Laborator 7 - Dictionare

Formatori:
Tutor: Militaru Mihai-Adrian
Tutor: Dragomir Titian-Cornel
+8

Data de începere a cursului:
25.09.2023
Utilizatori înscriși
Calendar
Note

Laborator 7 - Dictionare

Dicționare (dict)

Documentatia oficiala python: https://docs.python.org/3/tutorial/datastructures.html#dictionaries.

Definirea unui dictionar

Un dicționar constă dintr-o colecție de perechi cheie-valoare. Fiecare pereche cheie-valoare mapează cheia cu valoarea asociată.

Pentru a defini un dicționar prin folosirea unei liste, separate prin virgulă, de perechi cheie-valoare între acolade ({}). Două puncte (:) separă fiecare cheie de valoarea asociată.

```
dictionar_capitale = {
    'Bucuresti': 'Romania',
    'Budapesta': 'Ungaria',
    'Chisinau': 'Moldova'
}

print(dictionar_capitale)
```

Dictionarele au urmatoarele caracterisitici:

- o anumită cheie poate apărea într-un dicționar o singură dată, cheile duplicate nu sunt permise.
- cheia trebuie să fie de un tip imuabil (de exemplu, nu poate sa fie o lista)

Accesarea valorilor din dicționar

O valoare este accesată dintr-un dicționar prin specificarea cheii sale corespunzătoare între paranteze drepte ([]):

```
print(dictionar_capitale['Bucuresti'])
```

Dacă vă referiți la o cheie care nu este în dicționar, Python ridică o excepție:

```
print(dictionar_capitale['Madrid'])
    >> print(dictionar_capitale['Madrid'])
    >> KeyError: 'Madrid'
```

Pentru a actualiza o valoare atribuită unei chei, puteți doar să atribuiți o nouă valoare unei chei existente:

```
print(dictionar_capitale)  # {'Bucuresti': 'Romania', 'Budapesta': 'Ungaria', 'Chisinau': 'Moldova'}

dictionar_capitale['Bucuresti'] = 'RO'
print(dictionar_capitale)  # {'Bucuresti': 'RO', 'Budapesta': 'Ungaria', 'Chisinau': 'Moldova'}
```

Când parcurgeți dicționarele, cheia și valoarea corespunzătoare pot fi preluate în același timp folosind metoda items(). Aceasta metoda transforma dictionarul pe care se aplica in '`dict_items'(nu confundati dict_items cu listele).`

```
print(dictionar_capitale.items())  # dict_items([('Bucuresti', 'RO'), ('Budapesta', 'Ungaria'), ('Chisinau', 'Moldova')])
```

Pentru a obtine lista tuturor cheilor folositi metoda keys(), iar pentru valori, metoda values():

```
print(dictionar_capitale.keys())  # dict_keys(['Bucuresti', 'Budapesta', 'Chisinau'])
print(dictionar_capitale.values())  # dict_values(['RO', 'Ungaria', 'Moldova'])
```

Parcurgerea dictionarelor cu ajutorul functiei reduce():

```
elev_nota = {
    'Alex': 10,
    'Mihai': 9,
    'Ioana': 10
}

print(elev_nota.items())  # dict_items([('Alex', 10), ('Mihai', 9), ('Ioana', 10)])


def functie_suma(suma, elev):
    nume, nota = elev  # despachetam fiecare tuplu primit ca parametru (exemplu: ('Alex', 10))
    return suma + nota


def medie_elevi(dictionar):
    suma_note = functools.reduce(functie_suma, dictionar.items(), 0)
    return suma_note / len(dictionar)  # lungimea unui dictionar (numarul de elemente) se obtine cu ajutorul functiei len


print(medie_elevi(elev_nota))
```

Parcurgerea recursiva a dictionarelor. Pentru parcurgerea recursiva a dictionarelor, convertim dictionarul primit ca parametru in 'dict_items', apoi convertim 'dict_items' intr-o lista pe care o sa o parcurgem recursiv.

```
def suma_recursiva(dict_list):
    if len(dict_list) > 0:
        nume, nota = dict_list[0]
        return nota + suma_recursiva(dict_list[1:])
    else:
        return 0


def medie_elevi_recursiva(dictionar):
    suma_note = suma_recursiva(list(dictionar.items()))  # dictionarul primit ca parametru este convertit in dict_items,
apoi in lista
    return suma_note/len(dictionar)

print(medie_elevi_recursiva(elev_nota))
```

Exerciții rezolvate cu Dicționare:

1.Scrieți o funcție care ia o listă de asociere cu perechi de tip (șir, întreg) și creează un dicționar în care fiecare șir e asociat cu suma tuturor valorilor cu care e asociat în listă.

Exemplu:

Input: [("Ana",7), ("Alin",3), ("Ana",9)]

Output: {'Ana': 16, 'Alin': 3}

```
def transform(lista, dictionar = {}):
    if (lista == []):
        return dictionar
    if(lista[0][0] in dictionar):
        dictionar[lista[0][0]] = lista[0][1] + dictionar[lista[0][0]]
    else:
        dictionar[lista[0][0]] = lista[0][1]
    return transform(lista[1:],dictionar)

l = [("Ana",7), ("Alin",3), ("Ana",9)]

print(transform(l))
```

◄ Cod din curs - Grafuri - in Python

Sari la...

Exercitii - Saptamana 7 - Dictionare ►

Contactați serviciul de asistență

Sunteți conectat în calitate de Ciobanu Daria-Andreea (Delogare)
S1-L-AC-CTIRO1-LSD

Meniul meu
Profil
Preferinte
Calendar
ZOOM
Română (ro)
English (en)
Română (ro)

Rezumatul păstrării datelor
Politici utilizare site