Exercitii: pag. 323-326 si exemplele din cursul 2

Teste: pag. 134-135.

1. Exersati tipuri de operatii de baza in Python, in mod interactiv, in IDLE:

>>> 2 \*\* 16

65536

>>> 2 / 5, 2 / 5.0

(0.4, 0.4)

>>> 'spam' + 'eggs'

'spameggs'

>>> "eggs" + S

'eggsham'

>>> S \* 5

'hamhamhamhamham'

>>> S[:0]

''

>>> "green %s and %s" % ('eggs', S)

'green eggs and ham'

>>> 'green {0} and {1}'.format('eggs', S)

'green eggs and ham'

>>> ('x',)[0]

'x'

>>> ('x', 'y')[1]

'y'

>>> L = [1, 2, 3] + [4, 5, 6]

>>> L, L[:], L[:0], L[-2], L[-2:]

([1, 2, 3, 4, 5, 6], [1, 2, 3, 4, 5, 6], [], 5, [5, 6])

>>> ([1, 2, 3] + [4, 5, 6])[2:4]

[3, 4]

>>> [L[2], L[3]]

[3, 4]

>>> L.reverse(); L

[6, 5, 4, 3, 2, 1]

>>> L.sort(); L

[1, 2, 3, 4, 5, 6]

>>> L.index( 4 )

3

>>> help( L.index )

Help on built-in function index:

index(value, start=0, stop=9223372036854775807, /) method of builtins.list instance

**Return first index of value.**

Raises ValueError if the value is not present.

>>> {'a': 1, 'b': 2}['b']

2

>>> D = {'x': 1, 'y': 2, 'z': 3}

>>> D['w'] = 0

>>> D

{'x': 1, 'y': 2, 'z': 3, 'w': 0}

>>> D['x'] = D['w']

>>> D

{'x': 0, 'y': 2, 'z': 3, 'w': 0}

>>> D['x'] + D['w']

0

>>> D[(1, 2, 3)] = 4

>>> D

{'x': 0, 'y': 2, 'z': 3, 'w': 0, (1, 2, 3): 4}

>>> list(D.keys()), list(D.values()), (1, 2, 3) in D

(['x', 'y', 'z', 'w', (1, 2, 3)], [0, 2, 3, 0, 4], True)

>>> [[]], ["", [], (), {}, None]

([[]], ['', [], (), {}, None])

1. Fie o lista L cu patru elemente numerice (sau string), e.g. L=[1, 2, 3, 4]
   1. Efectuati o indexare in afara listei, e.g. L[4]
   2. Efectuati o decupare (slicing) in afara listei, e.g. L[-1000:100]
   3. Extrageti o secventa inversata, cu limita din stanga mai mare decat cea din dreapta, e.g. L[3:1]. Verificati si atribuirea L[3:1] = ['?']
2. Experimentati operatii de indexare, decupare si operatorul ***del***, asupra unei liste L cu patru elemente:
   1. L[2] = []
   2. L[2:3] = []
   3. del L[0]
   4. del L[1:]
   5. L[1:2] = 1 #este corect?!
3. Atribuiri de tuple:

>>> X = 'spam'

>>> Y = 'eggs'

>>> X, Y = Y, X

1. Construiti un dictionar indicand valorile corespunzatoare cheilor:

>>> D = {}

>>> D[1] = 'a'

>>> D[2] = 'b'

>>> D

{1: 'a', 2: 'b'}

Intregii sunt **imuabili**, deci pot fi chei (idem stringuri si tuple)

>>> D[(1, 2, 3)] = 'c'

>>> D

{1: 'a', 2: 'b', (1, 2, 3): 'c'}

1. Indexari in dictionare: construiti un dictionar D cu cheile ‘a’, ‘b’ si ‘c’.
   1. Ce se intampla cand se indexeaza o cheie inexistenta, e.g. D[‘d’] ?
   2. Ce se intampla cand se atribuie o valoare unei chei inexistente, e.g. D[‘d’] = ‘spam’
   3. Comparati cu situatia listelor.

>>> D = {'a':1, 'b':2, 'c': 3}

>>> D['a']

1

>>> D['d']

Traceback (most recent call last):

File "<pyshell#72>", line 1, in <module>

D['d']

KeyError: 'd'

>>> D['d'] = 'spam'

>>> D

{'a': 1, 'b': 2, 'c': 3, 'd': 'spam'}

1. Operatii generalizate:
   1. Ce se intampla la folosirea operatorului **+** asupra unor tipuri de date mixte (string + list, list + tuplu) – NU merge

>>> L = [1, 2]

>>> S = 'spam'

>>> L + S

Traceback (most recent call last):

File "<pyshell#77>", line 1, in <module>

L + S

TypeError: can only concatenate list (not "str") to list

>>> S + L

Traceback (most recent call last):

File "<pyshell#78>", line 1, in <module>

S + L

TypeError: can only concatenate str (not "list") to str

* 1. Este corecta folosirea operatorului **+** cand unul din operanzi este un dictionar ? – NU merge, fiindca dictionarul nu este secventa!

>>> {} + {}

Traceback (most recent call last):

File "<pyshell#79>", line 1, in <module>

{} + {}

TypeError: unsupported operand type(s) for +: 'dict' and 'dict'

* 1. Functioneaza metoda append() pe liste? Dar pe stringuri? (NU, fiindca stringurile sunt imuabile); Dar metoda keys(), merge pe liste? (NU, doar pe dictionare).

>>> L.append( 9 )

>>> L

[1, 2, 9]

>>> S.append( 'ham' )

Traceback (most recent call last):

File "<pyshell#82>", line 1, in <module>

S.append( 'ham' )

AttributeError: 'str' object has no attribute 'append'

>>> L.keys()

Traceback (most recent call last):

File "<pyshell#84>", line 1, in <module>

L.keys()

AttributeError: 'list' object has no attribute 'keys'

>>> list(D.keys())

['a', 'b', 'c', 'd']

* 1. Ce tip de obiect returneaza decuparea listelor si stringurilor sau concatenarea a doua liste sau stringuri ? (acelasi tip de obiect ca operandul/operanzii)

>>> [][:]

[]

>>> ""[:]

''

1. Indexarea stringurilor: definiti un string cu patru caractere, e.g. S = ‘spam’.
   1. Ce valoare are expresia S[0][0][0][0][0] ?
   2. Dar pentru lista L = [‘s’, ‘p’, ‘a’, ‘m’] ? L[0][0][0][0] = ??

>>> S = 'spam'

>>> S[0][0][0][0][0]

's'

>>> L = ['s', 'p', 'a', 'm']

>>> L[0][0][0][0][0]

's'

>>> #Fiindca lista L contine un string in pozitia 0… iar caracterele sunt de fapt stringuri de lungime 1.

1. Tipuri imuabile: definiti un string S de patru caractere, S = ‘spam’.
   1. Preschimbati S in ‘slam’ folosind doar decuparea si concatenarea.

>>> S = 'spam'

>>> S = S[:1] + 'l' + S[2:]

>>> S

'slam'

* 1. Doar indexare si concatenare

>>> S = S[0] + 'l' + S[2] + S[3]

>>> S

'slam'

* 1. Dar doar indexare, merge? – NU, stringurile sunt imuabile

1. Construiti o structura de date imbricata pentru prenume, initiala, nume, varsta, profesie:

>>> eu = {'nume': ('Dan', 'A', 'Schrager'), 'varsta': '?', 'job': 'profesor'}

>>> eu['job']

'profesor'

>>> eu['nume'][2]

'Schrager'

1. Reproduceti TOATE exemplele in cursul 2!