

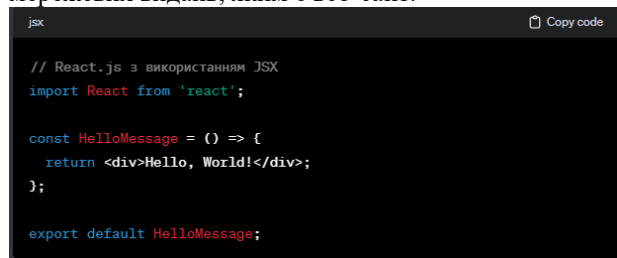
## ПЕРЕВАГИ ВИКОРИСТАННЯ ФРЕЙМВОРКУ REACT.JS У РОЗРОБЦІ НАВЧАЛЬНИХ ВЕБ-САЙТІВ

Фреймворк React.js відіграє ключову роль у спрощенні розробки та підтримки навчальних веб-сайтів. Він надає потужний інструментарій для створення динамічного та чуйного інтерфейсу, що є надзвичайно важливим у сучасному навчальному процесі.

Однією з ключових переваг React.js є його компонентний підхід до розробки. Завдяки компонентам, розробники можуть створювати частини веб-сайту, які можна повторно використовувати, що робить процес розробки більш ефективним та швидким. Наприклад, у навчальному веб-сайті можна створити компоненти для відображення уроків, завдань чи тестів, які можна легко використовувати на різних сторінках сайту. В цілому, модульна структура React.js дозволяє легко масштабувати і підтримувати сайт.

Також, React.js забезпечує чудову підтримку для роботи з великими обсягами даних. У навчальних веб-сайтах часто використовуються великі набори інформації, і фреймворк дозволяє ефективно керувати цими даними, забезпечуючи швидкий доступ до них та високу продуктивність веб-сайту [1].

Використання звичайного JavaScript генерує більший об'єм коду для досягнення тих же самих результатів. На відміну від цього React.js використовує декларативний підхід та JSX (розширення JavaScript), який дозволяє оптимізувати об'єм коду, що у свою чергу скорочує час завантаження - критично важливого показника для мережевих видань, яким є веб-сайт.



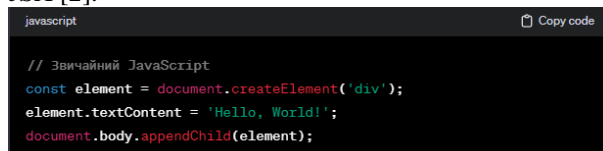
```
jsx
// React.js з використанням JSX
import React from 'react';

const HelloMessage = () => {
  return <div>Hello, World!</div>;
};

export default HelloMessage;
```

Рис. 1. Відрізок коду для створення простого елемента у React.js за допомогою JSX

Можна порівняти цей код з аналогічним у звичайному JavaScript без використання React.js та JSX [2].



```
javascript
// Звичайний JavaScript
const element = document.createElement('div');
element.textContent = 'Hello, World!';
document.body.appendChild(element);
```

Рис. 2. Відрізок JS-коду для створення простого елемента без використання React.js та JSX

Слід помітити, що у React.js код для створення елемента є більш декларативним та зрозумілим. Замість прямої маніпуляції DOM, React використовує JSX для створення структури компонентів, що робить код більш читабельним та зручним для розробки та обслуговування.

Ще одною перевагою React.js є типізація даних, що забезпечує більшу надійність та передбачуваність коду. У випадку суворої типізації даних виключається можливість виникнення runtime-помилки [3].

Узагальнюючи, слід зазначити, що React.js значно полегшує процес розробки навчальних веб-сайтів завдяки своїм передовим інструментам, компонентному підходу, ефективному керуванню даними та швидкості відображення змін. Це дозволяє розробникам створювати більш ефективні та користувальницько-орієнтовані освітні платформи, сприяючи покращенню якості навчання та зростанню зацікавленості до навчального матеріалу.

### Список літератури

1. Документація React: <https://reactjs.org/docs/getting-started.html>
2. "React - The Complete Guide (incl Hooks, React Router, Redux)" (Udemy): <https://www.udemy.com/course/react-the-complete-guide-incl-redux/>
3. Чому важлива типізація в JS: <https://foxminded.ua/tipizaciya-js/>

Науковий керівник: проф. Єгорова І.М.