Dictionare {cheie:valoare}

- cheie->unica
- key==lista nu
- cmp(d1, d2)
- len(d1, d2)
- str(dict)
- type(dict)
- del dict
- --------------------------
- clear()
- copy()
- fromkeys()->din list in key
- get(key)
- items()->tuplu
- keys()
- values()
- update(d1)
- key in dict:

Liste [] le pot schimba

- cmp(l1 ,l2)
- len(l)
- max(l)
- min(l)
- list(tuplu)
- ----------------
- append()
- extend()
- index()unde apare a
- insert()
- pop()
- remove()
- reverse()
- sort()
- count() a de cate ori in lista

Fisiere

file=open(nume_fis, „drepturi”,buff)

- acces: r, w, a
- file.(closed()
- mode()
- name()
- write(sh)
- close()
- read()
- tell()
- next()
- readlines()
- readline()
- writelines()

Tuple () nu le pot schimba

- un sg el->(10,)
- del tuplu
- len(t)
- t1+t2
- („a”)*4
- 3 in (1,2,3)
- for x in t:
- --------------
- cmp(t1, t2)
- len(t)
- max(t)
- min(t)
- tuple(list)

Conversii

chr(232)=>nu unicode
dict(s)
int(„7”)
list(tuple)
ord(„a”)=>ascii
set(list)
str(7)
tuple(list)

OS

- rename(old,new)
- remove(old,new)
- mkdir(nume)
- chdir(calea)
- getcwd()
- rmdir(nume)
- listdir(nume)
- unlink(cale)
- open

Module

- in acelasi folder
- import modulul_meu
- from modulul_meu import functie
- dir(math)
- input()=>str CONVERTESTE-l
- Join(„-„)

Set random

- add()
- pop()
- remove()
- clear()
- len(s)
- item in x
- set(lista)

- and
- or
- xor ^
- dif
- subset
- superset

Lambda

->lambda par1,par2: return 1 sg val
->nu face print
->lambda arg1, arg2:arg1+arg2
->sum=lambda s1,s2:s1+s2