

04_01 REACT 18.2

Fornisce metodi DOM uniformi, gerarchia più esplicita, componenti di livello più alto rispetto ad HTML, processamento automatico di eventi e aggiornamenti.

Approccio **dichiarativo**:

- Non si manipola esplicitamente il DOM
- Non si esplicita l'ordine delle operazioni
- Si definisce solo quando renderizzarsi

Approccio di **design funzionale**:

- Componenti come funzioni
- Re-renderizza tutto in caso di cambiamenti
 - Stato componente cambia
 - Nuova proprietà passata
- Gestione esplicita dello stato dell'applicazione

Caratteristica:

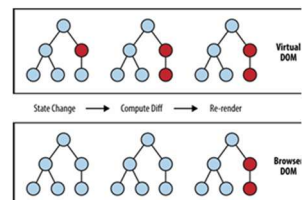
- Idempotente
- Immutabile

React si aspetta funzioni che siano pure (se devo renderizzare un componente, devo succedere solo questo e non devono esserci side-effects).

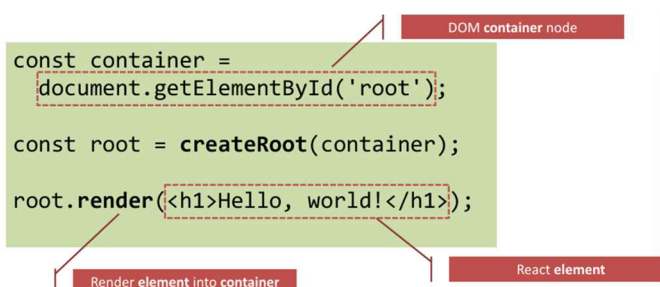
Componenti vengono renderizzati da 0, ogni volta che avviene un singolo cambiamento.

React ri-renderizza tutto ma dato che le manipolazioni del DOM sono costose, React e librerie simili a React implementano un **Virtual Dom**: layer presente sopra al DOM originale (Struttura dati interna, in memoria, ottimizzata per avere aggiornamenti molto veloci). Dunque, React non manipola il DOM direttamente ma lavora con il virtualDOM e, ad un certo punto, React manda al DOM. Inoltre, il Virtual DOM corregge alcune anomalie e assimetrie, gestisce eventi sintetici (click). Dopo che avviene una re-renderizzazione di un componente, React calcola la differenza tra ciò che è sulla pagina e cosa è presente nel DOM, e passa al DOM effettivamente solo le cose da modificare.

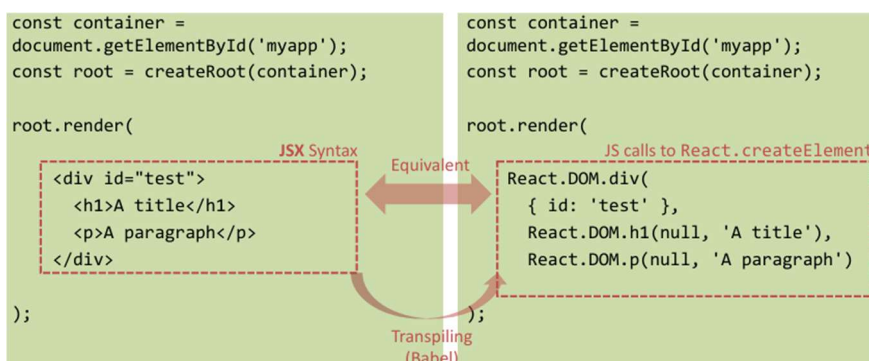
- Build new Virtual DOM tree
- Diff with old one
- Compute minimal set of changes
- Put them in a queue
- Batch render all changes to browser



Esempio base con React



→renderizza all'interno del DOM del container



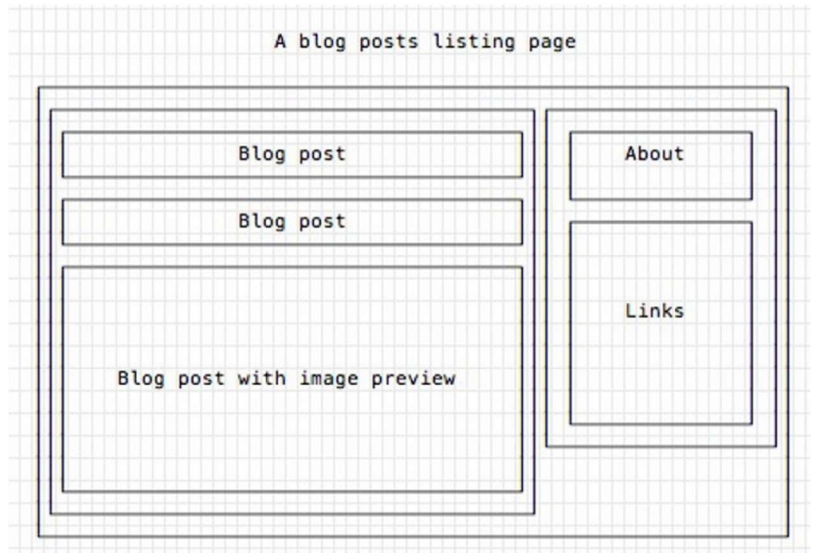
Noi usiamo quello a sx, il compilatore lo traduce nella sintassi js presente a dx.
Nota: non è html, sono simili ma operano sul virtual DOM.

Componenti:

Ogni cosa in una pagina React è un componente.

Ci sono 8 elementi ma 6 componenti in quanto blog post anche se presente 3 volte, è un componente che viene istanziato con un contenuto diverso per 3 volte.

Nota: elemento è un'istanza di componente.



Componente è una funzione che deve avere un return → altri elementi react che devono essere renderizzati.

As a function, returning DOM elements

```
const BlogPostExcerpt = () => {
  return (
    <div>
      <h1>Title</h1>
      <p>Description</p>
    </div>
  )
}
```

The function may receive some props

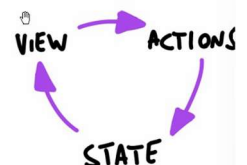
```
const BlogPostContent = (props) => {
  return (
    <div>
      <p>{props.content}</p>
    </div>
  )
}
```

Tipi di componenti:

- **Presentazione**
 - Generano nodi DOM da essere mostrati
 - Non gestiscono lo stato dell'applicazione
 - Possono avere qualche informazione interna per scopi legati alla presentazione
- **Container:**
 - Gestiscono stato di componenti figli
 - Possono interagire con back-end
 - Creano figli per visualizzare informazione vera e propria

Ogni componente può gestire **2 diverse proprietà:**

- **Props:** passati da padre, in sola lettura
 - **Valori:** per configurare come i componenti devono essere mostrati
 - **Funzioni:** per accedere ai metodi del componente padre
- **Stato:** insieme di variabili interne al componente, immutabile
 - Può essere inizializzato con valori di default o con valori props
 - Può essere modificato chiamando metodi specifici asincroni
 - Cambiamento di stato causa re-rendering del virtualDOM
 - Stato può essere passato ai figli come props
 - State is passed to the view and to child components
 - Actions are triggered by the view
 - Actions can update the state
 - The state change is passed to the view and to child component



Corollario: uno stato è sempre posseduto da un componente:

- Ogni dato che è affected da questo stato, può solo modificare componenti sotto esso, suoi figli.

Cambiamento di stato di un componente, influenza il proprio rendering e quello dei propri figli → infatti lo stato viene messo nel componente più alto possibile di react.

APPLICAZIONE REACT

1. `npm create vite@latest my-app`
2. From the menu, choose **React**, then **JavaScript**
3. `cd my-app`
4. `npm install`
5. ⌚ ... 65 Megabytes later ... ⌚
6. `npm run dev`
7. Visit <http://localhost:5173>

Nota: in react si può creare un css per ogni componente.

- Css generale
- Css specifico per componente

Nota: si può installare qualcosa che porta con se componenti, es: react-bootstrap.

- `npm install react-bootstrap`
- `npm install bootstrap`

```
import 'bootstrap/dist/css/bootstrap.min.css'
import { Col, Container, Row } from 'react-bootstrap';

import MyButton from './Button.jsx';

function App() {
  return (
    <Container>
      <Row>
        <Col>
          Premi qui: <MyButton lang='it' />
        </Col>
      </Row>
    </Container>
  );
}

export default App;
```

```
import { useState } from "react";
import { Button } from "react-bootstrap";

function MyButton(props) {
  let [buttonLang, setButtonLang] = useState(props.lang) ;

  if (buttonLang === 'it')
    return <Button variant='primary' onClick={()=>setButtonLang('en')}>Ciao!</Button>
  else
    return <Button variant='primary' onClick={()=>setButtonLang('it')}>Hello!</Button>
}

export default MyButton;
```