



Технологічний Стек Сучасності

Ваш комплексний шлях у світ передових технологій! Від штучного інтелекту до веб-розробки та блокчейну.



by Oleksandr Denishchuk

Мета

Цей курс допоможе вам зрозуміти основи ІТ

Що ви вивчите

- Дізнаємося що таке інтернет
- 🤖 Зрозумієте, як працює **Штучний Інтелект** (без складних термінів)
- 🐍 Спробуєте **ООП** на Python
- 🌐 Зробите простий сайт і запустите його в інтернеті
- 📊 Освоїте базові інструменти розробника (VS Code, Git)
- 💎 Дізнаєтесь про криптовалюти та створите свою

Ви побачите цифровий світ по-новому



Як працює Інтернет? (Магія зв'язку)



Клієнт-Сервер

Ваш браузер надсилає запит на сервер, де зберігається сайт.



IP-адреси та DNS

DNS перетворює назви сайтів на IP-адреси – унікальні "домашні адреси" в мережі.



Пакети даних

Інформація розбивається на маленькі пакети і збирається у місці призначення.



Протоколи

TCP/IP, HTTP/HTTPS – це "мови", якими спілкуються комп'ютери в мережі.

5

Вже знаєте команди **ping** та **tracert**

Значить можете працювати інсталятором у ваш

Інтернет – глобальна мережа комп'ютерів, що обмінюються інформацією за встановленими правилами.



Ми + ШІ = 🚀 (і трохи хайпу)

GitHub Copilot - наш вірний напарник

Інтегрували в VS Code з першого дня. Допомогає писати HTML, CSS та JavaScript без зайвих зусиль.

ChatGPT-промпти для веб-розробки

Створюємо запити для генерації Bootstrap-компонентів, налагодження коду та оптимізації Git-команд. Сучасна розробка без ШІ неможлива.





- 1. "Бау, він відповідає!" — ШІ як інтерактивна енциклопедія.
- 2. Інструмент для чернеток, оформлення та генерації матеріалів.
- 3. Напарник у проєктах — разом уточнюємо промпти й створюємо нове.
- 4. Розуміємо сильні сторони та обмеження, керуємо ШІ замість сліпої довіри.





Еволюція нашого ШІ-мислення

Наша подорож зі штучним інтелектом — від здивування до майстерності

-  Фаза "Ого!"
Перші кроки — запитували вірші, жартували, експериментували з простими запитам.
-  Фаза "Шо, він і це може?!"
Відкриття можливостей — ШІ створює таблиці, реферати та презентації блискавично.
-  Фаза "Гаразд, працюй зі мною"
Справжня співпраця — проєкти, код та контент створюються разом, як у стартапі.
-  Фаза "Стоп, не бреш"
Критичне мислення — виявляємо "галюцинації", перевіряємо інформацію, контролюємо процес.

Як алгоритм «думає»

Машинне навчання — не магія

Алгоритм отримує нові факти й змінює свою думку. Це як передумати про когось, коли дізнаєшся щось нове.

Функція втрат: виявлення помилок

Ми познайомились із поняттям **функції втрат**. Це як сигнал тривоги, який каже алгоритму: «Стоп! Ти щось переплутав!»

Основні типи завдань у машинному навчанні

- 1** **Класифікація** — розподілити на категорії (це песик чи котик?)
- 2** **Регресія** — передбачити число (яка завтра температура?)
- 3** **Кластеризація** — розділити дані на групи (хто дружить із ким?).





Практикуємо машинне навчання

1

Теорема Баєса на практиці

Ми вивчили важливі математичні концепції. Розібрались із парадоксом Монті Голла та задачкою про бібліотекаря, за яку дали Нобелівську премію

2

Лінійна регресія: прогнозування курсів валют

Ми розібрались, як це працює. Де тут теорема Піфагора, Евклідова відстань та функція втрат

3

K-найближчих сусідів (KNN)

Навчились знаходити схожі дані у багатовимірному просторі. Це працює схоже до системи рекомендацій фільмів.



Що таке ООП?

ООП — це спосіб програмування, коли програма складається з окремих «об'єктів», які взаємодіють між собою.

Це як уявити код не одним великим блоком, а групою персонажів, які знають, що робити, і спілкуються між собою.

- 1 Клас** — опис об'єкта, як креслення для будинку.
- 2 Об'єкт** — конкретний «будинок», створений за цим кресленням.
- 3 Інкапсуляція** — приховує деталі реалізації, як стіни приховують інженерні мережі.
- 4 Наслідування** — створення нових класів на основі старих, як створення нових моделей смартфонів на основі попередніх.
- 5 Поліморфізм** — одна дія може бути реалізована по-різному, як різні люди по-різному готують ту саму страву.

Наша практична робота

- Створили базовий клас Figure, який описує будь-яку геометричну фігуру.
- Навчилися створювати конкретні фігури на основі цього класу.

Ключові висновки

🔑 **Важливо запам'ятати:** ООП — це не просто шаблони, це **парадигма мислення**, що допомагає організувати складний код у зрозумілі об'єкти.

Наслідування та Інкапсуляція

👥 Наслідування — код у спадок

Це коли новий клас бере всі властивості й методи старого (базового), але додає щось своє.

Ми створили класи Circle, Line, Quadrilateral, які наслідували клас Figure.

Це як створити новий тип фігури, не переписуючи все з нуля — просто трохи змінюєш або додаєш.

🔒 Інкапсуляція — приховуємо зайве

Ми не повинні знати, як саме працює фігура всередині. Важливо, **що вона має метод .draw()**, і цього достатньо.

Це як мікрохвильова піч: натискаєш кнопку — вона гріє. Як саме — неважливо.

🔧 Практична реалізація принципів ООП

- Створили класи, які успадковують від Figure.
- Заклали в них методи для відображення та логіки малювання.
- Зробили так, щоб фігури "знали", як себе поводити — але ми не мусили знати всі деталі.

📌 Результати навчання

Ми навчилися будувати код, який можна **розширювати** і **перевикористовувати** без зайвого дублювання.



Поліморфізм — одна команда, різна поведінка

Визначення поліморфізму

Це коли **різні об'єкти** реагують на одну й ту ж команду по-своєму.

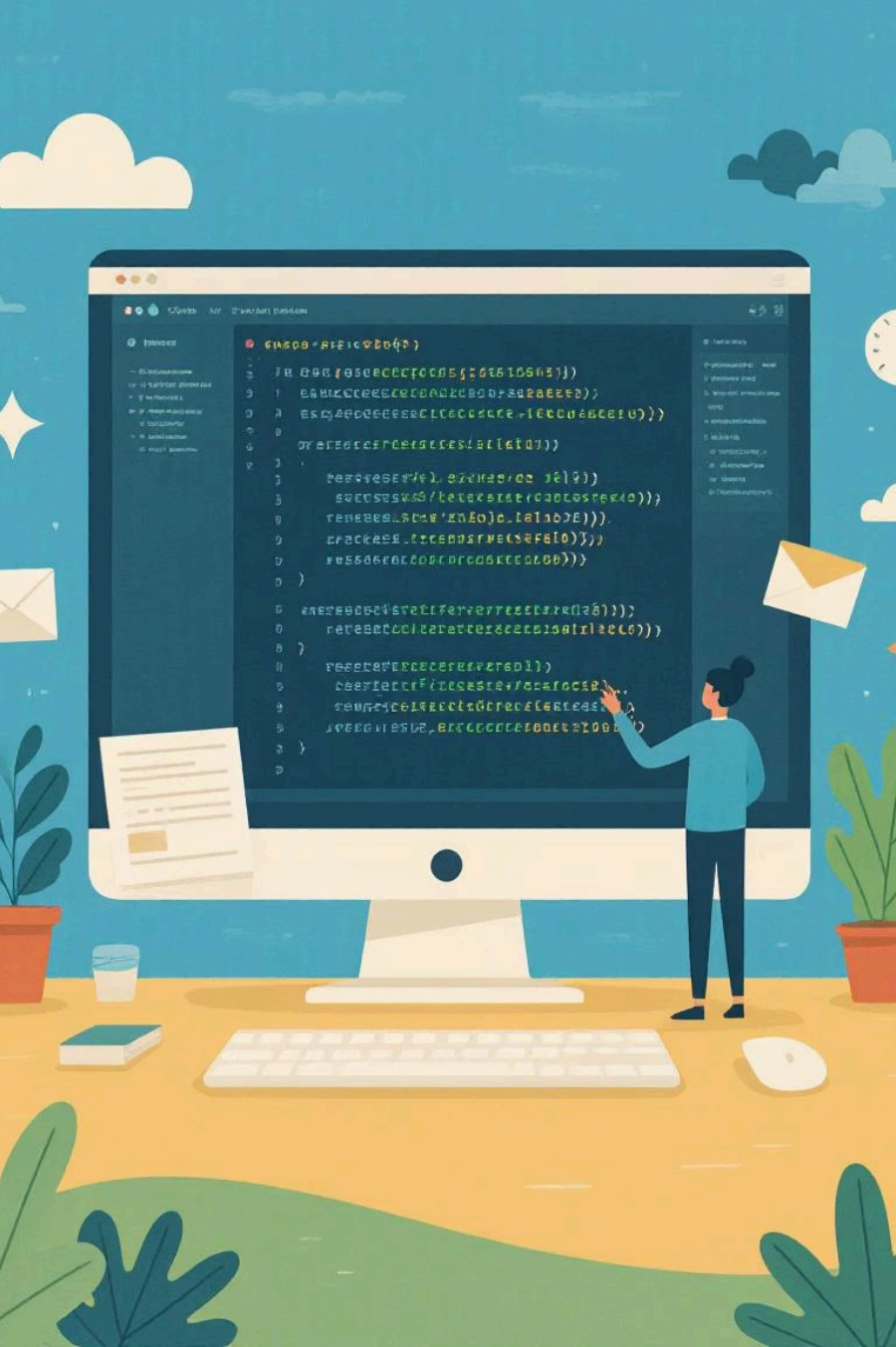
Уяви: у класі є Circle, Line, Quadrilateral. Усі вони мають метод `.draw()`, але кожна фігура малює себе по-іншому. Команда одна — результат різний.

Переваги в розробці

- Ми можемо написати **універсальний код**, який працює з різними типами фігур.
- Не потрібно знати, яку саме фігуру ми маємо — ми просто кажемо `.draw()` і фігура сама знає, що робити.

Практична реалізація

- Створили метод `draw()` у базовому класі Figure, а кожна фігура реалізувала його по-своєму.
- Створили список фігур і викликали `.draw()` для кожної — і кожна малювала себе, не питаючи дозволу в інших 😊



Веб-розробка – Готуємо сайт по рецепту

1



HTML – Структурний скелет

HTML — це скелет сторінки. Він тримає структуру, як каркас будівлі.

2



CSS – Стиль та оформлення

CSS — це стиль і краса. Він фарбує стіни, розставляє меблі, додає тіні та кольори.

3



JavaScript – Інтерактивність

JavaScript — це трохи магії. Він оживляє все, додає динаміку, анімацію, реакції на дії.

Ура! Ми зробили сторінку класу



Структура веб-сторінки

📄 Створили власну структуру **HTML-сторінки** (заголовки, блоки, списки, зображення)



Стилізація та дизайн

🎨 Оформлення через **CSS** Налаштували кольори, шрифти, відступи, розміщення елементів

✨ Додали трохи стилю: Псевдокласи типу `:hover`, м'які ефекти — сайт ожив!



Мобільна адаптація

📱 Адаптація для телефонів
Переконались, що наш сайт не "розвалюється" на мобільних



Bootstrap – зробили модний сайт за дві години

1 Готовий шаблон як фундамент

Взявши готовий шаблон **Bootstrap** і трохи адаптувавши його, ми отримали **повноцінний сайт** з усією необхідною структурою.

3 Автоматична адаптація під різні пристрої

Сайт однаково добре відображається на всіх пристроях.

Bootstrap автоматично адаптує всі елементи.

1

2

2 Готові компоненти для швидкої розробки

Всі елементи інтерфейсу **вже створені** — просто вставляй і налаштовуй.

Збирається як конструктор **LEGO** — швидко і надійно.

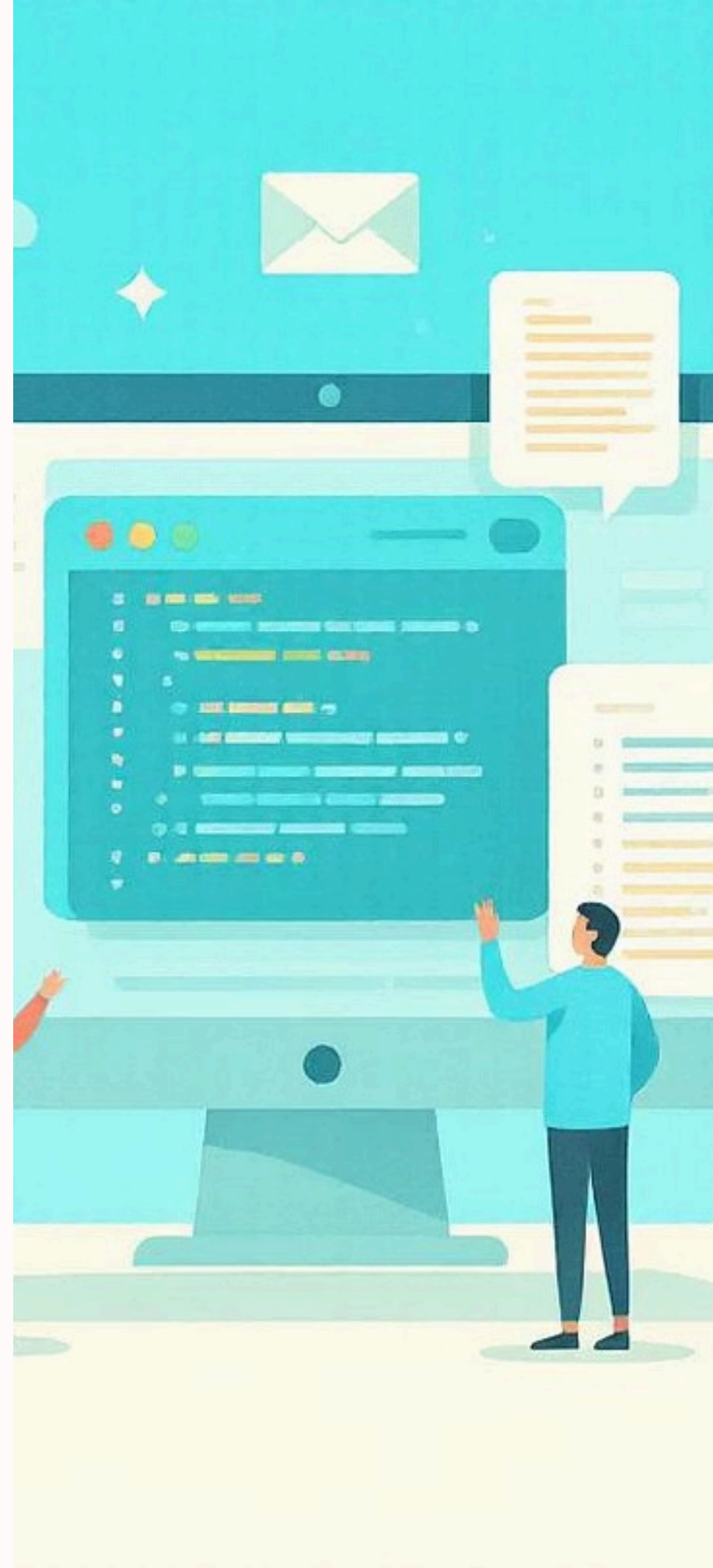
3

4

4 Професійний дизайн без спеціальних навичок

Стильний результат без використання Photoshop чи Figma.

📌 **Підсумок:** Готовий професійний сайт за один урок власними руками.



Git, GitHub та Аналіз Даних

git

Git & GitHub

Контроль версій: init, add, commit, push, pull.



GitHub Pages

Публікуємо свої проєкти онлайн.



Excel

Аналіз даних: таблиці, формули, фільтри.



Git та GitHub – контроль версій і публікація сайтів

Безпечне керування кодом

🔄 **Зберігаємо всі версії проєкту**
Якщо щось пішло не так — можна відкотитися назад. Спокійно, як у грі.

Публікація проєктів

🌐 **Викладаємо сайт онлайн за 3 хвилини** GitHub Pages = безкоштовна публікація. І це реально працює.

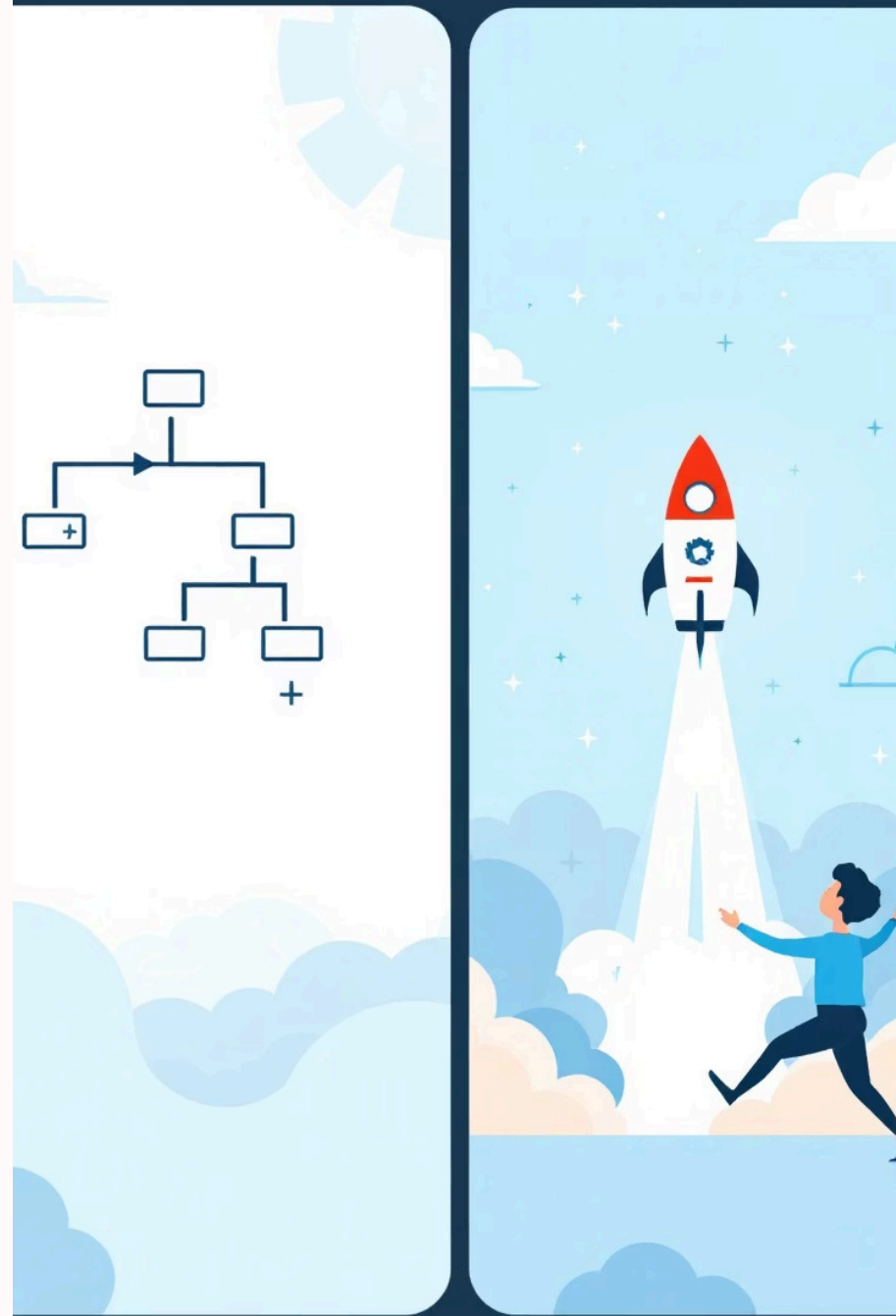
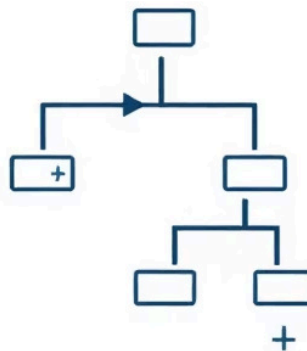
📌 **Результат:** Git — це вже не страшно. Це зручно.

Основні команди Git

💬 **Команди Git — вже не магія**
add, commit, push — як звичні команди. А reset --hard — вже не натискаємо просто так 😊

Гнучкість роботи

✂️ **Починаємо з нуля або підключаємо вже готове** Тепер ми самі керуємо своїм кодом — як справжні розробники.





Excel – як зрозуміти, куди пішли гроші на вечірку

PARTY EXPENSES

Venue Rental	\$19,600	Catering	\$15,950
Catering	\$3,800	Entertainment	\$15,000
Decorations	\$5,500	Entertainment	\$15,000
Decorations	\$9,750	Entertainment	\$18,950
Entertainment	\$,000	Entertainment	\$10,000
Entertainment	\$,000	Entertainment	\$20,000
Entertainment	\$,200	Entertainment	\$25,000
Miscellaneous	\$2,940	Entertainment	\$15,900
Miscellaneous	\$,600	Entertainment	\$15,000



Створювали таблиці

Зафіксували всі витрати по пунктах: піца, кульки, прикраси, напої.



Рахували бюджет

Використали формули для автоматичного обчислення. Ніхто не переплатив жодної гривні.



Аналізували дані

Фільтрували витрати та створювали графіки. Як у справжньому бізнес-звіті!



Результат: Не просто "погралися з табличками", а реально проаналізували дані й прийняли рішення на основі цифр.



Розуміємо Технологію Майбутнього

Що таке блокчейн?

Децентралізація, криптографія (хеші, підписи).

Bitcoin

Як працює перша криптовалюта?

Ethereum

Більше ніж гроші (смарт-контракти, EVM).

Токени **ERC-20**

Як створюються цифрові активи?

Наша перша криптовалюта – **ClassCoin**

Створили в **Remix**

Написали смарт-контракт мовою Solidity. Задали назву ClassCoin, символ та кількість токенів.

1




Запустили в **Ganache**

Розгорнули нашу монету в тестовій мережі. Витратили віртуальний Gas та отримали адресу контракту.

Працювали з **MetaMask**

Встановили криптогаманець у браузері. Імпортували ClassCoin та навчилися відправляти монети між адресами.

3

 **Результат:** Пройшли повний життєвий цикл створення та використання криптовалюти — від коду до транзакцій.



Що ми встигли за курс

1

Зрозуміли, як влаштований Інтернет та веб

2



Зрозуміли головні ідеї машинного навчання та як алгоритми "думають"

3



Освоїли ООП і створили фігури, що самі себе малюють

4



Розробили власний сайт класу та швидкий проєкт з **Bootstrap**

5



Опанували **Git, GitHub** - зберігаємо код професійно і хостимо швидко

6



Проаналізували реальні дані в **Excel** та створили бізнес-звіти

7



Створили власну криптовалюту **ClassCoin** та провели транзакції

Цей курс — про впевненість у цифровому світі. Ми не просто вивчали технології, а застосовували їх на практиці.

Курс "Технологічний Стек Сучасності": Концепція, Цілі та Очікувані Освітні Результати

Програма поглибленого вивчення сучасних інформаційних технологій для учнів старших класів, спрямована на формування ключових компетентностей XXI століття та профорієнтаційну підготовку.

I. Концептуальні засади курсу:

Курс "Технологічний Стек Сучасності" розроблений відповідно до принципів компетентнісного підходу в освіті та орієнтований на формування в учнів цілісного, системного уявлення про ключові напрями розвитку сучасних інформаційних технологій. Програма поєднує теоретичне осмислення фундаментальних концепцій, **зокрема принципів функціонування та практичного застосування штучного інтелекту (ШІ)**, з інтенсивною практичною діяльністю, забезпечуючи здобуття актуальних знань та навичок, необхідних для успішної навігації у цифровому світі та свідомого професійного самовизначення.

II. Мета та завдання курсу:

Мета курсу: Створення освітнього середовища для формування в учнів предметних ІТ-компетентностей та ключових компетентностей XXI століття через ознайомлення з сучасним технологічним стеком, **включаючи основи ШІ та його практичне застосування**, розвиток практичних навичок його використання та стимулювання інтересу до інноваційної діяльності в галузі інформаційних технологій.

Основні завдання:

- Формування предметної ІТ-компетентності (теоретичний аспект):**
 - Забезпечити розуміння учнями фундаментальних концепцій та взаємозв'язків у сферах:
 - Штучного інтелекту:** базові принципи роботи алгоритмів машинного навчання, можливості та обмеження сучасних ШІ-систем, етичні аспекти.
 - Об'єктно-орієнтованого програмування (на прикладі Python).
 - Веб-розробки.
 - Аналізу даних.
 - Технології блокчейн.
 - Ознайомити з принципами функціонування Інтернету та архітектурою сучасних веб-додатків.
- Розвиток предметної ІТ-компетентності (практичний аспект):**
 - Сформувати навички взаємодії зі штучним інтелектом:**
 - Опанування основ промпт-інжинірингу для ефективного використання генеративних моделей ШІ (наприклад, ChatGPT, GitHub Copilot).
 - Аналіз та застосування ШІ-інструментів для розв'язання практичних завдань у навчанні, програмуванні та повсякденній діяльності (генерація контенту, допомога в кодуванні, аналіз інформації).
 - Виконання практичних робіт з використанням ШІ для вирішення реальних кейсів.
 - Сформувати навички розробки статичних веб-сайтів (HTML, CSS).
 - Опанувати методи швидкої розробки адаптивних веб-інтерфейсів з використанням фреймворку Bootstrap.
 - Засвоїти принципи та інструментарій систем контролю версій (Git, GitHub) для командної та індивідуальної розробки.
 - Розвинути базові навички аналізу даних та візуалізації результатів засобами MS Excel.
 - Ознайомити з практичними аспектами створення та функціонування цифрових активів на основі технології блокчейн.
- Формування та розвиток ключових компетентностей:**
 - Інформаційно-цифрова компетентність:** Поглиблення навичок ефективного та критичного використання цифрових технологій, включаючи інструменти ШІ, для навчання, творчості та розв'язання проблем; формування розуміння доступності сучасних технологій та ефективності їх застосування.
 - Критичне та системне мислення:** Розвиток здатності аналізувати технологічні тренди, особливо у сфері ШІ, оцінювати можливості, ризики та етичні аспекти застосування ІТ.
 - Ініціативність та підприємливість:** Стимулювання до створення власних проєктів, генерації ідей та їх практичної реалізації, в тому числі з використанням ШІ.
 - Вміння вчитися впродовж життя:** Формування усвідомлення необхідності постійного оновлення знань у швидкозмінній сфері ІТ, особливо в контексті розвитку ШІ.
 - Профорієнтаційна компетентність:** Надання учням інформації для усвідомленого вибору майбутньої освітньої траєкторії та професійної кар'єри в ІТ або суміжних галузях; демонстрація міжгалузевого застосування ІТ-інструментів, **включаючи ШІ**.

III. Очікувані результати навчання:

По завершенню курсу учні **будуть здатні:**

- Демонструвати розуміння** базових принципів функціонування Інтернету, архітектури веб-додатків, ключових концепцій штучного інтелекту (включаючи промпт-інжиніринг) та технології блокчейн.
- Розуміти принципи об'єктно-орієнтованого програмування.
- Ефективно використовувати генеративні моделі ШІ** для вирішення навчальних та практичних завдань, формулюючи чіткі запити та аналізуючи отримані результати.
- Розробляти, тестувати та публікувати** прості статичні та адаптивні веб-сайти.
- На базовому рівні використовувати** систему контролю версій Git та платформу GitHub для управління програмними проєктами.
- Здійснювати базовий аналіз** наборів даних, інтерпретувати результати та представляти їх у візуалізованому вигляді.
- Описувати життєвий цикл** розробки простих цифрових продуктів та брати участь у його окремих етапах.
- Критично оцінювати інформацію технологічного характеру, включаючи можливості та обмеження ШІ, та визначати потенціал застосування ІТ для вирішення різноманітних завдань.**

IV. Освітня цінність та актуальність курсу:

Програма курсу "Технологічний Стек Сучасності" відповідає сучасним викликам інформаційного суспільства та запитам ринку праці. **Інтеграція блоку з вивчення штучного інтелекту, від теоретичних основ до практичного застосування, є особливо актуальною, оскільки ШІ трансформує всі сфери життя та професійної діяльності.** Курс забезпечує учнів не лише актуальними предметними знаннями, а й сприяє формуванню ключових компетентностей, визначених концепцією "Нової української школи" та європейськими освітніми рамками. Здобуті навички та знання є універсальними, підвищують адаптивність випускників до умов вищої освіти, розширюють їхні кар'єрні перспективи та закладають основи для навчання впродовж усього життя в епоху цифрових технологій.