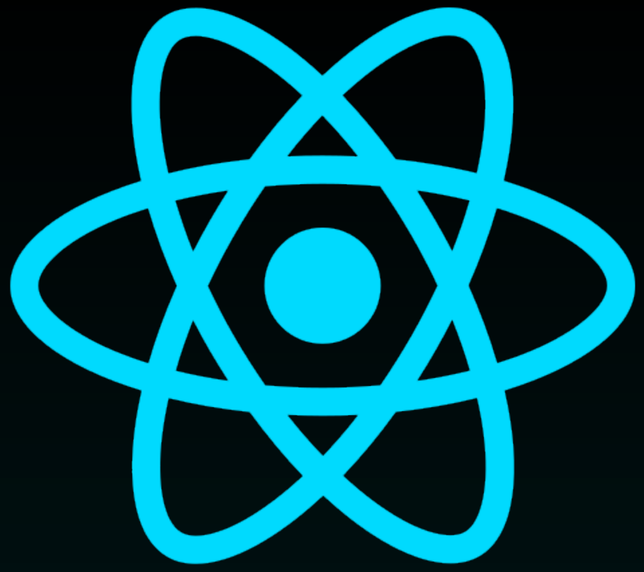


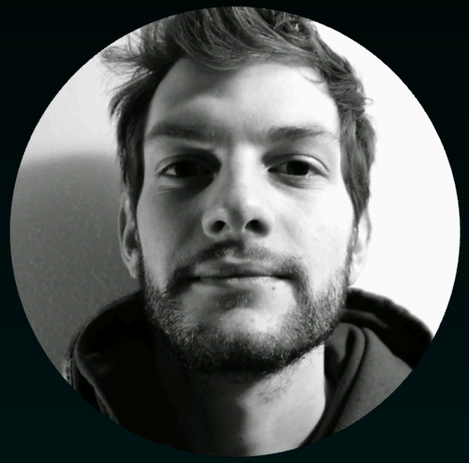
React

Una libreria JavaScript per creare interfacce utente

Febbraio - Marzo 2022
Marco Burrometo

Lezione 1 - 9 Feb 2022

Marco Burrometo



Frontend Developer @Levuro

https://marcoburrometo.cloud

React

- Libreria javascript per creare interfacce utente
  - ↘ Non è un Framework come Angular/Vue
- Mantenuta da Facebook ⟶ Solidità, sicurezza medio-lungo termine
- Community larga e in crescita ⟶ Supporto, documentazione, componenti aggiuntivi
- Non solo web (server side components / React Native)

https://it.reactjs.org/

React è *dichiarativo*

D*ichiariamo* quello che vogliamo fare, React si occuperà di farlo

“Progetta interfacce per ogni stato della tua applicazione.”

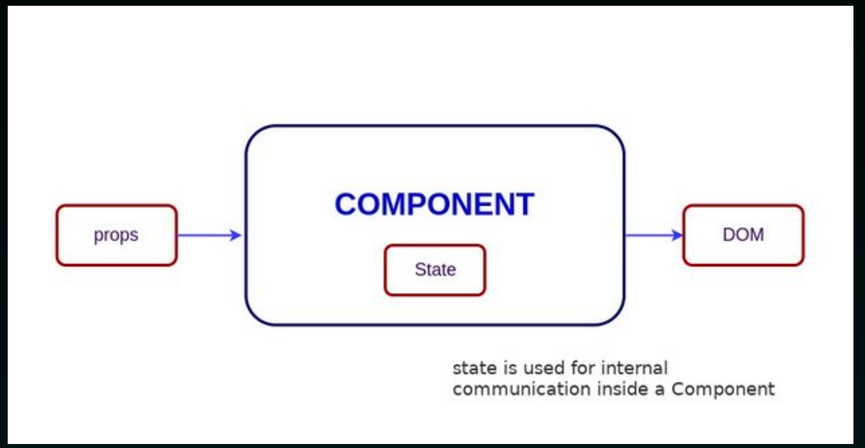
“Ad ogni cambio di stato React aggiornerà efficientemente solamente le parti della UI che dipendono da tali dati.”

Velocità, facilità e qualità di sviluppo
Non reinventiamo la ruota
Alte performance

Componenti

- I componenti sono parti di UI *indipendenti, isolati* e *riutilizzabili*

  Un componente è una funzione che può accettare dei dati in input (*props*) e avere il suo stato (*state*) e ritorna elementi React



- Possiamo creare UI complesse composte da più componenti semplici
- Ogni componente può contenere altri componenti

Function component

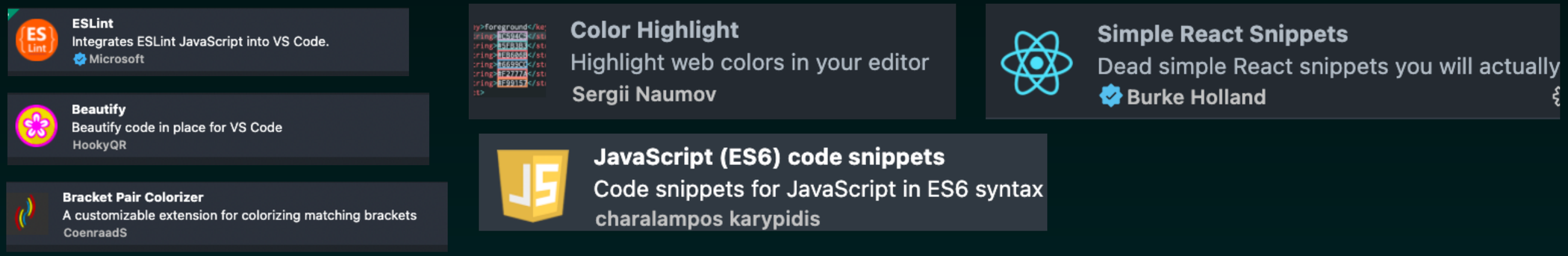
```
function Ciao(props) {
  return <h1>Ciao, {props.nome}</h1>;
}
```

Class component

```
class Ciao extends React.Component {
  render() {
    return <h1>Ciao, {this.props.nome}</h1>;
  }
}
```

Setup ambiente di sviluppo

- Node / ecosistema npm
  node_modules, packages
- VS Code
- Browser (estensioni di chrome)



Hello world

- https://it.reactjs.org/docs/create-a-new-react-app.html
- Struttura cartelle
- Primo componente (class vs function)
- Props
- Componenti innestati

JSX

JSX è un'estensione di Javascript che permette di scrivere del codice Javascript che sembra HTML, rendendolo più semplice da leggere e da gestire.

```
1  import React from "react";
2  import ReactDom from "react-dom";
3
4  const element = React.createElement(
5    "div",
6    {
7      className: "my-class"
8    },
9    "Hello world!"
10 );
11
12 ReactDom.render(element, document.getElementById("root"));
```

```
1  import React from "react";
2  import ReactDom from "react-dom";
3
4
5  const element = (
6    <div className="my-class">Hello world"</div>
7  );
8
9  ReactDom.render(element, document.getElementById("root"));
10
```

https://it.reactjs.org/docs/jsx-in-depth.html

JSX

- “React” deve essere importato

```
import React from 'react';
```

- I componenti HTML standard avranno sintassi invariata, i componenti definiti dall’utente dovranno avere la prima lettera maiuscola

```
// Correct! This is a component and should be capitalized:
function Hello(props) {
  // Correct! This use of <div> is legitimate because div is a valid HTML tag:
  return <div>Hello {props.toWhat}</div>;
}

function HelloWorld() {
  // Correct! React knows <Hello /> is a component because it's capitalized.
  return <Hello toWhat="World" />;
}
```

- class —> className
- kebab-case —> camel-case

```
const element = <div tabIndex="0"></div>;
```

JSX

- Per utilizzare javascript in jsx basta includerlo tra parentesi graffe { }

```
const name = 'Giuseppe Verdi';
const element = <h1>Hello, {name}</h1>;
```

- Formattazione condizionale

  . In linea con operatore ternario

```
return <div>{userLogged ? <div>Welcome</div> : <div>Please login</div>}</div>;
```

  . Diversi render

```
if (userLogged) {
  return <div>Welcome</div>;
}
return <div>Please login</div>;
```

JSX- Eventi

- In JSX, il gestore di eventi (*event handler*) viene passato come funzione, piuttosto che stringa.

```
<button onclick="attivaLasers()">
  Attiva Lasers
</button>
```

```
<button onClick={attivaLasers}>
  Attiva Lasers
</button>
```