



**АЛГОРИТМ ІЗ  
ПОВТОРЕННЯМ**

**ПОВТОРЕННЯМ**



# Використання структури повторення

Алгоритм, у якому передбачається *багаторазове* виконання одного й того самого набору команд, називають **циклічним**.

A diagram illustrating a cycle. On the left is a large white bucket with a red lid, labeled "80 л" in red text. On the right is a smaller metal bucket, labeled "10 л" in red text. A blue curved arrow points from the top of the large bucket to the top of the small bucket, and a grey box with the text "8 разів" is positioned between them, indicating that the smaller bucket is used 8 times to empty the larger one.

8 разів

80 л

10 л

A diagram illustrating a cycle. On the left, a green oval contains the text "Фарбуван ня стовпчика огоро" (Painting fence posts). To the right is a black and white illustration of a man in a hat painting a fence. Below the illustration is a small image of several paint cans and a brush.

Фарбуван  
ня  
стовпчика  
огоро

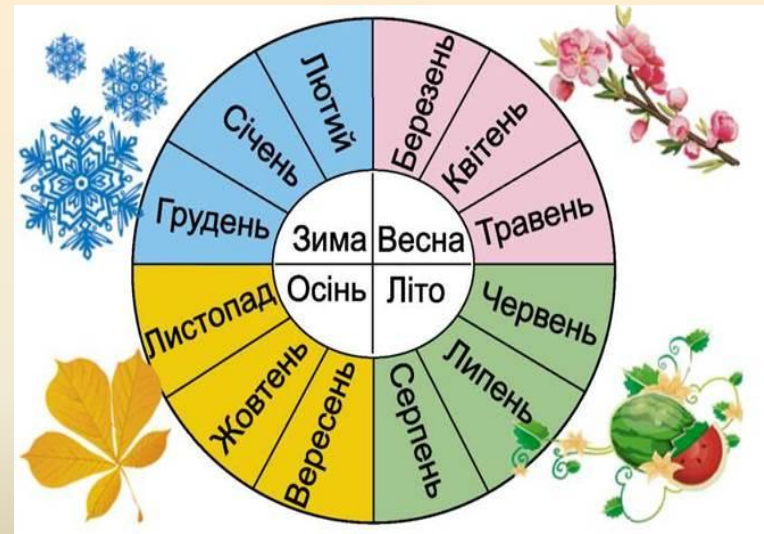
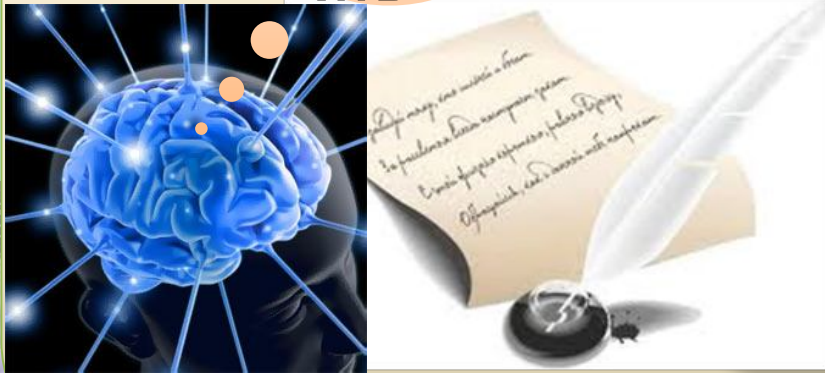
Повторення



# Використання структури повторення

Алгоритм, у якому передбачається *багаторазове* виконання одного й того самого набору команд, називають **циклічним**.

Вивчити  
вірш напам'  
ять



Повторення

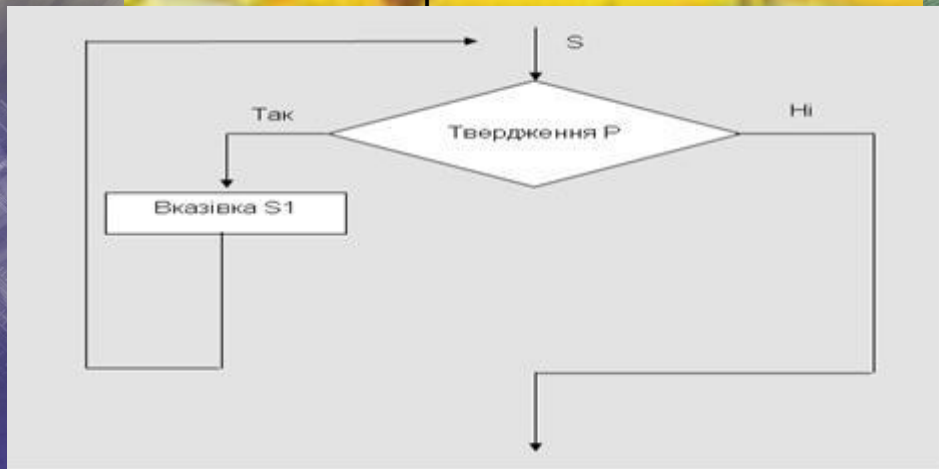


# Використання структури повторення

Для опису циклічних алгоритмів використовують алгоритмічну структуру повторення

**Повторення** – базова алгоритмічна структура, призначена для організації багаторазового виконання набору команд

«Багаторазове виконання» означає, що команди будуть виконуватися скінченну кількість разів





# Як розрізняти повторення?

## Кількість повторень

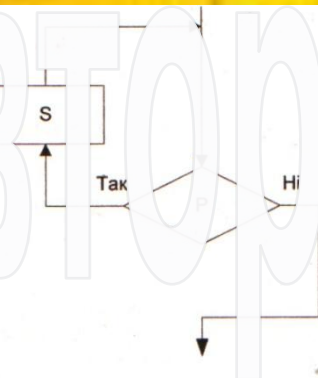
Команди циклу будуть повторюватись до досягнення **потрібної кількості** повторень

Повторення з **визначеною** кількістю повторень

Повторення з **невідомою** кількістю повторень

Для припинення циклу **задається** деяка **умова**, яка і забезпечує скінченність виконання команд, що **повторюються**

# ПОВТОРЕННЯ



**Повторення** — базова алгоритмічна структура, призначена для організації багаторазового виконання набору команд.





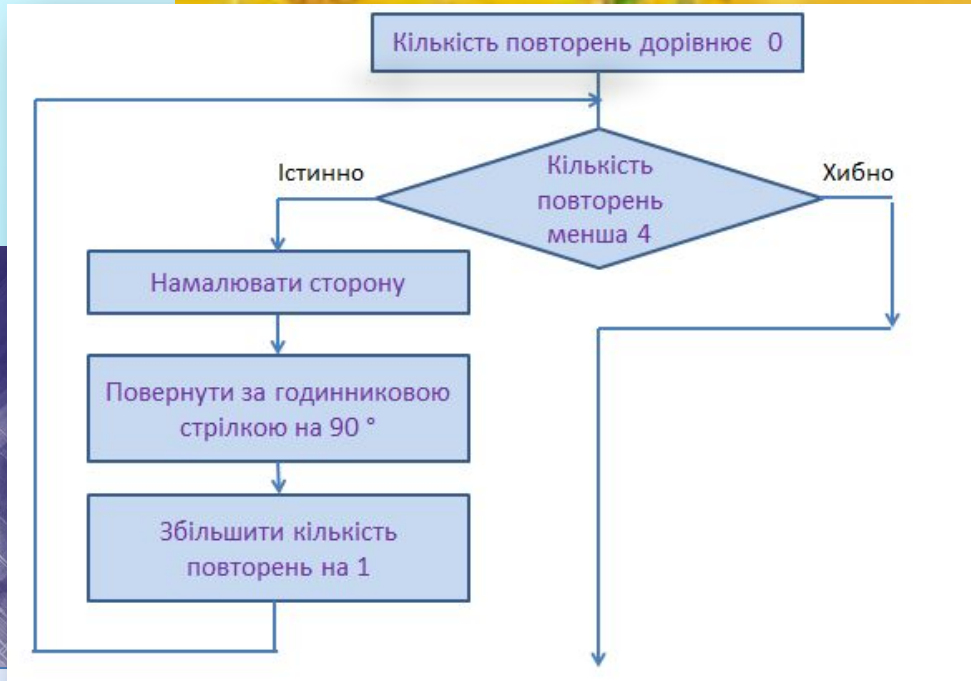
# Як розрізнати повторення?

## Кількість повторень

Команди циклу будуть повторюватись до досягнення **потрібної кількості** повторень

Повторення з визначеною **кількістю повторень**

Приклад



**Повторення** — базова алгоритмічна структура, призначена для організації багаторазового виконання набору команд.





# Як розрізнати повторення?

## Кількість повторень

Для припинення циклу **задається** деяка **умова**, яка і забезпечує скінченність виконання команд, що повторюються

Приклад

Повторення з **невідомою** кількістю повторень



**Повторення** — базова алгоритмічна структура, призначена для організації багаторазового виконання набору команд.



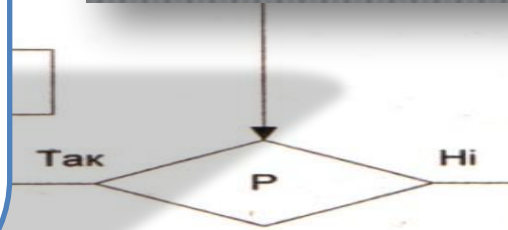
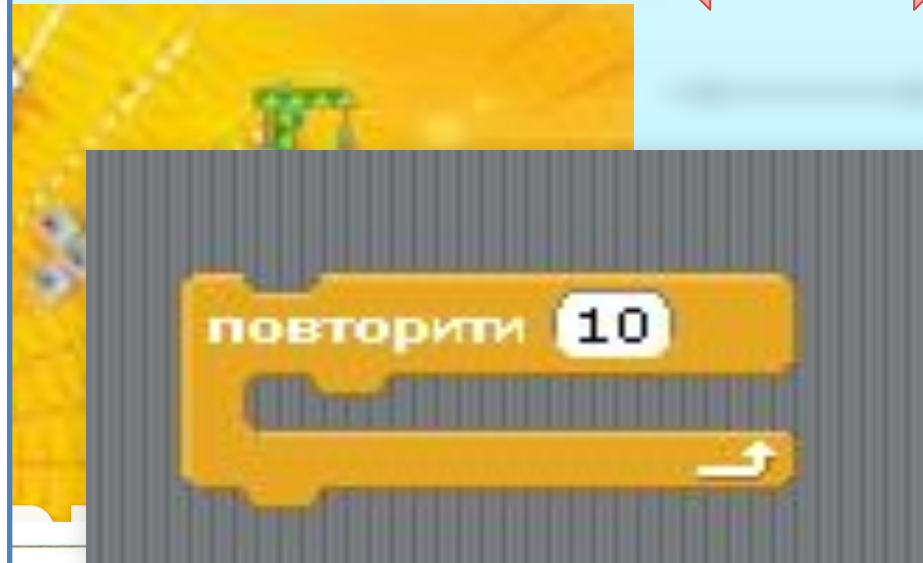
# Як створити циклічний алгоритм?

Визначена кількістю повторень

Для створення циклічних алгоритмів з визначеною кількістю повторень у середовищі Скретч використовують команда **Повторити  $K$** , де параметр  $K$  вказує на кількість повторень у тілі циклу.



Приклад



**Повторення** — базова алгоритмічна структура, призначена для організації багаторазового виконання набору команд.



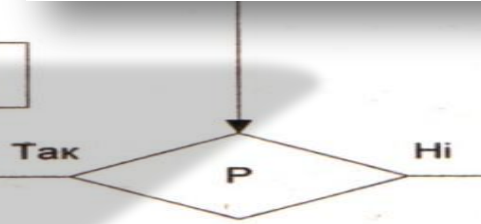
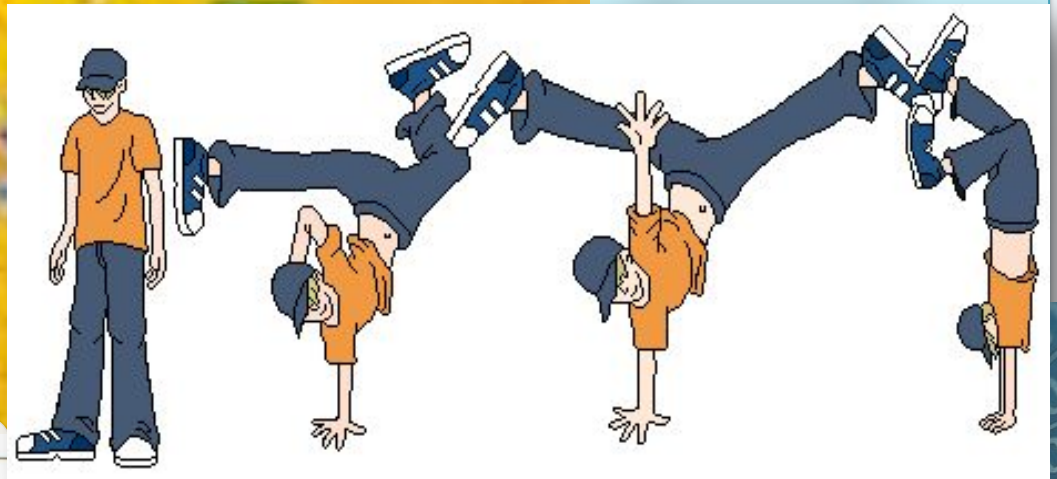


# Як створити циклічний алгоритм?

Визначена кількістю повторень

Приклад

Виконавець  
**Танцюрист**  
змінюватиме  
зовнішній вигляд  
4 рази з  
інтервалом 1  
секунду

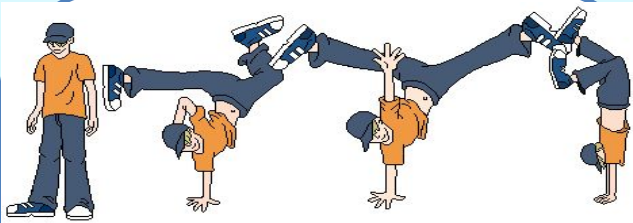


**Повторення** — базова алгоритмічна структура, призначена для організації багаторазового виконання набору команд.



# Як створити циклічний алгоритм?

Визначена кількістю повторень



Виконавець  
**Танцюрист**  
змінюватиме  
зовнішній вигляд 4  
рази з інтервалом 1  
секунду



Такий алгоритм можна  
подати графічно

Приклад

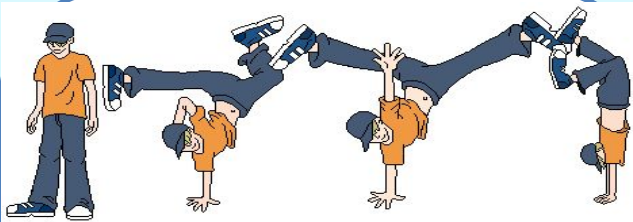


**Повторення** — базова алгоритмічна структура, призначена для організації багаторазового виконання набору команд.



# Як створити циклічний алгоритм?

Визначена кількістю повторень



Виконавець  
**Танцюрист**  
змінюватиме  
зовнішній вигляд 4  
рази з інтервалом 1  
секунду



Алгоритм у середовищі  
**Скретч**

Приклад

повторити 4

наступний образ

чекати 1 секунд

**Повторення** — базова алгоритмічна структура, призначена для організації багаторазового виконання набору команд.

FALSE

TRUE



# Як створити циклічний алгоритм?

Невизначена кількістю повторень

У разі, коли в алгоритмі **кількість повторів заздалегідь не відома**, у середовищі **Скретч** використовують команду

**Завжди**



Приклад

завжди

Так P Ні

**Повторення** — базова алгоритмічна структура, призначена для організації багаторазового виконання набору команд.

FALSE

TRUE

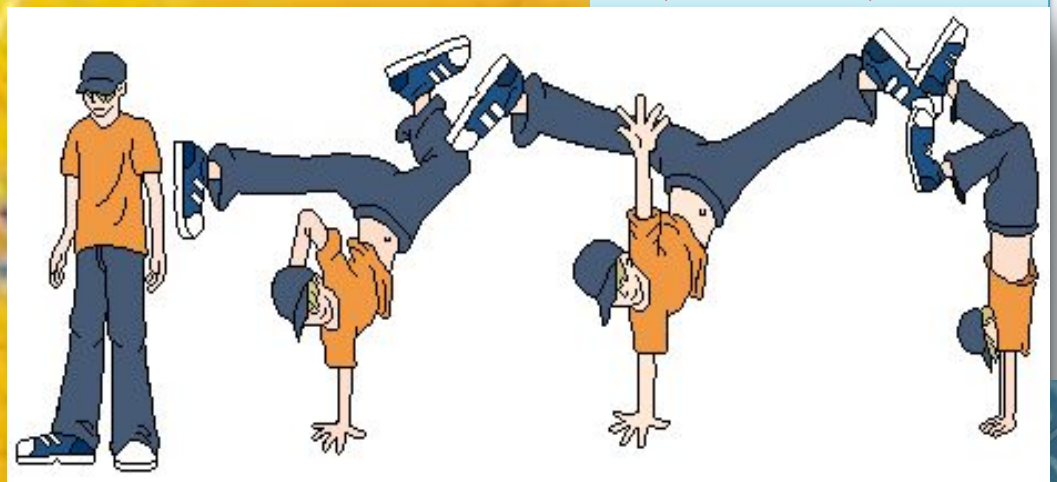


# Як створити циклічний алгоритм?

Невизначена кількістю повторень

Приклад

Виконавець  
**Танцюрист**  
змінюватиме  
зовнішній вигляд  
4 рази з  
інтервалом 1  
секунду



Щоб **зупинити** виконання команд, розміщених у тілі такого циклу, користувачу слід натиснути кноп



**Повторення** — базова алгоритмічна структура, призначена для організації багаторазового виконання набору команд.

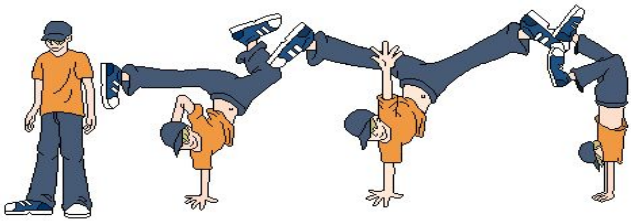
FALSE

TRUE



# Як створити циклічний алгоритм?

## Невизначена кількістю повторень



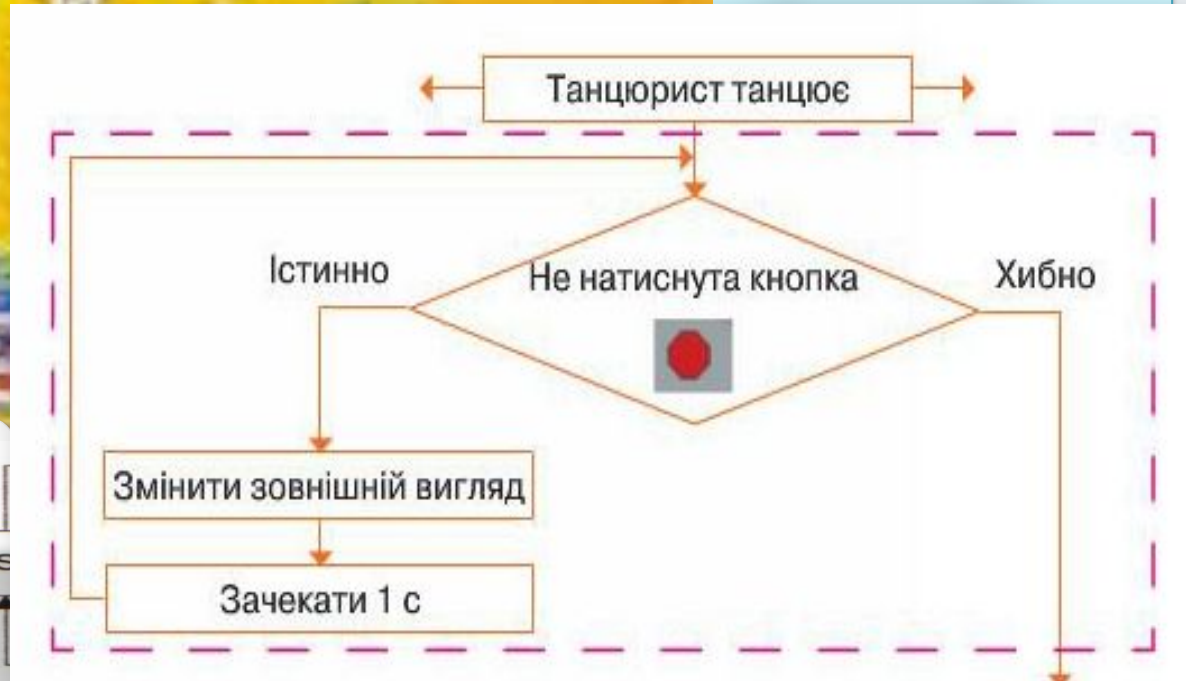
Виконавець  
**Танцюрист**  
змінюватиме  
зовнішній вигляд 4  
рази з інтервалом 1  
секунду

SCRATCH



Такий алгоритм можна подати графічно

Приклад

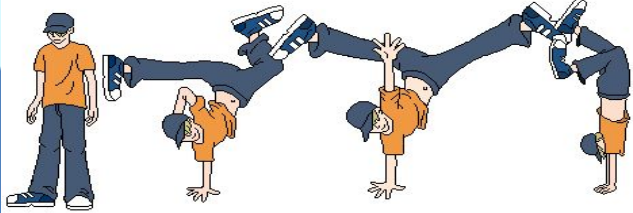


**Повторення** — базова алгоритмічна структура, призначена для організації багаторазового виконання набору команд.



# Як створити циклічний алгоритм?

## Невизначена кількістю повторень



Виконавець  
**Танцюрист**  
змінюватиме  
зовнішній вигляд 4  
рази з інтервалом 1  
секунду



Алгоритм у середовищі  
**Скретч**

Приклад

завжди

наступний образ

чекати 1 секунд

**Повторення** — базова алгоритмічна структура, призначена для організації багаторазового виконання набору команд.

FALSE

TRUE



# Домашнє завдання



Виконати перші 5 вправ  
в середовищі Blockly.