

ЛЕКЦІЯ 1.

1. Вступ.
2. Історія обчислювальної техніки
3. Двійкова система числення
4. Будова персонального комп'ютера
5. Мови програмування
6. Встановлюємо Python

Персональний комп'ютер - це не лише ноутбук, або коробка з монітором та клавіатурою. Аналогічні пристрої містяться у смартфонах, банкоматах, терміналах для оплати рахунків, сучасних телевізорах, автомобілях та більшості інших складних приладів.

ЕОМ - електронна обчислювальна машина, здатна виконувати обчислювальні задачі будь-якої складності за наявності відповідної програми.

Найперші обчислення в історії людства проводилися на пальцях. Саме тому число 10 лежить в основі всіх наших розрахунків - ми використовуємо **десяткову систему числення**.

З розвитком торгівлі і виникненням грошей з'явилася необхідність додавати і віднімати великі числа. Тому десь у 3-му тисячолітті до н.е. з'явився перший обчислювальний пристрій - [абак](#), прообраз сучасної рахівниці. Він являв собою дошку з "канавками", по яких можна було рухати кісточки, камінці або зернятка.

Стрімкий розвиток наук у Давній Греції призвів до появи різних механічних приладів, в тому числі обчислювальних, найвідомішим з яких є ["антикітерський механізм"](#) на основі зубчатих передач, призначений для астрономічних розрахунків.

Античні часи завершилися. Розпочалося Середньовіччя, темна доба в історії науки і техніки. Лише в епоху Відродження людство повернулося обличчям до історії і стало стрімко наздоганяти втрачений час.

Одним з результатів Першої науково-технічної революції став і прогрес в області обчислювальної техніки. В XVII столітті почали з'являтися арифмометри - настільні механічні машини для проведення арифметичних операцій (додавання та віднімання чисел, більш пізні вже вміли також множити та ділити): "обчислювальний годинник" Шикарда, "паскаліна" Паскаля, "Механічний калькулятор" Лейбніца. Далі - поява логарифмічного числення, нові моделі обчислювальних машин та промисловий випуск арифмометрів у 1820 р.

В 1804 р. французький ткач Джозеф Марі Жаккар (або Жаккард) винайшов верстат, здатний виробляти тканину з рельєфним візерунком, який задавався за допомогою [перфокарт](#) - картонних карток із отворами. Це дозволило налагодити промислове виробництво тканини, названої далі жаккардовою, принесло винахіднику премію від імператора Наполеона Бонапарта і ексклюзивне право на ліцензійні відрахування (одні з перших в історії) за кожний діючий верстат його конструкції у Франції. А сам винахід став **першим в історії програмованим пристроєм**.

В 1822 англійський математик Чарльз Бебідж побудував прототип ["різницевої машини"](#) для автоматичного обчислення логарифмічних таблиць. Під час роботи над нею він придумав **концепцію універсальної обчислювальної машини**, яку назвав "аналітичною". Аналітична машина Бебіджа складалася (майже як сучасні ЕОМ) з арифметичного пристрою, пам'яті та пристроїв вводу-виводу. За задумкою винахідника пристрій мав виконувати будь-які операції згідно програми, записаної на перфокартах, але за багатьох причин його проект так і не було втілено в життя. Разом із ним над винаходом активно працювала графиня Ада Лавлейс, донька поета Джорджа Байрона, яка і написала першу програму для (тоді ще не існуючого) комп'ютера. Таким чином, першим програмістом в історії була саме жінка ;-).

В 1936 англійський математик Алан Тьюрінг запропонував **математичну модель обчислювальної машини**, відому як [машина Тьюрінга](#). Довів, що вона здатна виконати будь-які обчислення, і ввів поняття **алгоритму** - послідовності дій, необхідних машині для досягнення результату. Сама ж **комп'ютерна програма** являє собою записаний набір команд, які задають алгоритм у формі, зрозумілій для машини.

В 1938 молодий німецький інженер [Конрад Цузе](#) побудував у себе вдома механічну цифрову обчислювальну машину Z1 з програмним управлінням та використанням двійкової системи числення - фактично **перший повноцінний комп'ютер**. Також до його досягнень відносяться декілька вдосконалених "зетів", чия конструкція лягла в основу наступних комп'ютерів, та **перша "алгоритмічна" мова програмування** Plankalkul, яка дозволяла використовувати для написання програми конструкції більш зрозумілі людині замість машинних команд. На жаль, це був початок Другої світової війни, коли Німеччина залишилася одна проти всіх, і більшість його досягнень ще довго залишалися невідомими в світі.

ПЕРШЕ ПОКОЛІННЯ ЕОМ (1946 Р.)

базувалося на електронних вакуумних лампах - великих пристроях з низькою ефективністю та надійністю. Великі ЕОМ. Для вводу-виводу даних використовувалися перфокарти та перфоленти, швидкодія сягала 10-20 тис. операцій в секунду.

1946 - завершення розробки [ENIAC'у](#). **Перший електронний цифровий комп'ютер**, побудований на замовлення армії США для обчислення

балістичних таблиць. Обчислювальна потужність - 350 операцій множення або 5000 операцій додавання на секунду. Вага - 27 тонн.

1950 - введення в експлуатацію "[Малої електронної обчислювальної машини](#)" в СРСР, побудованої під керівництвом академіка Сергія Лебедева в Інституті електротехніки АН УРСР. **Перша ЕОМ в Європі.**

ДРУГЕ ПОКОЛІННЯ ЕОМ (1955 Р.)

базувалося на транзисторах (перший кремнієвий транзистор виготовлено у 1954р.) - більш компактних та надійних елементах для побудови електронних схем. Для вводу-виводу даних використовувалася магнітна стрічка, швидкодія - 100 тис. - 1 млн. операцій в секунду.

1957 - **перший транзисторний комп'ютер** американської фірми NCR.

ТРЕТЄ ПОКОЛІННЯ ЕОМ (~1965 Р.)

стало можливим внаслідок винайдення в 1956 р. інтегральних мікросхем, які дозволили монтувати електронні схеми з дуже високою щільністю розташування елементів. Поява мініЕОМ. Швидкодія - 1-10 млн. операцій в секунду.

ЧЕТВЕРТЕ ПОКОЛІННЯ ЕОМ (70-ТІ РР.)

розпочалося після виникнення мікропроцесорів (1971р - перший масовий мікропроцесор фірми Intel) внаслідок розвитку електроніки та появи великих інтегральних схем. Мікропроцесори замінили десятки інших мікросхем, що дозволило ще зменшити розміри комп'ютерів. Поява мікроЕОМ, магнітних дисків та "дискет".

1976 - 26-річний американський програміст Стів Возняк створив персональний комп'ютер [Apple 1](#), який вважається сьогодні **першим комерційно успішним персональним комп'ютером**. Його шкільний друг Стів Джобс вигадує продавати такі "персональні комп'ютери" звичайним людям, вони створюють компанію Apple і комп'ютери стають масовим явищем.

За найбільш поширеною точкою зору четверте покоління ЕОМ триває досі, лише нарощуючи потужність та кількість елементів.

Паралельно ведуться дослідження альтернативних принципів побудови ЕОМ, але сам термін "[п'яте покоління ЕОМ](#)", що символізує принциповий якісний стрибок комп'ютерних технологій, досі не є чітко визначеним.

Отже, природньою для нас є десяткова система числення - це та, якою ми користуємося щодня. Для запису чисел у десятковій системі використовуються 10 знаків - арабські цифри від 0 до 9 (зверніть увагу: 9

на 1 менше за 10 - основу системи числення), кожен розряд числа записується однією цифрою і має певну "вагу".

Наприклад, візьмемо число 284 - воно має 3 розряди. Згідно того, чому нас вчили у школі, це одиниці, десятки та сотні. Відповідно, "вага" цих розрядів становить 1, 10 та 100; або, виражаючи їх через нашу основу числення, 10^0 , 10^1 , 10^2 (зверніть увагу: розряди для цього нумеруються справа наліво). Знаючи вагу розрядів, ми легко можемо віднайти саме число: $2*100 + 8*10 + 4*1 = 284$. Погодьтеся, для десяткової системи це було нескладно.

3 будь-якої системи числення у десяткову

В обчислювальній техніці застосовується двійкова система, основою якої є двійка, а для запису чисел, відповідно, використовуються 0 або 1. Чому так? Тому що 0 або 1 простіше передавати між елементами ніж 10 різних значень, і зберігати в пам'яті 0 або 1 також простіше. За тим самим принципом працює і вона.

Наприклад, число 10010 (у двійковій системі). Звичайно, ваги розрядів вже не будуть десятками та сотнями, але так саме залишаються степенями основи системи числення, тобто $1=2^0$, $2=2^1$, $4=2^2$, $8=2^3$, $16=2^4$. І тепер ми можемо легко перетворити це число у десяткову систему:

$$\begin{aligned} 1*16 + 0*8 + 0*4 + 1*2 + 0*1 = \\ 16 + 0 + 0 + 2 + 0 = \\ 18. \end{aligned}$$

Можете перевірити мене на калькуляторі.

Аналогічним чином можна перевести число з будь-якої системи числення у десяткову - це сума добутків значення кожного розряду на його вагу, де вага - це основа системи числення, піднесена у степінь, рівний порядковому номеру розряду справа наліво, починаючи з 0. Моторошно? Можемо порахувати ще трохи.

Шістнадцяткова система також дуже популярна - 2 розряди у ній позначають 1 байт даних. Так як арабських цифр не вистачає, в якості інших 6 символів використовуються латинські літери від A до F. Візьмемо 1A3F і переведемо в десяткову систему:

$$\begin{aligned} 1*16^3 + A*16^2 + 3*16^1 + F*16^0 = \\ 1*16^3 + 10*16^2 + 3*16^1 + 15*16^0 = \\ 1*4096 + 10*256 + 3*16 + 15*1 = \\ 4096 + 2560 + 48 + 15 = \\ 6719. \end{aligned}$$

Власне, система числення може мати будь-яку основу.

Наприклад, в трійковій системі для позначення розрядів використовуватимуться 0, 1 та 2. Візьмемо яке-небудь 210120:

$$\begin{aligned} 2*3^5 + 1*3^4 + 0*3^3 + 1*3^2 + 2*3^1 + 0*3^0 = \\ 2*243 + 1*81 + 0*27 + 1*9 + 2*3 + 0*1 = \\ 486 + 81 + 0 + 9 + 6 + 0 = \\ 582. \end{aligned}$$

З десяткової системи числення у будь-яку іншу

Для того ж, щоб перевести число з десяткової системи у будь-яку іншу, слід виконати зворотні дії: ділити початкове число на ваги розрядів і отримати таким чином їх значення.

Або можна згадати, що кожний наступний розряд - по суті перенос того, що не вмістилося у попередній, і трохи спростити розрахунки: брати не початкове число, а те, що залишилося від попереднього ділення, ділити його прямо на основу і брати остачу від ділення.

Переведемо наше десяткове 284 в двійкову систему:

$284 / 2 = 142$ (це те, що не вміщається в нульовий розряд), остача 0 (те що вмістилося)

142 (те, що не вмістилося в попередній розряд) $/ 2 = 71$, остача 0

$71 / 2 = 35$, остача 1

$35 / 2 = 17$, остача 1

$17 / 2 = 8$, остача 1

$8 / 2 = 4$, остача 0

$4 / 2 = 2$, остача 0

$2 / 2 = 1$, остача 0

$1 / 2 = 0$, остача 1, далі немає куди ділити

Тепер ці остачі від останньої до першої - це і є розряди двійкового числа: 100011100

Або 582 назад у трійкову:

$582 / 3 = 194$, остача 0

$194 / 3 = 64$, остача 2

$64 / 3 = 21$, остача 1

$21 / 3 = 7$, остача 0

$7 / 3 = 2$, остача 1

$2 / 3 = 0$, остача 2

Записуємо остачі від останньої до першої - 210120 - те саме число, яке ми переводили з трійкової системи, отже все вірно.

З будь-якої в будь-яку

Все так само :-). Просто арифметичні операції доведеться проводити не в десятковій системі. Тому простіше перевести число з початкової системи числення в десяткову, а те, що отримаємо - в ту, яка нам необхідна.

Архітектура фон Неймана



Така схема побудови комп'ютера (також відома як [прінстонська](#)) була розроблена цілим колективом авторів під час роботи над ENIAC'ом і названа на честь одного з розробників - Джона фон Неймана. Крім неї існують і альтернативні варіанти, найвідомішим з яких є гарвардська архітектура. Головною особливістю схеми фон Неймана є використання єдиної пам'яті для запису даних та команд, що робить її найбільш простою для реалізації.

Отже, комп'ютер обов'язково містить операційний та керуючий блоки, які об'єднуються в **центральний процесор** або CPU (Central Processing Unit) - той самий, що вважається серцем вашого комп'ютера. *Операційний блок* уміє виконувати деякий обмежений набір простих команд, як наприклад, прочитати команду з комірки пам'яті А, прочитати дані з комірки пам'яті В, скласти 2 числа, відняти, розділити, записати результат в комірку С та інші. *Керуючий блок* управляє діями операційного блоку і слідує, щоб програма виконувалася у заданій послідовності. *Акумулятор* (або регістр) також є частиною процесора і слугує для збереження операндів та проміжних результатів обчислень, для того щоб не звертатися за ними кожного разу до основної пам'яті.

Крім того, процесор містить *тактовий генератор*, який продукує сигнали для синхронізації роботи всіх мікросхем, - це і є ті самі "гігагерци", за якими ми звикли вимірювати швидкість комп'ютера. Тобто, коли кажуть, що процесор має частоту 2GHz, це значить, що він генерує 2 мільярди тактових сигналів в секунду і теоретично виконує 2 мільярди елементарних операцій в секунду. Але більшість команд є складнішими і можуть

потребувати по декілька тактів, тому така характеристика швидкодії є приблизною.

У **пам'яті** зберігаються програми та дані, з якими має справу комп'ютер (а якщо абстрагуватися, то в архітектурі фон Неймана програма - це такі самі дані, як і ті, що нею обробляються). Там же зберігаються всі проміжні дані, необхідні для виконання програми. Пам'ять складається з великої кількості комірок, кожна з яких має адресу і містить 1 байт (тобто 8 біт інформації). Це дозволяє в будь-який момент часу звернутися до будь-якої комірки і зчитати або записати туди дані. Втім є один недолік - дані зберігаються лише поки наявне живлення, а при знеструмленні комп'ютера всі данні стираються. Саме тому ми знаємо її як *оперативну пам'ять* або RAM (random access memory). Об'єм оперативної пам'яті сучасного комп'ютера знаходиться у межах 1-16 гігабайт, тобто 1-16 мільярдів байт.

А для того, щоб дані нікуди не зникали після завершення роботи комп'ютера, використовуються додаткові запам'ятовуючі пристрої з нижчою швидкістю доступу до інформації. Найбільш розповсюдженим варіантом є *жорсткий диск* - власне на ньому і зберігаються всі ваші картинки, музика і фільми. там же записана операційна система і всі інші програми, що є на вашому комп'ютері. Під час запуску програми вона копіюється у оперативну пам'ять і далі виконується звідти. Всі дані, з якими працює програма, також завантажуються з жорсткого диска та назад за вимогою. Саме тому, при вимкненні комп'ютера слід зберегти всі відкриті документи - в цей час всі ваші зміни існують лише в оперативній пам'яті і зникають з неї разом із струмом.

Жорсткий диск, як і всі інші пристрої, що підключаються до комп'ютера, відноситься до **пристроїв вводу-виводу**. Кожен з них має власний набір команд для взаємодії з комп'ютером і потребує для роботи спеціальну програму, що називається драйвером. Якщо в операційній системі встановлений відповідний драйвер, вона здатна "спілкуватися" з підключеним пристроєм і ви можете його використовувати. Найбільш розповсюджені пристрої мають стандартизовані набори команд і драйвери для них вже включені в операційну систему, це монітор, миша, клавіатура, жорсткий диск. Але, якщо ви, наприклад, підключите до свого комп'ютера круту мишку з 5 клавішами, одразу працюватимуть лише 2 або 3 з них, а для повноцінної роботи необхідне встановлення драйверів, що записані на диску у коробці.

Взаємодію між усіма компонентами забезпечує так звана **системна шина**. До неї підключається все обладнання, і вона передає сигнали між окремими пристроями. Фізично вона розташовується на *материнській платі*, навкруги якої і збирається сучасний комп'ютер.

Звичайно, це дуже загальний опис і в реальності все набагато складніше - все-таки не цьому присвячений наш курс. Але нашою метою було хоча б часткове викриття магії персонального комп'ютера і, я сподіваюся, ми її досягнули.

Якщо ви користуєтеся комп'ютером під керівництвом операційної системи Linux або MacOS, вам пощастило - інтерпретатор Python вже встановлений у вашій системі. Скоріше за все, це версія 2.7.x і всі наші приклади мають нормально у вас працювати.

Якщо у вас, як і у мене, встановлено Windows, доведеться встановити інтерпретатор Python вручну. Процес установки викладено у відео лекції, але продублюю тут основні кроки для вашої зручності:

1. **Завантажуємо** установник:

1. Заходимо на офіційний сайт [Python](https://www.python.org/), обираємо пункт меню "Downloads". Натискаємо на кнопку для скачування версії 2.x.x (на момент написання тексту це 2.7.9), обираємо папку, де ви хочете зберегти установник.

2. Чекаємо на завантаження, п'ємо каву.

2. **Встановлюємо** інтерпретатор на комп'ютері:

1. Запускаємо файл-установник із папки, де він був збережений. Next.

2. Обираємо папку для установки. Раджу залишити папку "за умовчанням", так як наявність кирилиці або пробілів у шляху може в подальшому призвести до проблем із запуском деяких програм, які потребують Python для роботи.

3. На етапі "Customize Python" краще включити опцію "Add python.exe to Path" - це дозволить викликати інтерпретатор прямо за іменем python, не вказуючи повний шлях до програми. Зверніть увагу: ця можливість з'явиться лише після перезавантаження комп'ютера.

4. Чекаємо на завершення установки. Finish.

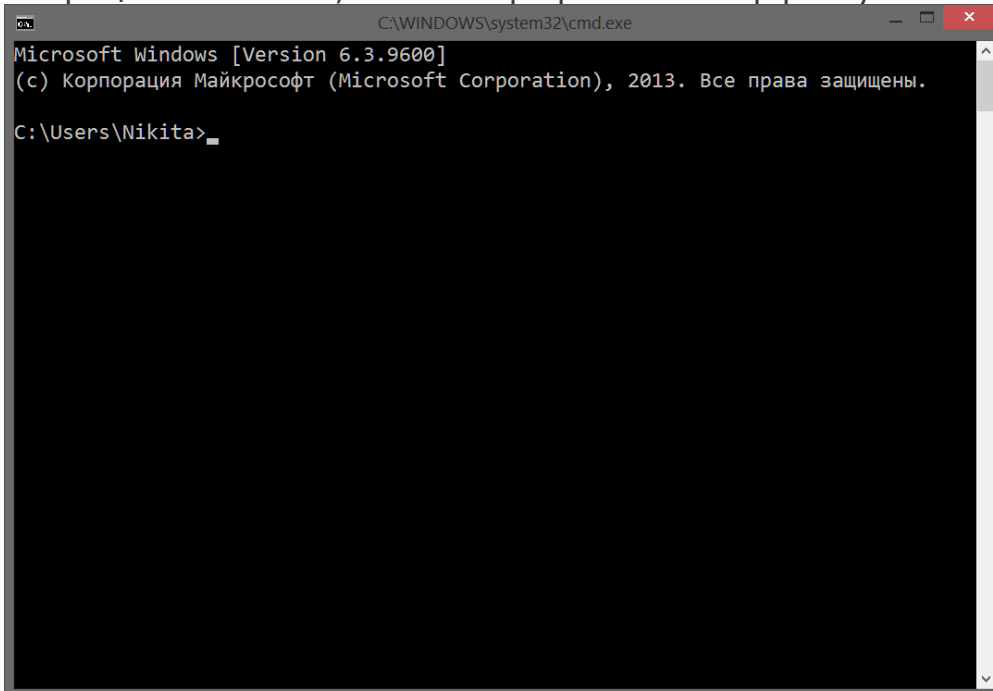
3. Перезавантажуємо машину та **перевіряємо** працездатність інтерпретатора:

1. Запускаємо консоль. Ця дія знадобиться вам і далі, так як всі найпростіші програми працюють в консолі і ваші наступні лабораторні роботи також запускатимуться саме в ній:

1. для Windows: меню "пуск", виконати, вводимо "cmd" та тиснемо Ентер.

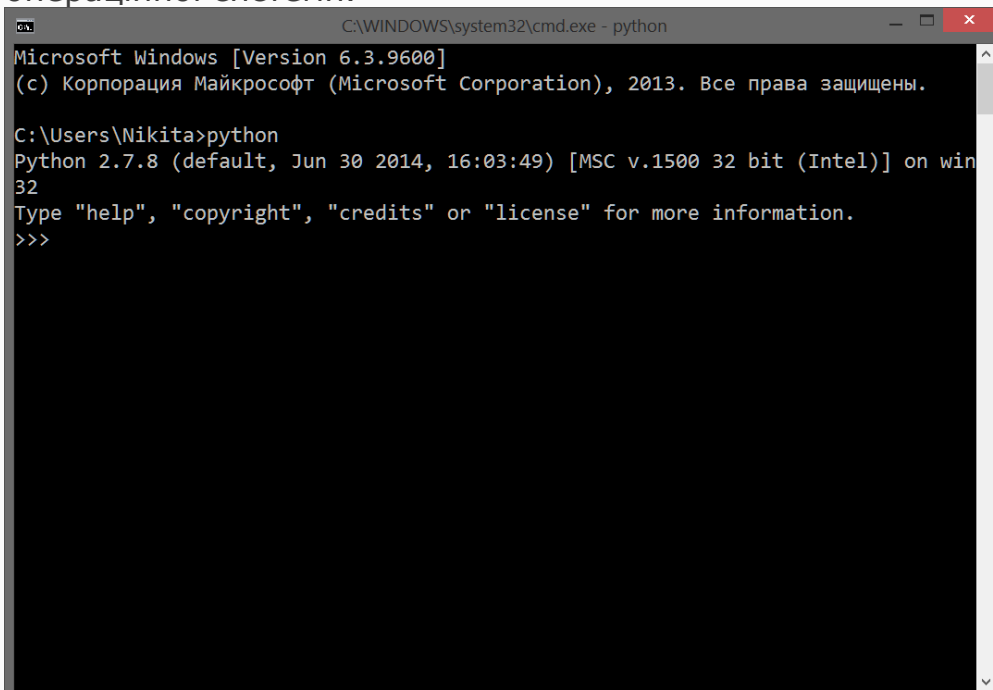
2. для Linux та MacOS: знайдіть програму, яка називається "Термінал" або "Консоль" - точна назва може відрізнятися, в залежності від виду та версії вашої операційної системи.

2. Перед вами з'явилося вікно з контрастним текстом. Це - командний рядок, в якому можна запускати програми та вводити різноманітні програми - фактично можна робити все, що дозволяє операційна система, але без графічного інтерфейсу.



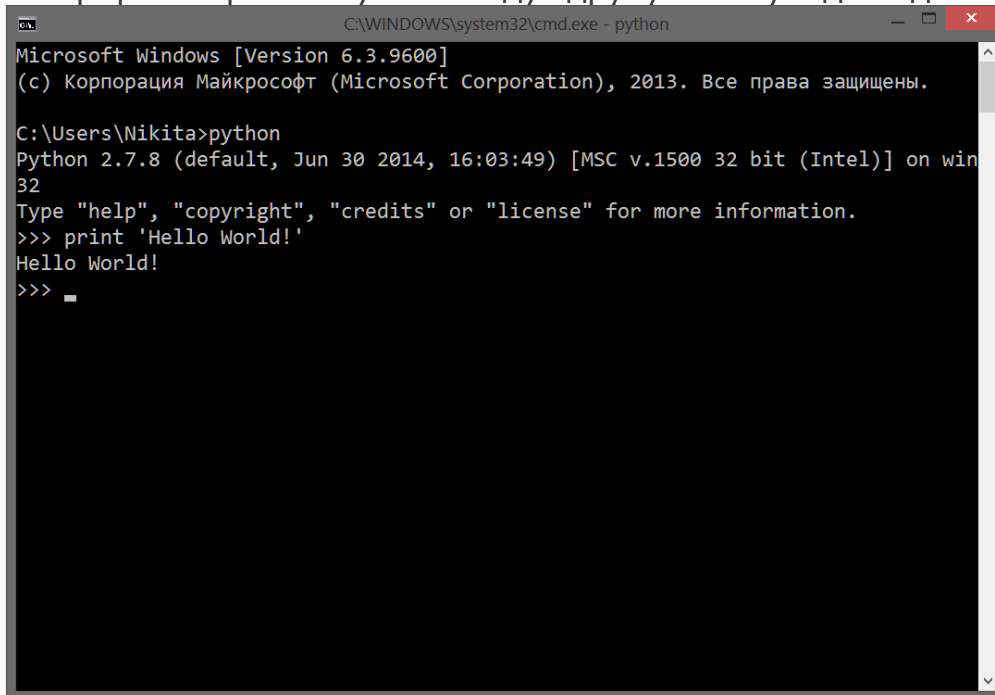
```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) Корпорация Майкрософт (Microsoft Corporation), 2013. Все права защищены.
C:\Users\Nikita>
```

3. Якщо ви включили опцію "Add python.exe to Path", в консолі тепер можна просто набрати python та натиснути Ентер - буде запущено сесію інтерпретатора Python в інтерактивному режимі. Це як ще одна консоль в консолі, лише яка сприймає команди мови Python замість команд операційної системи.



```
C:\WINDOWS\system32\cmd.exe - python
Microsoft Windows [Version 6.3.9600]
(c) Корпорация Майкрософт (Microsoft Corporation), 2013. Все права защищены.
C:\Users\Nikita>python
Python 2.7.8 (default, Jun 30 2014, 16:03:49) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

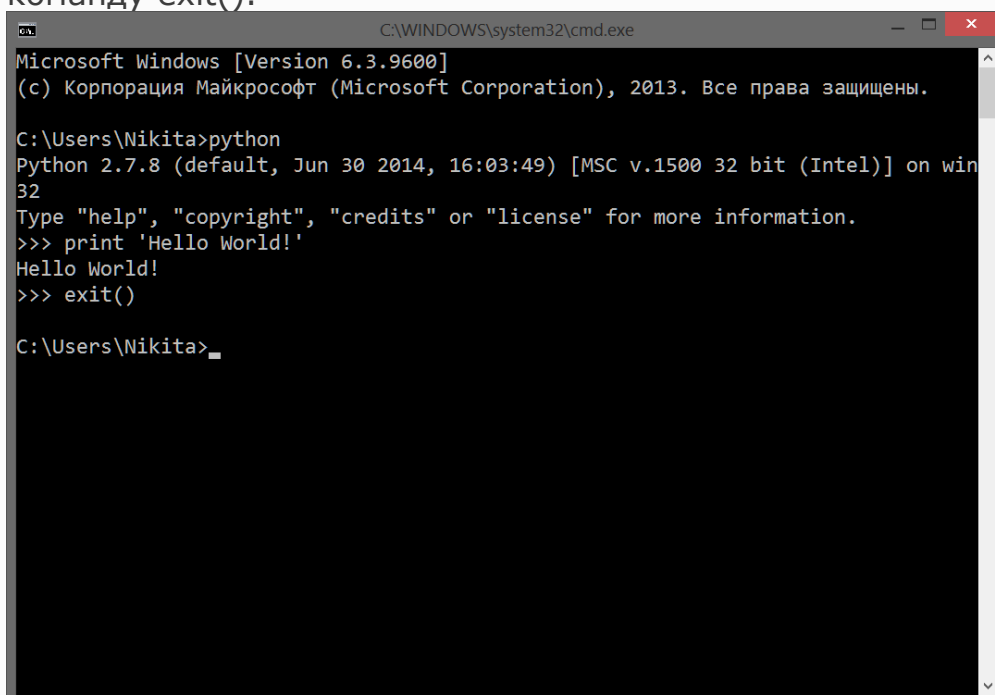
4. Вводимо просту команду: `print 'Hello World!'`, тиснемо Ентер. Інтерпретатор виконує команду: друкує нам у відповідь "Hello World!"



```
C:\WINDOWS\system32\cmd.exe - python
Microsoft Windows [Version 6.3.9600]
(c) Корпорация Майкрософт (Microsoft Corporation), 2013. Все права защищены.

C:\Users\Nikita>python
Python 2.7.8 (default, Jun 30 2014, 16:03:49) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print 'Hello World!'
Hello World!
>>> _
```

5. Для завершення інтерактивної сесії слід набрати і виконати команду `exit()`.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) Корпорация Майкрософт (Microsoft Corporation), 2013. Все права защищены.

C:\Users\Nikita>python
Python 2.7.8 (default, Jun 30 2014, 16:03:49) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print 'Hello World!'
Hello World!
>>> exit()

C:\Users\Nikita>_
```

4. Поздоровляю!