ІНФОРМАТИКА ТА ПРОГРАМУВАННЯ

Тема 6. Списки

Списки

- **Списки** у Python є послідовностями, що складаються з елементів різних типів.

- Списки, на відміну від рядків, є такими, що змінюються (mutable). Тобто кількість елементів та самі елементи можуть змінюватись під час виконання програми.

Носій типу список

- Визначимо множину послідовностей з елементів множин $M_1, \ldots, M_k$, яку позначимо *Seq*, наступним чином:
  1. Порожня послідовність $\Lambda \in Seq$
  2. Якщо $A \in Seq$, $c \in M_1 \cup \ldots \cup M_k$, то $Ac \in Seq$, де Ac – результат приписування елемента c праворуч до послідовності A.
- Нехай $len(A)$ – довжина послідовності A, або кількість елементів у послідовності A.
- Тоді позначимо

$Seq_n = \{A: A \in Seq, len(A) \le n\}$

- Ця множина Seq_n і є носієм типу список.
- У Python обмеження n, як і для цілих чисел, залежить тільки від об’єму доступної пам’яті.

Основні операції для списків

- Більша частина операцій повторює аналогічні операції для рядків

Операція	Опис
[x_1, … , x_n]	Створити список з елементів x_1, … , x_n
[]	Порожній список
list(x)	Перетворення x у список (x повинно належати типу, що ітерується)
s + t	конкатенація s та t
s * n або n * s	n зчеплених копій s
s[i]	i-й елемент s, починаючи з 0, якщо i < 0, то повертає (-i) елемент з кінця списку
s[i:j]	Вирізка з s від i до j (підсписок, що починається з i –го елементу та закінчується j -1 елементом)
s[i:j:k]	Вирізка з s від i до j з кроком k
len(s)	довжина s

Основні операції для списків.2

Операція	Опис
min(s)	Найменший елемент списку s
max(s)	Найбільший елемент списку s
s.index(x[, i[, j]])	Індекс першого входження x до s (починаючи з індекса i та перед індексом j)
s.count(x)	Кількість входжень x до s

Відношення для списків

- Для списків визначено відношення 6 стандартних відношень з множини *Rel* = {==, !=, >, <, >=, <=}.
- Відношення a == b означає попарну рівність всіх елементів двох списків a, b.
- Відношення a < b визначається рекурсивно:
  1. Якщо a == [ ], b != [ ], то a < b == True
  2. Якщо b == [ ], то a < b == False
  3. Якщо a != [ ], b != [ ], a[0] != b[0] то a < b ≡ a[0] < b[0]
  4. Якщо a != [ ], b != [ ], a[0] == b[0] то a < b ≡ a[1:] < b[1:]
- Інші відношення з множини *Rel* визначається через бульові операції та відношення == та <.
- Для обчислення відношень з множини *Rel* відповідні елементи 2 списків повинні мати порівнювані типи (наприклад, обидва числові або обидва рядки і т.д.)
- Обчислення відношень для непорівнюваних типів дає помилку.

Відношення для списків.2

- Окрім відношень з множини *Rel*, для списків визначено відношення:

x in a, x not in a

  - де x – елемент, a – список.

- x in a == True, коли x входить у a
- x not in a == True, коли x не входить у a

Інструкції для списків

- Для списків визначено присвоєння та виведення.

```
a = e, print(a)
```

- Введення не визначено, тому треба вводити список поелементно.
- Визначено також цикл по всіх елементах списку

```
for x in a:
    P
```

- Окрім цього, визначено ще ряд інструкцій

Інструкції для списків.2

Інструкція	Опис
s[i] = x	і –й елемент s замінюється на x
s[i:j] = t	вирізка s від і до j замінюється значенням t (t повинно належати типу, що ітерується)
del s[i:j]	Те ж саме, що і s[i:j] = []
s[i:j:k] = t	елементи s[i:j:k] замінюються відповідними елементами t (кількість елементів s[i:j:k] та t повинна бути однаковою)
del s[i:j:k]	Видаляє елементи s[i:j:k] зі списку
s.append(x)	приписує x у кінець списку (те ж саме, що й s[len(s):len(s)] = [x])
s.sort()	Впорядковує список x за неспаданням

Реалізація масивів на базі списків

- Списки можуть бути використані для реалізації масивів.
- **Масиви** – це багатовимірні таблиці, які складаються з однотипних елементів.
- Елемент масиву доступний за його індексами.
- Кількість індексів дорівнює кількості вимірів таблиці.
- Одновимірний масив ще називають вектором, а двовимірний – матрицею.
- Наприклад,

v = [1, 3, 7, 11]

- v - вектор з 4 цілими елементами. v[1] == 3

a = [[0.5, 2.0, -3.2], [2.1, 4.6, 0.1]]

- a – матриця 2×3 з дійсних чисел. a[1][2] == 0.1

Приклади

- Обчислити скалярний добуток векторів a, b, що складаються з n дійсних компонент.

- Обчислити значення мінімального елемента матриці m×n

Взаємозв’язок рядків та списків

- Python має потужні засоби, які зв’язують рядки та списки.
- Використовуючи ці засоби, можна легко розв’язувати задачі, пов’язані з аналізом текстів.
- Розглянемо декілька операцій над рядками, в яких використовуються списки.

Взаємозв'язок рядків та списків.2

Операція	Опис
s.join(t)	Повертає рядок з усіма елементами t, між якими в якості розділювача стоїть рядок s
s.split(sep=None, maxsplit=-1)	Повертає список, у якому рядок s розбито на підрядки рядками-розділювачами sep. Якщо sep не вказано, то мається на увазі рядок з довільної кількості пропусків. Якщо вказано maxsplit!=-1, то виконується рівно maxsplit розбиттів.
s.splitlines([keepends])	Повертає список рядків, який є розбиттям s на підрядки, обмежені символами кінця рядка (\n, \r або \r\n).
s.rsplit(sep=None, maxsplit=-1)	Діє як split, але при вказанні maxsplit!=-1 рахує розбиття не від початку, а від кінця рядка s

Приклади

- Словами будемо називати послідовності символів рядка, які не містять пропусків всередині та розділені пропусками. Обчислити кількість слів у рядку.

- Видалити повторні входження слів у рядку та показати всі різні слова рядка у алфавітному порядку.

Додаткові інструкції для списків

Інструкція	Опис
s.clear()	Вилучає всі елементи з s (те ж саме, що й del s[:])
s.copy()	Робить копію s (те ж саме, що й s[:])
s.extend(t)	розширює s значенням t (те ж саме, що й s[len(s):len(s)] = t)
s.insert(i, x)	вставляє x у s у позицію, задану i (те ж саме, що й s[i:i] = [x])
s.pop([i])	Повертає елемент з позиції i а також вилучає його зі списку s (за угодою i == len(s)-1, тобто буде вилучений останній елемент)
s.remove(x)	Видаляє перший такий елемент s, для якого s[i] == x
s.reverse()	Переставляє елементи s у зворотному порядку

Спискоутворення

- Спискоутворення (list comprehension) – це вираз, результатом якого є список.
- Вираз має такий синтаксис:

```
[e(x) for x in t if F]
```

  - де $e(x)$ – вираз, який залежить від x; t – вираз типу, що ітерується, F – умова.
- Python вибирає всі x з t, які задовольняють умову F, застосовує до кожного x вираз $e(x)$ та повертає отриманий список.
- Якщо умова F відсутня, то if F опускають.

Функція enumerate

- Функція enumerate(s) застосовується для того, щоб окрім елементів списку s можна було у циклі for аналізувати індекси цих елементів.
- Ця функція повертає пари (індекс, елемент).

- Наприклад:

```
for i, c in enumerate(s):
    if i == c:
        print (i)
```

Приклади

- Обчислити значення мінімального елемента матриці m×n (версія 2)

- Видалити повторні входження слів у рядку та показати всі різні слова рядка у алфавітному порядку (версія 2)

Резюме

- Ми розглянули:
  1. Списки: носій, операції, відношення та інструкції для списків.
  2. Реалізацію масивів на базі списків
  3. Взаємозв’язок між рядками та списками
  4. Спискоутворення. Функцію enumerate

Де прочитати

1. A Byte of Python (Russian) Версия 2.01 Swaroop C H (Translated by Vladimir Smolyar), http://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.01.pdf
2. Марк Лутц, Изучаем Python, 4-е издание, 2010, Символ-Плюс
3. Python 3.4.3 documentation
4. Бублик В.В., Личман В.В., Обвінцев О.В.. Інформатика та програмування. Електронний конспект лекцій, 2003 р.,