

Лабораторна Робота 3 – Структури: Рядки, списки.

Мета роботи – вивчити та засвоїти базові навички використання базових структур даних у Python. Дізнатися про принципи взаємодії з рядками та списками, а також про основні вбудовані методи для їх обробки.

1. Рядки

1.1 Базові поняття про рядки

Рядок - це послідовність символів з довільним доступом. Рядки в мові Python неможливо змінити - в цьому випадку говорять, що це immutable тип. Спроба змінити символ в певній позиції або підстроку викличе помилку:

```
word = 'GG EZ'
word[1] = 'J'
```

```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-1-0c6fa3432d82> in <module>
      1 word = 'GG EZ'
----> 2 word[1] = 'J'

TypeError: 'str' object does not support item assignment
```

Але потрібно, то змінити можна, наприклад, так:

```
[6]: word = 'GG EZ'
      word = word[:2] + '!' + word[3:]
      print(word)

GG!EZ
```

Індекси можуть мати негативні значення для відліку з кінця - відлік починається з -1:

```
[7]: print(word[-1])

Z
```

Рядки в Пітоні можна укладати як в одинарні, так і в подвійні лапки, причому лапки одного типу можуть бути довільно вкладені в лапки іншого типу:

```
[9]: word1 = '123'
      word2 = "7'8''9"
      print(word1)
      print(word2)

123
7'8''9
```

Великі набори рядків і цілі тексти можна укладати в потрібні лапки:

```
[11]: print("""
      One
      Two
      Three
      """)

One
Two
Three
```

Зворотний слеш в рядках використовується для так званої escape-послідовності. Після слеша може іти один або кілька символів. У наступному прикладі комбінація '\ n' - це новий рядок, '\ t' - це табуляція:

```
[12]: s = 'HI\nGL\tHF'
      print(s)

HI
GL      HF
```

1.2 Зрізи

Зріз - це механізм гнучкого управління рядком на основі індексації. Можна отримати будь-який символ рядка по його індексу. Перший символ має індекс 0. Підрядок може бути визначена за допомогою зрізу - двох індексів, розділених двокрапкою:

```
[15]: word3 = 'Hello people'
      print(word3[4])
      print(word3[0:4])
      print(word3[4:8])

o
Hell
o pe
```

Якщо в зрізі опущений перший символ, значить, він дорівнює нулю; якщо опущений останній символ - він дорівнює довжині рядка:

```
[17]: print(word3[:5])
      print(word3[6:])

Hello
people
```

Можна вибирати послідовність символів з рядка з певним кроком:

```
[18]: print(word3[::-2])
      print(word3[::-1])

Hlpo!
elpoep olleH
```

1.3 Операції з рядками

Рядки можна склеювати за допомогою оператора "+" (між двох рядків поспіль взагалі можна нічого не ставити, і вони будуть сконкатеніровані), рядки можна множити за допомогою оператора "*", а також рядки можна порівнювати за допомогою операторів <, <=, =, !=, >, >=.

```
[22]: word4="Hello" + "Word"
      word5= "Hello " + "People"
      print(word4)
      print(word5)
      word5*=3
      print(word5)
      if(word5==word4):
          print("Yes")
      else:
          print("No")

HelloWord
Hello People
Hello PeopleHello PeopleHello People
No
```

1.4 Методи для обробки рядків

Є велика кількість методів для обробки рядків, далі будуть наведені основні з них:

find – знаходить підрядок в рядку - повертає позицію входження рядка, або -1:

```
[30]: word6 = 'у меня нет друзей'
      buf=(word6.find('друзей'))
      print(buf)

11
```

join - об'єднує через роздільник набір рядків:

```
[36]: word7 = ['я', 'друзья']
      sep1 = '+'
      sep2 = '='
      word9=(sep1.join(word7))
      print(word9)
      word8 = [word9, "Веселая компания"]
      word9=(sep2.join(word8))
      print(word9)

я+друзья
я+друзья=Веселая компания
```

split - це зворотна функція для **join**, розбиває рядок на послідовність:

```
[37]: print(word9.split("+"))

['я', 'друзья=Веселая компания']
```

replace - замінює в рядку одну підстроку на іншу:

```
[50]: word10=word9.replace("Я"," кто-то другой ")
      print(word10)

      кто-то другой + друзья = Веселая компания
```

strip - видаляє пробіли зліва і справа:

```
[52]: print("(" + word7[0] + ")")
      print("(" + word7[0].strip(" ") + ")")

      ( Я )
      (Я)
```

1.5 Форматування рядків

Для форматування в рядках використовується стандартний оператор – символ «%». Зліва від відсотка вказуємо рядок, праворуч - значення або список значень:

```
[55]: s = 'Hello %s' % 'word'
      print(s)
      s = 'one %d %f' % (2 , 3.5)
      print(s)
      x = 4/3
      print('%10.5f' % x)
      '%10.5f' % x

      Hello word
      one  2 3.500000
          1.33333
[55]: '  1.33333'
```

2. Списки

2.1 Базові поняття про списки

Для угруповання безлічі елементів в Пітоні використовується список list, який може бути записаний як індексована послідовність значень, розділених комами, укладена в квадратні дужки. Списки мають довільну вкладеність, тобто можуть включати в себе будь-які вкладені списки. Фізично список являє собою масив покажчиків (адрес) на його елементи.

Елементи списку не обов'язково повинні бути одного типу.

```
[62]: mas = ['hello', 'word', 65, 12.34, 5]
      print(mas)
      ['hello', 'word', 65, 12.34, 5]
```

Як і для рядків, для списків нумерація індексів починається з нуля. Для списку можна отримати зріз, об'єднати кілька списків і так далі:

```
[64]: print(mas[3])
      print(mas[0:2])
      print(mas[1:3])
      12.34
      ['hello', 'word']
      ['word', 65]
```

Можна міняти як окремі елементи списку, так і діапазон:

```
[66]: mas[1]="people"
      mas[2:5]=[2,4]
      print(mas)
      ['hello', 'people', 2, 4]
```

Також можна робити вставку у довільне місце у списку:

```
[79]: mas[1:1]=["funny"]
      print(mas)
      ['hello', 'funny', 'people', 2, 4]
```

2.2 Операції зі списками

Копіювання списку:

```
[91]: mas1=mas[:]      #створення копії списку
      mas2=list(mas)   #створення копії списку
      mas3=mas         #створення копії посилання на список
      mas[1]=1
      print(mas)|
      print(mas1)
      print(mas2)
      print(mas3)

['hello', 1, 'people', 2, 4]
['hello', 'funny', 'people', 2, 4]
['hello', 'funny', 'people', 2, 4]
['hello', 1, 'people', 2, 4]
```

Додавання або конкатенація списків:

```
[92]: mas4=mas3+mas2
      print(mas4)|

['hello', 1, 'people', 2, 4, 'hello', 'funny', 'people', 2, 4]
```

Множення, або повторення списків:

```
[93]: mas5=mas2*3
      print(mas5)

['hello', 'funny', 'people', 2, 4, 'hello', 'funny', 'people', 2, 4, 'hello', 'funny', 'people', 2, 4]
```

2.3 Вбудовані функції

Списки мають великий набір функцій:

- **append, extend** - додавання;
- **insert** - вставка;
- **index** - знайти індекс першого входження конкретного елемента;
- **count** - підрахунок повторів елемента;
- **remove, del** - видалення елемента;
- **sort** - сортування;
- **reverse** - реверс;
- **pop** - витяг елемента;
- **len** - довжина списку;
- **max** - максимальний елемент;
- **min** - мінімальний елемент;
- **in** - перевірка елемента на входження.

Додавати можна як одинарні елементи, так і набір елементів. Списки можуть бути вкладеними - вкладений список.

```
[9]: mas6 = ['hi', 'I', 'am', 'here']  
    mas.append(mas6)  
    print(mas)  
['hello', 1, 'people', 2, 4, ['hi', 'I', 'am', 'here']]
```

```
[20]: mas.insert(5, 'chh')  
    print(mas)  
['hello', 1, 'people', 2, 4, 'chh', ['hi', 'I', 'am', 'here']]
```

```
[40]: print(mas.count(1))
```



```
[37]: print(mas.index('people'))
```

2

```
[36]: if "people" in mas:
      print("true")
      else:
      print("false")
      if "cat" in mas:
      print("true")
      else:
      print("false")
```

true
false

```
[41]: mas.remove('people')
      print(mas)
```

['hello', 1, 2, 4, 1, ['hi', 'I', 'am', 'here']]

```
[42]: del mas[5][3]
      print(mas)
```

['hello', 1, 2, 4, 1, ['hi', 'I', 'am']]

```
[49]: mas_int=[1,3,5,4,7,2,1,5,6,8,4,2,4]
      mas_int.sort()
      print(mas_int)
```

[1, 1, 2, 2, 3, 4, 4, 4, 5, 5, 6, 7, 8]

```
[51]: mas_int.reverse()
      print(mas_int)
```

[8, 7, 6, 5, 5, 4, 4, 4, 3, 2, 2, 1, 1]

```
[52]: print(len(mas_int))
      print(max(mas_int))
      print(min(mas_int))
```

13
8
1

```
[54]: print(mas_int.pop())
      print(mas_int)
      print(mas_int.pop(0))
      print(mas_int)
```

1
[8, 7, 6, 5, 5, 4, 4, 4, 3, 2, 2, 1]
8
[7, 6, 5, 5, 4, 4, 4, 3, 2, 2, 1]

2. Варіанти завдання

Таблиця 1.2 – Варіанти завдань

№	Завдання
1.	<p>Дано рядок A=" if two witches were watching two watches, which witch would watch which watch?" – замінити всі голосні літери цього рядка порядковим номером цієї літери в алфавіті B="abcdefghijklmnopqrstuvwxyz".</p> <p>Надрукувати рядок використовуючи форматування цифр у числовий тип.</p> <p>Задані дві матриці однакового розміру $X=(5 \times 5)$, $Y=(5 \times 5)$ зі значеннями елементів від -10 до 10. Сформувати матрицю Z, що є сумою матриць X та Y.</p>
2.	<p>Дано рядок A=" the thirty-three thieves thought that they thrilled the throne throughout thursday." – замінити всі перші літери кожного слова цього рядка порядковим номером цієї літери в алфавіті B="zyxwvutsrqponmlkjihgfedcba".</p> <p>Надрукувати рядок використовуючи форматування цифр у числовий тип.</p> <p>Задані дві матриці однакового розміру $X=(4 \times 5)$, $Y=(5 \times 4)$ зі значеннями елементів від -10 до 10. Сформувати матрицю Z, що є добутком матриць X та Y.</p>
3.	<p>Дано рядок A=" You know New York, you need New York, you know you need unique New York." – замінити всі літери верхнього регістру порядковим номером наступного символу в алфавіті B="abcdefghijklmnopqrstuvwxyz".</p> <p>Надрукувати рядок використовуючи форматування цифр у числовий тип.</p> <p>Задані дві матриці однакового розміру $X=(6 \times 3)$, $Y=(6 \times 3)$ зі значеннями елементів від -10 до 10. Сформувати матрицю $Z=(6 \times 3)$, яка буде формуватися перенесенням відповідного стовпчику з матриць (X або Y) в якому середнє арифметичне цього стовпця буде більшим (при однакових значеннях віддавати перевагу матриці X)</p>
4.	<p>Дано рядок A=" he thrusts his fists against the posts and still insists he sees the ghosts." – замінити всі входження підрядка "th" на "st" і навпаки, замінити кожен пробіл на номер наступної за ним літери в алфавіті B="abcdefghijklmnopqrstuvwxyz".</p> <p>Надрукувати рядок використовуючи форматування цифр у числовий тип.</p> <p>Задані дві матриці однакового розміру $X=(5 \times 3)$, $Y=(5 \times 3)$ зі значеннями елементів від -10 до 10. Сформувати матрицю $Z=(5 \times 3)$, яка буде формуватися перенесенням відповідного елементу з матриць (X або Y) модуль якого буде більшим(при однакових значеннях віддавати перевагу Y).</p>

5.	<p>Дано рядок A=" i wish to wish the wish you wish to wish, but if you wish the wish the witch wishes, i won't wish the wish you wish to wish. " – замінити всі входження підрядка "wi" на "sh" і навпаки, замінити кожен пробіл на номер наступної за ним літери в алфавіті B="zyxwvutsrqponmlkjihgfedcba".</p> <p>Надрукувати рядок використовуючи форматування цифр у числовий тип.</p> <p>Задані дві матриці однакового розміру $X=(4 \times 4)$, $Y=(4 \times 4)$ зі значеннями елементів від -10 до 10. Сформувати матрицю $Z=(2 \times 4)$, першим рядком якої буде список середніх арифметичних рядків матриці X, а другий список середніх арифметичних стовпців матриці X</p>
6.	<p>Дано рядок A=" can you can a canned can into an uncanned can like a canner can can a canned can into an uncanned can?" – підрахувати у рядку усі подвоєння літер, також вивести усі варіанти подвоєнь які зустрічаються у рядку, замінити усі подвоєння на порядковий номер цієї літери в алфавіті B="zyxwvutsrqponmlkjihgfedcba".</p> <p>Надрукувати рядок використовуючи форматування цифр у числовий тип.</p> <p>Задані дві матриці однакового розміру $X=(3 \times 4)$, $Y=(3 \times 4)$ зі значеннями елементів від -10 до 10. Сформувати матрицю $Z=(3 \times 4)$, яка буде формуватися перенесенням відповідного рядку з матриці (X або Y) в якому середнє арифметичне цього стовпця буде меншим (при однакових значеннях віддавати перевагу матриці Y)</p>
7.	<p>Дано рядок A=" if you must cross a course cross cow across a crowded cow crossing, cross the cross coarse cow across the crowded cow crossing carefully." – замінити в кожному слові рядка, всі букви які зустрічаються більше одного разу (в межах слова) на їх номер в алфавіті B="zyxwvutsrqponmlkjihgfedcba".</p> <p>Підрахувати суму всіх отриманих в рядку чисел.</p> <p>Надрукувати рядок використовуючи форматування цифр у числовий тип.</p> <p>Задана матриця $Z=(6 \times 6)$, виконати дзеркальне відображення матриці відносно головної діагоналі.</p>
8.	<p>Дано рядок A=" brisk brave brigadiers brandished broad bright blades, blunderbusses, and bludgeons balancing them badly. " – замінити всі входження підрядка "br" на "bl" і навпаки, замінити кожен пробіл на номер наступної за ним літери в алфавіті B="zyxwvutsrqponmlkjihgfedcba".</p> <p>Надрукувати рядок використовуючи форматування цифр у числовий тип.</p> <p>Задані дві матриці однакового розміру $X=(4 \times 5)$, $Y=(4 \times 5)$ зі значеннями елементів від -10 до 10. Сформувати матрицю $Z=(4 \times 5)$, кожен елемент якої буде формуватися як сума модулів відповідних елементів із матриць, зі знаком «+», якщо знаки елементів були однаковими, та «-» у зворотному випадку.</p>

9.	Дано рядок A="imagine an imaginary menagerie manager managing an imaginary menagerie. " – в кожному слові рядка розташувати слова в зворотному порядку, пробіли між словами замінити на порядковий номер пробілу в реченні.
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця $Z=(5 \times 5)$, помножити кожен елемент на модуль середнього арифметичного рядку в якому він розташований.
10.	Дано рядок A="six sick hicks nick six slick bricks with picks and sticks. " – замінити всі слова коротше чотирьох символів на номер входження слова в речення.
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця $Z=(7 \times 7)$, виконати дзеркальне відображення матриці відносно центрально рядка.
11.	Дано рядок A="send toast to ten tense stout saints' ten tall tents. " – замінити кожен останню букву слова на літеру, а кожен першу на її номер в алфавіті B="abcdefghijklmnopqrstuvwxyz".
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця $Z=(4 \times 6)$, помножити кожен елемент на модуль середнього арифметичного стовпчика в якому він розташований.
12.	Дано рядок A="how can a clam cram in a clean cream can?" – замінити кожне друге слово речення на порядковий номер слова в реченні.
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця $Z=(4 \times 5)$, виконати зрушення кожного стовпця на 2 позиції вниз.
13.	Дано рядок A="i wish to wash my irish wristwatch." –замінити всі голосні літери цього рядка порядковим номером цієї літери в алфавіті B="zyxwvutsrqponmlkjihgfedcba".
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця $Z=(5 \times 5)$, виконати дзеркальне відображення матриці відносно центрально стовпчика.
14.	Дано рядок A="how can a clam cram in a clean cream can?" – замінити всі останні літери кожного слова цього рядка порядковим номером цієї літери в алфавіті B="abcdefghijklmnopqrstuvwxyz".
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця $Z=(4 \times 5)$, виконати зрушення кожного рядку на 2 позиції вліво.

15.	Дано рядок A="send toast to ten tense stout saints' ten tall tents." – замінити центральну букву кожного слова на її порядковий номер в алфавіті B="zyxwvutsrqponmlkjihgfedcba" (якщо в слові парна кількість букв - замінити дві центральні) .
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця Z=(6 x 6), виконати дзеркальне відображення матриці відносно побічної діагоналі.
16.	Дано рядок A="six sick hicks nick six slick bricks with picks and sticks." – замінити кожну першу літеру слова на літеру верхнього регістру, а кожну останню на її номер в алфавіті B="abcdefghijklmnopqrstuvwxyz".
	Надрукувати рядок використовуючи форматування цифр у числовий тип.
	Задана матриця Y = (5 x 5), створити вектор Z, що складається з максимальних елементів кожного стовпця матриці.

3. Приклад

Варіант - 16.

```
[35]: a="six sick hicks nick six slick bricks with picks and sticks"#вхідний рядок
      alf1="abcdefghijklmnopqrstuvwxyz"#алфавіт нижнього регістру
      alf2="ABCDEFGHIJKLMNOPQRSTUVWXYZ"#алфавіт верхнього регістру
      sp=" " #пробіл
      i=0
      a=alf2[alf1.find(a[0])] + a[1:] #зміна регістру першого символу
      while i<len(a): #доки рядок не закінчився
          if(a[i]==sp): #пошук пробілу у рядку
              a=a[:i+1]+alf2[alf1.find(a[i+1])] + a[i+2:] #зміна регістру символу після пробілу
              k=alf1.find(a[i-1])
              a=a[:i-1]+"%d"%k+a[i:] #заміна символу до пробілу на його номер у алфавіті
              if(k>9): #перевірка чи змінився розмір рядку
                  i+=1
              i+=1
      a=a[:-2]+"%d"%alf1.find(a[-1]) #заміна останнього символу до на його номер у алфавіті
      i=0;
      print(a)
```

Si23 Sic10 Hick18 Nic10 Si23 Slic10 Brick18 Wit7 Pick18 An3 Stic18

```
[36]: #форматування цифр
      for i in range(len(a)):
          if(alf1.find(a[i])!=-1 and alf2.find(a[i])!=-1 and sp.find(a[i])!=-1):
              print("%d" %a[i], end = '')
          else:
              print(a[i], end = '')
```

Si23 Sic10 Hick18 Nic10 Si23 Slic10 Brick18 Wit7 Pick18 An3 Stic18

```
[37]: y= [[3,4,2,5,4],\
          [4,5,8,5,7],\
          [5,4,5,6,7],\
          [3,3,5,8,8],\
          [9,4,5,6,7]] #вхідна матриця
      z=list(); #результуючий вектор
      sum=0
      for j in range(5):
          sum=0;
          for i in range(5):
              sum+=y[i][j] # підрахунок суми елементів
          z+= [sum/5]; # формування результуючого вектору
      print(z);
```

[4.8, 4.0, 5.0, 6.0, 6.6]

4. Контрольні запитання

1. Що таке зрізи, та як вони застосовуються у рядках та списках.
2. Які існують можливі операції з рядками.
3. Що таке конкатенація рядків, які існують способи конкатенації.
4. Які існують методи обробки рядків, наведіть приклади.
5. Форматування - як застосовується та для чого.
6. Які існують можливі операції з списками.
7. Якого типу можуть бути елементи в межах одного списку.
8. Наведіть приклади вбудованих функцій для роботи зі списками.
9. Опишіть механізми звернення, заміни та видалення елементу рядка.
10. Множення списків, як виконується та для чого.