

React JS

React è una libreria che permette di creare interfacce utente.

(Pinterest usa react <3)

Caratteristiche:

- **Virtual dom**
- **Componenti**: un'app fatta con react è un insieme di componenti
- **JSX**: linguaggio di React che combina Java, HTML e CSS

Dom

Struttura dati che rappresenta la gerarchia delle pagine web, una volta fatto posso modificarlo ma fa "pesare" il server, perchè per renderlo dinamico faccio "aggiornare" il server e viene mandato di nuovo anche se uguale.

Virtual dom

React prede il dom salvato sul Browser e lo salva, alla modificazione del Dom si crea un altro Virtual dom. React confronta i due Dom e, se trova delle modifiche e solo in questo caso, modifica il Dom "effettivo" e lo sostituisce con il **Real Dom**.

Componenti

Sono come delle classi che accettano dei valori (props) e restituiscono in output HTML.

props

Oggetti le cui chiavi corrispondono alle proprietà.

NON cambiano.

Possono essere passate di "padre in figlio" ma non viceversa.

(Prima lettera in Maiuscolo)

Es

Lista di impiegati:

Abbiamo più componenti:

- La pagina degli impiegati
- La lista degli impiegati
- Gli elementi della lista
- La barra di ricerca

Stato

Oggetto contenente dati dinamici che vengono "osservati", ad ogni cambiamento viene aggiornato in tempo reale.

Per modificare uno stato si usa **setState** che prende in input il nuovo stato.

```
this.setState({click= true});
```

Ciclo di vita

- `construct()`
- `componentWillUpdate()`
- `render()`
- `componentDidMount()`

Tipi di componenti:

Funzionali

I componenti funzionali sono:

- **semplici da scrivere**
- **limitati**: non posso accedere al ciclo di vita e non ha neanche uno stato.

Classe

- stato
- ciclo di vita

Componente **contenitore**.

Es

Per chiamare JS all'interno dell'output devo usare le {}

Babel

Precompila il codice e lo rende leggibile dal Browser