



Лекция 5

Библиотеки.

Фреймворки. React

Мартин Комитски



ЦРИТО
Центр Развития
ИТ-Образования



План на сегодня

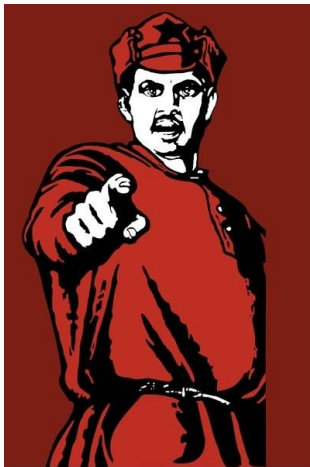


- Введение
- Теория
 - Библиотека
 - Фреймворк
 - Инструменты
 - Введение в React
 - React Компоненты
 - JSX
- Практика
 - Используемые инструменты
 - Разворачиваем свое окружение для разработки React приложений
 - Разбор примеров по React (Компоненты и все остальное)

Минутка бюрократии



- Внимание
- Отметки о посещении занятий
- Обратная связь о лекциях





Библиотека — это структурированный набор полезного функционала.

Содержит функции для работы со следующими вещами:

- Строки
- Даты
- DOM-элементы
- События
- Cookie
- Анимации
- Запросы
- Многое другое

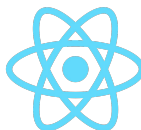


Плюсы:

- Увеличивает скорость разработки
- Снижает порог входа в проект (и сами технологии)
- Кроссбраузерность

Минусы:

- Могут быть ошибки
- Быстрое устаревание/обновление



zepto.js



Фреймворки



Фреймворк — это *каркас приложения*. Он обязывает разработчика выстраивать архитектуру приложения в соответствии с некоторой логикой.

Предоставляет функционал для:

- Событий
- Хранилищ
- Связывания данных

Фреймворки



Плюсы:

- Более высокий уровень абстракции
- Можно построить около 80% вашего приложения

Его минусы:

- Немалые трудности из-за ограничений
- Быстрое устаревание/обновление





Инструмент — это вспомогательное средство разработки, но он не является неотъемлемой частью проекта.

Виды:

- системы сборки
- компиляторы
- транспайлеры
- механизмы развертывания
- препроцессоры
- линтеры
- тесты
- многое другое

Инструменты



Плюсы:

- Невероятное упрощение работы
- Контроль разных этапов разработки
- Автоматизация

Минусы

- Их много
- Надо знать
- Надо настраивать



Введение в React



React – это декларативная, эффективная и гибкая JavaScript библиотека для разработки интерфейсов. Она позволяет составлять сложные визуальные интерфейсы из атомарных кусочков, называемых “компонентами”.

Императивный – как сделать.

Декларативный – что сделать.

Появился в 2011 в Facebook, в ленте Instagram в 2012.

Virtual DOM
JSX

Введение в React. Компоненты. JSX



```
class ShoppingList extends React.Component {  
  render() {  
    return (  
      <div className="shopping-list">  
        <h1>Shopping List for {this.props.name}</h1>  
        <ul>  
          <li>Instagram</li>  
          <li>WhatsApp</li>  
          <li>Oculus</li>  
        </ul>  
      </div>  
    );  
  }  
}  
  
// Example usage: <ShoppingList name="Mark" />
```

Введение в React. Понятие JSX



```
const element = <h1>Hello, world!</h1>;
```

JSX - ни строка, ни HTML.

JSX - представляет собой расширение JavaScript при помощи XML синтаксиса.

JSX - нормальное, полноценное JS выражение.

```
const name = 'Martin Komitsky';  
const element = <h1>Hello, { name }</h1>;
```

```
ReactDOM.render(  
  element,  
  document.getElementById('root')  
)
```



Hello, Martin Komitsky

Введение в React. Понятие JSX. Выражения



JSX может содержать JS выражение внутри { }.

```
function formatName(user) {  
  return user.firstName + ' ' + user.lastName;  
}  
  
const user = {  
  firstName: 'Martin',  
  lastName: 'Komitsky'  
};  
  
const element = (  
  <h1>  
    Hello, Sir { formatName(user) }!  
  </h1>  
)  
);  
  
ReactDOM.render(  
  element,  
  document.getElementById('root')  
);
```



Hello, Sir Martin Komitsky!

Введение в React. Понятие JSX. Дочерние элементы, атрибуты



JSX может содержать многострочные конструкции, тогда оборачивается в ().

```
const element = (  
  <div>  
    <h1>Hello!</h1>  
    <h2>Good to see you here.</h2>  
  </div>  
);
```

В **JSX** можно в выражениях выставлять любые **HTML** атрибуты, но в *camelCase* (кроме `class` – там будет `className`).

```
const element = <img src={ user.avatarUrl }></img>;
```

JSX защищает от внедрения опасного кода.

```
const title = response.xss;  
// This is safe:  
const element = <h1>{ title }</h1>;
```

Введение в React. Компоненты. JSX



Вопросы?



Введение в React. Отрисовка элементов



Довольно! Перемещаемся в текстовый редактор!



Все примеры будут позже в репозитории.

Введение в React. Используемые инструменты



- [React DevTools](#)
- [Webpack](#)
- Babel
- [Create React App](#)



“
Перерыв!

Преподаватель (с)

Домашнее задание № 5



1. Закрепить знания React.js
2. Изучить Create React App
3. Подключить к проекту Create React App
4. Переписать компоненты на React
5. Восстановить стили для всех компонентов

Срок сдачи

FS-11: 8 ноября

FS-21: 5 ноября



Спасибо за внимание!



**Организационные
вопросы, работа с git, js?**