Основы JSX

Цель работы

Познакомить со способом описания визуального кода посредством комбинации кода на JavaScript и разметки XML.

Задания для выполнения

1. Подключите библиотеки для работы с React.
2. В фигурных скобках JSX поместить любое корректное выражение JavaScript. Например, 5 + 2, user.firstName, или formatName(user) — всё это допустимые выражения в JavaScript.
3. Выведите свои ФИО, возраст и добавьте стилей с использованием формат JSX.
4. Загрузить созданную страницу на GitHub в репозиторий Student, используя формат в названии Фамилия (латинскими буквами)_3.

Методические указания

1. Чтобы подключить вспомогательные библиотеки скопируйте ссылки:

```
<!DOCTYPE html>

<html>

<head>

   <meta charset="utf-8" />

   <title>Hello React</title>

<script crossorigin src="https://unpkg.com/react@16/umd/react.production.min.js"></script>

<script crossorigin src="https://unpkg.com/react-dom@16/umd/react-dom.production.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/6.25.0/babel.min.js"></script>
```

```
</head>

<body>

…

</body>
</html>
```

2. Чтобы создать и поместить объект можно воспользоваться синтаксисом:

```
<!--опишем объект-->

<script type="text/babel">

 ReactDOM.render(

    <h2>5 + 2 = {5 + 2}</h2>,

    document.getElementById("app")

 )

</script>

<!--поместим описанный объект-->
<div id="app"> </div>
```

3. Стили можно задать:

```
const userClassName = "user-info";

const styleObj = {

    color:'green',

    fontFamily:'Verdana'
};
```

Или inline: `<div className={userClassName} style={{color:'red', fontFamily:'Verdana'}}>`

Для создания ФИО и возраст можно воспользоваться:

```
const user = {
    id : 5,
    age: 33,
    firstName: 'Alevtina',
    lastName: 'Shatalova',
    getFullName: function(){
        return `${this.firstName} ${this.lastName}`;
    }
};
```

Для сборки:

```
ReactDOM.render(
    <div className={userClassName}  style={styleObj}>
        <p>Полное имя: {user.getFullName()}</p>
        <p>Возраст: {user.age}</p>
    </div>,
    document.getElementById("app")
)
```

4. При работе с JSX следует учитывать ряд моментов, в частности, в JSX для установки класса применяется атрибут `className`*, а не* `class`*. Второй момент: атрибут* `style` *в качестве значения принимает объект javascript. И третий момент: в JSX используется camel-case, то есть если мы хотим определить стилевое свойство для шрифта, например, свойство* `font-family`*, то соответствующее свойство в объекте стиля будет называться* `fontFamily`*, то есть дефис отбрасывается, а следующая часть слова начинается с заглавной буквы.*

Контрольные вопросы

1. Чем JSX отличается от JS?

Дополнительные задания

1. Дополнительно: изменить стиль реакт-компонентов и добавьте новые на html-страницу.