



# Become QA Auto

## СПИСКИ

Бутенко Сергій



```
le.py > ...
x = "PiMyLifeUp" #First assign the 'x' variable a string
type(x) #Output the data type of our 'x' variable
# 43 #Now assign it a integer value
type(x) #Output its type again

days = 5
print("There are " + str(days) + " days r

a, b, c = "Apple", "Banana", "Cherry"

b = c = "Lime"

print("Hello World")
# 453
int(x)
int(y)

print("Life Up")
```



# План лекції



**Поняття списку**



**Доступ до елементів списку**



**Основні етапи роботи зі списками**



**Вилучення елементів списку**



**Додавання елементів до списку**



**Автоматизація опрацювання списків за допомогою циклів**



# Програмне забезпечення для газопостачальної компанії



## Функціонал:

- Облік споживання
- Обчислення вартості спожитого газу
- Визначення максимальної заборгованості по оплаті

## Недоліки збереження даних у незалежних змінних:

- Кількість споживачів змінюється
- Необхідно змінювати кількість змінних
- Необхідно вводити нові змінні у розрахункові формули
- Неможливість автоматизації обчислень



# Списки – аналоги таблиць



N рахунку	Спожити й газ (значенн я)
0	153
1	220
2	0
3	23
4	120
5	34
6	56
...	
9998	150
9999	183

індекс	Список gas_volume
0	153
1	220
2	0
3	23
4	120
5	34
6	56
...	
9998	150
9999	183

**Список** – це впорядкована динамічна колекція елементів довільного типу.

- Список - динамічна структура
- Список може містити дублікати

## Основні етапи роботи із списками:

- Оголошення та ініціалізація
- Опрацювання
- Вивід результату

## Оголошення списку:

`<ім'я> = [<елемент1>, <елемент2>]` або

`<ім'я> = list(<елемент1>, <елемент2>)`

Приклад:

`gas_volume = [153, 220...] або`

`gas_volume = list((153, 220...))`



# Додавання елементів до списку



індекс	Список gas_volume
0	153
1	220
2	0
3	23
4	120
5	34
6	56
...	
9998	150
9999	183

## Синтаксис додавання елемента до списку:

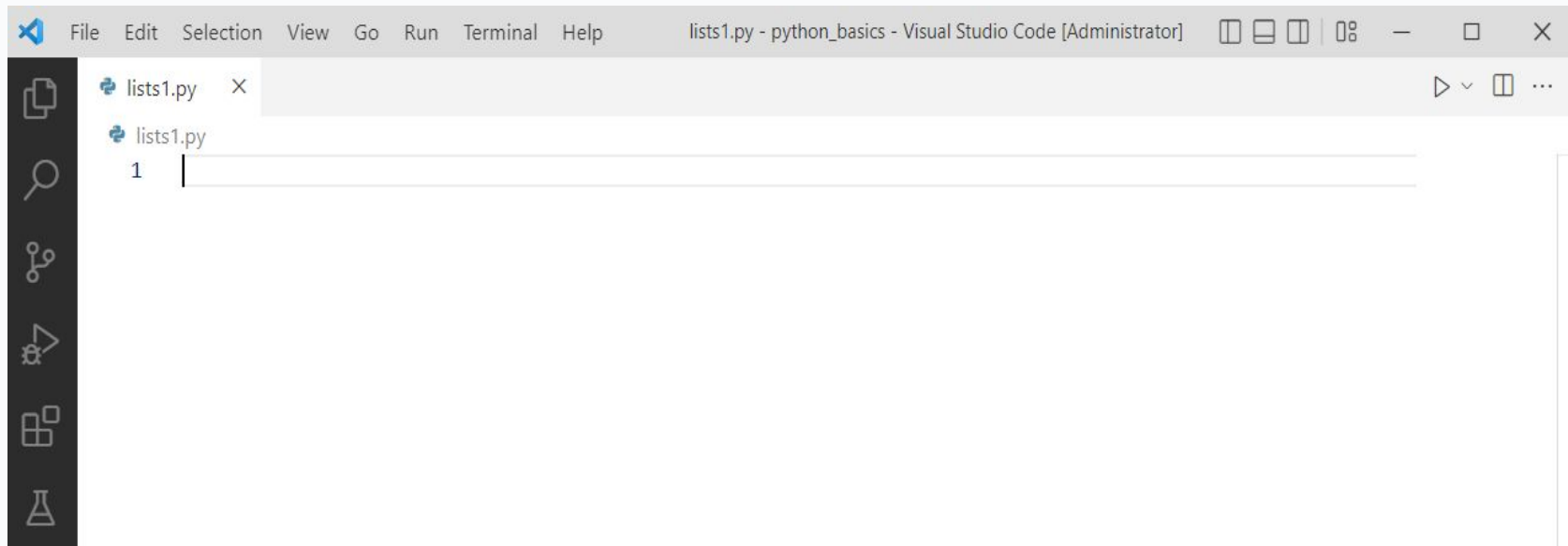
```
<ім'я_списку>.append(<новий_елемент>)
```

## Наприклад:

```
gas_volume = [153, 220, 0]
print(gas_volume)
>> [153, 220, 0]
gas_volume.append(777)
print(gas_volume)
>> [153, 220, 0, 777]
```



# Практика





## Практика



lists1.py X

lists1.py > ...

```
1 gas_volume = [153, 220, 0]
2 print(gas_volume)
3 gas_volume.append(777)
4 print(gas_volume)
5
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS C:\Users\sergii.butenko\python_basics> & C:/Users/sergii.butenko/python_basics/n.exe c:/Users/sergii.butenko/python_basics/lists1.py
```

```
[153, 220, 0]
```

```
[153, 220, 0, 777]
```

```
PS C:\Users\sergii.butenko\python_basics>
```



# Доступ до елемента списку



індекс	Список gas_volume
0	153
1	220
2	0
3	23
4	120
5	34
6	56
...	
9998	150
9999	183

## Синтаксис доступу до елемента списку:

`<ім'я_списку>[<індекс_елемента>]`

## Наприклад:

```
gas_volume[3]=23  
gas_volume[9998]=150
```





# Практика



lists2.py X

lists2.py > ...

```
1 gas_volume = [153, 220, 0]
2 print(gas_volume[0])
3 print(gas_volume[1])
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

```
PS C:\Users\sergii.butenko\python_basics> & C:/Users/sergii.butenko/ocal/Programs/Python/Python310/python.exe c:/Users/sergii.butenko/python_basics/lists2.py
```

```
153
```

```
220
```

```
PS C:\Users\sergii.butenko\python_basics> █
```



# Вилучення елементів списку



індекс	Список gas_volume
0	153
1	220
2	0
3	23
4	120
5	34
6	56
...	
9998	150
9999	183

## Способи вилучення елементів із списку:

1. Вилучення елемента за індексом
2. Вилучення елемента за значенням

```
gas_volume = [153, 220, 0]
del gas_volume[0]
print(gas_volume)
>> [220, 0]
```

```
gas_volume.remove(0)
print(gas_volume)
>> [220]
```



# Практика

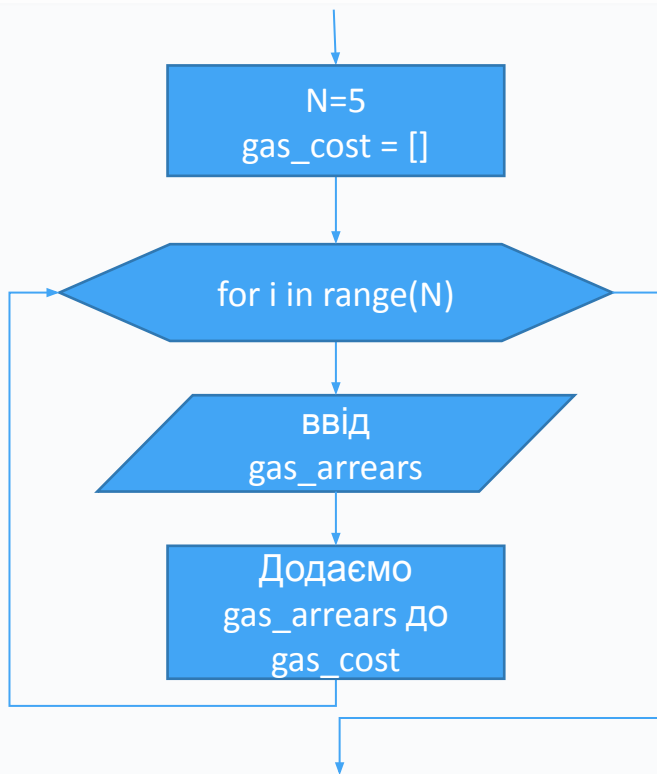


```
lists3.py X
lists3.py > ...
1 gas_volume = [153, 220, 0]
2 print(gas_volume)
3
4 del gas_volume[0]
5 print(gas_volume)
6
7 gas_volume.remove(0)
8 print(gas_volume)
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
PS C:\Users\sergii.butenko\python_basics> & C:/Users/sergii.butenko/Programs/Python/Python310/python.exe c:/Users/sergii.butenko/lists3.py
[153, 220, 0]
[220, 0]
[220]
PS C:\Users\sergii.butenko\python_basics> 
```

# ⦿ Автоматизація заповнення списку



## Задача:

Оператору компанії необхідно ввести значення заборгованостей по оплаті за газ.

## Розв'язок задачі:

1. Кількість споживачів газу `N=5`
2. Створення списку `gas_cost`
3. В циклі для `i`, що змінюється від 0 до `N-1` з кроком 1:  
Ввести `gas_arrears` для споживача з номером `i`  
Додати це значення до списку `gas_cost`



# Практика



lists4.py X

8.1 Списки > lists4.py > ...

```
1 N = 5
2 gas_cost = []
3
4 for i in range(N):
5     gas_arrears = float(input("Введіть заборгованість: "))
6     gas_cost.append(round(gas_arrears, 2))
7
8 print(gas_cost)
9
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS C:\Users\sergii.butenko\Desktop\python_basics> & C:/Users/sergii.but
/python_basics/8.1 Списки/lists4.py"
```

```
Введіть заборгованість: 99
```

```
Введіть заборгованість: 100.56896
```

```
Введіть заборгованість: 9.87
```

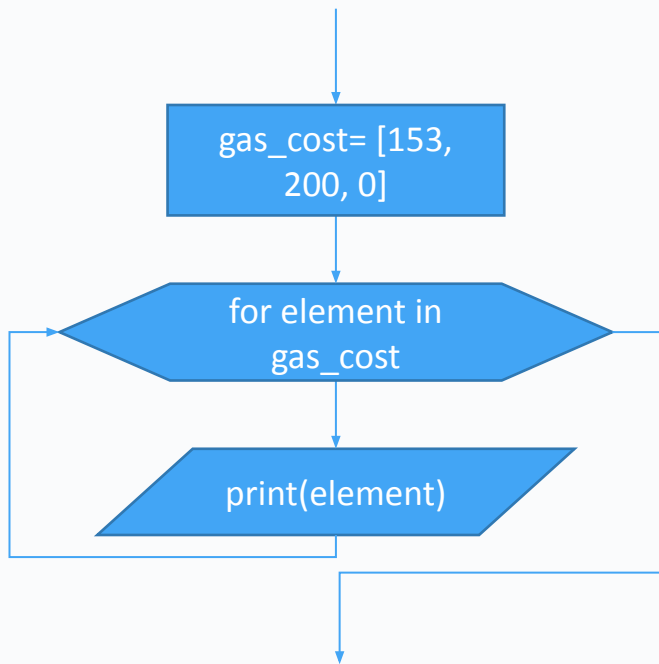
```
Введіть заборгованість: 9
```

```
Введіть заборгованість: 100000
```

```
[99.0, 100.57, 9.87, 9.0, 100000.0]
```

```
PS C:\Users\sergii.butenko\Desktop\python_basics> █
```

# ⦿ Автоматизація виводу елементів списку



## Задача:

Оператору компанії необхідно вивести кожне значення заборгованостей по оплаті за газ в окремому рядку.

## Розв'язок задачі:

В циклі для змінної `element`, що набуває значення кожного елемента списку по черзі:

вивести значення `element` на екран



# Практика



lists5.py X

8.1 Списки > lists5.py > ...

```
1 gas_cost = [153, 200, 0]
2
3 for element in gas_cost:
4     print(element)
5
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

```
PS C:\Users\sergii.butenko\Desktop\python_basics> & C:/Users/
/python_basics/8.1 Списки/lists5.py"
```

153

200

0

```
PS C:\Users\sergii.butenko\Desktop\python_basics> █
```



# Підсумки



**Список** – це впорядкована динамічна колекція елементів довільного типу.



**Всі дії над списками виконуються поелементно.**



**Основні етапи роботи зі списками:**

- Оголошення та ініціалізація
- Опрацювання
- Вивід результату