

# Основи Scratch

Автор: Марцін Піотрович

## Урок 2:

# Перший скрипт в Scratch

Друге заняття з програмування в Scratch – це продовження праці з анімацією, створеною на першому уроці. Учні починають користуватися все більшою кількістю скриптів, працюють над анімацією тла. Ми також впроваджуємо перші умовні інструкції, котрі дозволять нам створювати більш удосконаленні блоки скриптів.

### Цілі занять:

Учень повинен:

- Знати різні способи запуску скриптів,
- Користуватися блоками з бібліотек рух, вигляд, контроль, події,
- Змінювати тло, впроваджувати елемент анімації,
- Користуватися умовними інструкціями.

### Ключові поняття:

- Сцена, спрайт
- Скрипт
- Контроль, події, сенсор
- Умовна інструкція
- Осі «x» і «y»

### Методи роботи:

- Лекція, дискусія, ведення,
- Презентація результатів роботи,
- Практичні вправи на комп'ютері,
- Мозковий штурм,
- «Навчання через особистий досвід – обмежуємо кількість інформації, яку надаємо, до необхідного мінімуму і намагаємось допомогти учню розпочати самостійну працю» [А. Валат, *Вибрані проблеми дидактики інформатики*].

### Допоміжні матеріали:

- Scratch - [www.scratch.mit.edu](http://www.scratch.mit.edu)
- Оснащення комп'ютерного класу (2 учнів на 1 робоче місце)

Час реалізації занять 45 хв.



### Зміст програми (зв'язок з основною програмою)

Базова програма загальної освіти для початкової школи – II етап навчання – IV–VI класи. Детальний зміст комп'ютерних занять:

3.1 Пошук та використання інформації з різних джерел. Учень: шукає інформацію в різних електронних джерелах (словники, енциклопедії, бібліотеки, технічна документація та засоби інтернету);

4.1 Опрацювання за допомогою комп'ютера малюнків, історій, текстів, анімацій, мультимедійних презентацій і цифрових даних. Учень:

- створює малюнки і теми за допомогою графічного реактора (використовуючи форми, кольори, зміну малюнків, фрагменти інших малюнків);

1. Вирішення проблем і прийняття рішень з використанням комп'ютера. Учень: за допомогою ряду команд створює прості теми, або керує об'єктом на екрані.

2. Вирішення проблем і прийняття рішень з використанням комп'ютера. Учень: приймає участь в командній роботі, під час реалізації спільного проекту спілкується з іншими особами, приймає рішення в межах своїх завдань і дозволів.

3. Використання комп'ютера та інформаційно-комунікаційних технологій для розвитку своїх улюблених занять, застосування комп'ютера в повсякденному житті, окреслення загроз і обмежень, пов'язаних з користуванням комп'ютером та інтернетом. Учень: шанує приватність і працю інших осіб;

4. Використання комп'ютера та інформаційно-комунікаційних технологій для розвитку своїх улюблених занять, застосування комп'ютера в повсякденному житті, окреслення загроз і обмежень, пов'язаних з користуванням комп'ютером та інтернетом. Учень: дотримується етичних та юридичних правил, пов'язаних з користуванням комп'ютером та інтернетом, оцінює можливі загрози.

## Перебіг заняття:

### 1. Введення в тематику та інтеграція групи – 12 хв.

Заняття починаємо з пригадування результатів нашої роботи з першого заняття.

Учні входять в свої облікові записи. Тих, хто працював над своїм проектом вдома, заохочуємо до демонстрації своєї міні-програми. Ті, хто завантажив чужий проект і здійснив у ньому зміни, також можуть похвалитися своїми досягненнями. Попросіть учнів, щоб показали, яким чином вони запрограмували окрему функцію програми.

Безумовно, для частини дітей це буде момент пригадування собі способу логування на сторінці [scratch.mit.edu](https://scratch.mit.edu), чи закінчення роботи з попереднього заняття. Ми дозволяємо учням на індивідуальний підхід до роботи.

### 2. Основна частина – 25 хв.

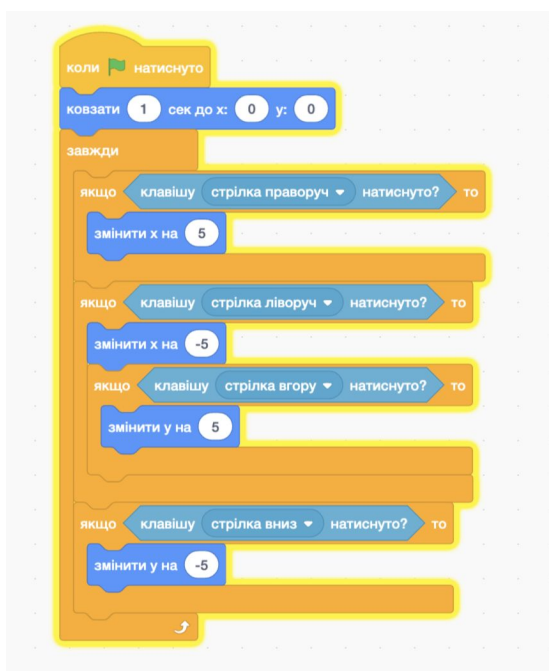
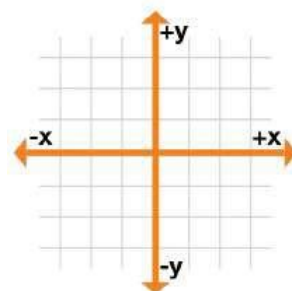
#### Управління спрайтом

На цьому етапі роботи ми повертаємось до нашої анімації з першого заняття.

Питаємо учнів, чи грають вони в гру Minecraft (там задіяні координати  $x$ ,  $y$ ,  $z$ ), чи мають досвід гри в шахи/шашки або заняття з графічними диктантами. Нехай поміркують над тим, що об'єднує всі ці види діяльності, звідки ми знаємо де знаходиться фігура в шахах, чи окремий колір в графічному диктанті. Щоб краще керувати нашим спрайтом в Scratch ми впроваджуємо декартові координати і пояснюємо учням, що кожна крапка на сцені має свою «адресу» (так само, як кожна клітинка на шаховій дошці чи в графічному диктанті). Можна на дошці накреслити решітку з вісями  $x$  і  $y$  і дозволити учням визначити положення кількох об'єктів.

Модифікуємо скрипт з попереднього уроку так, щоб стало можливим керування спрайтом за допомогою стрілок клавіатури. Дозволяємо учням провести мозковий штурм і запропонувати, як міг би виглядати такий скрипт. Варто також додати блок, котрий призведе до того, що спрайт завжди на початку гри буде стартувати з середини екрану. В запропонованому на першому уроці скрипті це був водолаз. Розмірковуємо на тим, що має статися, коли водолаз торкнеться краю екрану?

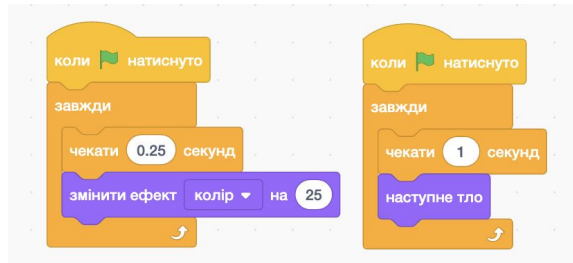
Пояснюємо учням, що представляють собою блоки умовних інструкцій (в скриптах «контроль»). Зразок виконання завдання може виглядати наступним чином:



### «Анімація» тла

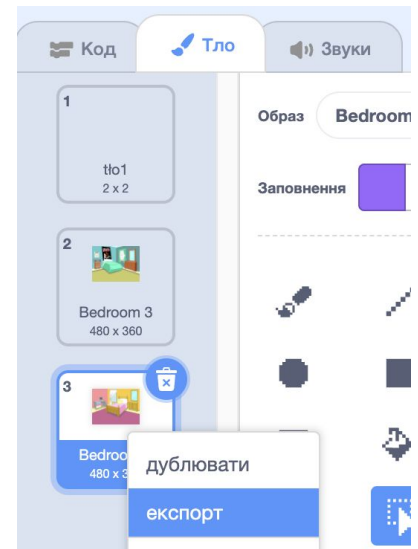
Варто запитати учнів, чи задоволені вони постійним тлом, яке не змінюється? Чи можна його якось анімувати? Учні можуть запропонувати два рішення:

Змінюючи колірний ефект



Змінюючи фон сцени

В цьому випадку варто показати спосіб модифікування тла: клацаємо на сцені, входимо в закладку «образи», далі кілька разів дублюємо сцену і в кожному дублі робимо якісь зміни (напр., домальовуємо якісь елементи).



### 3. Підсумок і оцінка – 8 хв.

Вчитель перевіряє прогрес, досягнений учнями в праці. Заохочує їх до закінчення завдань вдома. Вчитель просить учнів перед закінченням праці, вписати до опису коротку інструкцію гри.

Вчитель задає також оціночні питання:

- хто навчився чомусь новому в Scratch;
- хто відкрив для себе щось нове самостійно;
- що особливо зацікавило при виконанні завдань;
- що можна було б змінити на наступних уроках.

### Зауваження/альтернативи:

Ймовірно, деякі учні раніше познайомились з програмуванням в Scratch. Якщо вони виконали швидко згадані вище завдання, то можуть самостійно впроваджувати нові елементи. В нашій грі це може бути, наприклад, якась перешкода, котру потрібно обминати, простий скрипт якої може виглядати наступним чином:

