

# Основи роботи з React компонентами

**№ уроку:** 3 **Курс:** React Базовий

**Засоби навчання:** Visual Studio Code, браузер, пакет менеджер node.

## Огляд, ціль та призначення уроку

Ознайомитись з компонентами та їх типами в реакті. Навчитись розробляти компоненти багаторазового використання. Ознайомитись з методами передачі даних між компонентами. Попрактикуватись в проектуванні та розробці веб-застосунків з використанням компонентів для футеру, навігації, контенту та хедеру.

## Вивчивши матеріал даного заняття, студент зможе:

- Розробляти React компоненти.
- Володіти технологією передачі даних між компонентами.
- Вміти структурувати проекти на реакті.

## Зміст уроку

1. Визначення компонентів в React.
2. Ознайомлення з функціональними та класовими компонентами.
3. Передача даних між компонентами (робота з об'єктом props).
4. Застосування отриманих знань на практиці.

## Резюме

- Компоненти - це незалежні елементи інтерфейсу, які можуть функціонувати та розроблятися незалежно один від одного.
- Головною перевагою такого методу побудови веб-застосунків є можливість повторного використання коду (компонентів) та розподіленість при розробці.
- Компоненти в реакті можна створювати двома способами: у функціональному стилі програмування та у об'єктно орієнтованому стилі програмування. На сьогоднішній день найбільш актуальним способом створення компонентів є функціональний стиль, але в деяких випадках повністю відмовитись від ООП стилю неможливо.
- Архітектура реакт-застосунків дозволяє передавати дані між компонентами. Для цього використовується об'єкт props. Пропси призначені лише для читання, їх не можна змінювати,
- Пропси передаються тільки в односторонньому порядку, хоч і існують способи для передачі даних в батьківський компонент із дочірнього, робити це не передбачено документацією бібліотеки React.

## Закріплення матеріалу

- За допомогою якого стилю програмування розробляються компоненти в сучасних React-застосунках?

- Як передати дані з одного компоненту в інший?
- Чи можна змінювати об'єкт props?
- Чи дозволяється передавати дані із дочірнього компоненту в батьківський?
- Чи можна використовувати тип файлу .jsx при створенні компоненту?

### Додаткове завдання

#### Завдання

Модифікувати практичний приклад. За аналогією до компоненту Articles.js розробити компонент багаторазового використання Topics.js, який буде повертати посилання на сторінки для компоненту Sidebar.js.

### Самостійна діяльність студента

#### Завдання 1

Ознайомитись з основними поняттями, розглянутими на уроці.

#### Завдання 2

Повторити всі практичні приклади уроку.

#### Завдання 3

Взяти будь-який веб-сайт та проаналізувати які компоненти могли бути використані при його розробці.

### Рекомендовані ресурси

Компоненти в реакт

[https://www.w3schools.com/react/react\\_components.asp#](https://www.w3schools.com/react/react_components.asp#)

Об'єкт props

[https://www.w3schools.com/react/react\\_props.asp](https://www.w3schools.com/react/react_props.asp)

Компоненти і пропси

<https://uk.reactjs.org/docs/components-and-props.html#gatsby-focus-wrapper>