

Sviluppo front end per il web con JavaScript e React

Simone Capodivento

February 2026

Contents

Introduction	1
HTML	1
CSS	1
JavaScript	1
AJAX	1
JQuery	1
React	2
Formato jsx	2
Exercises	2
Installazione delle dipendenze	2
app.jsx	3
products.jsx	4
product.jsx	5
main.jsx	6
Annotazioni personali	6
Risorse	6

Introduction

L'obiettivo di questo documento è quello di riassumere all'osso quello che ho visto durante il corso di Sviluppo frontend per il web con JavaScript e React. Era basato così: partiti da html, abbiamo visto come rendere dinamica una pagina web con appunto JavaScript e le sue varie librerie per poi districare tutti i concetti utilizzando React.

HTML

Questo è un linguaggio di sviluppo di pagine web, tutte le pagine **Google**. Possono essere ispezionate con il tasto destro e poi Ispeziona. Per dichiarare una funzione si utilizza `jQuery`. È inoltre un linguaggio di markup: ogni contenuto è dentro un tag, anche detto linguaggio prolioso. Inoltre è un linguaggio gerarchico.

CSS

È un linguaggio dedicato allo stile delle pagine web.

1

1

È buona abitudine dividere i file html, css e js

1

JavaScript

1

Possiede varie caratteristiche in comune con Java. È utile per aiutare l'utente finale che non sa come navigare nella nostra applicazione. È uno script che viene eseguito al refresh della page. Inoltre possiede queste caratteristiche:

1

- interpretato
- OO

2

2

Il principale utilizzo di JavaScript è quello di dinamicizzare una web page.

2

2

AJAX

3

È un insieme di tecniche di sviluppo web che permettono di aggiornare parti di una pagina web in background, senza ricaricarla interamente.

6

```
$.ajax({
  type: "GET",
  success: function(data) {
    console.log(data);
  },
  error: function(richiesta, stato, errori) {
    alert("Errore nel caricamento dati");
  }
});
```

Questo è come si richiama AJAX¹.

JQuery

Questa è una libreria di javascript utilizzata per semplificare la manipolazione del DOM e soprattutto la gestione degli eventi.

¹Il web funziona con json files, invece del formato XML.

React

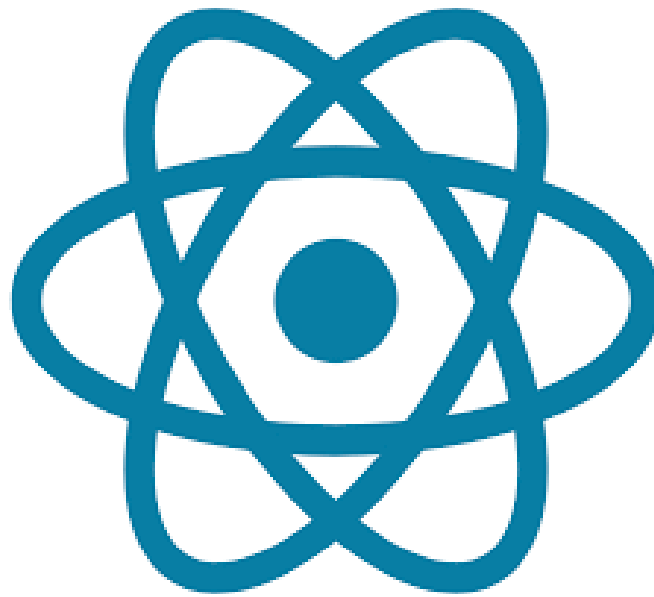


Figure 1: React

È stato ideato da Facebook ²

Formato jsx

È un linguaggio basato su JavaScript

Exercises

Abbiamo iniziato con l'installazione del template in terminale di vscode:

```
npm create vite@latest nameProject --template react
```

Installazione delle dipendenze

npm install

Per installare tutte le librerie e tutto l'occorrente per la nostra web app.

npm run dev per lanciare la nostra webapp

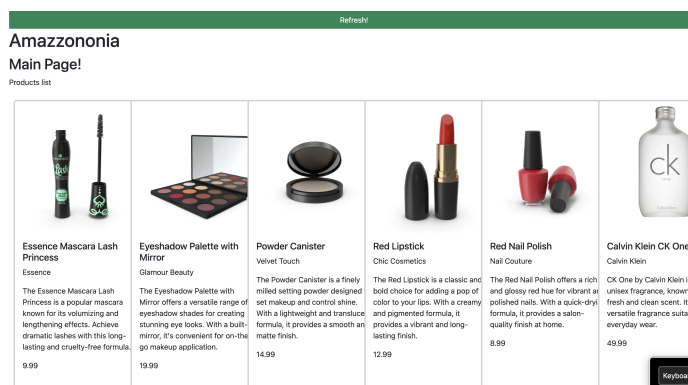


Figure 2: La webapp completa

Qui abbiamo usato l'ispezione per vedere se le attività che vogliamo che la nostra webapp svolga vengano compiute correttamente, come il loading button, la chiamata api, il bottone refresh e il caricamento delle immagini.

²Angular è la controparte ideata da Google. Il cui compito è quello di gestire le interfacce utente. SI utilizza nel JavaScript; inoltre, è utile per la sicurezza e scardina alcuni concetti ³ e introduce il concetto di stato.

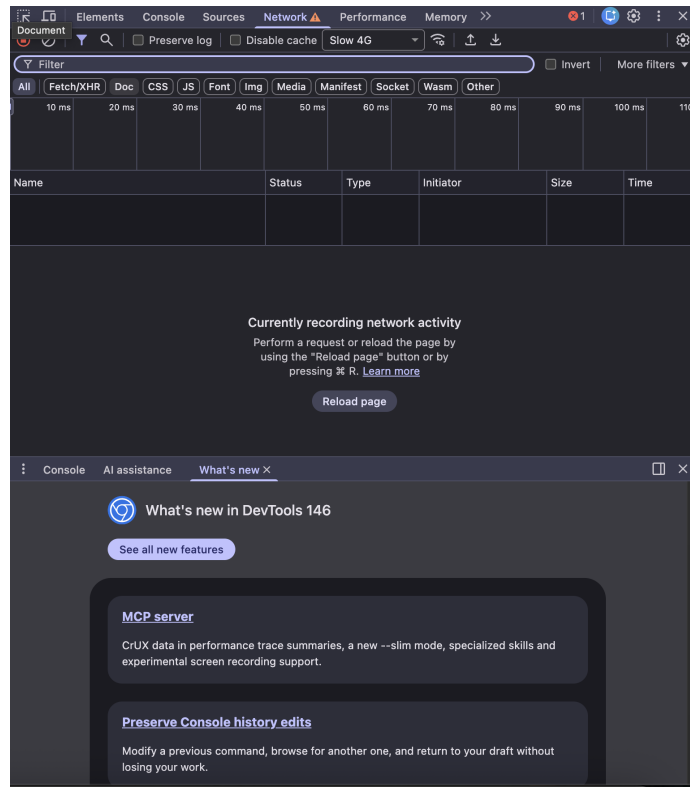


Figure 3: Devtools

Passando al codice, abbiamo `app.jsx`, `products.jsx`, `product.jsx` e il `main`.

`app.jsx`

```
import 'bootstrap/dist/css/bootstrap.min.css';  
import Products from './components/Products';
```

```
function App()  
{  
  console.log("APP()");  
  
  {  
  
  }  
  
  return (  
    <>  
  
    <Products/>  
  
  </>  
)
```

```
}
```

```
export default App;
```

products.jsx

```
import Product from "../Product";
import { useState, useEffect } from "react";
import axios from "axios";
import Row from "react-bootstrap/Row";
import Col from "react-bootstrap/Col";

function Products() {
  const [products, setProducts] = useState([]);
  const [error, setError] = useState(null);
  const [refresh, setRefresh] = useState(0);
  const [loading, setLoading] = useState(null)

  useEffect(() => {
    async function fetchProducts() {
      setLoading(true);
      try {
        const response = await axios.get("https://dummyjson.com/products");
        setProducts(response.data.products);
      } catch (error) {
        console.dir(error);
        setError(error);
      } finally {
        setLoading(false);
      }
    }

    fetchProducts();

    return () => {
      console.log("Component unmounted");
    };
  }, [refresh]);
  function changeRefresh()
  {
    setRefresh(refresh + 1);
  }

  /*render() {
    let { loading } = this.state; // *** Get the flag

    let theList = this.state.map((item, id) => {
      */
  return (

    <section>

      <Row>
      <button class='btn btn-success pull-right'
      onClick={changeRefresh}>Refresh!</button>

      <h1>Amazzononia</h1>
      <h2>Main Page!</h2>
    </Row>
    <p>Products list </p>
  )
}
```

```
{error && <p style={{ color: "red" }}>Error: {error.message}</p>}
```

```
{loading && (
  <div className="d-flex justify-content-center my-4">
    <div className="spinner-border text-primary" role="status">
      <span className="visually-hidden">Loading... </span>
    </div>
  </div>
)}

<Row xs={2} md={4} lg={6}>
  {products.length > 0 ? (
    products.map((product, index) => (
      <Col key={index} className="flex-grow-1">
        <Product productP={product} />
      </Col>
    ))
  ) : (
    <Col><p>No products!</p></Col>
  )}
</Row>
</section>
);
}
```

```
export default Products;
```

product.jsx

```
import Card from 'react-bootstrap/Card';
```

```
function Product( {productP} ) {
```

```
  return (
    <Card style={{ width: '18rem',borderWidth: '3px', margin: '10px', minHeight: '100px', min
      <Card.Img variant="top" src={productP.images[0]}
    />
    <Card.Body>
      <Card.Title>{productP.title}</Card.Title>
      <Card.Text>
        {productP.brand}
      </Card.Text>
      <Card.Text>
        {productP.description}
      </Card.Text>
      <Card.Text>
        {productP.price}
      </Card.Text>
    </Card.Body>
  </Card>
  );
}
```

```
/*export default BasicExample;
```

```
  <p>product details.</p>
  <h2>{productP.title}</h2>
  <p>brand: {productP.brand}</p>
  <p>price: {productP.price}</p>
```

```

        <p>description: {productP.description}</p>
    )
}
    */
export default Product;

```

main.jsx

```

    //import { StrictMode } from 'react'
import { createRoot } from 'react-dom/client'
//import './index.css'
import App from './App.jsx'

createRoot(document.getElementById('root')).render(
  //<StrictMode>
  <App />
  //</StrictMode>,
)

```

Annotazioni personali

tecnologie utili.

- ReactBootstrap
- React Router
- Next.js

Risorse

VITE <https://vite.dev/guide/> Node js <https://nodejs.org/en> <https://axios-http.com/docs/intro>
<https://flowbite-react.com/> <https://reactnative.dev/> [https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Spread_syntax)
[US/docs/Web/JavaScript/Reference/Operators/Spread_syntax](https://docs/Web/JavaScript/Reference/Operators/Spread_syntax) [https://developer.mozilla.org/en](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules) -
[US/docs/Web/JavaScript/Guide/Modules](https://docs/Web/JavaScript/Guide/Modules)