# Работа с Node.js. Условный рендеринг

### Цель работы

Познакомить со способом подключения внешнего файла с React-компонентами (создание своего App), а также со способами условного включения JSX.

### Задания для выполнения

1. Для работы необходимо установить [Node.js](https://nodejs.org/en/) (<https://nodejs.org/en/>). Пакет npm поставляется вместе с Node.js, поэтому его устанавливать отдельно не нужно.
2. Откройте командную строку/терминал и перейдите с помощью команды **cd** к папке, в которой будет создании ваш App. Затем выполните команды в терминале поочередно:

### npx create-react-app my-appcd my-appnpm start

1. Используя предыдущие работы и понятия «Условный рендеринг», «Изменяемое состояние» – создайте меню для ресторана по примеру согласно тех. заданию:



Тех. Задание:

- меню на русском;

- верстка по примеру скрина (допускается без бокового меню);

- для каждого блюда возможность отобразить дополнительно цену, ингредиенты и.т.д.;

- блюда берутся из массива объектов.

1. Загрузить созданную страницу на GitHub в репозиторий https://github.com/al-shatalova/Lab-5.git, используя формат в названии Фамилия (латинскими буквами).

### Методические указания

Создание своего App

После установки Node.js в командной строке программы выполните команды:

### npx create-react-app my-app – создание приложенияcd my-app – переход в папку приложенияnpm start – запуск приложения

### В терминале появится сообщение:



### Результатом работы программы будет запущенное в браузере React-приложение с адресом: <http://localhost:3000/>



В папке вашего приложения автоматически появятся следующие файлы:

 

Теперь можно открывать в любой IDE файл App.js из папки src, редактировать его, обновляя App в браузере.

## Условный рендеринг

В React нет специального синтаксиса для записи условий. Вместо этого вы будете использовать те же методы, что и при написании обычного кода JavaScript. Например, вы можете использовать оператор [if](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/if...else) для условного включения JSX:

let content;

if (isLoggedIn) {

 content = <AdminPanel />;

} else {

 content = <LoginForm />;

}

return (

 <div>

 {content}

 </div>

);

Если вы предпочитаете более компактный код, вы можете использовать [условный ? оператор.](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Conditional_Operator) В отличие if от этого, он работает внутри JSX:

<div>

 {isLoggedIn ? (

 <AdminPanel />

 ) : (

 <LoginForm />

 )}

</div>

Когда вам не нужна else ветвь, вы также можете использовать более короткий [логический && синтаксис](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Logical_AND#short-circuit_evaluation):

<div>

 {isLoggedIn && <AdminPanel />}

</div>

Все эти подходы также работают для условного указания атрибутов. Если вы не знакомы с некоторыми элементами этого синтаксиса JavaScript, вы можете начать с использования всегда if...else.

## Краткое описание

* В React вы управляете логикой ветвления с помощью JavaScript.
* Вы можете возвращать выражение JSX условно с помощью if инструкции.
* Вы можете условно сохранить некоторый JSX в переменную, а затем включить его в другой JSX, используя фигурные скобки.
* В JSX это {cond ? <A /> : <B />} означает *“если*cond*, рендерить*<A />*, в противном случае*<B />*”*.
* В JSX это {cond && <A />} означает *“если*cond*, рендерить*<A />*, в противном случае ничего”*.
* Ярлыки являются обычными, но вам не обязательно использовать их, если вы предпочитаете простой рендеринг if.

### Показывать или скрывать другой компонент при нажатии в React

import {useState} from 'react';

let App = function Appp() {

 const [isShown, setIsShown] = useState(false);

 const handleClick = event => {

 setIsShown(current => current=true);

 };

 return (

 <div>

 <button onClick={handleClick}>Click</button>

 {isShown && (

 <div>

 <h2>Some content here</h2>

 </div>

 )}

 </div>

 );

}

const root = createRoot(document.getElementById('root'));

root.render(

 <StrictMode>

 <App />

 </StrictMode>

);

### Контрольные вопросы

1. Можно ли использовать другой веб-сервер вместо Node.js?
2. Что компилятор предпочитает – ярлыки или конструкцию if?

**Полезные ссылки:**

1. https://react.dev/learn/conditional-rendering

2. https://bobbyhadz.com/blog/react-onclick-show-component