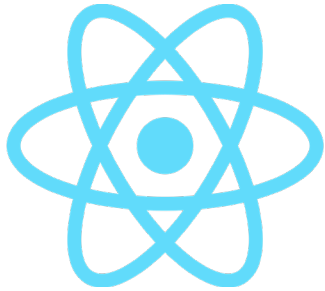




**Microsoft** Partner  
Silver Learning



# React Базовий

Розробка застосунку для списку нотатків

# React Базовий

## Introduction

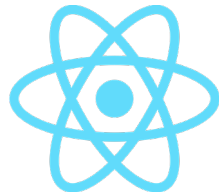


Кочев Денис

Web-розробник в Data Bridge

Магістр комп'ютерних наук Національного аерокосмічного університету ім. М.Є Жуковського

 [denkochev](https://www.linkedin.com/in/denkochev)



# React Базовий

## Тема уроку

Розробка застосунку для списку нотатків

# React Базовий

## План уроку

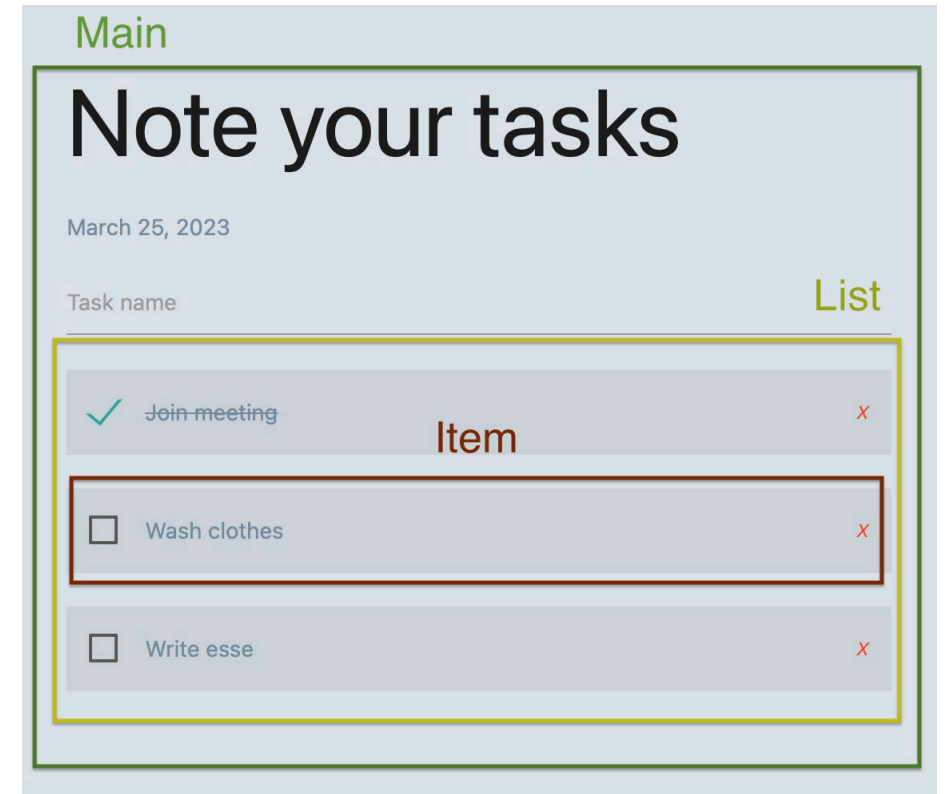
1. Структура застосунку
2. Правила ефективного коду
3. Функціонал додатку
4. Хуки
5. Візуальна складова застосунку

# React Базовий

## Структура застосунку

Перш за все, створюючи систему з комплексним функціоналом, потрібно розподілити розробку на декілька етапів:

- Визначити архітектуру проекту.
- Розподілити компоненти у відповідності до їх функціоналу.
- Визначити пропси до базових компонентів та створити стани.
- Синхронізувати додаток із зовнішніми системами за допомогою useEffect.



*Розбиття на компоненти*

# React Базовий

## Правила ефективного коду

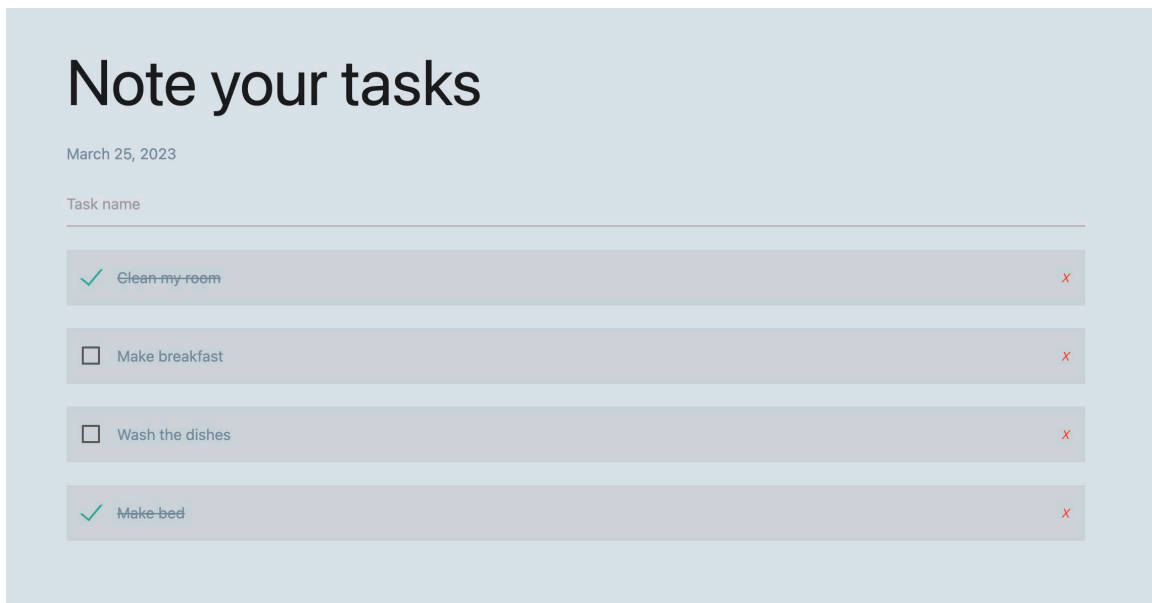
Існує декілька правил розробки React-застосунків, дотримуючись яких додаток буде працювати ефективніше та матиме код, що легко сприйматиметься сторонніми розробниками. Якісно написаний код легше масштабувати та підтримувати.

1. Використовуйте нативні JS методи для роботи з масивами. Якщо вам потрібно обробити масив даних, спробуйте зробити це через `map` або `filter` замість `for` або `while`.
2. Надавайте вашим об'єктам унікальні ідентифікатори, не використовуйте як ідентифікатор позицію елементу в масиві. Для унікального нумерування елементів в React існує метод `uuidv4()`.
3. При роботі з компонентами багаторазового використання або списками як атрибут `key` не використовуйте його індекс.

# React Базовий

## Функціонал додатку

Застосунок для списку нотатків повинен мати можливість додавати нові завдання до списку, відмічати виконані та видаляти записи зі списку. Записи даного застосунку будуть зберігатись в локальному сховищі `localStorage`, що дасть змогу зберігати записи при кожному оновленні сторінки.



*Інтерфейс застосунку*

# React Базовий

## Хуки

Для розробки додатку такого рівня функціоналу нам знадобляться хуки `useEffect` та `useState`.

Хуки станів `useState` в даному застосунку будуть зберігати всі записи:

```
const [tasks, setTodos] = useState(() => {
  const storedTodos = localStorage.getItem('tasks');
  if(!storedTodos){
    return []
  } else {
    return JSON.parse(storedTodos)
  }
});

const [tasksTitle, setTaskTitle] = useState('');
```

За допомогою хуку `useEffect` ми зможемо синхронізувати взаємодію стану та локального сховища:

```
useEffect(()=>{
  localStorage.setItem('tasks',JSON.stringify(tasks))
},[tasks])
```



# React Базовий

## Візуальна складова застосунку

Щоб отримати найбільш оптимальний набір стилів за обмежений проміжок часу, ми скористаємося front-end фреймворком Materialize. Для роботи з пакетом стилів даного фреймворку його потрібно встановити та імпортувати.

Для встановлення скористаємося наступною npm командою:

```
npm install materialize-css jquery
```

Імпортувати materialize потрібно в файл з css класами та в index.js :

```
@import "materialize-css/dist/css/materialize.min.css";
```

```
import 'materialize-css';
```

# Інформаційний відеосервіс для розробників програмного забезпечення



# Перевірка знань

TestProvider.com



Перевірте, як ви засвоїли даний матеріал на [TestProvider.com](https://testprovider.com)

TestProvider – це online-сервіс перевірки знань з інформаційних технологій. За його допомогою ви можете оцінити свій рівень та виявити слабкі місця. Він буде корисним як у процесі вивчення технології, так і для загальної оцінки знань IT-спеціаліста.

Успішне проходження фінального тестування дозволить вам отримати відповідний сертифікат.

# React Базовий

Дякую за увагу! До нових зустрічей!



Кочев Денис  
Web-розробник в Data Bridge

