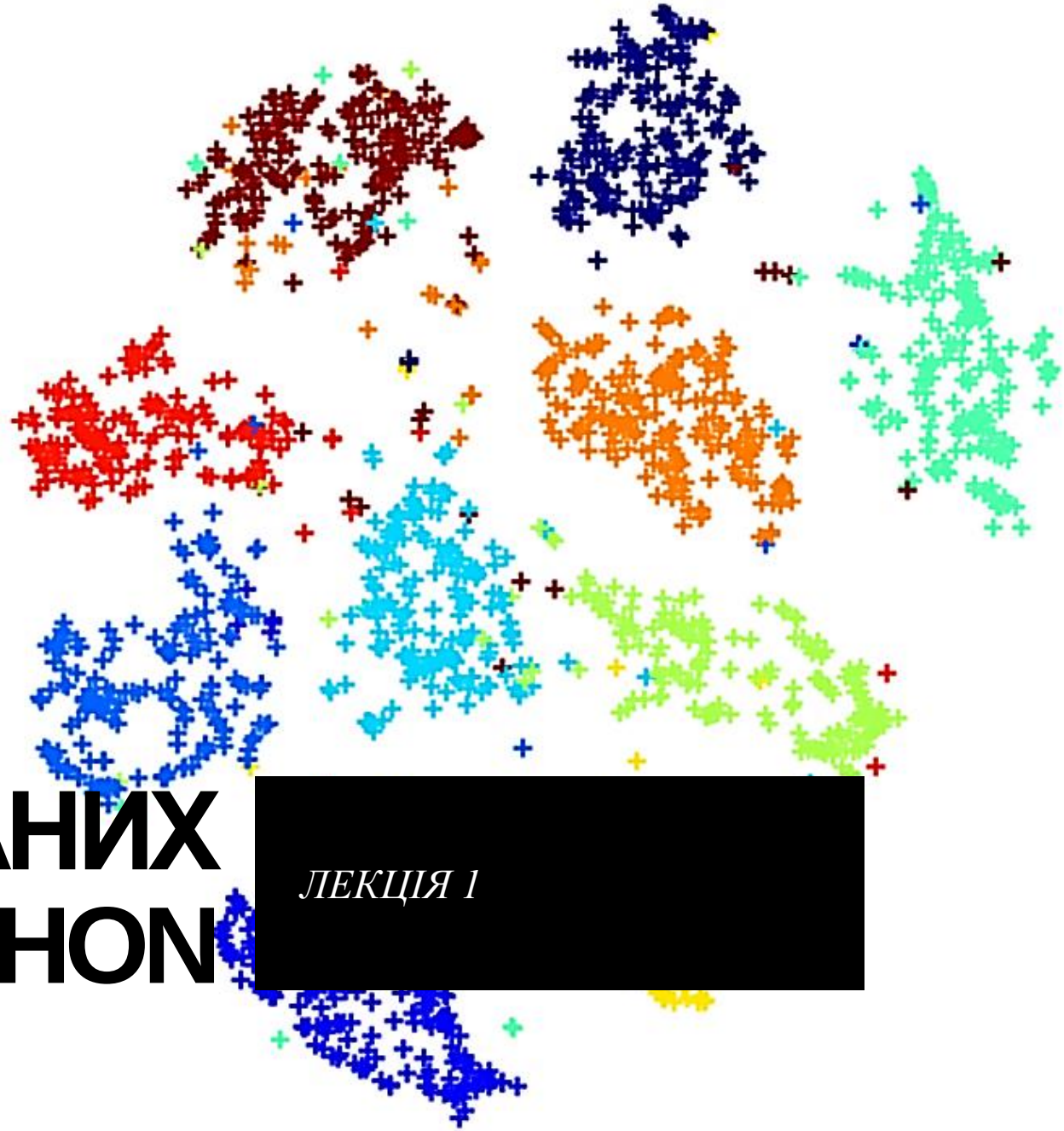
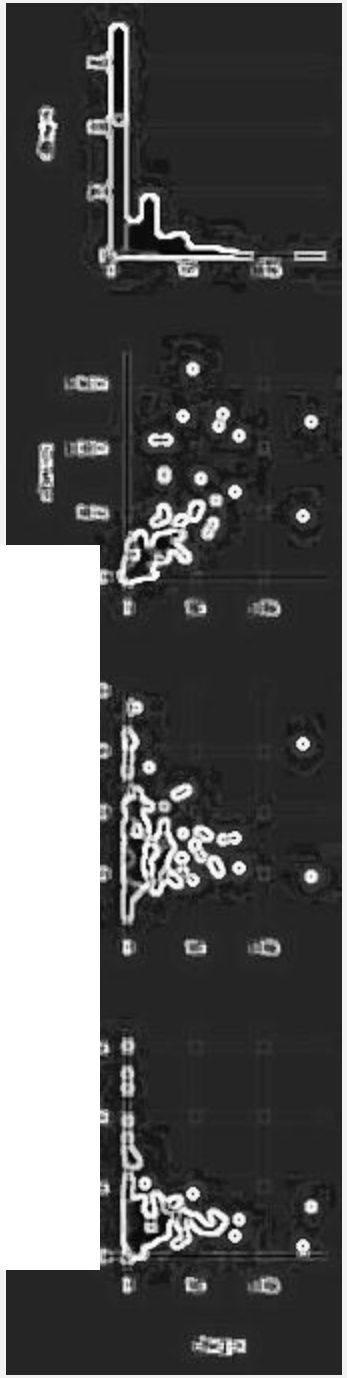
ЖИТОМИРСЬКА ПОЛІТЕХНІКА
ДЕРЖАВНИЙ УНІВЕРСИТЕТ



АНАЛІЗ ДАНИХ PYTHON

ЛЕКЦІЯ 1



ПРОЕКТ

https://www.kaggle.com/

План

- План курсу.
- Дані і обмін даними.
- Що таке аналіз даних?
- Процес аналізу даних.
- Помилки при роботі з даними.
- Типи даних.
- Чому саме Python?

ПЛАН КУРСУ

6 годин лекцій
6 годин лабораторні роботи
залік

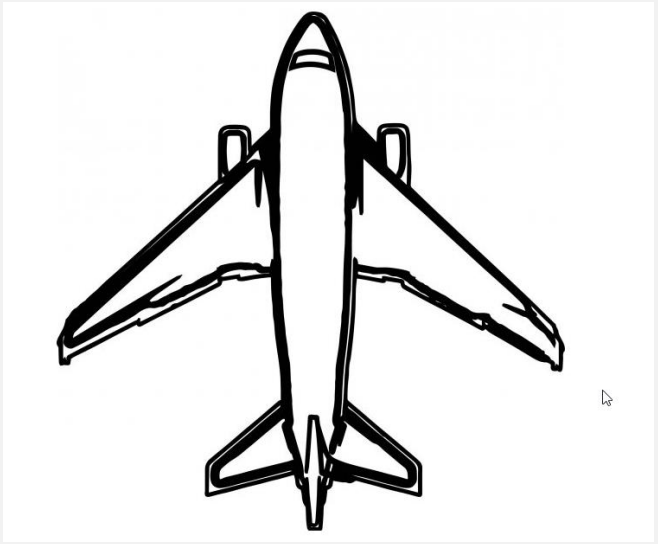
Лабораторна робота 20 (20*3=60)
Тести – 3 (10*3=30)
Сертифікат курсів - 20

МОВИ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

Портал - https://learn.ztu.edu.ua/enrol/index.php?id=2303

Безкоштовний доступ до освітньої платформи Udemy для здобувачів вищої освіти Житомирської політехніки:
https://news.ztu.edu.ua/2022/10/bezkoshtovnyj-dostup-do-osvitnih-platform-coursera-ta-udemy-dlya-zdobuvachiv-vyshhoyi-osvity-ta-vykladachiv-zhytomyrskoyi-politehniky

ДАННІ І ОБМІН ДАНИМИ



0,5 ТВ даних під час польоту Boeing 787, за сутки 100 000 перельотів в середньому

За 1 хв.

- Google - приблизно 2.5 млн. запитів
- Facebook – приблизно 700 000 тисяч користувачів заходить

надають інформацію

- Uber - найбільша в світі служба таксі, не володіє жодним авто
- Alibaba - retailer, нічого не виробляє, діяльність – торгові операції
- Airbnb - сервис короткострокової оренди житла, не володіє жодним помешканням

ДАННІ І ОБМІН ДАНИМИ

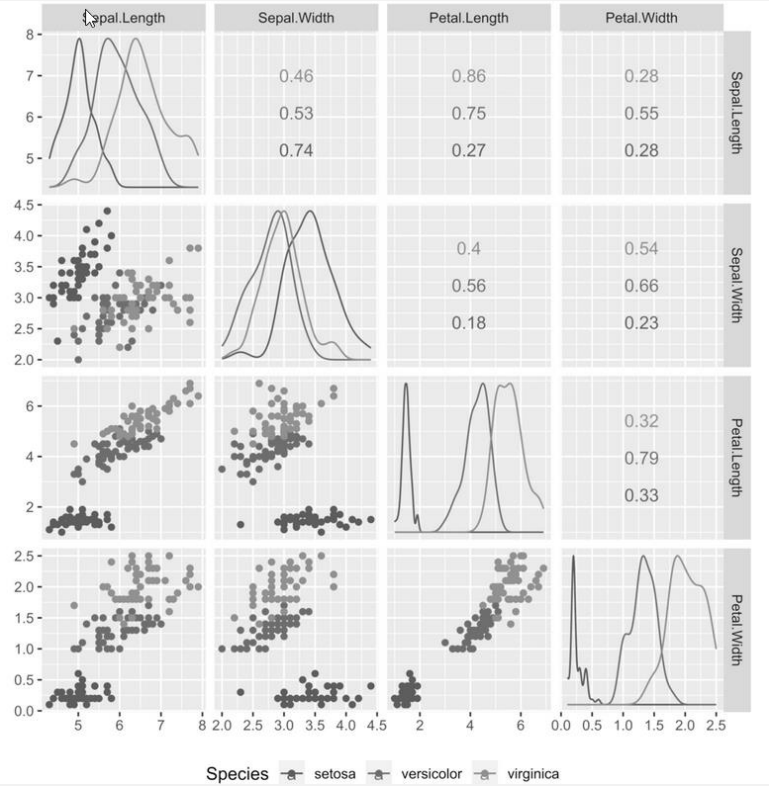
Data - Driven Company

Data - Driven Company – це підприємство, кероване даними – дата-орієнтована Agile-компанія, бізнес-процеси та організаційна структура якої побудовані на основі інформаційних потоків та їх безперервної, прогнозної аналітики

https://mode.com/blog/becoming-a-data-driven-company/#:~:text=A%20data%2Ddriven%20organization%20is,answers%20to%20important%20business%20questions.
https://www.phdata.io/blog/examples-of-data-driven-companies/

ДАННІ

Коли дані мають сенс



- оцінка ризику трансгенних продуктів;
- оцінка ризику нових вакцин;
- передбачення кількості захворювань на грип або інші захворювання по країнах, регіонах тощо;
- передбачення результатів виборів;
- голосові асистенти мобільних телефонів;
- самокеровані автомобілі тощо.

ПОНЯТТЯ «ІНФОРМАЦІЯ» І «ДАНІ»

дані

Під **даними** розуміють невпорядковані спостереження, числа, слова, звуки, зображення. Це набір дискретних, об'єктивних фактів.

Термін *дані* походить від слова data – факт.

інформація

Інформація

– це сукупність даних, впорядкована з певною метою, що додає їм сенс;

– це результат перетворення і аналізу даних;

Термін *інформація* (information) означає роз'яснення, виклад.

Перетворення і обробка даних дозволяє отримати інформацію, тобто коли дані організовані, впорядковані, згруповані, розбити по категоріям, вони стають інформацією.

ЩО ТАКЕ АНАЛІЗ ДАНИХ?

- **Аналіз даних** - розділ математики, що займається розробкою методів обробки даних незалежно від їх природи.
- **Інтелектуальний аналіз даних** це сучасна концепція аналізу даних. Інтелектуальний аналіз даних - це обробка інформації та виявлення в ній моделей і тенденцій, які допомагають приймати рішення

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ

Поняття аналізу даних з'явилось у 1978 році. Найбільшої популярності термін «аналіз даних» набув у першій половині 90-х років

Інтелектуальний аналіз даних

Інтелектуальний аналіз даних це сучасна концепція аналізу даних, яка припускає, що дані можуть бути неточними, неповними (містити пропуски), бути суперечливими, різнорідними, непрямими, і при цьому мати гігантські обсяги.

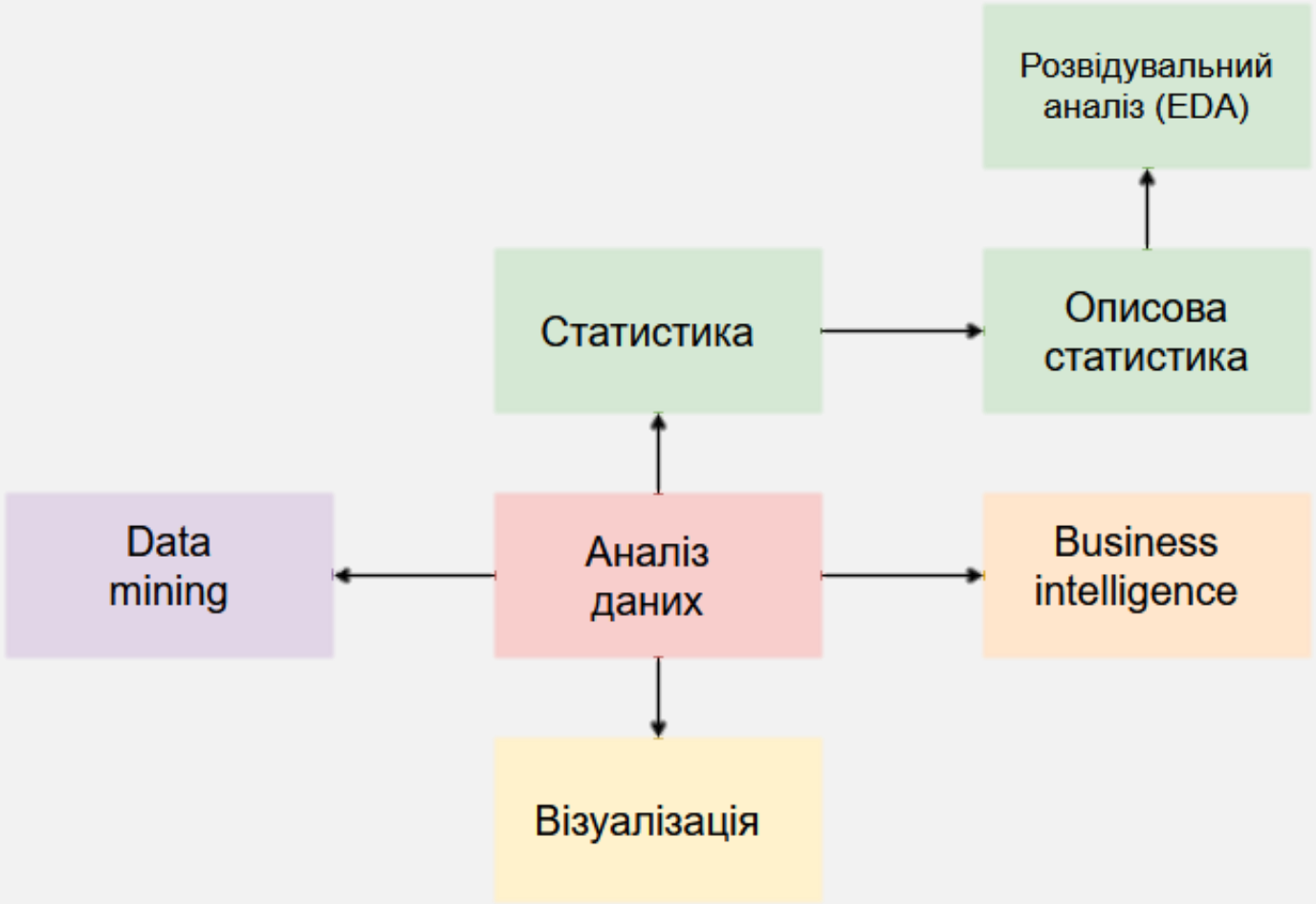
Необхідність

Необхідність інтелектуального аналізу даних виникла в результаті **поширення інформаційних технологій**, що дозволяють детально *протоколювати процеси бізнесу і виробництва*.

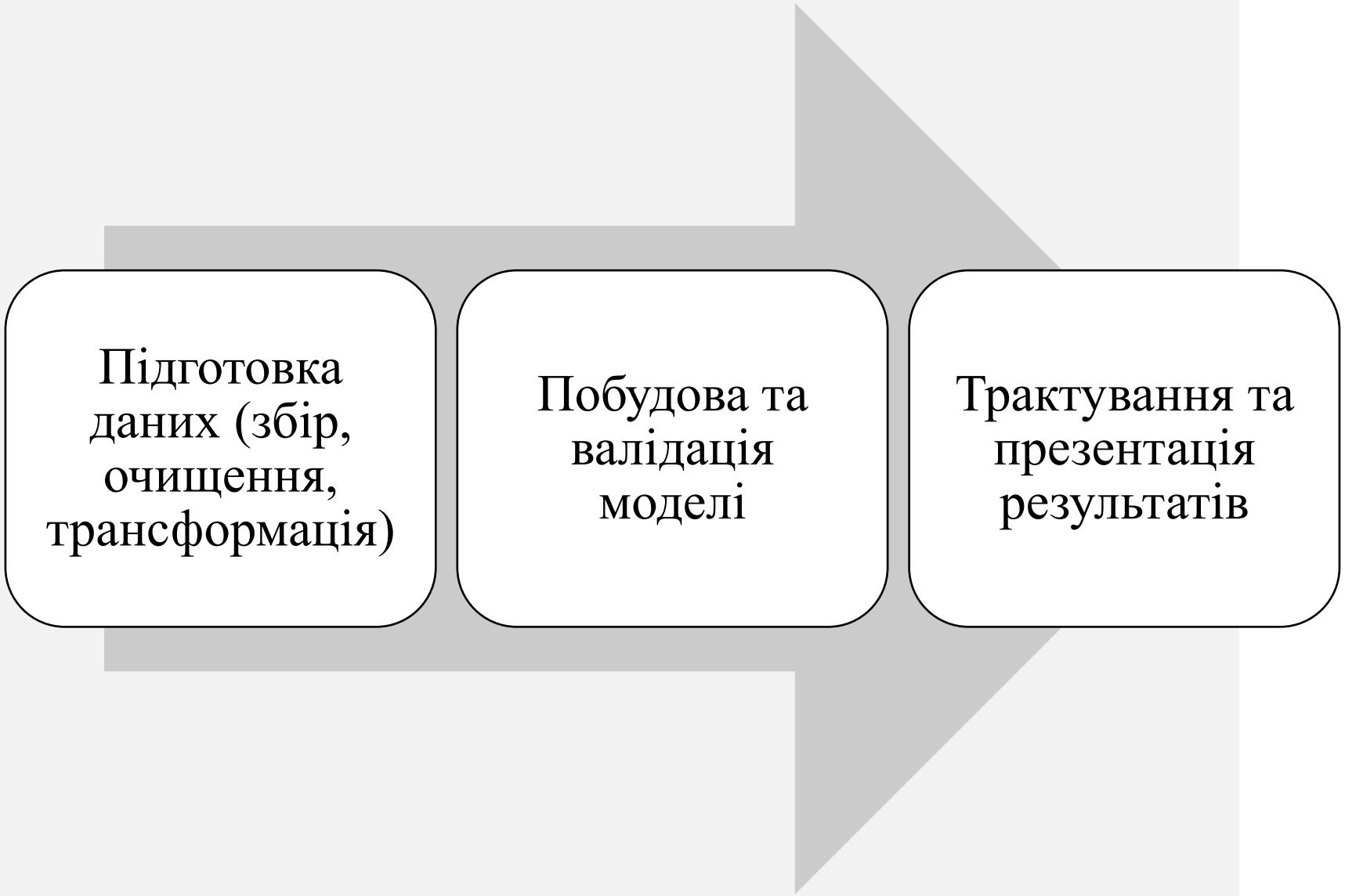
Великі обсяги даних, широта і різноманітність інформації привели до вибухового зростання популярності методів інтелектуального аналізу даних

АНАЛІЗ ДАНИХ

АНАЛІЗ ДАНИХ



ПРОЦЕС АНАЛІЗУ ДАНИХ



ТРАКТУВАННЯ ДАНИХ

Парадокс Сімпсона

Парадокс Сімпсона названо на честь дослідника Едварда Сімпсона, який у 1951 описав цей феномен. Хорошою ілюстрацією буде ситуація, що склалася в університеті Берклі в 1973.
Тоді університет звинуватили в гендерній нерівності. Для ілюстрації трохи спростимо умови. Нехай в університеті є всього два факультети: A та B.

Факультет A			
	Подано заяв	прийнято	Відсоток прийнятих
чоловіки	900	450	50
жінки	100	80	80
Факультет B			
чоловіки	100	10	10
жінки	900	180	20

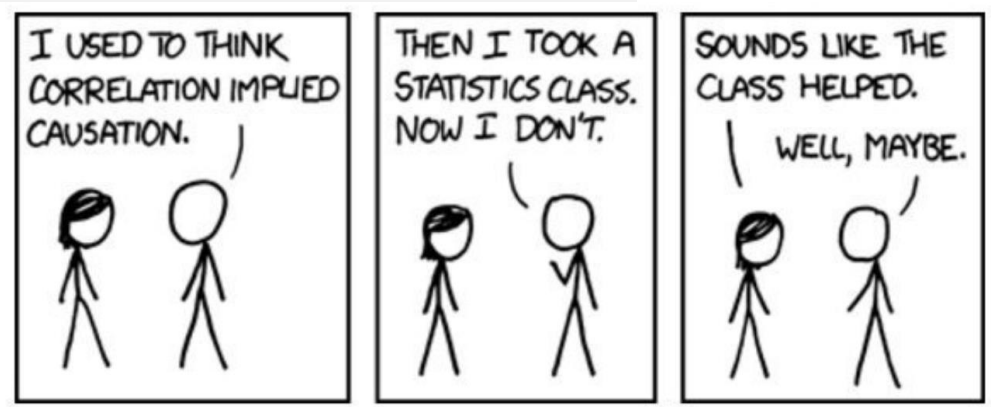
ТРАКТУВАННЯ ДАНИХ

Парадокс Сімпсона

Разом факультет A і B			
	Подано заяв	прийнято	Відсоток прийнятих
чоловіки	1000	460	46
жінки	1000	260	26

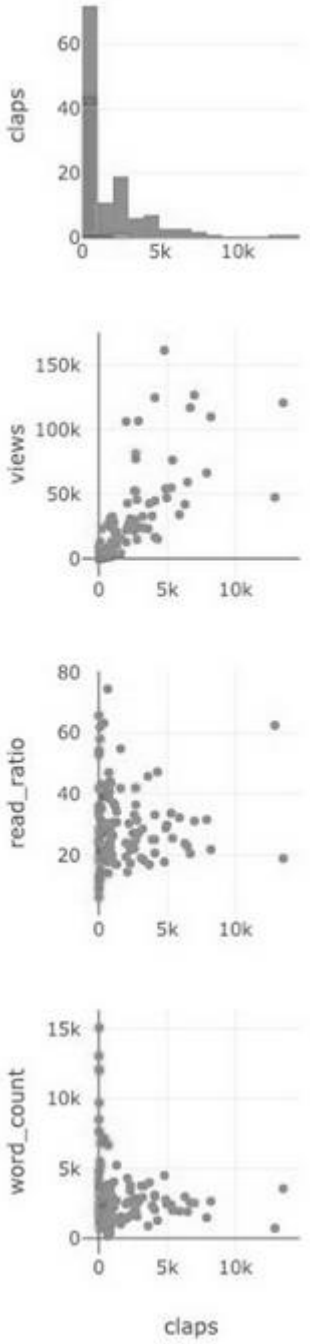
ПОМИЛКИ ПРИ РОБОТІ З ДАНИМИ

- помилки підготовки даних;
- обмежений доступ до даних у середині компанії;
- використання нерелевантних даних (неструктуровані, полу-структуровані);
- помилки вибірки;
- помилки кореляцій;
- уявна кореляція між непов'язаними факторами;
- втрата даних;
- помилки моделювання;
- неправильна інтерпретація даних;
- екстраполяція окремого випадку на загальну ситуацію;
- зайва віра в дані;
- отримати некоректні висновки через велику кількість викидів.



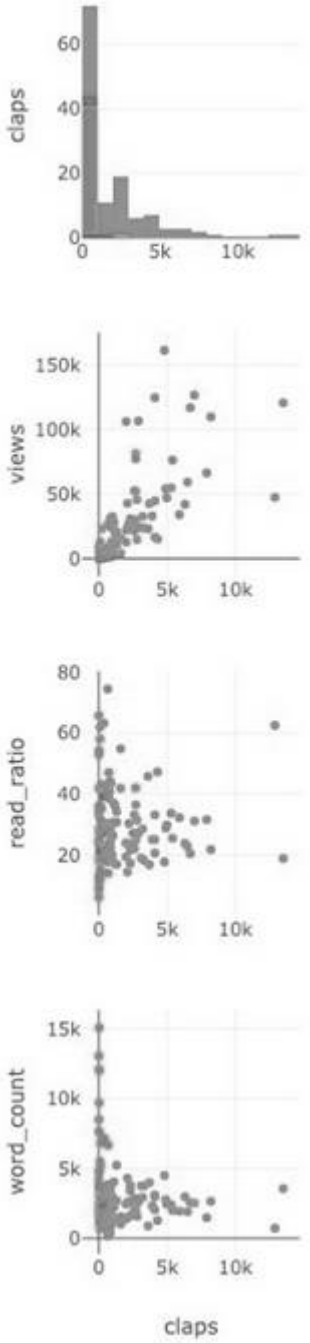
ТИПИ ДАНИХ

Кількісні	**дискретні**	кількість дітей у сім'ї
		кількість медалей олімпійської збірної
		кількість учнів
		кількість хворих
	неперервні	ріст
		вага
		заробітна плата
		швидкість
		площа



ТИПИ ДАНИХ

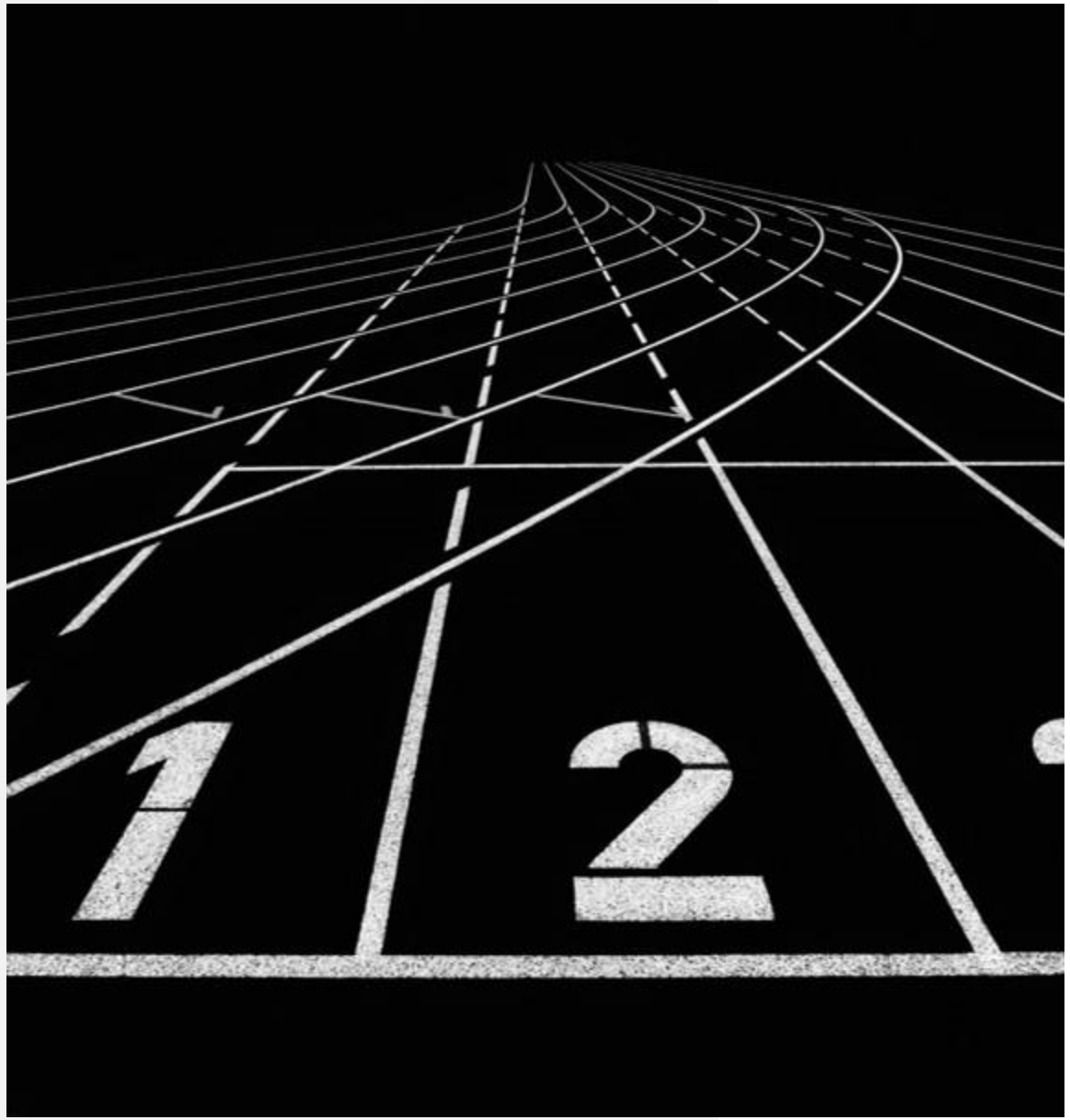
Категоріальні	**невпорядковані**	імена
		стать
		область
		назви міст
		групи крові
	впорядковані	бакалавр, магістр, доктор філософії
		погоджуюсь, не погоджуюсь, важко відповісти
	бінарні	так, ні
		0, 1
		є, немає



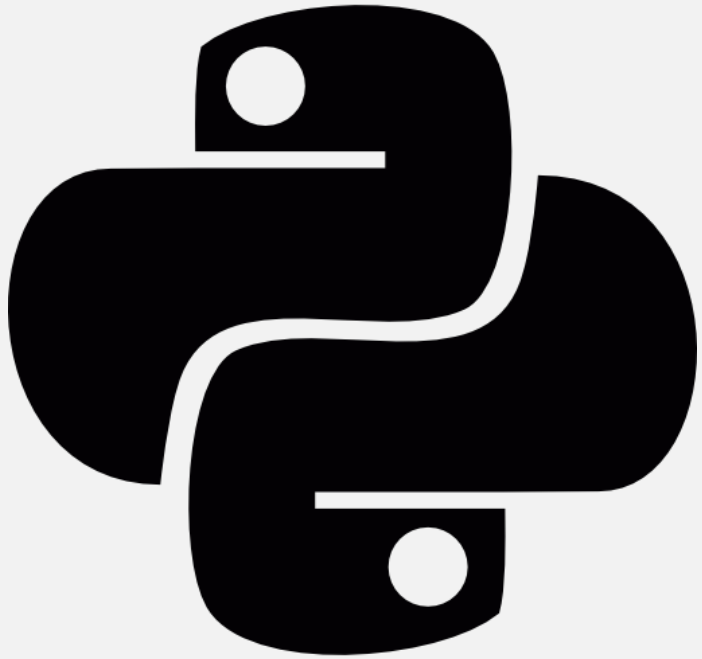
PYTHON

Чому саме **Python?**

- Простий, але виразний синтаксис.
- Багатий вибір бібліотек. І йдеться не лише про бібліотеки алгоритмів машинного навчання — на Python розробляють хмарні сховища, стрімінгові сервіси, ігри.
- Висока культура документації.



PYTHON



- Python - інтерпретована мова програмування
- Динамічна типізація
- Гарна підтримка модульності
- Підтримка об'єктно-орієнтованого програмування
- Вбудована підтримка Unicode в рядках
- Автоматичне прибирання сміття
- Інтеграція з C/C++, якщо можливостей Python недостатньо
- Зрозумілий та лаконічний синтаксис, що сприяє ясному відображенню коду
- Величезна кількість модулів, які входять в стандартну поставку Python 3
- Кросплатформеність.

PYTHON


python
Pandas
NumPy
SciPy
Scikit Learn
Matplotlib
Seaborn
Statsmodels
Keras
Plotly
TensorFlow

1. Версії
2. Google Colaboratory
3. Основи Python
    1) Типи даних
    2) Введення – виведення
    3) Умовний оператор
    4) Оператор вибору в Python (if else)
    5) Цикли
    6) Функції
    7) Рядки
    8) Списки
    9) Масиви
    10) Множини
    11) Модулі

Python

Основи мови

ВЕРСІЇ PYTHON

Що потрібно знати?

Хоча основні випуски неповністю сумісними, другорядні випуски зазвичай сумісні. Версія 3.6.1 повинна бути сумісна, наприклад, з 3.7.1. Остання цифра позначає останні виправлення та оновлення.

Python 2.7 і 3.7 - різні версії. Програмне забезпечення, написане в одній версії, часто вже не буде правильно працювати в іншій версії. При використанні Python важливо знати, яка версія потрібна і яка у вас версія.

Python 2 перестане публікувати оновлення безпеки та виправлення після 2020 року. Вони продовжили термін через велику кількості розробників, що використовують Python 2.7. Python 3 включає утиліту 2-3, яка допомагає перекладати код Python 2 в Python 3.

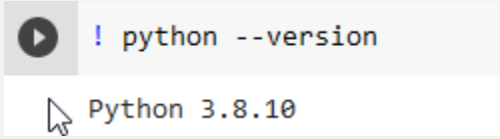
Python 1.0 - 01.1994 - > Python 2.0 — 16.10.2000 - > Python 3.0 — 3.12. 2008 - > Python 3.6 — 23.12.2016 -> Python 3.9 — 5.10.2020 …..

Коли ви дивитеся на номер версії, зазвичай потрібно читати три цифри:

основна версія. другорядна версія. мікро версія

GOOGLE COLABORATORY

Що б перевірити версію в Colaboratory необхідно використовувати команду:



Google Colaboratory - це дослідницький інструмент для навчання та досліджень в області машинного навчання. Це середовище ноутбука Jupyter, для використання якого не потрібні налаштування.

1 - Він забезпечує безкоштовну обчислювальну потужність

2 - Налаштування не потрібні

3 - Простота обміну і спільної роботи

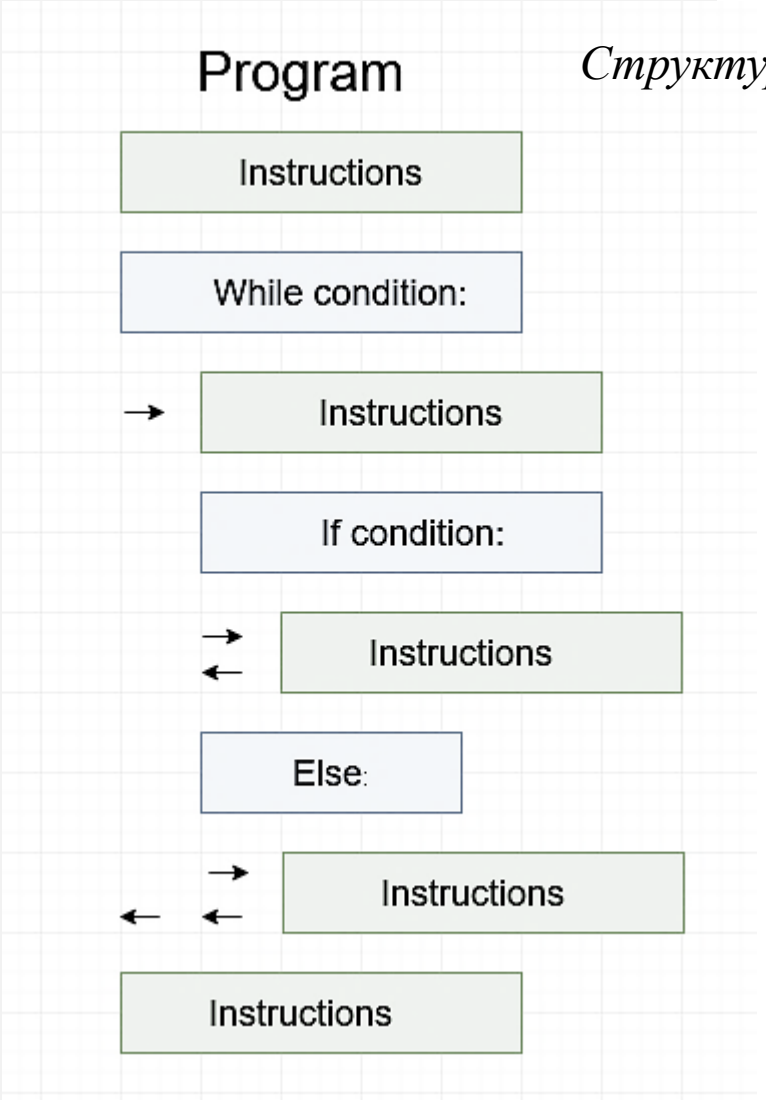
4 - Підтримує Python 2.7 і Python 3.6

5- Інтегрований з GitHub

ОСНОВИ PYTHON

Не містить операторних дужок (begin..end в pascal або {..} в Ci), замість цього блоки виділяються відступами: пробілами або табуляцією, а вхід в блок з операторів здійснюється двокрапкою.

Однорядкові коментарі починаються зі знака «#», багаторядкові - починаються і закінчуються трьома подвійними лапками «”””” …. “””””».



Структура програми

ТИПИ ТА СТРУКТУРИ ДАНИХ

В Python реалізовані вбудовані типи:

булевий тип;

рядок;

ціле число довільної точності;

число з плаваючою комою;

комплексне число.

Також є і готові колекції:

списки (lists);

кортежі (tuples)(незмінні списки);

словники (dictionaries);

множини.

ВВЕДЕННЯ-ВИВЕДЕННЯ В PYTHON

Введення даних здійснюють з допомогою функції **input().**

Функція input() повертає *текстовий рядок*. Для зчитування цілого числа потрібно виконати перетворення типу за допомогою функції int():

```
x=input()
a=int(x)
```

або

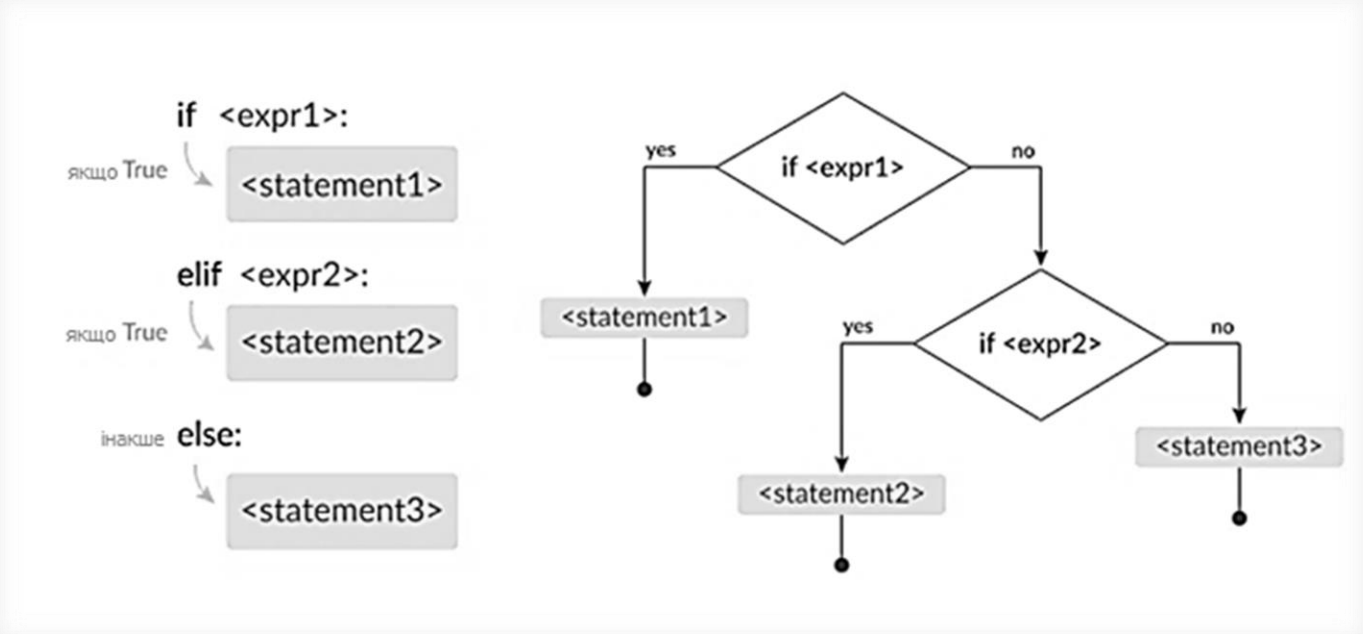
```
a=int(input(x))
```

Для зчитування кількох значень з одного рядка використовують метод **split()**

Виведення даних здійснюють з допомогою функції **print().**

```
print(2+2**2)
```

ОПЕРАТОР ВИБОРУ В PYTHON (if else)




```
x, y = 8, 2
if (x<y):
  print('x<y')
elif (x>y):
  print('x>y')
else:
  print('x=y')
```

```
x>y
```

КОНСТРУКЦІЯ SWITCH CASE В PYTHON

В Python немає простої конструкції switch-case.

У наведеному прикладі, в залежності від значення змінної argument, в стандартному виведення буде відображатися повідомлення. У випадку, коли argument =0, буде надруковано «Sunday».

Можна використовувати словник:

```
[91] def f(x):
         return {
             'a': 1,
             'b': 2,
         }[x]
     f('b')
```

```
[93] def get_week_day(argument):
       switcher ={
            0: "Sunday",
            1: "Monday",
            2: "Tuesday",
            3: "Wednesday",
            4: "Thursday",
            5: "Friday",
            6: "Saturday"
       }
       return switcher.get(argument, "Invalid day")

     print (get_week_day(6))
     print (get_week_day(8))
     print (get_week_day(0))
```

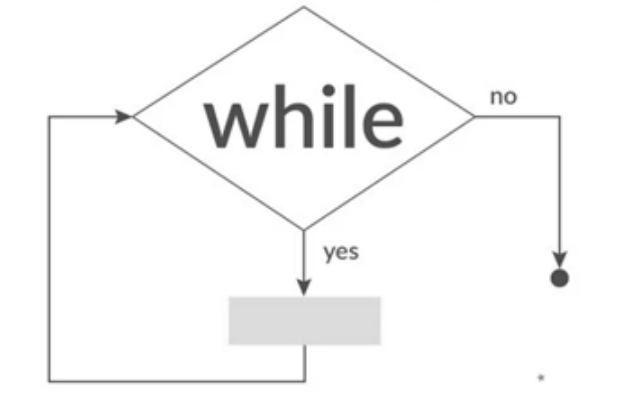
КОНСТРУКЦІЯ SWITCH CASE В PYTHON

Можна використовувати лямбда:

```
[60] import math

     print ("Please enter an integer:\n")
     x = int(input())
     print("Choices:\n 1 - Square\n 2 - Cube\n 3 - Square Root")
     print ("Please enter your choice:\n")
     choice = int(input())
     def switch_func(value, i):
         return {
             1: lambda val: math.pow(val, 2),
             2: lambda val: math.pow(val, 3),
             3: lambda val: math.sqrt(val),
         }.get(value)(i)
     print(switch_func(choice, x))
```

ЦИКЛИ В PYTHON

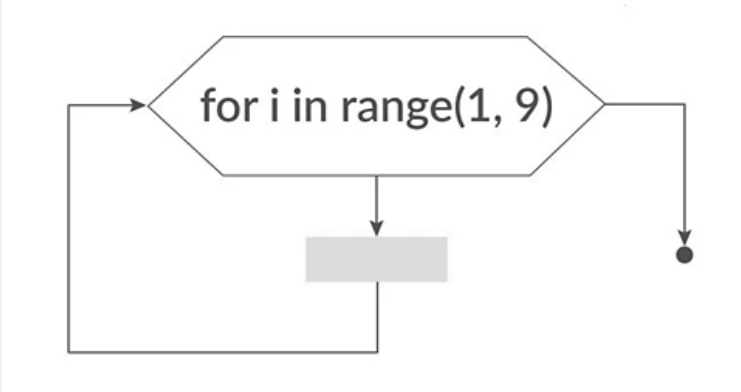



```
[16] i = 1
     while i < 6:
       print(i)
       i += 1
```

```
[17] i = 1
     while i < 6:
       print(i)
       if i == 3:
         break
       i += 1
```

```
[18] i = 0
     while i < 6:
       i += 1
       if i == 3:
         continue
       print(i)
```

ЦИКЛИ В PYTHON




```
[19] fruits = ["apple", "banana", "cherry"]
     for x in fruits:
       print(x)
```

```
[21] for x in "banana":
       print(x)
```

```
[28] for x in range(7):
       print(x)
```

```
[31] s=0
     for x in range(1,5):
       s+=x
     print(s)
```

Функція **range ()** повертає послідовність чисел, що починається з 0 за замовчуванням, зі збільшенням на 1 (за замовчуванням) і закінчується зазначеним числом.

```
adj = ["red", "big", "tasty"]
fruits = ["apple", "banana", "cherry"]

for x in adj:
  for y in fruits:
    print(x, y)
```

ФУНКЦІЇ В PYTHON

Функція - це блок коду, який запускається тільки при виклику.
В Python функція визначається за допомогою ключового слова **def:**

```
[36] def my_function():
       print("Hello from a function")

     my_function()
```

```
[10] def main():
       x,y = 1,8
       st = "x is less than y" if (x < y) else "x is greater than or equal to y"
       print(st)

     if __name__ == "__main__":
       main()
```

САМОСТІЙНА РОБОТА

Лекція_1

Colaboratory? https://colab.research.google.com/