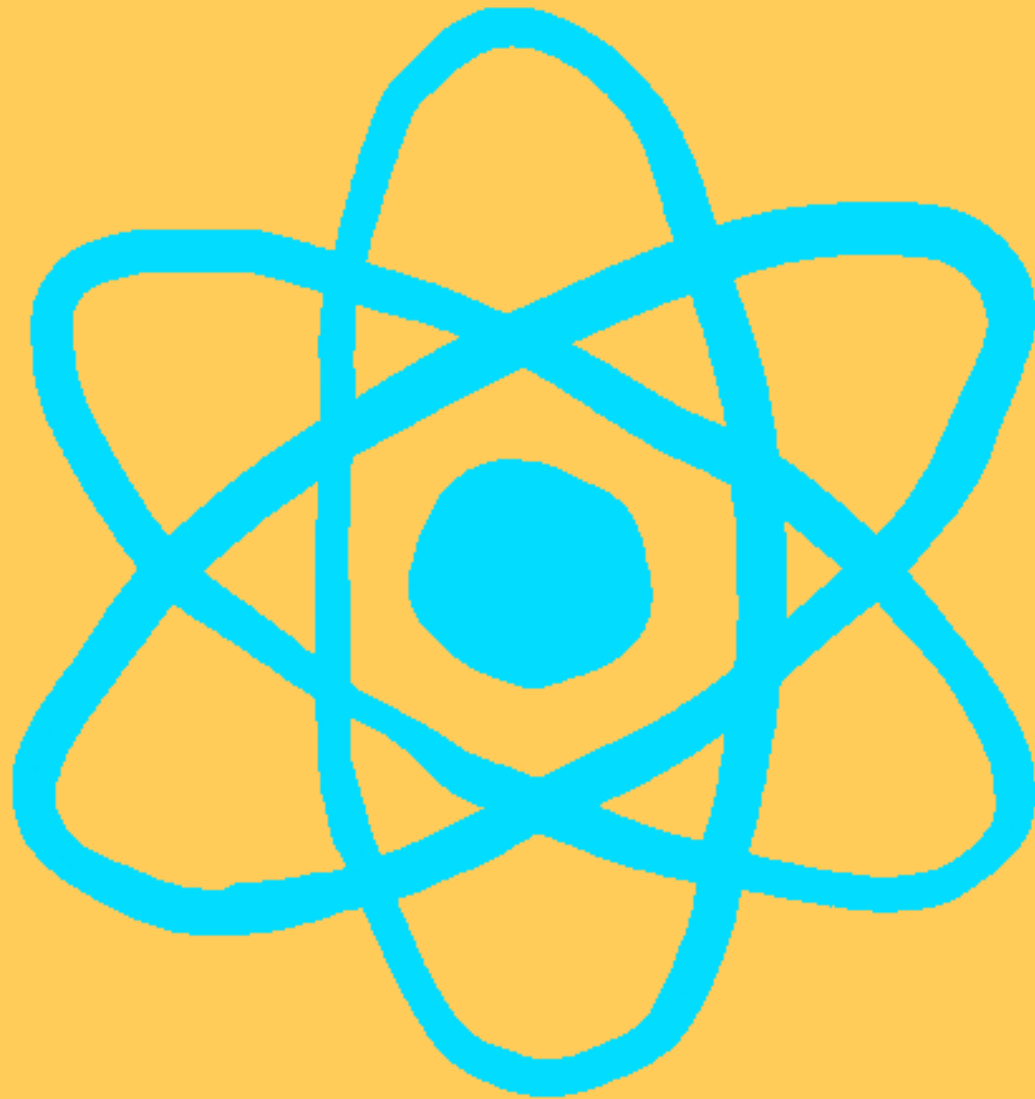


*FRONTEND. REACT. ЛЕКЦІЯ 7*

# CSS, CSS IN JS





CSS

# CSS

- CSS



# CSS

- CSS
- CSS modules

# CSS

- CSS
- CSS modules
- Preprocessors

# CSS

- CSS
- CSS modules
- Preprocessors
- CSS in JS

# CSS

## Глобальні стилі

```
import './index.css';

export function Root() {
  return (
    <form>
      <label>Ім'я:</label>
    </form>
  );
}
```

# CSS

## Локальні стилі

```
1 import './index.css';
2
3 export function Count(props) {
4     const { count } = props;
5     return (<p classname="count-item">Наявна кількість: {count}</p>);
6 }
```

```
1 .count-item {
2     border: 1px solid yellowgreen;
3     padding: 5px 0;
4 }
```

# CSS

## Локальні стилі

```
1 import './index.css';
2
3 export function Count(props) {
4     const { count } = props;
5     return (<p classname="count-item">Наявна кількість: {count}</p>);
6 }
```

```
1 .count-item {
2     border: 1px solid yellowgreen;
3     padding: 5px 0;
4 }
```

# CSS. НАЙКРАЩІ ПРАКТИКИ

# CSS. НАЙКРАЩІ ПРАКТИКИ

- Тека з назвою компоненту



# CSS. НАЙКРАЩІ ПРАКТИКИ

- Тека з назвою компоненту
- index.js

# CSS. НАЙКРАЩІ ПРАКТИКИ

- Тека з назвою компоненту
- index.js
- index.css

# CSS. ПЕРЕВАГИ

# *CSS. ПЕРЕВАГИ*

- Простий в розробці

# CSS. ПЕРЕВАГИ

- Простий в розробці
- Непотрібна додаткова конфігурація

# CSS. НЕДОЛІКИ

# CSS. НЕДОЛІКИ

- Класи глобальні, а не локальні

# CSS. НЕДОЛІКИ

- Класи глобальні, а не локальні
- Необхідні методології типу БЕМ



# *CSS. НЕДОЛІКИ*

- Класи глобальні, а не локальні
- Необхідні методології типу БЕМ
- Перевикористання (перебивання) стилів

# *CSS. НЕДОЛІКИ*

- Класи глобальні, а не локальні
- Необхідні методології типу БЕМ
- Перевикористання (перебивання) стилів
- Зайвий CSS

# INLINE CSS

```
1 import { useState } from 'react';
2
3 export function Price(props) {
4   const [stateSmth, setStateSmth] = useState(randomNumber());
5
6   return (<div style={{backgroundColor:" color"}}=>
7     Кількість кліків: {stateSmth}<br>
8   </div>);
9 }
```

# INLINE CSS. ПЕРЕВАГИ

# *INLINE CSS. ПЕРЕВАГИ*

- Швидко

# *INLINE CSS. ПЕРЕВАГИ*

- Швидко
- Можливо динамічно конфігурувати

# INLINE CSS. НЕДОЛІКИ

# *INLINE CSS. НЕДОЛІКИ*

- Неможливо перебити стилі



# *INLINE CSS. НЕДОЛІКИ*

- Неможливо перебити стилі
- Неможливе перевикористання стилів

# INLINE CSS. НЕДОЛІКИ

- Неможливо перебити стилі
- Неможливе перевикористання стилів
- Запис CamelCase, немає MediaQueries

# CSS МОДУЛІ

- Вирішення проблеми перевикористання стилів

```
1 import styles from './index.module.css';
2
3 export function Count(props) {
4     const { count } = props;
5     return (<p classname="{styles.count}">Наявна кількість: {count}</p>);
6 }
```

```
1 .count {
2     border: 1px solid yellowgreen;
3     padding: 5px 0;
4 }
```

# CSS МОДУЛІ

- Вирішення проблеми перевикористання стилів

```
1 import styles from './index.module.css';
2
3 export function Count(props) {
4     const { count } = props;
5     return (<p classname="{styles.count}">Наявна кількість: {count}</p>);
6 }
```

```
1 .count {
2     border: 1px solid yellowgreen;
3     padding: 5px 0;
4 }
```

# CSS МОДУЛІ

# CSS МОДУЛІ

- Унікальні імена

# CSS МОДУЛІ

- Унікальні імена
- Непотрібна система типу БЕМ

# CSS МОДУЛІ

- Унікальні імена
- Непотрібна система типу БЕМ
- Непотрібна специфічність селекторів



# CSS МОДУЛІ

- Унікальні імена
- Непотрібна система типу БЕМ
- Непотрібна специфічність селекторів
- Легко знаходити CSS, що не використовується

# ПРЕПРОЦЕСОРИ

CSS препроцесор – це надбудова над CSS, яка додає додаткові можливості за допомогою синтаксичних конструкцій

# ПРЕПРОЦЕСОРИ

CSS препроцесор – це надбудова над CSS, яка додає додаткові можливості за допомогою синтаксичних конструкцій

- SCSS

# ПРЕПРОЦЕСОРИ

CSS препроцесор – це надбудова над CSS, яка додає додаткові можливості за допомогою синтаксичних конструкцій

- SCSS
- SASS

# ПРЕПРОЦЕСОРИ

CSS препроцесор – це надбудова над CSS, яка додає додаткові можливості за допомогою синтаксичних конструкцій

- SCSS
- SASS
- LESS

# ПРЕПРОЦЕСОРИ

CSS препроцесор – це надбудова над CSS, яка додає додаткові можливості за допомогою синтаксичних конструкцій

- SCSS
- SASS
- LESS
- Stylus

# ПРЕПРОЦЕСОРИ. МОЖЛИВОСТІ

# ПРЕПРОЦЕСОРИ. МОЖЛИВОСТІ

- Міксини



# ПРЕПРОЦЕСОРИ. МОЖЛИВОСТІ

- Міксини
- Змінні

# ПРЕПРОЦЕСОРИ. МОЖЛИВОСТІ

- Міксини
- Змінні
- Вкладеність

# CSS IN JS

```
npm i styled-components
```

# CSS IN JS

```
1 import { createPortal } from "react-dom";
2 import styled from "styled-components";
3
4 const modalRootEl = document.querySelector('#modal');
5 export function Portal(props) {
6     const { open, onClose } = props;
7
8     const Title = styled.h1`
9         font-size: 1.5em;
10        text-align: center;
11        color: palevioletred;
12    `;
13
14    return open ? createPortal(
15        <div classname="modal-background" onclick="{onClose}">
16        <div classname="modal-content">
17            <title>Test</title>
18            {props.children}
```

# CSS IN JS

```
1 import { createPortal } from 'react-dom',
2 import styled from "styled-components";
3
4 const modalRootEl = document.querySelector('#modal');
5 export function Portal(props) {
6   const { open, onClose } = props;
7
8   const Title = styled.h1`
9     font-size: 1.5em;
10    text-align: center;
11    color: palevioletred;
12  `;
13
14  return open ? createPortal(
15    <div classname="modal-background" onclick="{onClose}">
16      <div classname="modal-content">
17        <title>Test</title>
18        {props.children}
19      </div>
20    </div>
```

# CSS IN JS

```
6     const { open, onClose } = props;
7
8     const Title = styled.h1`
9         font-size: 1.5em;
10        text-align: center;
11        color: palevioletred;
12    `;
13
14    return open ? createPortal(
15        <div classname="modal-background" onclick="{onClose}">
16            <div classname="modal-content">
17                <title>Test</title>
18                {props.children}
19            </div>
20        </div>,
21        modalRootEl
22    ) : null;
23 }
```

# CSS IN JS. ПЕРЕВАГИ

# *CSS IN JS. ПЕРЕВАГИ*

- Контроль CSS з JS



# CSS IN JS. ПЕРЕВАГИ

- Контроль CSS з JS
- Відсутність конфлікту імен

# CSS IN JS. НЕДОЛІКИ

# CSS IN JS. НЕДОЛІКИ

- Синтаксис

# CSS IN JS. НЕДОЛІКИ

- Синтаксис
- Швидкодія. Збільшення кількості JS

# CSS IN JS. БІБЛІОТЕКИ

# *CSS IN JS. БИБЛИОТЕКИ*

- Styled Components

# CSS IN JS. БИБЛИОТЕКИ

- Styled Components
- emotion/react

# CSS IN JS. БИБЛИОТЕКИ

- Styled Components
- emotion/react
- JSS



# CSS IN JS. ЗАЛЕЖНІСТЬ ВІД ДАНИХ

```
1 import styled from "styled-components";
2
3 const TitleProps = styled.h1`
4     font-size: 1.5em;
5     text-align: center;
6     color: ${props => props.type === "regular" ? 'red' : 'green'};
7 `;
8
9 return open ? createPortal(
10     <div classname="modal-background" onclick="{onClose}">
11         <div classname="modal-content">
12             <title>Test</title>
13             <titleprops type="regular">Test</titleprops>
14             {props.children}
15         </div>
16     </div>,
17     modalRootEl
18 ) : null;
```

# CSS IN JS. ЗАЛЕЖНІСТЬ ВІД ДАНИХ

```
1 import styled from "styled-components";
2
3 const TitleProps = styled.h1`
4     font-size: 1.5em;
5     text-align: center;
6     color: ${props => props.type === "regular" ? 'red' : 'green'};
7 `;
8
9 return open ? createPortal(
10     <div classname="modal-background" onclick="{onClose}">
11         <div classname="modal-content">
12             <title>Test</title>
13             <titleprops type="regular">Test</titleprops>
14             {props.children}
15         </div>
16     </div>,
17     modalRootEl
18 ) : null;
```

# CSS IN JS. ЗАЛЕЖНІСТЬ ВІД ДАНИХ

```
2
3 const TitleProps = styled.h1`
4     font-size: 1.5em;
5     text-align: center;
6     color: ${props => props.type === "regular" ? 'red' : 'green'};
7 `;
8
9 return open ? createPortal(
10     <div classname="modal-background" onclick="{onClose}">
11         <div classname="modal-content">
12             <title>Test</title>
13             <titleprops type="regular">Test</titleprops>
14             {props.children}
15         </div>
16     </div>,
17     modalRootEl
18 ) : null;
19 }
```

# CSS IN JS. РОЗШИРЕННЯ СТИЛІВ

```
1 import styled from "styled-components";
2
3 const Title = styled.h1`
4     font-size: 1.5em;
5     text-align: center;
6     color: palevioletred;
7 `;
8
9 const ExtendStyles = styled(Title)`
10     background: darkblue
11 `;
12
13 return open ? createPortal(
14     <div classname="modal-background" onclick="{onClose}">
15         <div classname="modal-content">
16             <title>Test</title>
17             <titleprops type="regular">Test</titleprops>
18             <extendstyles type="regular">Test</extendstyles>
```

# CSS IN JS. РОЗШИРЕННЯ СТИЛІВ

```
1 import styled from 'styled-components',
2
3 const Title = styled.h1`
4     font-size: 1.5em;
5     text-align: center;
6     color: palevioletred;
7 `;
8
9 const ExtendStyles = styled(Title)`
10     background: darkblue
11 `;
12
13 return open ? createPortal(
14     <div classname="modal-background" onclick="{onClose}">
15         <div classname="modal-content">
16             <title>Test</title>
17             <titleprops type="regular">Test</titleprops>
18             <extendstyles type="regular">Test</extendstyles>
19         </div>
20     </div>
```

# CSS IN JS. РОЗШИРЕННЯ СТИЛІВ

```
7 ;  
8  
9 const ExtendStyles = styled(Title)`  
10     background: darkblue  
11 `;  
12  
13 return open ? createPortal(  
14     <div classname="modal-background" onclick="{onClose}">  
15         <div classname="modal-content">  
16             <title>Test</title>  
17             <titleprops type="regular">Test</titleprops>  
18             <extendstyles type="regular">Test</extendstyles>  
19             {props.children}  
20         </div>  
21     </div>,  
22     modalRootEl  
23 ) : null;  
24 }
```

*ЩО ВИБРАТИ?*

*ЩО ВИБРАТИ?*

CSS modules



*ЩО ВИБРАТИ?*

CSS modules

CSS modules + SCSS

## ЩО ВИБРАТИ?

CSS modules

CSS modules + SCSS

CSS in JS

## ЩО ВИБРАТИ?

CSS modules

CSS modules + SCSS

CSS in JS

Pure CSS

