új fájlt csinál

1. *Sztringek:* 
   1. egy type hint, nem változékony ,nem lehet az értékét változtatni, csak cserélni
      1. **Műveletei:** **indexelés[]**

**szeletelés**[tól: ig]

**benne van-e**: „in”

**hossz:** len()

**sortörés:** ’\n’

**soremelés(line break)**: sor végéről egyből atugrik a következőre

1. *Lista[]:*
   1. **rendezett sorrendje az elemeknek**
   2. **lista és sztingek különbsége:** a listában többféle típusú karakter is lehet, a sztringben nem
   3. **lista műveletek:** **len():** darabszám

**min():** minimum

**max():**maximum

**sum():** szummázza, ha a listában számok vannak

**lista.append():**hozzáadás pl:lista.append(„z”)

**lista.insert(hányadik helyre, mit):** megadott helyre pakol elemet

**lista.extend(ujlista):**hozzáad egy másik listát az eredeti végéhez

**lista.pop(x):** megadja a 0tól x-ig indexű elemeket

**lista.remove(mit):** kiszedi az első „x” értékű elemet

**lista.sort():** abc sorrendbe rakja a listát

**lista.reverse():**megfordítja a listádat

1. *Típusok:* 
   1. **tuple:** nem változékony listák, csak cserélni lehet őket

**létrehozás**: my\_tuple: 1,2 VAGY my\_tuple: (1,)

* 1. **szótár kulcs elérése**: dict[key]

**létrehozás**: my\_dict={„ati”:„3cm”,”miki”:”2cm”,”doma”:”20cm”}

* 1. **halmaz:**

**létrehozás:** my\_set=set(’abc’) vagy my\_set={’a’,’b’,’c’}

halmazokba különbőző típusu adatok is lehetnek

**a dupla ugyan olyan elemek azonnal törlődnek** ,pl: my\_set={’a’,’a’} == print(my\_set) ={a}

**hozzáadni**: .add(elem)

**törölni:** .clear() =teljes halmaz üres lesz

**.discard(elem)** lehet törölni, nem baj ha nincs olyan elem

1. *Kész feladatok:*
   * 1. **type hint:** az elem típusa
     2. **kozvetlen rekurzív függvényhívás**: a függvénben már önmagában meg van hívva a függvény
     3. **függvény készítés:** def név(feldolgozandó adat):
     4. **raise:** egy kivételt adhatunk meg, hogy akkor milyen hibakódot írjon ki a program
     5. **print():** csak kimutatja a megkapott értéket a képernyőre, nem olyan mint a return
     6. **logikai típusú elem:** and , or, not
     7. python fájltípus:?????????????
     8. függvénysáv:????????????