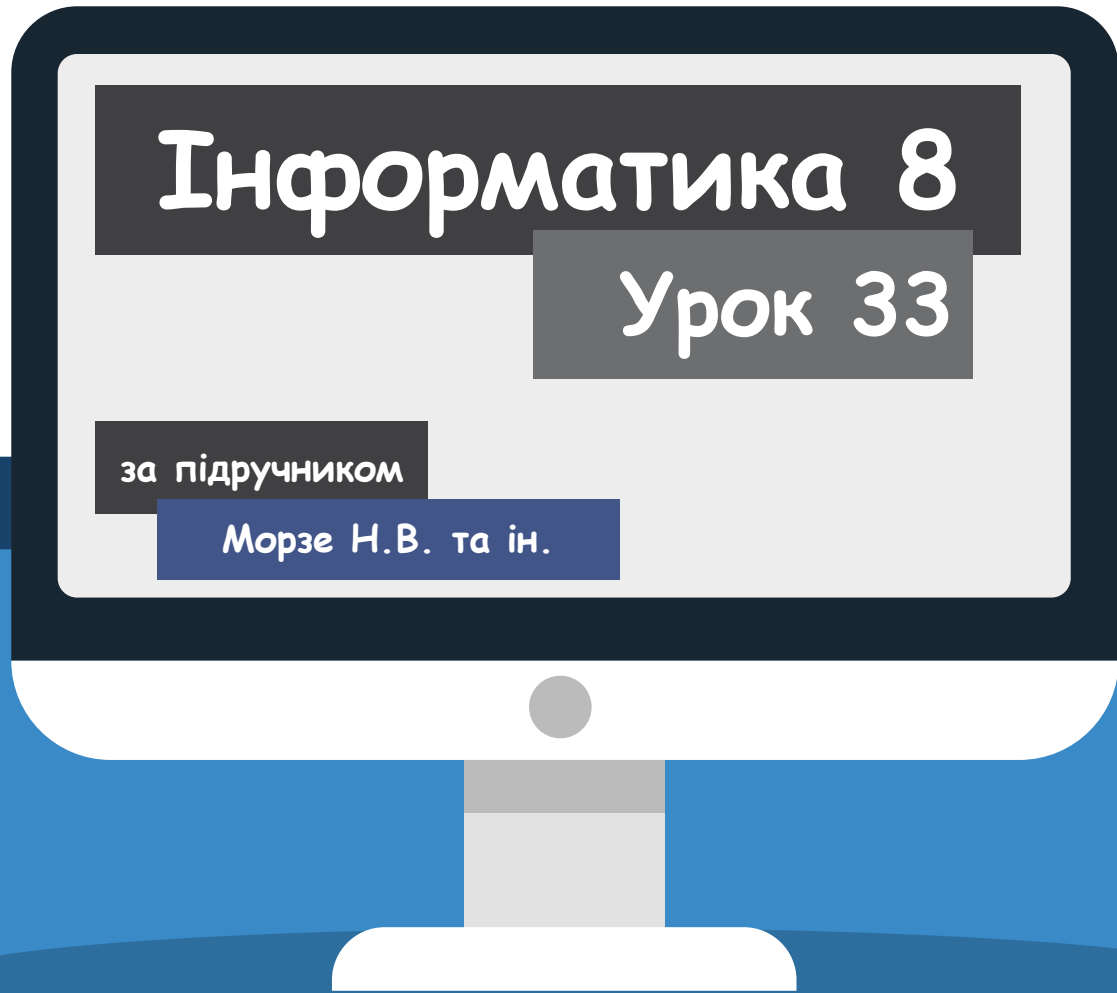


Робота з даними у мові програмування (списки, масиви, кортежі)



Нова українська школа



Як працюють із масивом даних у мові програмування?

Мотиваційний вступ

- *Робота з набором даних у програмуванні дає змогу ефективно керувати великими обсягами даних та автоматизувати їх опрацювання. Це дозволяє швидко знаходити закономірності, аналізувати дані та приймати точні рішення.*

Завдання

- *Опанувавши роботу з даними, ви зможете створювати програми для різних завдань: від дослідження тенденцій до прогнозування майбутніх результатів.*

Як у мові програмування Python описують послідовність даних?

*Ви вмiєте створювати програми мовою програмування **Python** для опрацювання певної кількості змінних.*

Їхні значення вводять

Результати опрацювання виводяться

```
Thonny - C:\Users\ProMa\Desktop\Test.py @ 8:24
Файл  Правка  Вигляд  Виконати  Інструменти  Допомога
Test.py * x
1  #Периметр п'ятикутника
2  a=int(input())
3  b=int(input())
4  c=int(input())
5  d=int(input())
6  f=int(input())
7  p=a+b+c+d+f
8  print('Периметр p=', p)
```

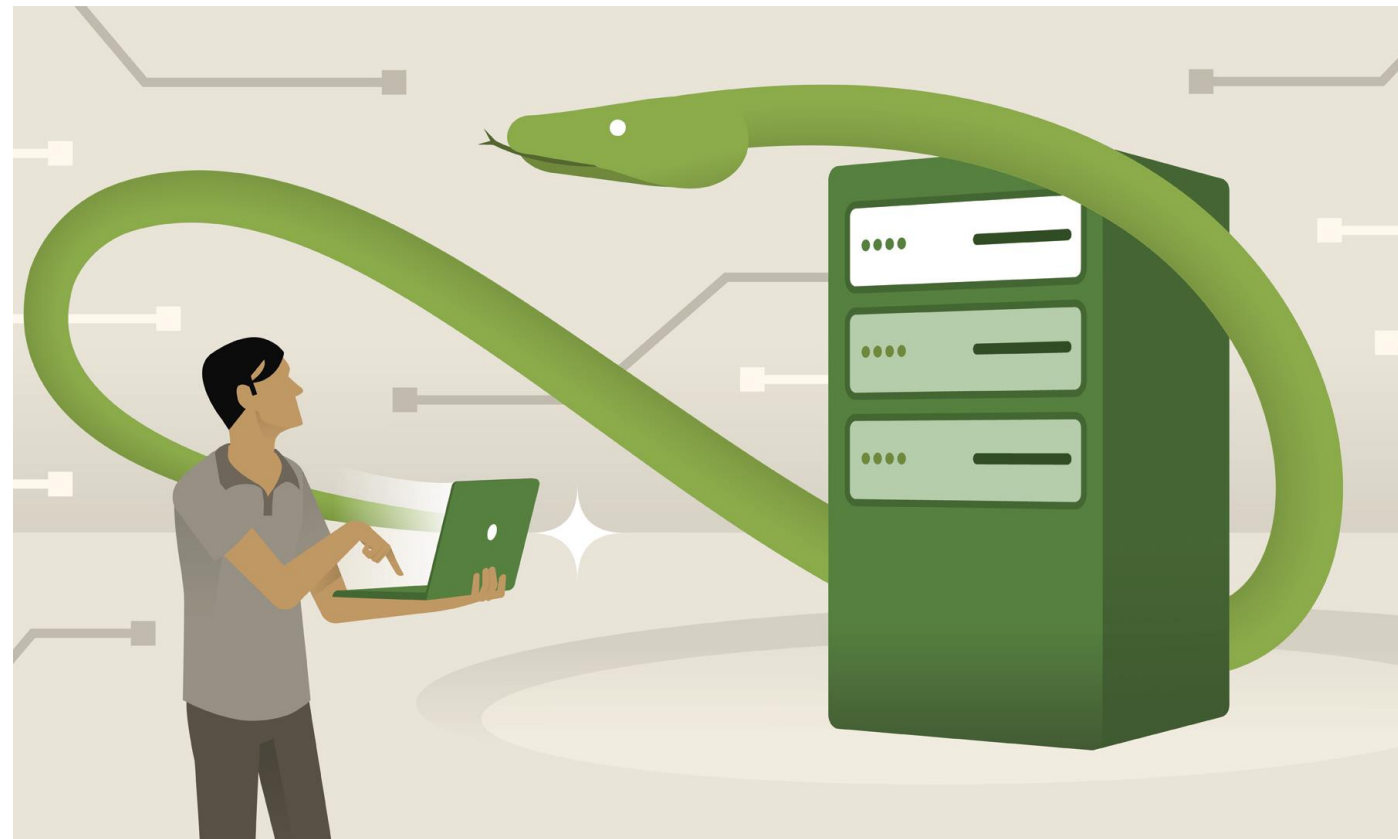
*за допомогою команди введення **input()***

*командою **print()***

Як у мові програмування Python описують послідовність даних?

Але часто потрібно опрацьовувати велику кількість змінних або множину значень однієї змінної.

*Для цього в мові програмування **Python** використовують **списки (list)** — впорядковану послідовність із декількох значень об'єктів будь-яких типів: числа, рядки, списки тощо.*



Як у мові програмування Python описують послідовність даних?

*В інших мовах програмування для опису сукупності однотипних даних, об'єднаних спільним іменем, яким можна надавати значення в процесі виконання програми, використовують **масиви**.*

Цікаво

Масив у програмуванні — це тип структури даних, що має складені значення. (З Оксфордського словника англійської мови, 1995 р.)

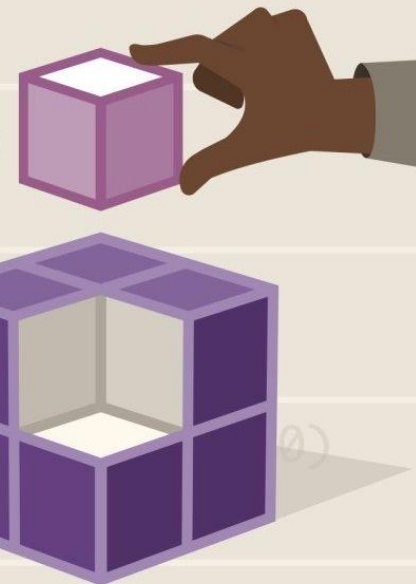
```
import numpy as np
```

```
integers_array=np.arange(10)  
integers_array
```

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
integers_second_array=np.arange(10, 110)  
integers_second_array
```

```
array([100, 101, 102, 103, 104, 105, 106, 107, 108, 109])
```




Як у мові програмування Python описують послідовність даних?

На відміну від рядків, списки складаються не з символів, а з різних об'єктів (значень, даних) і вкладаються не в лапки, а в квадратні дужки

[]

```
eng_di= {}  
#Adding lists as values  
eng_di['solitude']= ['loneliness', 'isolation', 'being alone', 'being  
eng_di['hope']= ['aspiration', 'ambition', 'desire', 'wish', 'yearning',  
print (eng_di)
```



```
#Creating a dictionary with an empty  
eng_di.clear()
```

Як у мові програмування Python описують послідовність даних?

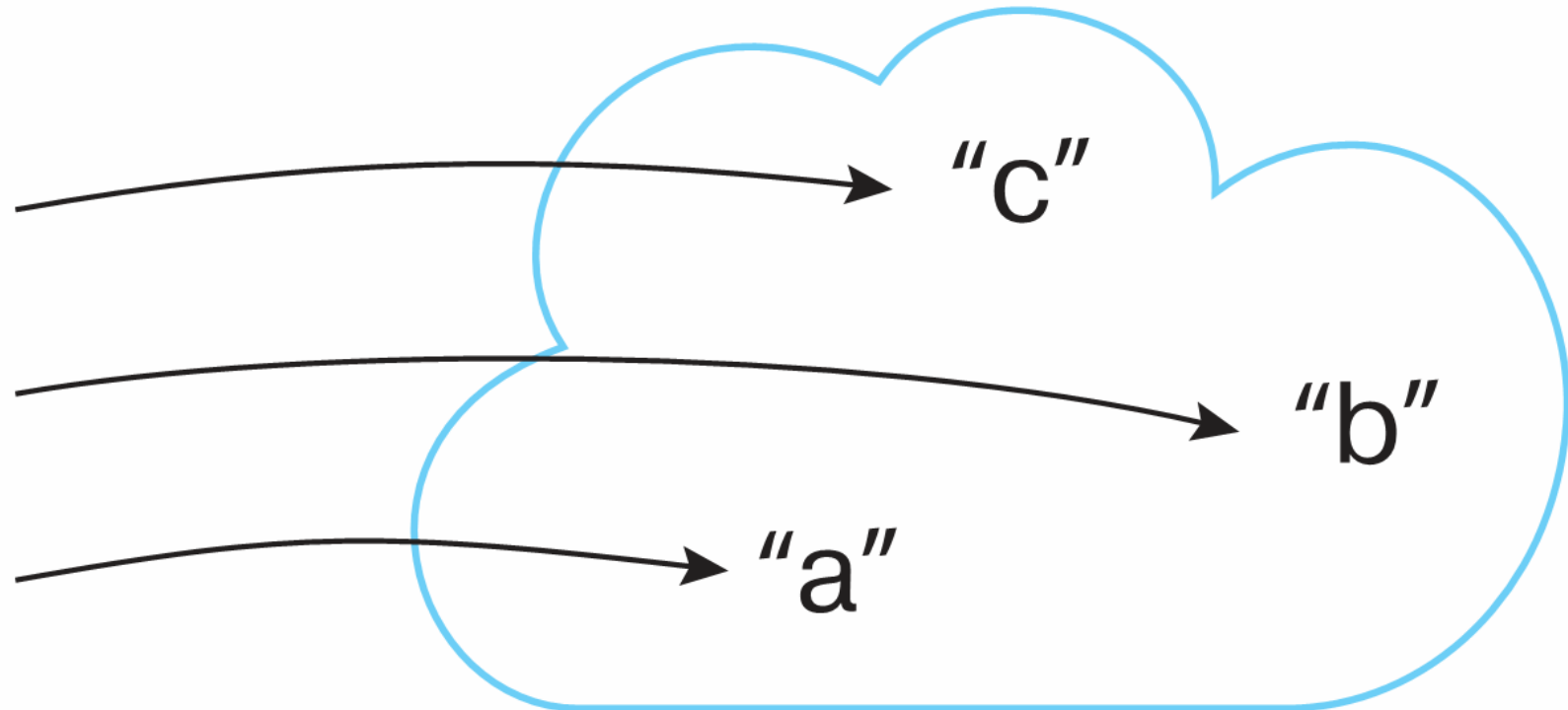
Взагалі **Python** розміщує елементи списку в пам'яті, а сам список містить вказівники на об'єкти. Таким чином, список в **Python** — це масив вказівників.

List

2

1

0



Як у мові програмування Python описують послідовність даних?

У мові **Python** елементи списку розділяються комами.
Наприклад,

```
empty_list = []
```

*порожній список
з ім'ям empty_list*

```
weekdays = ['Monday', 'Friday',  
'Wednesday', 'Thursday', 'Friday']
```

*СПИСОК ДНІВ ТИЖНЯ
з ім'ям weekdays*

```
animals = ['camels', 'bats',  
'elephants', 'dolphins', 'bears']
```

*Список тварин з ім'ям
animals*

Як у мові програмування Python описують послідовність даних?

Продовження...

```
mix=['Oksana', 15,165]
```

*Мішаний список,
що містить текстові
та числові дані
з ім'ям mix*

```
bin=[[0, 0, 0], [0, 0, 1], [0, 1, 0]]
```

*Список, що складається зі
списків з ім'ям bin*

Як у мові програмування Python описують послідовність даних?

У мові програмування **Python** вивести всі елементи списку *a* можна за допомогою команди **print(a)**, при цьому будуть виведені значення елементів списку і поміщені в квадратні дужки через кому.

```
a = [1, 2, 3]
b = [4, 5]
c = a + b
d = b * 3
print(c)
print(d)
print([7, 8] + [9])
print([0, 1] * 3)
```

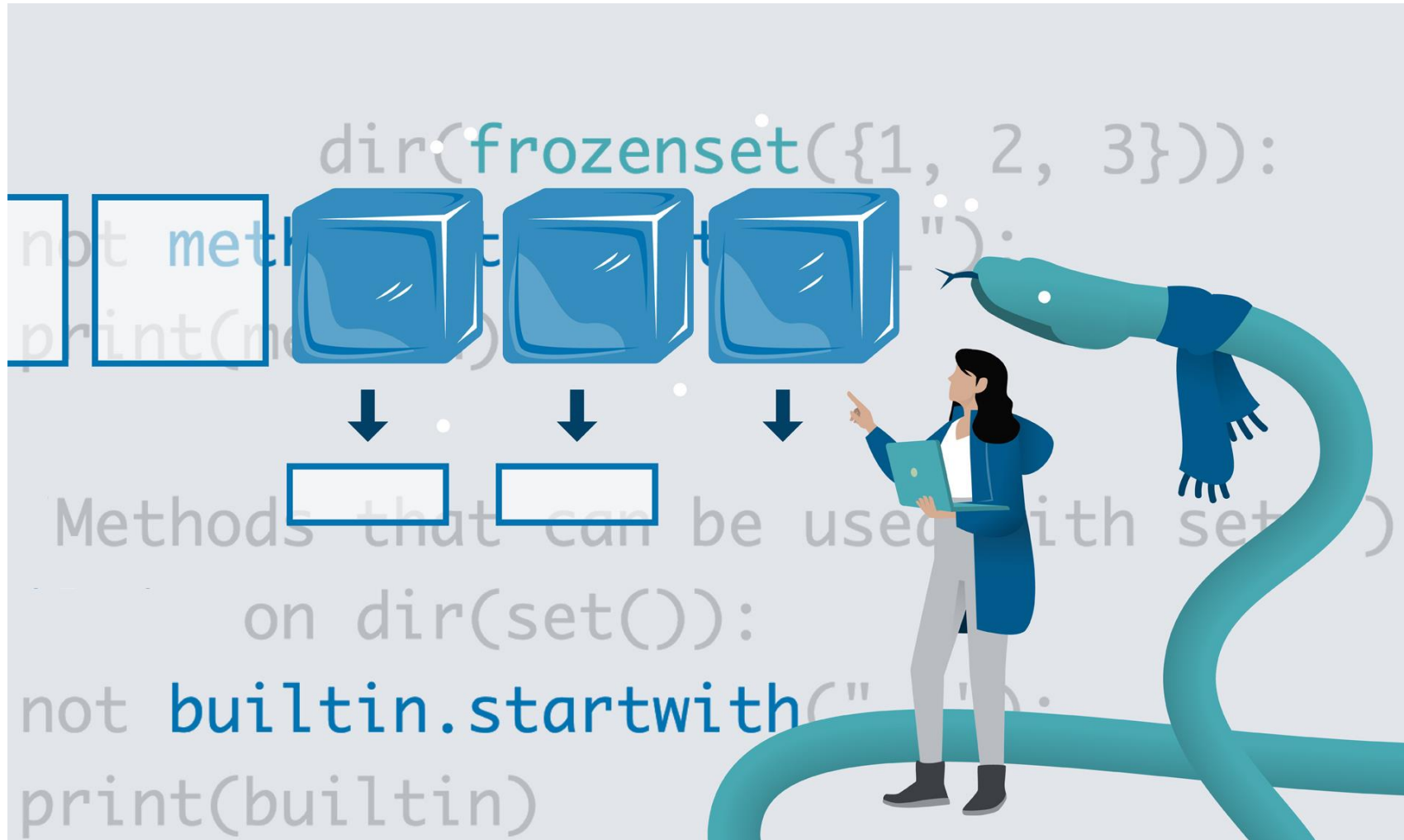
Програмний код

Результат

```
[1, 2, 3, 4, 5]
[4, 5, 4, 5, 4, 5]
[7, 8, 9]
[0, 1, 0, 1, 0, 1]
```

Як звернутись до окремого елемента списку?

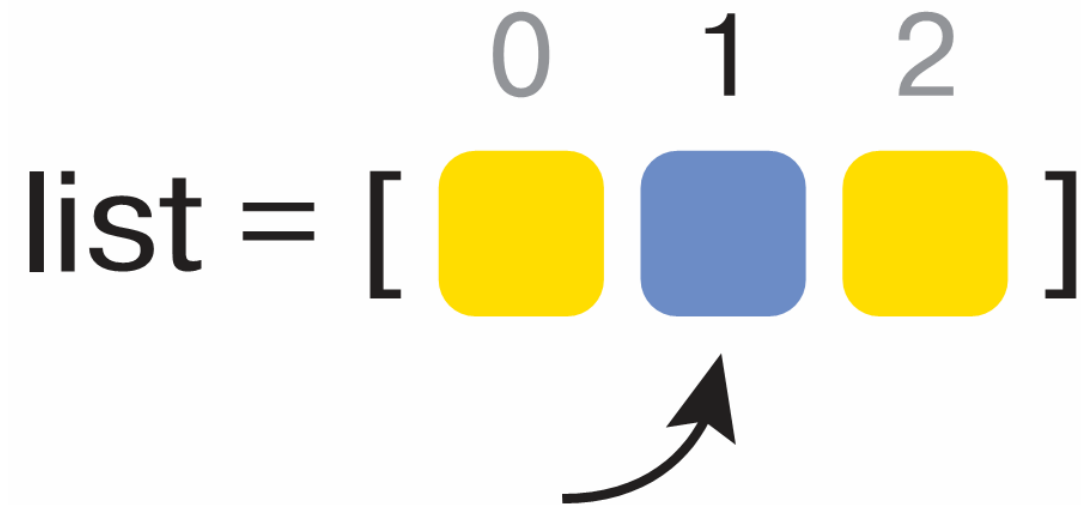
Списки є **впорядкованими** наборами даних, тому для доступу до будь-якого елемента списку слід повідомити **Python** позицію (**індекс**) потрібного елемента.



Як звернутись до окремого елемента списку?

Індекси приймають тільки цілочисельні значення.

Щоб звернутися до елемента в списку, вказують ім'я списку, за яким слідує індекс елемента в квадратних дужках. Як і для рядків, зі списку можна отримати конкретне значення, вказавши його індекс.



Якщо вказати позицію, яка міститься перед списком або після нього, буде згенеровано помилку.



Як звернутись до окремого елемента списку?

0

1

2

```
color_model = ['red', 'green', 'blue']  
print(color_model [1])
```

Результат

green



Індекс може бути як > 0 , так $i < 0$. Останній символ має номер -1 , передостанній -2 , і т. д.

Як організувати обмін даними у різних списках?

Якщо присвоїти один список більш ніж одній змінній, то зміни у списку в одному місці спричинять за собою його зміни в інших.

Фрагмент коду

Результат

```
>>> a = [1, 2, 3]
>>> a
```

```
[1, 2, 3]
```

```
>>> b = a
>>> b
```

```
[1, 2, 3]
```

```
>>> a[0] = 'new'
>>> a
```

```
['new', 2, 3]
```

```
>>> b
```

```
['new', 2, 3]
```

Як організувати обмін даними у різних списках?

Значення списку можна скопіювати в незалежний новий список за допомогою одного з таких способів:

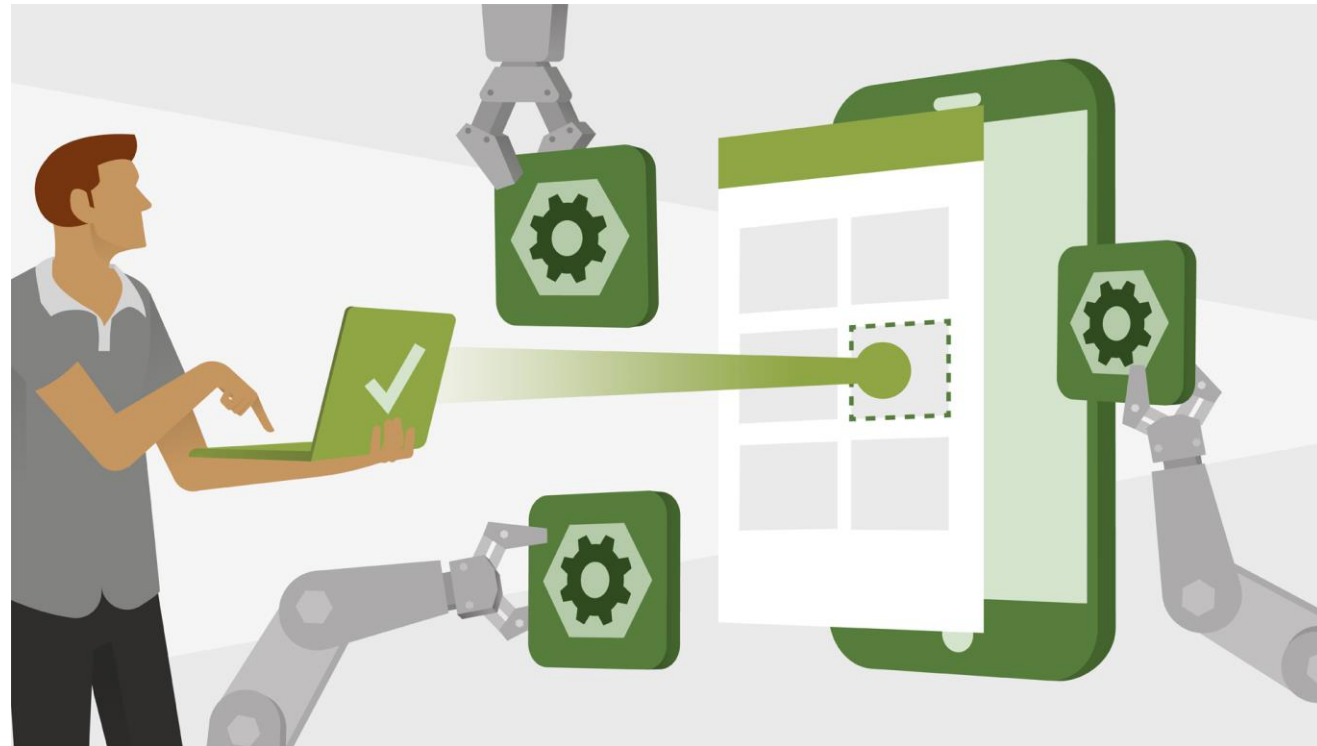
- функції **copy()**;
- функції **list()**;
- розділенням списку за допомогою **[:]**.



Як організувати обмін даними у різних списках?

Наприклад, оригінальний список буде присвоєний змінній **a**, а інші списки — **b**, **c**, **d** — будуть копіями списку **a**.

Зауважимо, що **b**, **c**, **d** — це нові об'єкти, що мають свої значення, які не пов'язані з оригінальним списком елементів **[1, 2, 3]**, на який посилається змінна **a**. Зміни в **a** не впливають на копії **b**, **c**, **d**.



Як організувати обмін даними у різних списках?

Фрагмент коду

```
>>> a = [1, 2, 3]
>>> b = a.copy()
>>> c = list(a)
>>> d = a[:]
>>> a
>>> b
>>> d
```

Результат

```
[1, 2, 3]
[1, 2, 3]
[1, 2, 3]
```

Як організувати обмін даними у різних списках?

Фрагмент коду

```
>>> a = [4, 2, 3]
>>> d = [10]
>>> a
>>> b
>>> c
>>> d
```

Результат

```
[4, 2, 3]
[1, 2, 3]
[1, 2, 3]
[10]
```

Розгадайте ребус

rebus1.com / ua



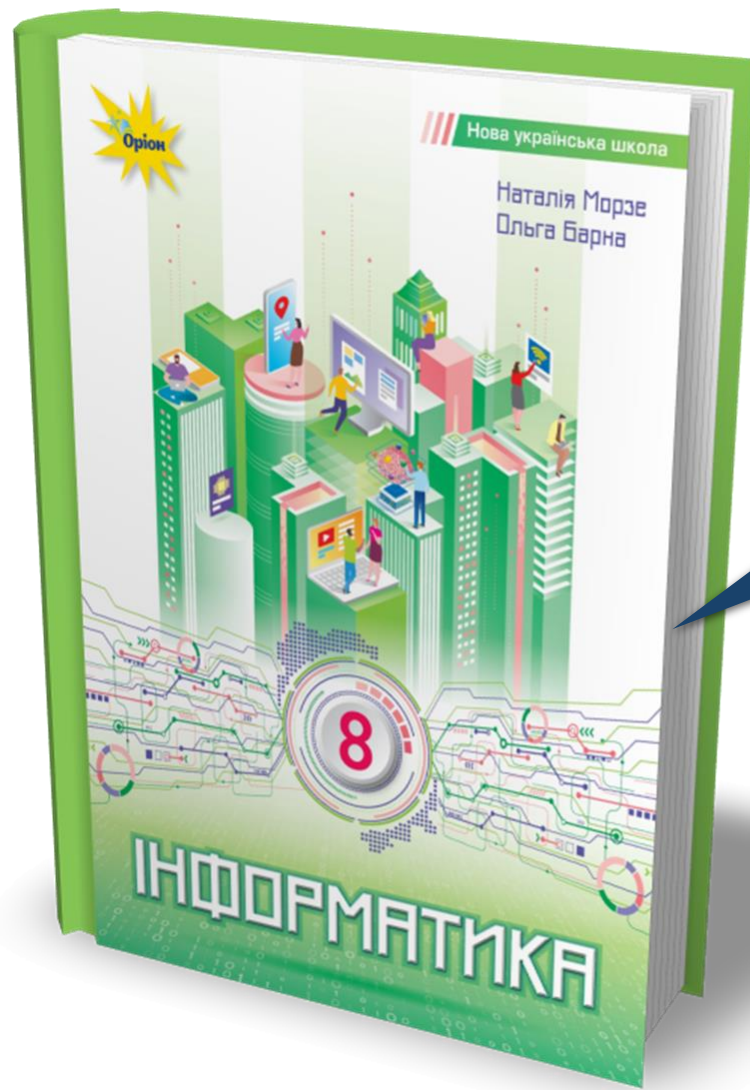
”



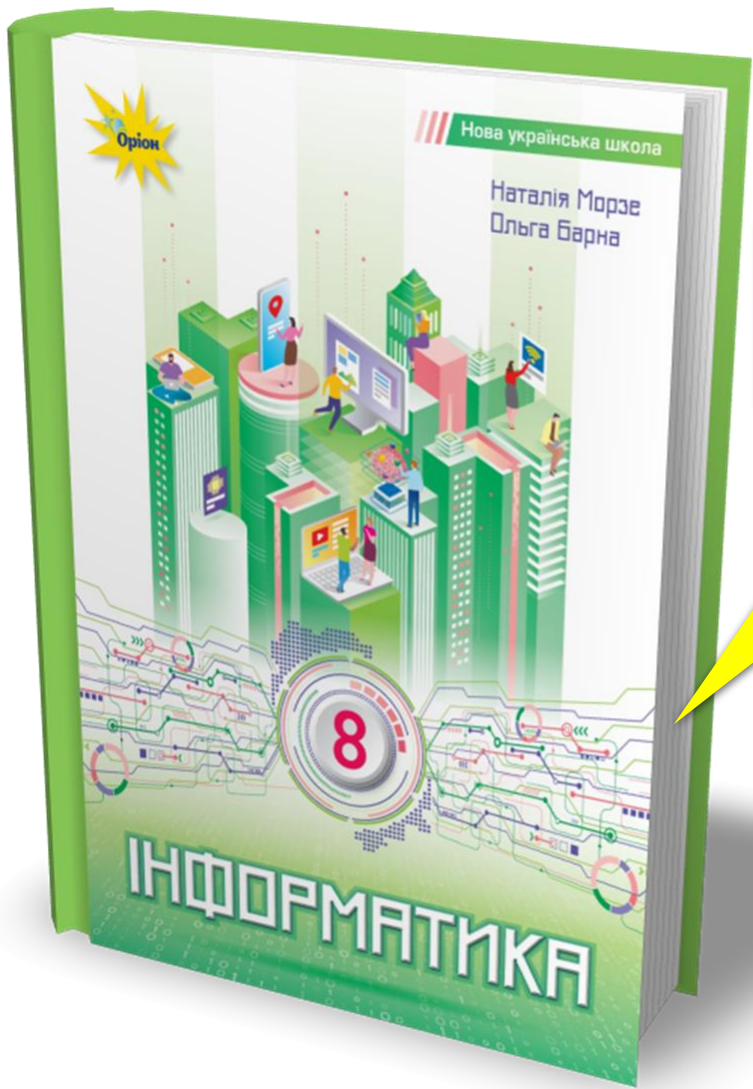
Масив



Домашнє завдання

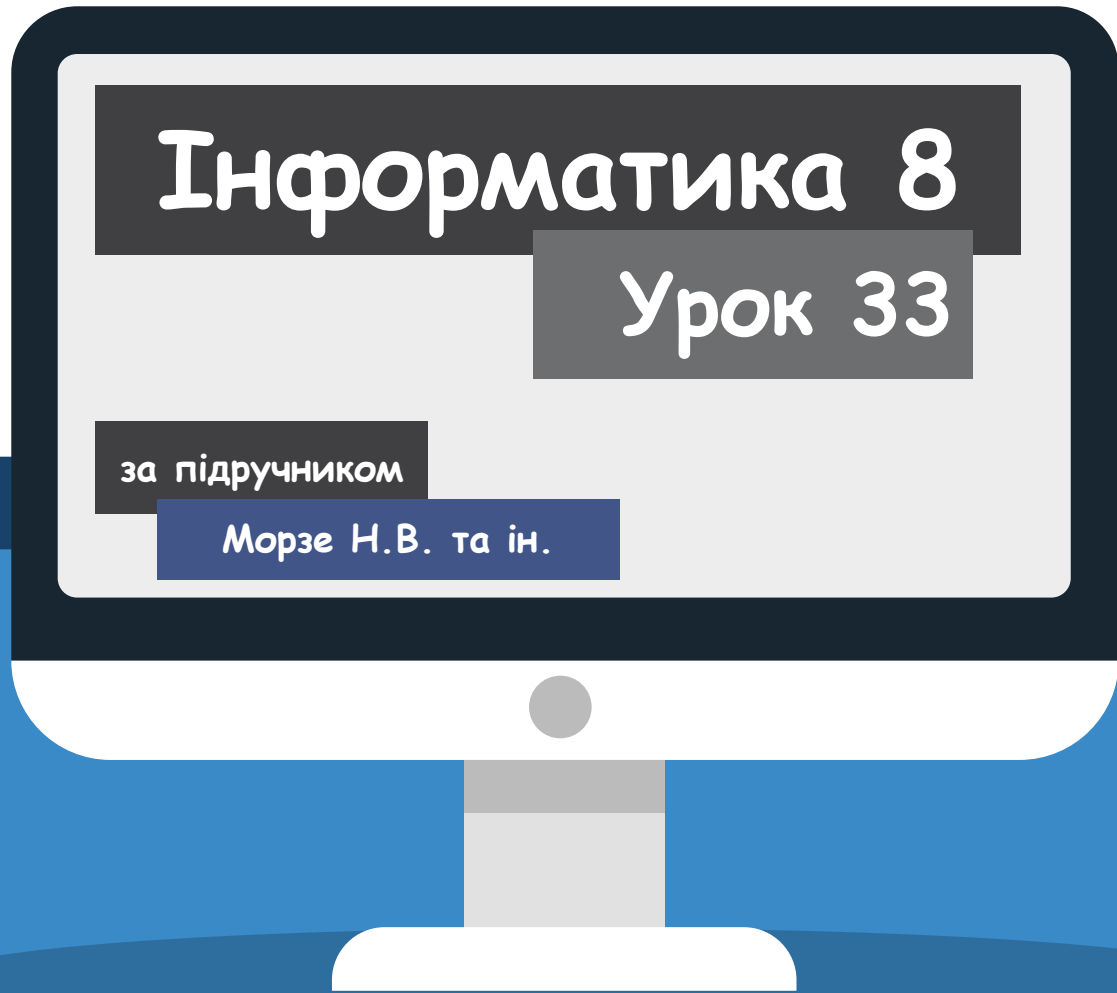


Проаналізувати
с. 110-114



**Сторінка
112-114**





Успіхів у навчанні!

Нова українська школа

