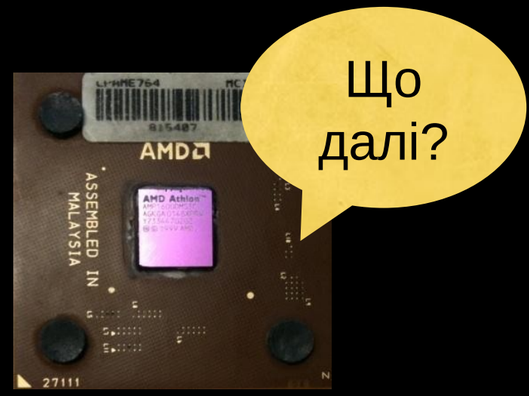
Навіщо програмувати?

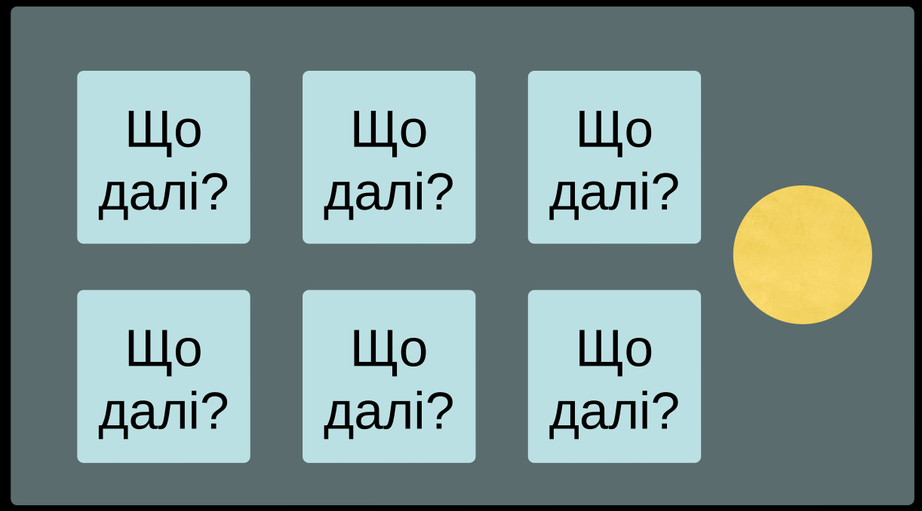
Розділ 1

Python для всіх
www.py4e.com

Комп’ютери хочуть бути корисними……

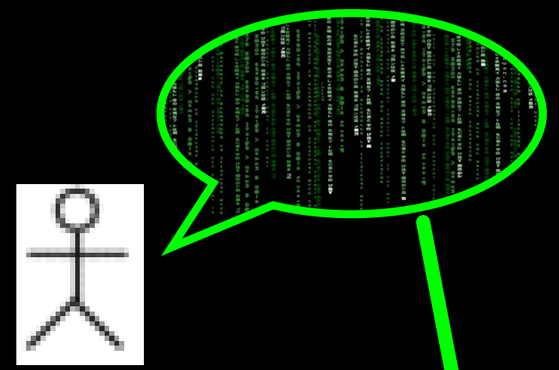
- Комп’ютери розроблені, щоб допомагати нам
- Але нам треба знати їхню мову, тоді ми зможемо пояснити, що треба зробити
- Користувачам простіше - хтось вже надав комп’ютеру інструкції (застосунки, програми), тож нам треба лише обрати, яку застосовувати





Програмісти прогнозують потреби

- Застосунки для iPhone – це ринок
- Застосунки для iPhone завантажували понад 3 мільярди разів
- Програмісти звільняються з робіт, аби працювати розробниками мобільних застосунків
- Програмісти знають, як працюють застосунки

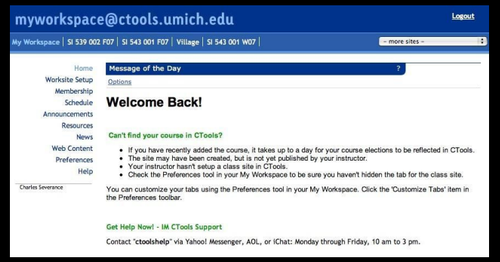


Користувачі проти Програмістів

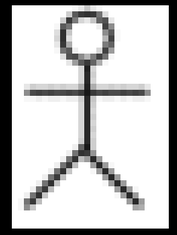
- Користувачі сприймають комп’ютери як перелік інструментів – текстовий редактор, таблиця, мапа, список справ тощо
- Програмісти досліджують як комп’ютер працює та його мову
- Програмісти мають інструменти, які дозволяють їм створювати нові інструменти
- Часом програмісти створюють купу інструментів для користувачів, й іноді - маленьких «помічників» для автоматизації власних задач

Для чого ставати програмістом?

- Аби зробити щось – ми можемо бути й користувачами, і програмістами
  - Видалити дані опитування
- Щоб створити придатне для застосування іншими – тільки професійними програмістам
  - Вирішити проблему з продуктивністю програмного забезпечення Сакаі
  - Додати гостьову книгу на вебсайт



Користувач



Комп’ютер

Апаратне забезпечення + Програмне забезпечення

Програміст

Дані

Інформація

....

Мережі

З точки зору розробника програмного забезпечення ми будуємо ПЗ. Кінцеві користувачі (зацікавлені особи / виконавці) – ті, кого нам потрібно задовольнити – часто платять кошти, коли хочуть, що ми щось для них зробили. Але ми маємо використовувати дані, інформацію, мережі, щоб виконати дії користувача. Апаратне і програмне забезпечення – наші друзі та союзники у цьому квесті.

Що таке код? Програмне забезпечення? Програма?

- Послідовний набір інструкцій
  - Це часточка нашого інтелекту в комп’ютері
  - Ми щось визначаємо, потім кодуємо це, і потім передаємо це комусь іншому, аби зберегти їхній час і енергію на з’ясування цього

- Трохи креативності – особливо, коли ми ретельно враховуємо поведінку користувача (UI/UX)

Програми для людей...

https://www.youtube.com/watch?v=XiBYM6g8Tck

Програми для людей...

Поки грає музика:

Ліву руку витягнути і підняти вгору
Праву руку витягнути і підняти вгору
Перекид лівою рукою
Переворот правою рукою
Ліву руку до правого плеча
Праву річку до лівого плеча
Ліва рука на потилицю
Удар правою рукою по потилиці
Удар лівою рукою вправо
Удар правою рукою в лівий бік
Ліва рука на лівому стерні
Правою рукою по правому стерну
Похитуйся.
Похитуйся.
Стрибок.

https://www.youtube.com/watch?v=XiBYM6g8Tck

Програми для людей...

Поки грає музика:

Ліву руку витягнути і підняти вгору
Праву руку витягнути і підняти вгору
Перекид лівою рукою
Переворот правою рукою
Ліву руку до правого плеча
Праву річку до лівого плеча
Ліва рука на потилицю
Удар правою рукою по потилиці
Удар лівою рукою вправо
Удар правою рукою в лівий бік
Ліва рука на лівому стерні
Правою рукою по правому стерну
Похитуйся.
Похитуйся.
Стрибок.

https://www.youtube.com/watch?v=XiBYM6g8Tck

Програми для людей...

Поки грає музика:

Ліву руку витягнути і підняти вгору
Праву руку витягнути і підняти вгору
Перекид лівою рукою
Переворот правою рукою
Ліву руку до правого плеча
Праву руку до лівого плеча
Ліва рука на потилицю
Удар правою рукою по потилиці
Удар лівою рукою вправо
Удар правою рукою в лівий бік
Ліва рука на лівому стегні
Правою рукою по правому стегну
Похитуйся.
Похитуйся.
Стрибок.

https://www.youtube.com/watch?v=XiBYM6g8Tck

Програми для Python...

Image: https://www.flickr.com/photos/allan_harris/4908070612/ Attribution-NoDerivs 2.0 Generic (CC BY-ND 2.0)

Програми для Python...

the clown ran after the car and the car ran into the tent and
the tent fell down on the clown and the car

```
name = input('Enter file:')
handle = open(name)

counts = dict()
for line in handle:
    words = line.split()
    for word in words:
        counts[word] = counts.get(word,0) + 1

bigcount = None
bigword = None
for word,count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count

print(bigword, bigcount)
```

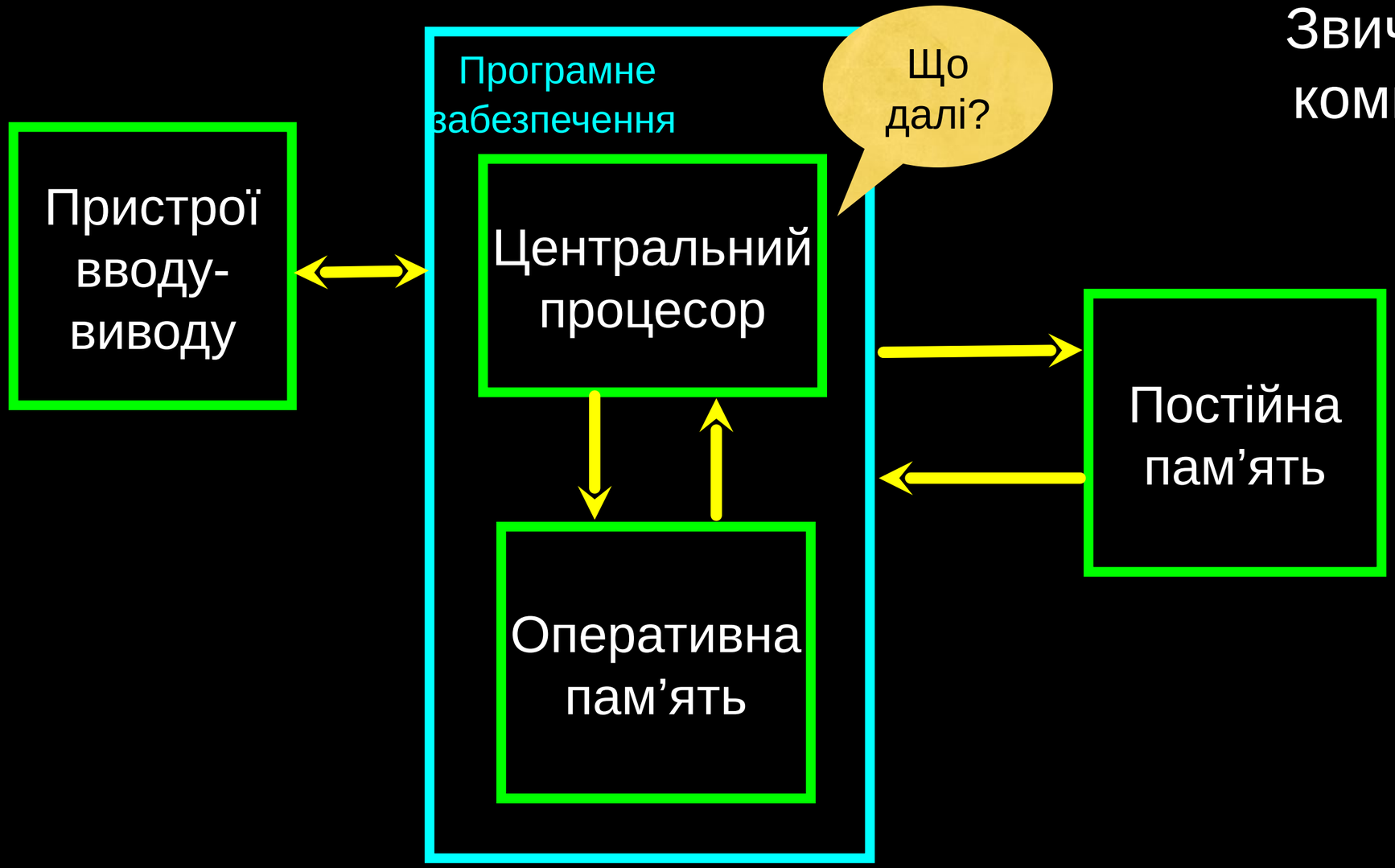
python words.py
Вхідний файл: words.txt
to 16

python words.py
вхідний файл: clown.txt
the 7

Архітектура (будова) апаратного забезпечення

http://upload.wikimedia.org/wikipedia/commons/3/3d/RaspberryPi.jpg

Звичайний комп’ютер



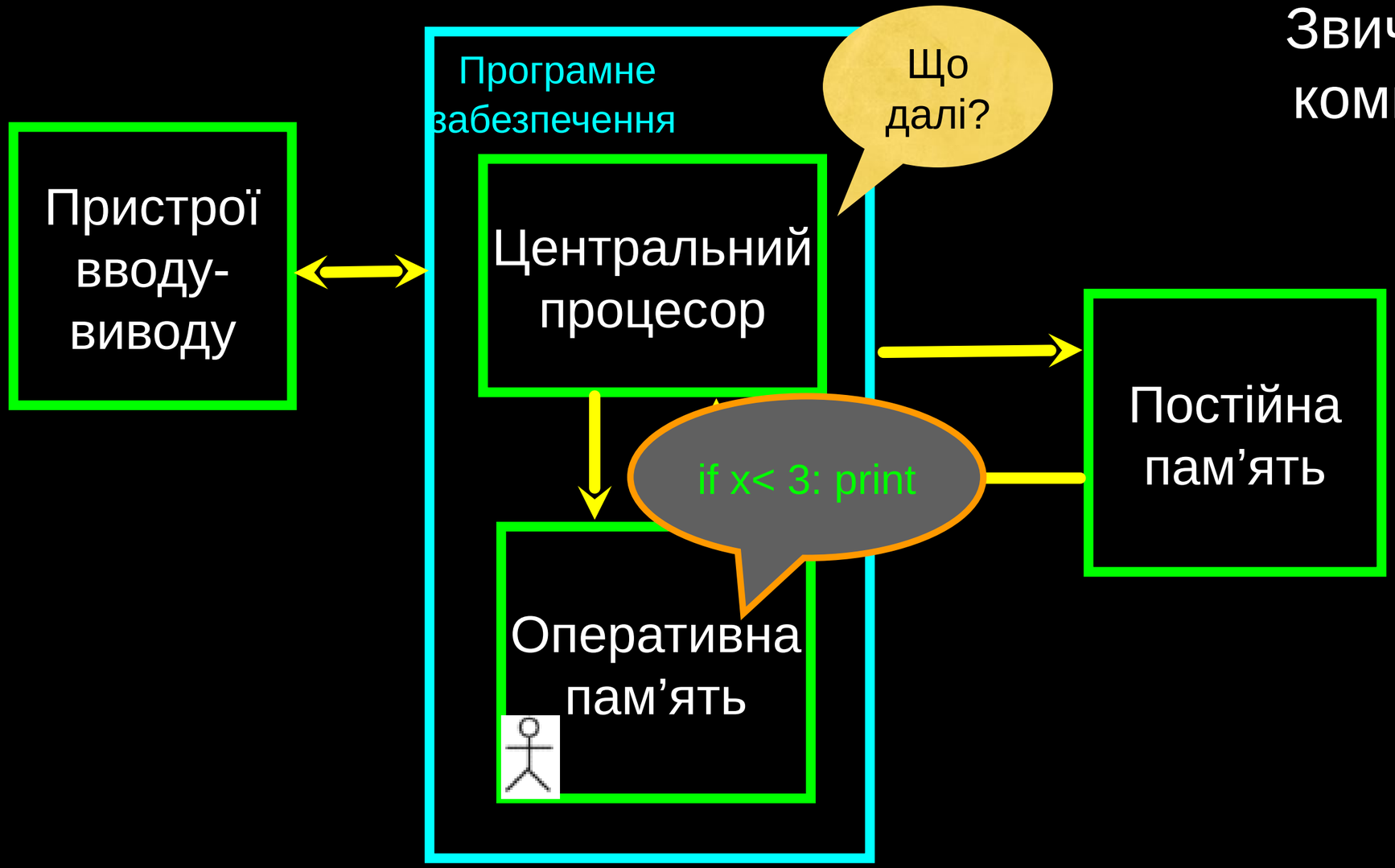
Визначення понять

- Центральний процесор: запускає програми – Процесор завжди ставить собі питання «Що робити далі?». Не зовсім мозок – дуже тупий, але швидкий

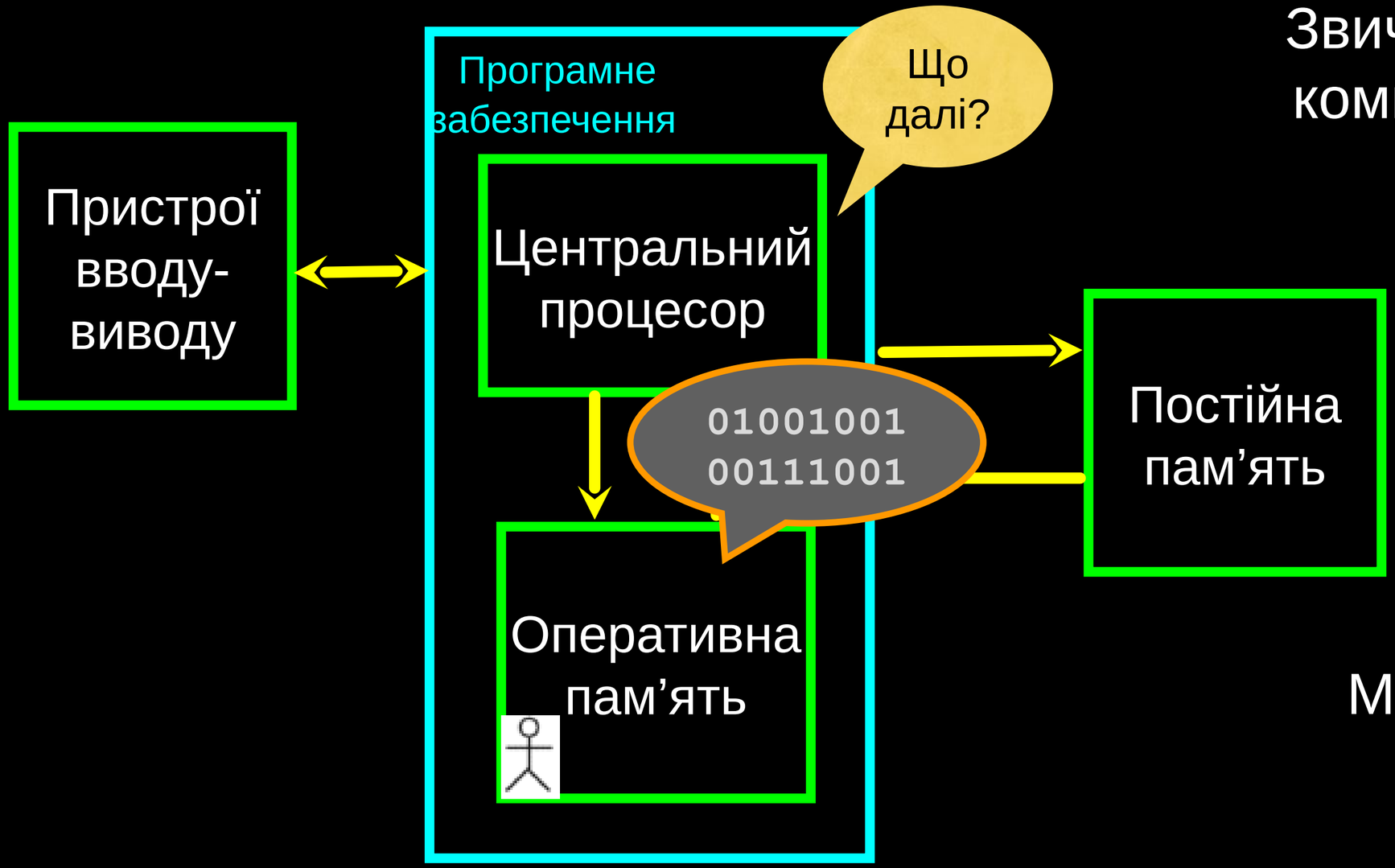
Що далі?

- Пристрої вводу: клавіатура, мишка, сенсорний екран
- Пристрої виводу: монітор, екран, динаміки, принтер, дисковод
- Оперативна пам’ять: швидке, невелике сховище даних, які втрачаються при перезавантаженні ПК, також відома як пам'ять з довільним доступом
- Постійна пам’ять: повільна, масштабна, постійна пам’ять, де дані зберігаються аж до видалення – диски, флешки, карти пам’яті

Звичайний комп’ютер

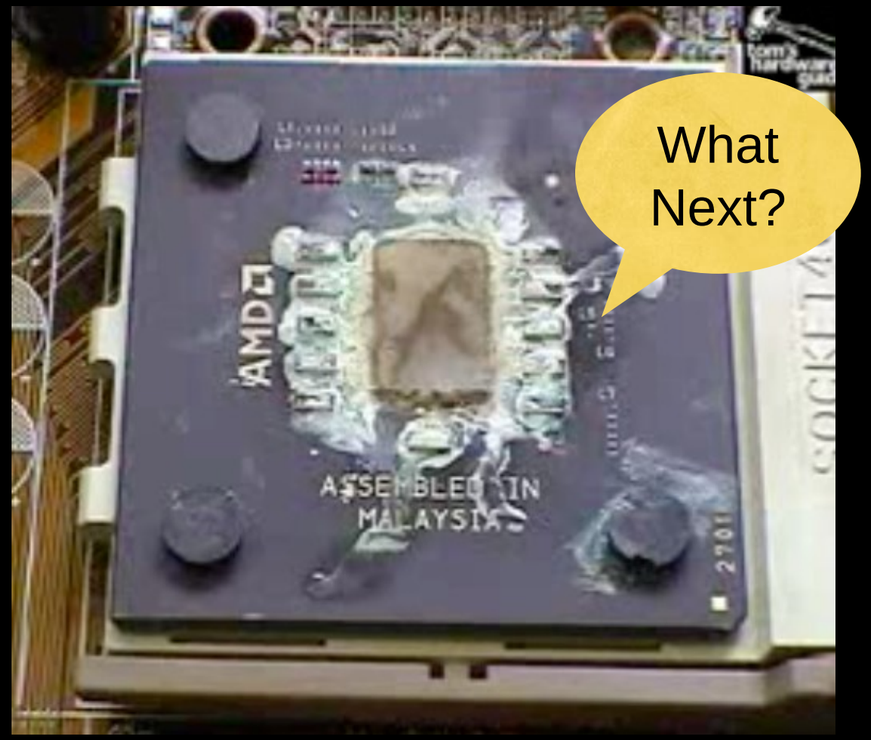


Звичайний комп’ютер



Машинна мова

Дуже гарячий центральний процесор



http://www.youtube.com/watch?v=y39D4529FM4

Робота жорстких дисків

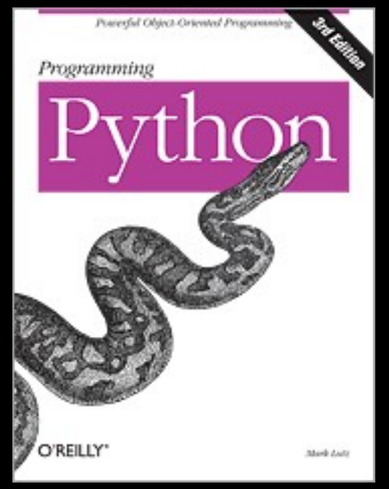
http://www.youtube.com/watch?v=9eMWG3fwiEU

Python як мова

Парселтон це мова змій і тих, хто вміє з ними говорити. Людей, які розмовляють парселтонською, називають парселентомовними. Це дуже незвичне вміння, і можливо, воно передається у спадок. Майже всім відомо, що парселтонмовці – нащадки Салазара Слизерина.

http://harrypotter.wikia.com/wiki/Parseltongue

Python – мова інтерпретатора Пайтон і тих, хто її опанував. Людей, що можуть спілкуватися мовою, називають пайтоністами. Це також дуже незвичне вміння, що може передаватися у спадок. Переважна більшість Пайтоністів використовує програмне забезпечення, створене Ґвідо ван Россумом.



Початківцям: Синтаксичні помилки

- Щоб пояснити комп’ютерові задачу мовою Python, необхідно вивчити її. На початку ми можемо багато помилятися та спілкуватися тарабарщиною як маленькі діти.
- Коли ви помиляєтеся, Python почувається розгубленим. Він каже «синтаксична помилка», тобто що він розуміє, що ви лише вивчаєте мову. Однак може здаватися, що Python злий, і не розуміє вас.
- Пам’ятайте: ви розумні, ви здатні навчатися. Комп'ютери дуже прості й швидкі, але не вміють навчатися. Це означає, що вам легше вивчити Python, ніж комп’ютеру вивчити українську.

Поговоримо з Python

```
csev$ python3
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  5 2015, 21:12:44)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwinType
"help", "copyright", "credits" or "license" for more information.
>>>
```

Що далі?

```
csev$ python3
Python 3.5.1 (v3.5.1:37a07cee5969, Dec  5 2015, 21:12:44)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwinType
"help", "copyright", "credits" or "license" for more information.
>>> x = 1
>>> print(x)
1
>>> x = x + 1
>>> print(x)
2
>>> exit()
```

Це хороший тест, аби впевнитися, що ви правильно встановили Python. Однак зауважте, що quit() також можна використовувати для завершення інтерактивної сесії.

Що сказати?

Елементи мови Python

- Словник / Слова – змінні та ключові слова (Розділ 2)
- Структура речень – правильні синтаксичні шаблони (Розділи 3-5)
- Скрипт / сценарій – набір інструкцій для створення певної програми

```
name = input('Enter file:')
handle = open(name)

counts = dict()
for line in handle:
    words = line.split()
    for word in words:
        counts[word] = counts.get(word,0) + 1

bigcount = None
bigword = None
for word,count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count

print(bigword, bigcount)
```

Коротка «історія» про те, як рахувати слова у Python

python words.py
Вхідний файл: words.txt
to 16

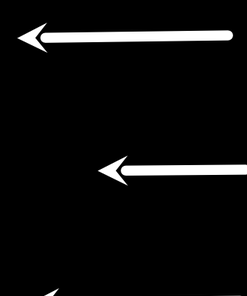
Зарезервовані (ключові) слова

Ви не можете використовувати ключові (зарезервовані) слова як назви змінних / ідентифікатори

```
False   class   return  is      finally
None    if      for     lambda  continue
True    def     from    while   nonlocal
and     del     global  not     with
as      elif    try     or      yield
assert  else    import  pass
break   except  in      raise
```

Речення чи рядки коду

```
x = 2
x = x + 2
print(x)
```



Інструкція присвоювання

Присвоювання з виразом

Функція «принт»

Змінна Оператор Константа Функція

Частини програмування

Скрипти Python

- Інтерактивна консоль найкраще за все підходить для експериментів та програм у 3-4 рядки
- Більшість програм набагато довші, тож їх друкують у файлі й наказують Python запускати команди з цього файлу
- Ми називаємо це «запускати скрипт чи сценарій Python»
- Для зручності ми додаємо «.py» після назви файлу, щоб позначити, що програма написана на Python

Консоль або сценарії

- Консоль

  - Ви пишете один рядок за раз і одразу ж чекаєте відповіді від Python

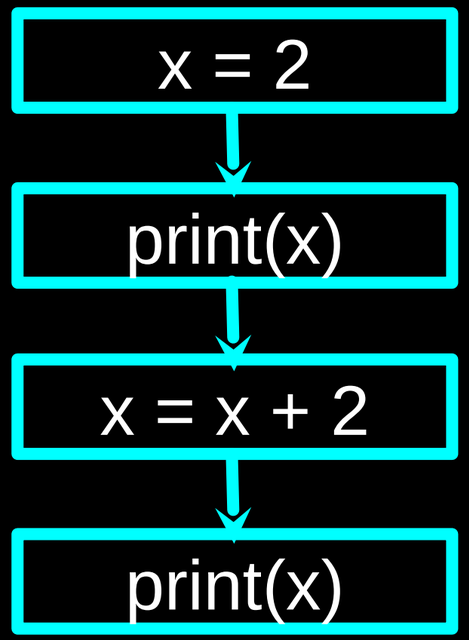
- Сценарій

  - За допомогою текстового редактора ви послідовно записуєте кілька рядків (інструкцій) у файл і просите Python виконати зазначені інструкції

Серія кроків чи базові моделі

- Як за рецептом для приготування страви або згідно з інструкцією зі складання меблів, ви вказуєте послідовність кроків, які треба виконати по черзі.
- Деякі кроки умовні – їх можна пропустити.
- Деколи крок або серію кроків треба повторити.
- За потреби ми зберігаємо серію кроків і наказуємо комп'ютеру повторити їх мільйони разів у різних місцях програми (Розділ 4).

Послідовні кроки



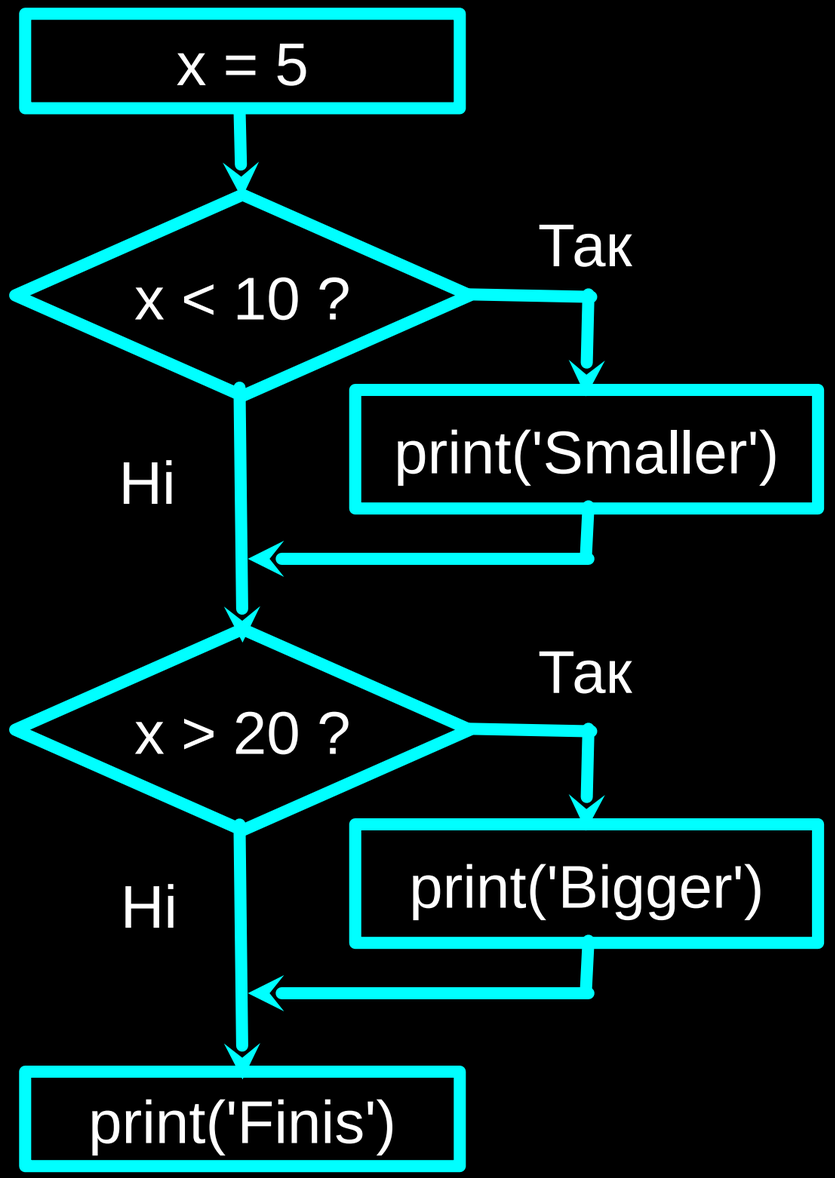
Програма:

```
x = 2
print(x)
x = x + 2
print(x)
```

Вивід:

```
2
4
```

Коли програма запущена, вона послідовно виконує крок за кроком. Ми вказуємо що за чим виконувати: спочатку це, потім те, перейти до наступного – і так далі.



Умовні кроки

Програма:

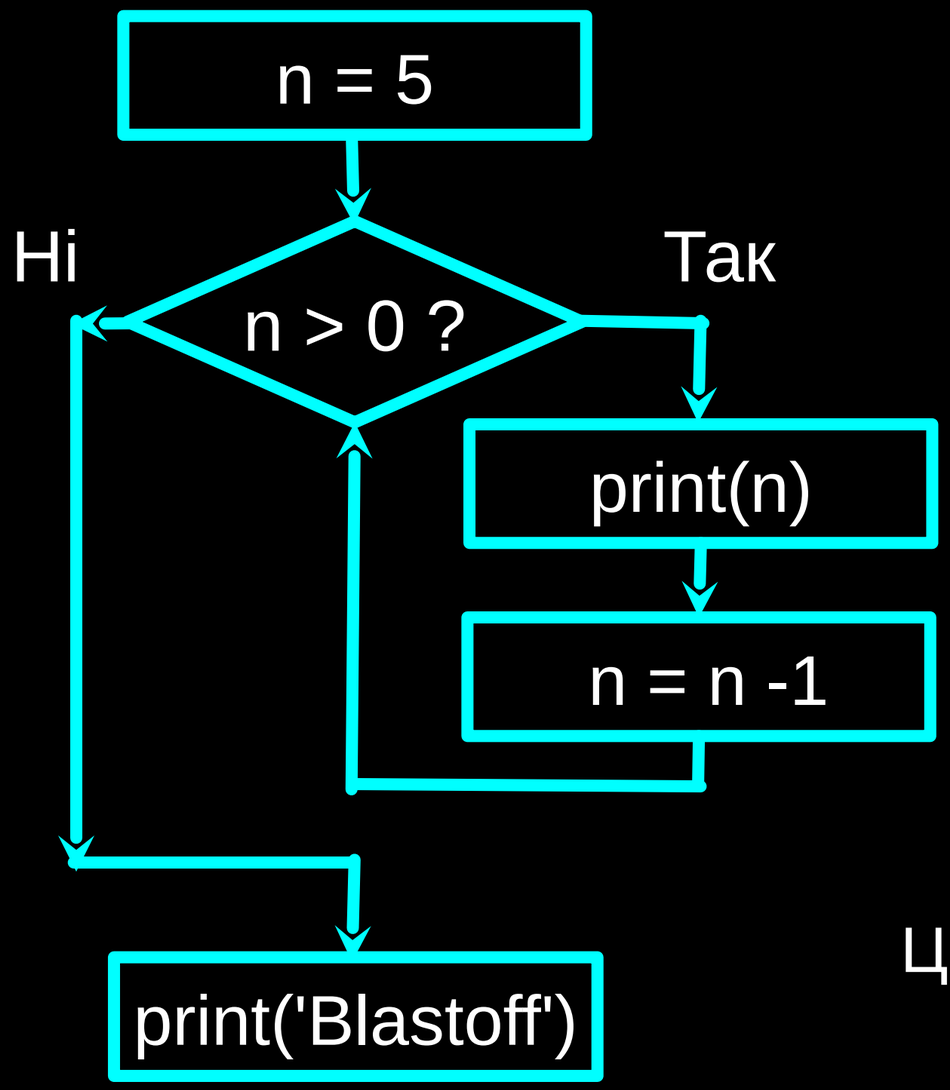
```
x = 5
if x < 10:
    print('Smaller')
if x > 20:
    print('Bigger')

print('Finis')
```

Вивід:

```
Smaller
Finis
```

Циклічні кроки



Програма:

```
n = 5
while n > 0 :
    print(n)
    n = n - 1
print('Blastoff!')
```

Вивід:

```
5
4
3
2
1
Blastoff!
```

Цикли (повторювані кроки) мають ітераційні змінні, які змінюються на кожній ітерації протягом циклу.

```
name = input('Enter file:')
handle = open(name, 'r')

counts = dict()
for line in handle:
    words = line.split()
    for word in words:
        counts[word] = counts.get(word,0) + 1

bigcount = None
bigword = None
for word,count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count

print(bigword, bigcount)
```

Послідовні

Циклічні

Умовні

```
name = input('Enter file:')
handle = open(name, 'r')

counts = dict()
for line in handle:
    words = line.split()
    for word in words:
        counts[word] = counts.get(word,0) + 1

bigcount = None
bigword = None
for word,count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count

print(bigword, bigcount)
```

Коротка «історія» про те, як рахувати слова у Python

Вхідні дані від Користувача

Речення про послідовне оновлення даних

Абзац про те, як знайти набільше значення у списку

Висновки

- Це короткий огляд Розділу 1
- Ми будемо повертатися до цієї інформації впродовж курсу
- Ми розглянули загальну картину

Права власності / Застереження