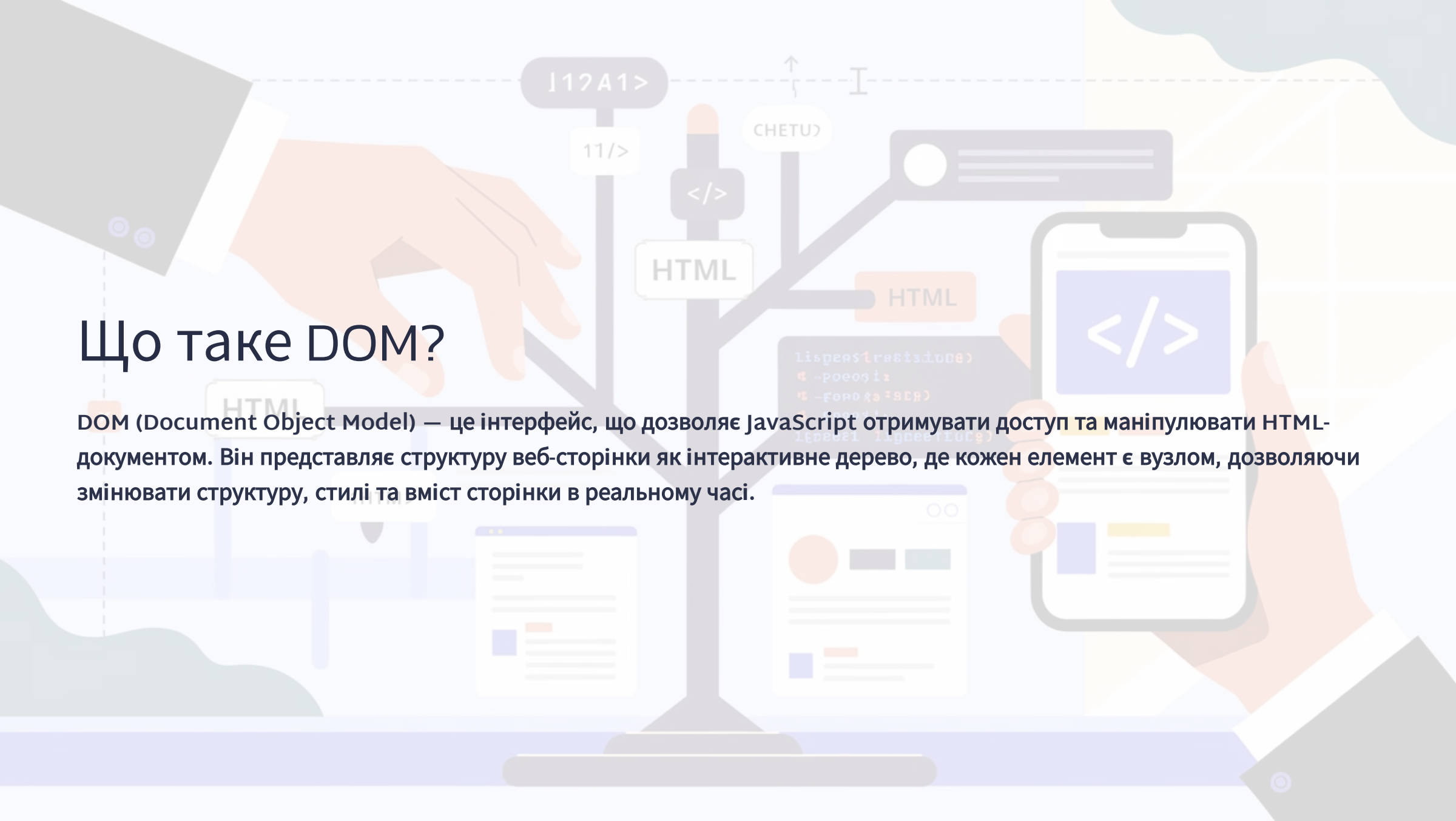


# Основи роботи з DOM

DOM - це програмна інтерфейсна модель для HTML-документів. Вона представляє структуру веб-сторінки як дерево об'єктів, дозволяючи JavaScript маніпулювати елементами в реальному часі.

# Що таке DOM?

DOM (Document Object Model) — це інтерфейс, що дозволяє JavaScript отримувати доступ та маніпулювати HTML-документом. Він представляє структуру веб-сторінки як інтерактивне дерево, де кожен елемент є вузлом, дозволяючи змінювати структуру, стилі та вміст сторінки в реальному часі.



# Доступ до елементів: querySelector

`document.querySelector`

Вибирає перший елемент, що відповідає заданому CSS-селектору. Наприклад, `document.querySelector('#header')` вибере елемент з `id="header"`.

`document.querySelectorAll`

Вибирає всі елементи, що відповідають заданому CSS-селектору, і повертає `NodeList`. Наприклад, `document.querySelectorAll('.item')` вибере всі елементи з `class="item"`.



# Доступ до елементів: getElement методи



## getElementsByClassName



## getElementsByTagName

Вибирає елементи за тегом.

# Зміна вмісту елементів

## innerHTML

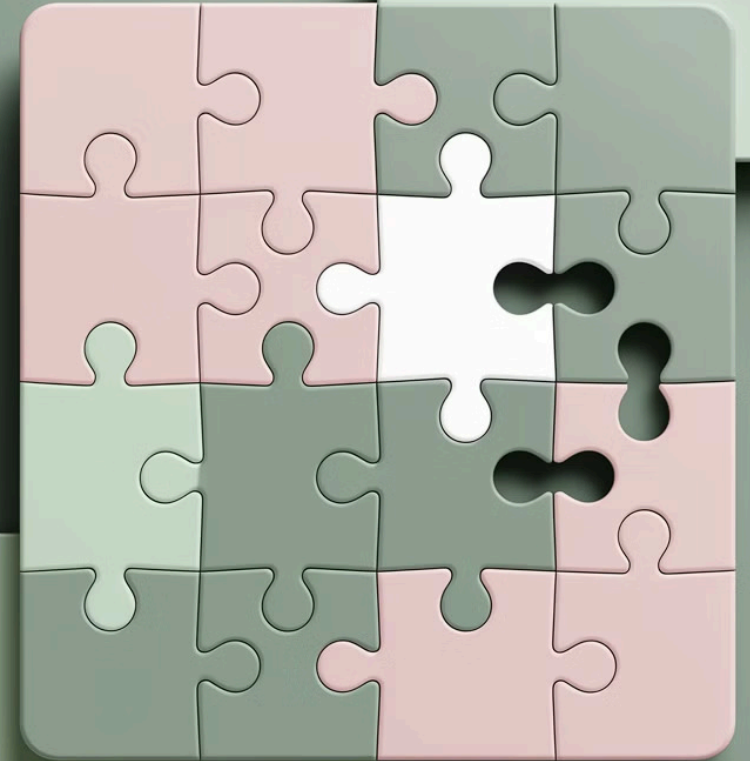
- **innerHTML:** Змінює HTML-вміст елемента. Наприклад: `document.getElementById('example').innerHTML = '<p>Новий текст</p>';` замінить вміст `div` на параграф з текстом "Новий текст".

## textContent

- **textContent:** Змінює лише текст в елементі, ігноруючи HTML-теги. Наприклад: `document.getElementById('example').textContent = '<p>Новий текст</p>';` відобразить текст "<p>Новий текст</p>" як звичайний текст. Безпечніше, якщо не потрібно виконувати HTML.

# Додавання та видалення елементів

- 1 `createElement`  
Створює новий елемент.
- 2 `appendChild`  
Додає новий елемент як дочірній до існуючого.
- 3 `removeChild`  
Видаляє елемент зі сторінки.



# Обробка подій

## addEventListener

Використовується для приєднання обробника подій до вказаного елемента. Наприклад,  
``element.addEventListener('click', handleClick);``

## Типи подій

Клік (``click``), наведення миші (``mouseover``), зміна форми (``change``), завантаження сторінки (``load``), натискання клавіші (``keydown``) тощо.

## Функція-обробник

Функція, яка викликається при виникненні події. Може приймати об'єкт події (``event``) як аргумент для отримання додаткової інформації.

1

2

3



# Робота з атрибутами

1

`getAttribute`

Отримання значення атрибута

2

`setAttribute`

Встановлення значення атрибута

3

`removeAttribute`

Видалення атрибута



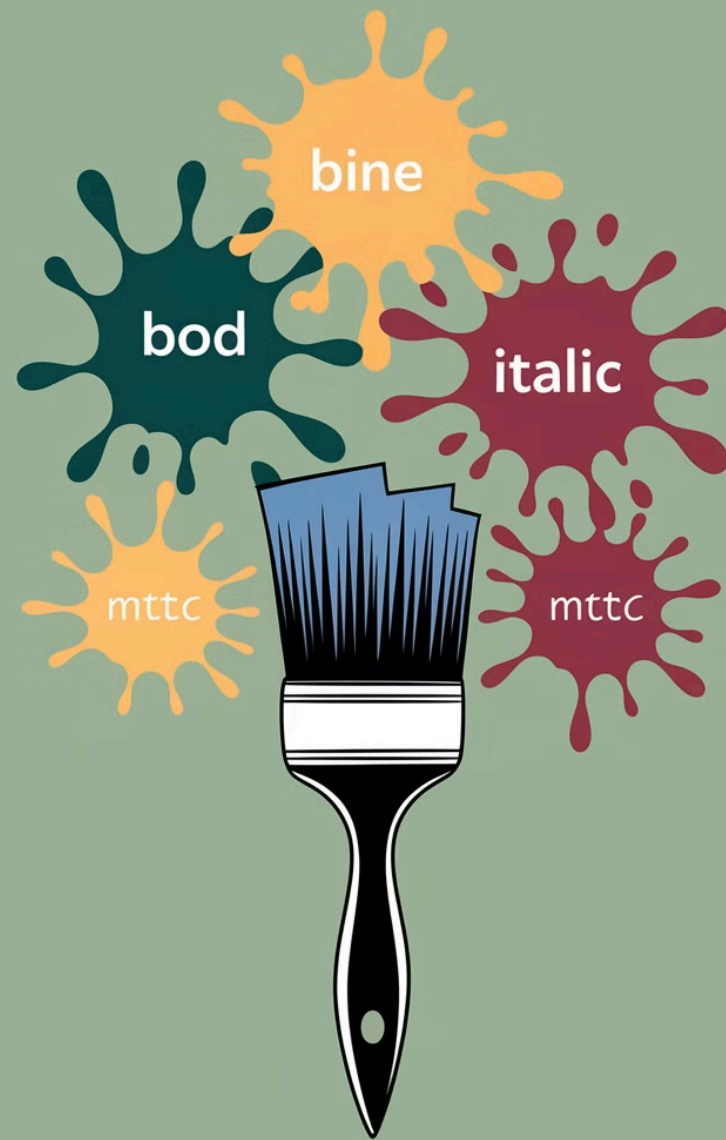
# Зміна стилів

`style`

Змінює інлайн-стили елемента.

`classList`

Дозволяє додавати, видаляти або перемикати класи.





# Взаємодія з формами

## Н

value

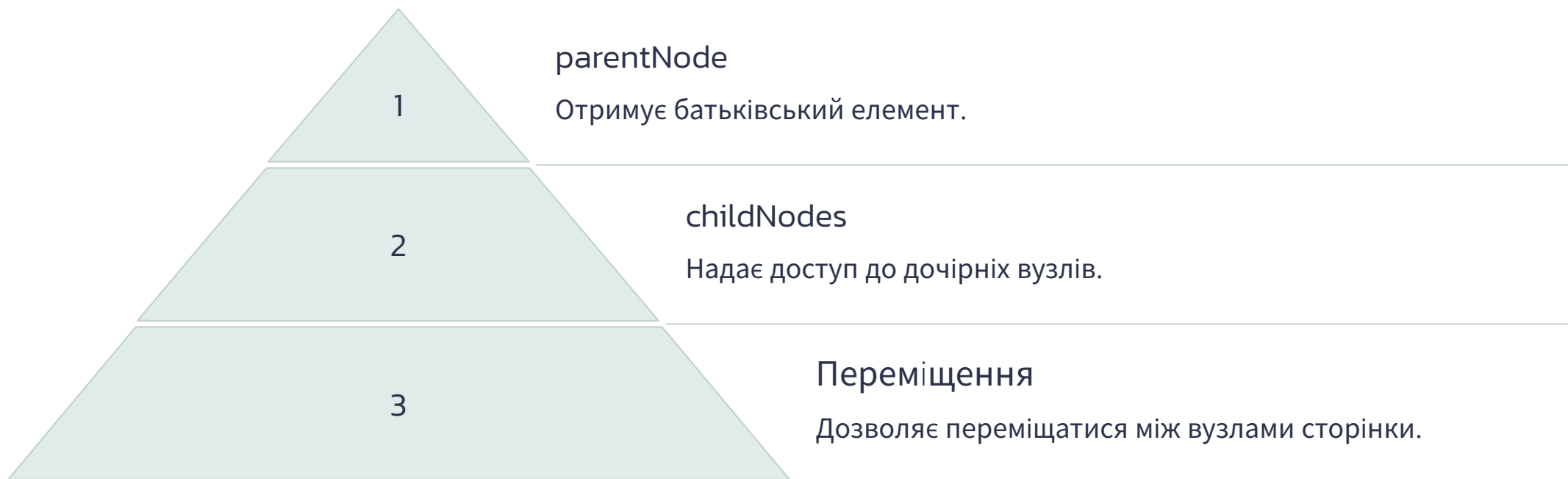
Отримує або змінює значення елементів форм.

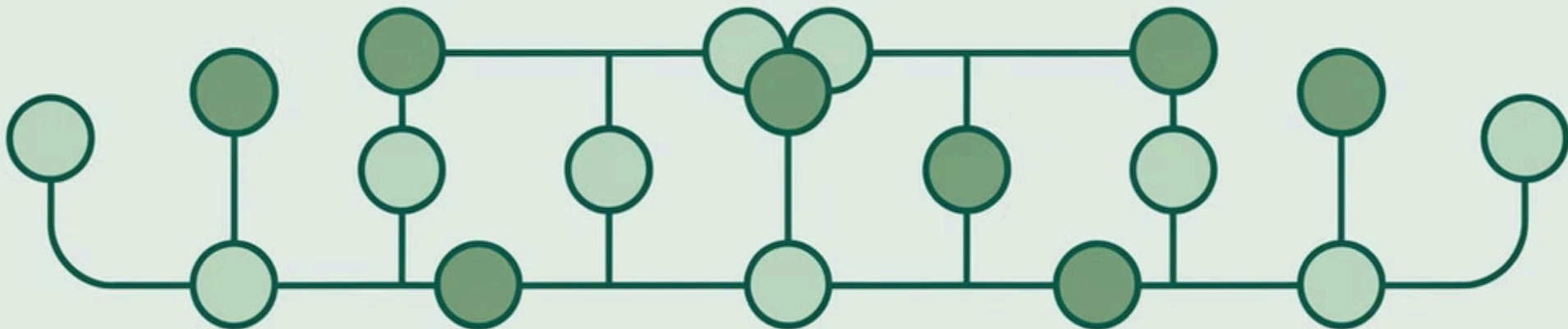
## Н

submit

Обробляє дані, введені у форми.

# Навігація DOM: Батьківські елементи





## Навігація DOM: Сусідні елементи

1

`nextSibling`

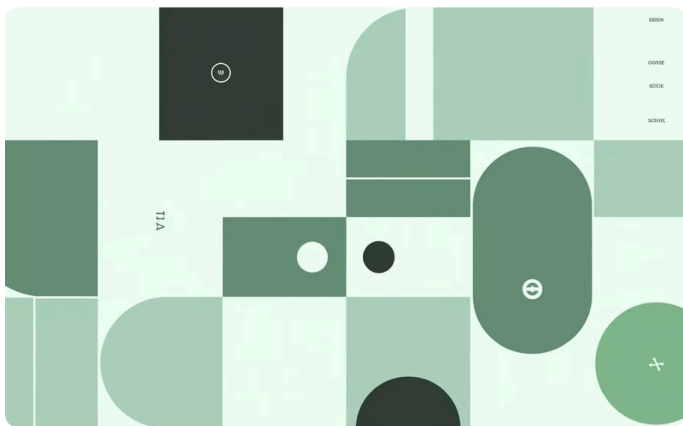
Отримує наступний сусідній вузол.

2

`previousSibling`

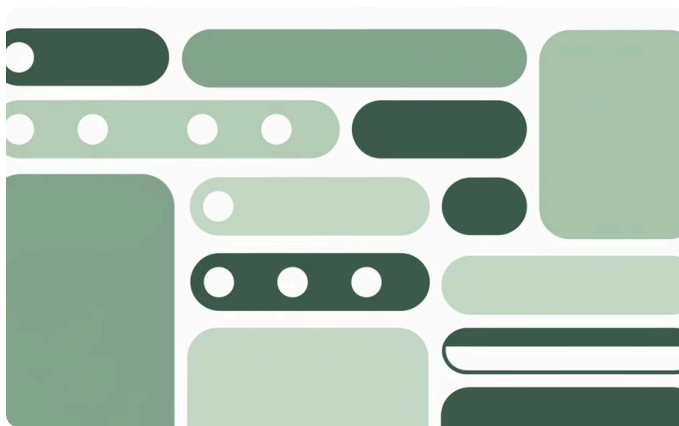
Отримує попередній сусідній вузол.

# Практичне застосування DOM



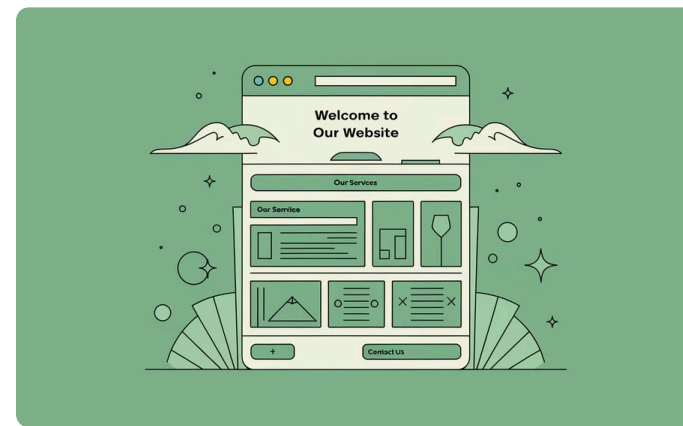
## Динамічні веб-сторінки

Створення інтерактивних елементів без перезавантаження.



## Валідація форм

Перевірка введених даних у реальному часі.



## Анімації

Створення плавних переходів та ефектів.

# Висновки

- 1 Потужний інструмент  
DOM - ключовий елемент  
для створення динамічних  
веб-сторінок.
- 2 Широкі можливості  
Дозволяє маніпулювати  
структурою, стилями та  
вмістом HTML.
- 3 Основа веб-розробки  
Важливий для розуміння та ефективної роботи з JavaScript.

