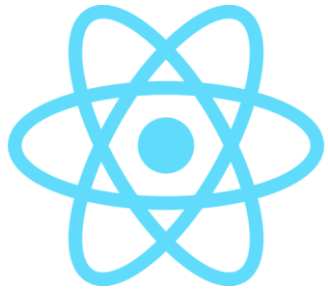




**Microsoft** Partner  
Silver Learning



# React Базовий

Знайомство з React

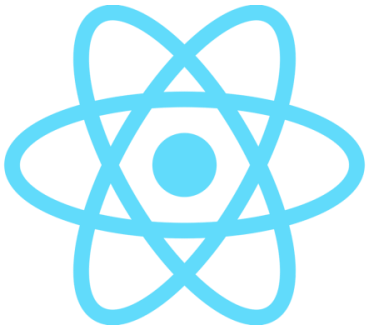
# React Базовий

## Тема уроку

### Знайомство з React

# React Базовий

## React



**React** – це бібліотека з відкритим кодом, яка розширює можливості мови програмування JavaScript. Фреймворк суттєво полегшує розробку користувацької частини вебдодатків та дозволяє використовувати ресурси браузера більш ефективно.

Вебфреймворк React був створений програмістом з фейсбуку Джорданом Валке. Перше його застосування в комерційній компанії припадає на 2011 рік. Стрічка новин у Facebook та Instagram розроблена за допомогою реакту.



*Джордан Валке*

# React Базовий

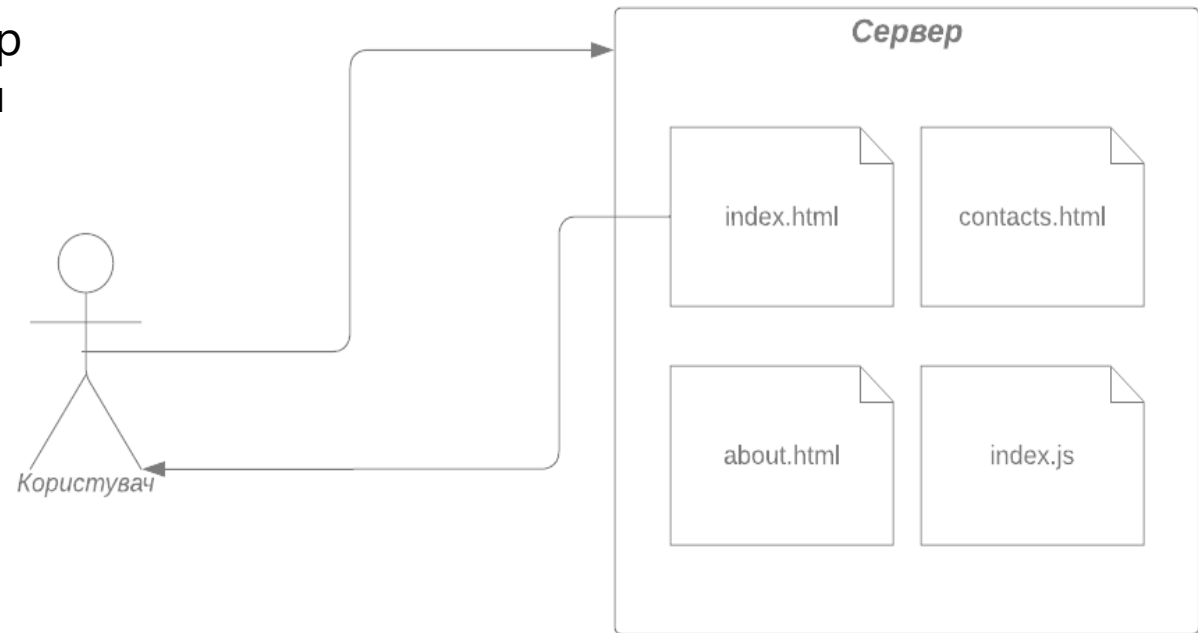
## Які задачі покликаний вирішити React

1. Front-end.
2. Розробка односторінкових вебсайтів.
3. Ефективна та швидка робота з різноманітними API.
4. Масштабованість за рахунок контейнеризації компонентів.
5. Розробка кросплатформених інтерфейсів.

# React Базовий

## Механізм роботи інтерфейсів вебдодатків

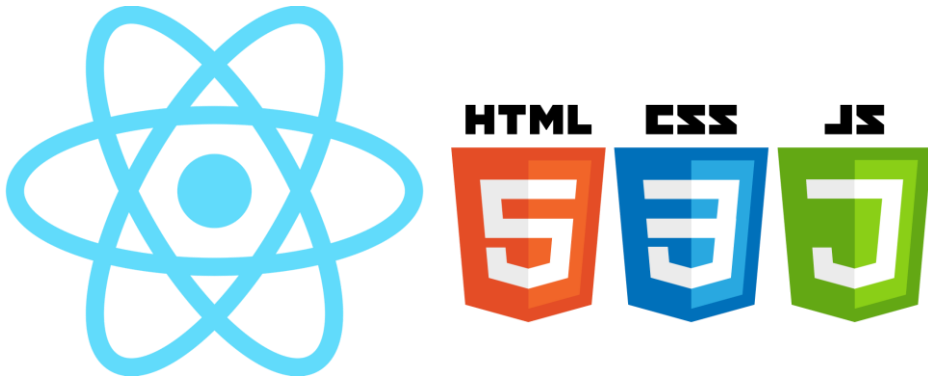
Перші вебресурси розроблялись, як набір окремих сторінок, які відображались при запитах користувачів до серверу по його адресі.



*Механізм роботи інтернет-ресурсів до появи PHP та React*

# React Базовий

## Сучасний підхід до розробки вебзастосунків



Frontend - відповідає за інтерфейсну частину додатку.



Backend - відповідає за логістику, взаємодію з базою даних

# React Базовий

## Середовище розробки

Даний курс розроблений в середовищі Visual Studio Code. Для роботи з пакетами використовується пакет-менеджер node.



Visual Studio Code

<https://code.visualstudio.com/>



Node

<https://nodejs.org/uk>

# React Базовий

## Огляд документації

Вся документація по бібліотеці React знаходиться на офіційному інтернет-ресурсі проєкту. Знайти її можна за наступними посиланнями:

<https://uk.reactjs.org/docs/getting-started.html>

<https://reactjs.org/docs/hello-world.html>

<https://uk.reactjs.org/tutorial/tutorial.html>

```
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(<h1>Hello, world!</h1>);
```

*Приклад побудови реакт-елементу, заголовку Hello world!*



# React Базовий

## Створення додатку

1. React - бібліотека з відкритим кодом, яка розширює можливості мови програмування JavaScript. Даний фреймворк суттєво полегшує розробку користувацької частини вебдодатків та дозволяє використовувати ресурси браузера більш ефективно.

2. Щоб створити реакт-застосунок, потрібно ввести наступну команду в терміналі:

```
npx create-react-app <назва_додатку>
```

3. Щоб запустити реакт-застосунок, потрібно ввести наступну команду в терміналі:

```
npm start
```

# React Базовий

## Структурування React-додатків



Одним з найбільш популярних способів структуризації файлової системи React-застосунків є розбиття проєкту на логічно відокремлені директорії.

В даному прикладі каталог **components** зберігає компонент багаторазового використання `Goods.js`, а каталог **containers** містить контейнери, з яких будується основна інтерфейсна частина.

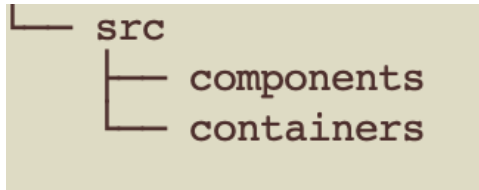
# React Базовий

## Правила ефективного React-додатку

1. Розподіляйте компоненти проєкту
2. Описуйте лише один компонент всередині одного файлу
3. Не перезавантажуйте застосунок
4. Дотримуйтесь однієї логіки розташування файлів
5. Не витрачайте більше 5 хвилин на вибір файлової структури проєкту

# React Базовий

## Як обрати структуру майбутнього проєкту



*Розповсюджена структура реакт-додатків*

1. Наскільки великим буде проєкт?
2. Які технології, окрім реакту, він буде використовувати?
3. Наскільки масштабований застосунок ви хочете отримати?

# React Базовий

## Що таке JSX

```
return (  
  <>  
    <h1>Hello World!</h1>  
  </>  
);
```

*Приклад використання JSX*

**JSX** – це розширення мови програмування JavaScript, яке дозволяє використовувати синтаксис HTML всередині JavaScript коду.

React також дозволяє розробляти застосунки без синтаксису JSX, але його використання суттєво спрощує розробку структури користувацького інтерфейсу.

# React Базовий

## Відмінності JSX від HTML

**CamelCase** для атрибутів.

Назви властивостей стилів, які містять дефіс в JSX позначається виключно в кемелкейсі: font-family стає fontFamily, font-size стає fontSize і так далі.

Атрибут for перетворюється на htmlFor.

Екранування дозволяє убезпечити місця інтерфейсу де повинен бути тільки текст, а не елемент розмітки.

Атрибути за замовчуванням, якщо атрибут передається в компонент без значення, то воно становиться **true** за замовчуванням.

```
<p style={{fontSize:10,fontStyle: 'italic'}}>  
  My name is Denys!  
</p>
```

*Inline-стили в React*

```
<input type="text" readOnly={true} />  
<input type="text" readOnly />
```

*Еквівалентні приклади*

# React Базовий

## Як JSX працює "під капотом"

Все, що ми пишемо в react-компонентах, переписується в JavaScript код за допомогою транспайлеру Babel.



Даний код в синтаксисі JSX

```
<>  
  <h1>Hello World!</h1>  
</>
```

Переписується в наступний JavaScript код

```
React.createElement(React.Fragment, null, React.createElement("h1", null, "Hello World!"))
```

# React Базовий

## Елементи в React

Елементи в React можна створювати без використання JSX синтаксису. Для цього використовується метод `React.createElement()`.

```
const element = React.createElement('h1',{
  color:'green',
  'style': {
    'fontSize':45,
    'color':'green',
    'background':'brown',
    'fontFamily':'italic'
  }
},'some text');
```

*Елемент h1 без використання JSX*

Аргументи методу `createElement` наступні :

`React.createElement(<-назва_тегу->, <-аргументи->, <-вкладений_текст_або_елемент->)`



# React Базовий

## Підсумки уроку

1. Документація React дозволяє використовувати індивідуальну файлову структуру для кожного проєкту. Файлова структура корпоративних застосунків передбачає окремі директорії для стилізації, компонентів, сторінок, даних, контексту та хуків.
2. JSX є розширенням мови програмування JavaScript, яке дозволяє створювати React-елементи за допомогою тегів, наближених до HTML.
3. Документація React рекомендує використання синтаксису JSX при побудові компонентів, але також надає можливість створювати react-застосунки без використання JSX.

# 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.