

# 6-oy 2-dars

React.js



- DAVOMAT
- UY VAZIFASI
- VIDEO YOZISH

1. JavaScriptda event handler nima ?
2. onclick event handler nima vazifani bajaradi va uni qanday qo'llash usullari bor ?
3. unload va onunload event handlerlari nima uchun kerak ?
4. oninput va onchange event handlerlari nima uchun kerak ?
5. onmouseover va onmouseout event handlerlar nima uchun kerak ?
6. onmouseup va onmousedown event handlerlar nima uchun kerak ?
7. onkeydown, onkeypress, onkeyup eventlari nima uchun kerak ?
8. `<button onclick="clickFunc(this)" >Click</button>` - shu yerdagi this kalit so'zi nima vazifani bajaradi ?

# Node.js

Avvallari javaScript kodlar faqat browserlarda ishlagan browserlardan boshqa hech qayerda ularni ishlatib bo'lmagan.

Js kodini qandaydir mashinada, operatsion tizimda, mobil qurulmalarda ishlatishning imkoni bo'lmagan. Ushbu muammoni hal qilish uchun Node.js dan foydalaniladi.

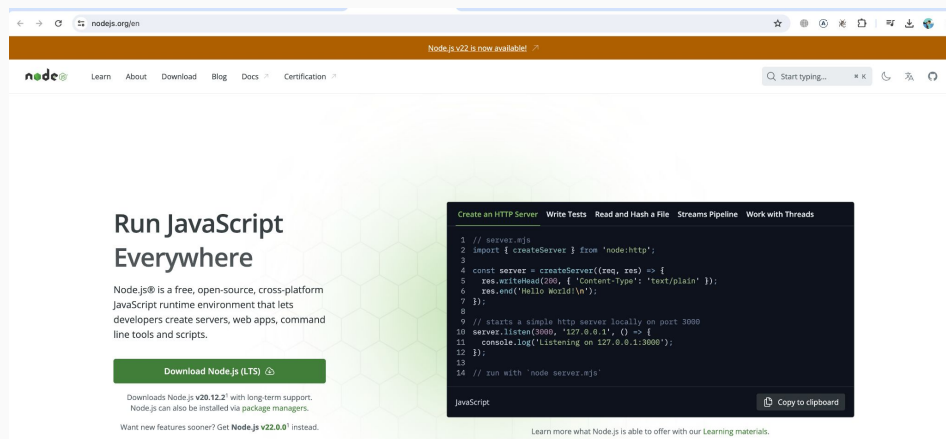
Node.js bu ochiq kodli, javascriptni kodni ko'plab qurulmalar yoki platformalarda ishlatib beruvchi runtime muhiti xisoblanadi.

Biz odatda Node.js ni server tomonida backend serverlarni amalga oshirishda ishlatamiz.



# Node.js ni o'rnatish

nodejs.org web sahidasiga kiri node.js o'rnatish dasturini yuklab oling. Va yuklangan dasturni kompyuteringizga o'rnating.



Dasturni o'rnatkach kompyuter terminalini ochib **node --version** buyrug'ini bering. Agar terminalda node.js ning siz o'rnatkan bersiyasi chiqsa dema u tog'ri ishlamoqda

# package manager npm and yarn

qachonki siz node.js ni o'rnatkaningizda avtomatik ravishda sizda npm ham o'rnatilgan bo'ladi. Uni tekshirish uchun **npm --v** buyrug'i ishlatiladi.

NPM - bu Node Package Manager . JavaScript dasturchilar biror bir muammoni hal qilishda yozgan kodlarini npm ga joylashtirishlari. Boshqa dasturchilar esa ana shu npmga joylashtirilgan kodlarni npm yordamida o'zlariga o'rnatib unda foydalanishlari mumkin.

Bu ko'rinishdagi npm ga joylashtirilgan maxsus bir vazifani bajaruvchi kodlar **package** yoki **module** deb ataladi.

Zamonaviy js dasturlari bir necha o'n/yuz/ming lab packagelarni ishlatkan bo'lishi mumkin.

O'zingiz uchun kerakli bo'lgan package larni siz **npmjs.com** saytida topa olasiz va o'zingiz yaratkan package ni unga joylashtira olasiz.

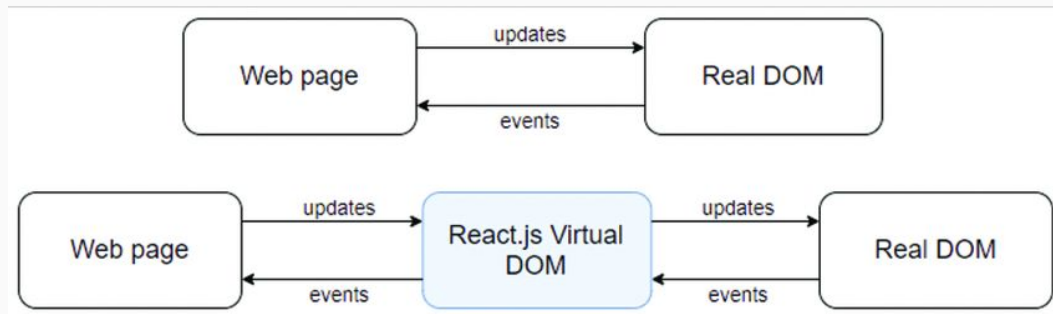
**yarn** bu npm kabi u xam package manager hisoblanadi. Yarn npm kabi node.js bilan avtomatik o'rnatilmaydi. Dasturchilar yoki npm ni yoki yarn ni ishlatishadi. Shu sabab siz ham ulardan istalgan birini tanlab ishlatishingiz mumkin. Umumiy fikrlarga ko'ra yarn tezroq va yangi chiqqan package manager xisoblanadi. Uni npm yordamida o'rnatish kerak.

# Real DOM va Virtual DOM

Biz shu paytkacha siz bilan JS yordamida DOM da element qo'shish, olib tashlash stil berish attributini o'zgartirish vazifalarini bajardik bu vazifalarning hammasi **Real DOM** da sodir bo'ladi. Real domda siz js yodamida qandaydir ishni qilasiz , oz'garish sabab butun DOM tree yangittan hosil bo'ladi va u tezda browserda ko'rinadi.

**Virtual DOM** ham aslida Real DOM bilan ishlaydi farqi Virtual DOMda DOM tree ni o'zgartirishdan avval optimal bo'lgan xisob kitoblar va Real DOM ni o'zgartirishning eng optimal bo'lgan yo'li xisoblanib keyin Real DOM o'zgartiriladi.

Real DOM da o'zgarish bo'lsa butun DOM tree o'zgaradi. Virtual DOM yordamida esa DOM treening o'zgarmagan qismi olib qolinib o'zgarishi kerak bo'lgan qismi o'zgartiriladi. bu esa ancha samarali va tezkor bo'ladi. Zamonaviy SPA dasturlar Virtual DOM bilan ishlaydi.

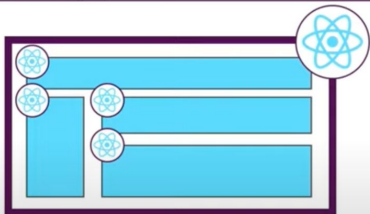


Web sahifalar asosan 2 turga bo'linadi. Single page applicaton SPA va Multi page application MPA.

## Two Kinds of Applications

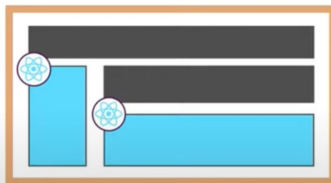
### Single Page Applications

Only ONE HTML Page, Content is (re)rendered on Client



### Multi Page Applications

Multiple HTML Pages, Content is rendered on Server



### SPA vs MPA

	SPA	MPA
SEO	-	+
Initial Load	-	+
Speed	+	-
Server	+	-



# Single page application Library va Frameworklar



Vue.js



## React haqida

- React.js bu Library (Kutubxona) lekin u Framework emas.
- React.js bu JavaScript Library
- React.js Single Page Application turida ishlaydi.
- React.js Virtual DOM usubini qo'llaydi.
- React.js da browserda ko'rinib turgan UI qismi, ko'rinish bir nechta qismlarga bo'linadi. Masalan: Navbar, sidebar, content, footer kabilar.
- Bu bo'lingan qismchalar Component deb ataladi.

## Yangi React loyihani yaratish

Yangi react loyihani yaratish uchun odatda terminalda

**npm install create-react-app**

**npx create-react-app <loyiha nomi>**

buyrug'i beriladi.

Bu buyruq siz terminalni ochka papka ichida <loyiha nomi> bilan yangi papka ochiadi va uning ichida react loyiha ning barcha kerakli fayllari saqlanadi.

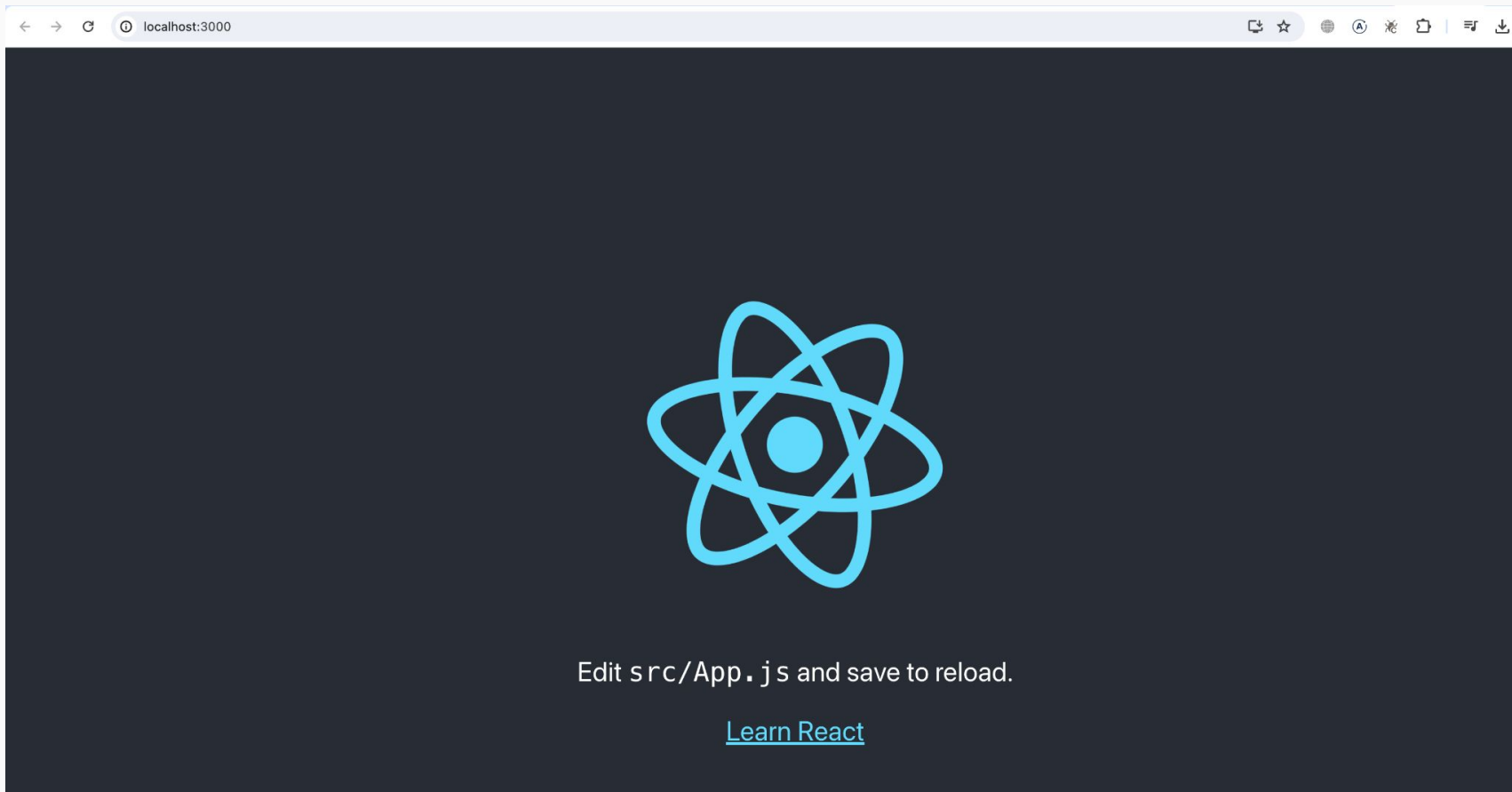
React loyiha yaratilgach ana shu loyiha papaksi ichida terminalni ochib

**npm start**

buyrug'i beriladi.

Buyruq browserning **localhost:3000** portida react loyihani ishlatib beradi.

# Ishka tushkan react loyihaning ilk ko'rinishi



- React da biz browserda hosil bo'ladigan text, rasm, forma kabi elementlarni hosil qilish uchun HTML dan foydalanmaymiz.
- Uning o'rniga HTML sintaksisiga juda o'xshash JSX dan foydalanamiz.
- JSX bu HTML bilan juda oxshash bo'lgan ammo HTML sintaksi asosida JavaScript kodni ishlatib beruvchi uskuna xisoblanadi.
- JSX yordamida siz HTML kodi bilan deyarli bir xil kodni yozasiz. Ammo u Reactga murojaat qilib uning **React.createElement(type, props, ...children);** metodiga murojaat qiladi.

## JSX qoidalari

- Doim React componentda ishlatilinishi kerak
- fayl nomi **.js** yoki **.jsx** bilan nomlanishi kerak
- jsx da yozilgan kod doim biror umumiy elementga o'ralgan bo'lishi kerak. Hayotiy misolda bu barcha elementlar umumiy bir idishka solingan bo'lishi

```
// It is JSX format

<div>
  <div>
    <h1>Mukhriddin</h1>
  </div>
  <div>
    <h1>@mukhriddinweb</h1>
  </div>
</div>
```

# Component

React loyihalar asosan Componentlardan tashkil topkan.

Componentlar bu js funksiyalariga o'xshash bir marotaba yozib ko'p marotaba ishlatishk mo'ljallangan.

Componentlar bu biror ui qismini hosil qlish uchun kerak bo'lgan kodlarni yig'indisidir

Qoidalar:

- Component nomlari doim Katta harflar bilan boshlanadi
- Componentlar doim html teglar kabi `<ComponentNomi></ComponentNomi>` ko'rinishida ishlatilindi.
- Componentlar doim JSX turidagi kodni return qiladi.

Turlari

- Reactda 2 xil turdagi componentlar qo'llaniladi
- Class va Function component
- Class component bu class sintaksisiga asoslagan component va u hozirda kam ishlatilindi
- Function component bu js funksiyalariga juda o'xshash va unda funksiya sintaksisi qo'llaniladi

## Class component (Stateful)

```
import React from 'react'

export class Hello extends React.Component {
  render() {
    return (
      <h1>Hello, {this.props.name}</h1>
    )
  }
}
```

## Functional Component (Stateless before hooks)

```
import React from 'react'

export const Hello = (props) => {
  return (
    <h1>Hello, {props.name}</h1>
  )
}
```

# Kerali VS Code extensionlar

- ES7+ React/Redux/React-Native snippets
- ESLint

Bu extensionlar sizga React loyihalarda componentlar yaratish va loyihada ishlashda yordami beradi.



# Rect component yaratish va ularga stillar berish

Amaliy vazifa:

- Navbar
- Sidebar
- Content
- Footer

componentlarni yaratish

- ularga stillar berish
- hosil bo'lgan componentlarni App.js da chaqirib bilashtirish