

[Github URL](#)

[Github URL for Vercel](#)

[Vercel URL](#)

W13-P1: Demo prop drilling

=> Chrome, remove 4th person

The screenshot shows a web application running on localhost:3000. The page title is 'Prop Drilling'. The main content area displays a list of names: John, Peter, and Susan. Each name is followed by a 'Remove' button. A red box highlights the 'Remove' button for John. The right side of the image shows the DevTools component inspector and props panel. The component inspector shows the 'PropDrilling_11' component, and the props panel shows the state of the application, including the list of names.

```
components
  Fragment key="c"
    SegmentViewNode key="c-page"
      SegmentViewNode key="page"
        ClientPageRoot
          PropDrilling_11
            Tutorial_11_wrapper
              List
                SinglePerson key="1"
                SinglePerson key="2"
                SinglePerson key="3"
            Next.MetadataOutlet
              next_outlet_boundary
                Suspense name="Next.MetadataOut..."
                SegmentBoundaryTriggerNode
                  SegmentViewStateNode key="/tutorials/cont..."
                    BoundaryTriggerNode
                      PropDrilling_11
                        props
                          params: bound Promise
                          searchParams: bound Promise
                          new_entry: ""
                        hooks
                          1 State: [(-), (-), (-)]
                            0: {id: 1, name: "john"}
                            1: {id: 2, name: "peter"}
                            2: {id: 3, name: "susan"}
                          new_entry
                        rendered by
                          <ClientPageRoot>
                            hydrateRoot()
```

=> relevant code

```
src > app > (root) > tutorials > context > 1_propdrilling_11 > page.jsx > PropDrilling_11 > reo
1 'use client';
2
3 import { useState } from 'react';
4 import Wrapper from '../_assets/_wrapper/Tutorials_11';
5
6 import { data } from '../_assets/_data/data';
7
8 // more components
9 // fix - context api, redux (for more complex cases)
10
11 const PropDrilling_11 = () => {
12   const [people, setPeople] = useState(data);
13   console.log('data', data);
14
15   const removePerson = (id) => {
16     setPeople((people) => {
17       return people.filter((person) => person.id !== id);
18     });
19   };
20
21   return (
22     <Wrapper>
23       <div className='container'>
24         <h3>prop drilling</h3>
25         <List people={people} removePerson={removePerson} />
26       </div>
27     </Wrapper>
28   );
29 };
30
31 const List = ({ people, removePerson }) => {
32   return (
33     <div>
34       {people.map((person) => {
35         return (
36           <div>
37             <SinglePerson
38               key={person.id}
39               {...person}
40               removePerson={removePerson}
41             />
42           </div>
43         );
44       })}
45     </div>
46   );
47 };
48
49 const SinglePerson = ({ id, name, removePerson }) => {
50   return (
51     <div className='item'>
52       <h4>{name}</h4>
53       <button className='btn' onClick={() => removePerson(id)}>
54         remove
55       </button>
56     </div>
57   );
58 };
59
60 export default PropDrilling_11;
```

3a020ce Hao Yi Chen

Mon Dec 15 00:54:19 2025 +0800 W13-P1: Demo prop drilling

W13-P2: Context API demo for W13-P1

=> Chrome, show PeopleContext.Provider by remove last person

The screenshot shows a web application running on localhost:3000/tutorials/context/2_context_people_11. The application displays a list of names: John, Peter, and Susan, each with a 'Remove' button. The DevTools component inspector is open, showing the component tree. The 'Context.Provider' component is highlighted, and its props are visible. The props include a 'value' object with 'people' (an array of three objects) and 'removePerson' (a function). The 'people' array contains three objects: {id: 1, name: 'john'}, {id: 2, name: 'peter'}, and {id: 3, name: 'susan'}.

Context API Demo

John
Remove
Peter
Remove
Susan
Remove

Context.Provider

```
props
  children: <ForwardRef ./>
  value: {people: Array(3), removePerson: t1() {}}
    people: [{id: 1, name: 'john'}]
    people: [{id: 2, name: 'peter'}]
    people: [{id: 3, name: 'susan'}]
    removePerson: t1() {}
    new entry: --
    new entry: --
  rendered by
    PeopleContextProvider_11 @ _context.jsx:19
    <PeopleContextProvider_11>
    PropDrilling_11 @ page.jsx:12
```

=> relevant code

```
src > app > (root) > tutorials > context > 2_context_people_11 > _context.jsx > PeopleContextPro
1 'use client';
2
3 import { useState, createContext, useContext } from 'react';
4 import { data } from '../../assets/_data/data';
5
6 const PeopleContext = createContext();
7
8 export const PeopleContextProvider_11 = ({ children }) => {
9   const [people, setPeople] = useState(data);
10  console.log('data', data);
11
12  const removePerson = (id) => {
13    setPeople(people => {
14      return people.filter((person) => person.id !== id);
15    });
16  };
17
18  return (
19    <PeopleContext.Provider value={{ people, removePerson }}>
20      {children}
21    </PeopleContext.Provider>
22  );
23
24 export const usePeopleContext_11 = () => {
25   return useContext(PeopleContext);
26 }
27
src > app > (root) > tutorials > context > 2_context_people_11 > page.jsx > ...
5 import { usePeopleContext_11, PeopleContextProvider_11 } from './
6 _context';
7
8 // more components
9 // fix - context api, redux (for more complex cases)
10
11 const PropDrilling_11 = () => {
12   return (
13     <PeopleContextProvider_11>
14       <Wrapper>
15         <div className='container'>
16           <h3>Context API Demo</h3>
17           <List />
18         </div>
19       </Wrapper>
20     </PeopleContextProvider_11>
21   );
22
23 const List = () => {
24   const { people } = usePeopleContext_11();
25   return (
26     <>
27       {people.map((person) => {
28         return <SinglePerson key={person.id} {...person}/>;
29       })}
30     </>
31   );
32
33 const SinglePerson = ({ id, name }) => {
34   const { removePerson } = usePeopleContext_11();
35   return (
36     <div className='item'>
37       <h4>{name}</h4>
38       <button className='btn' onClick={() =>
39         removePerson(id)>
40         remove
41       </button>
42     </div>
43   );
44
45 }
```

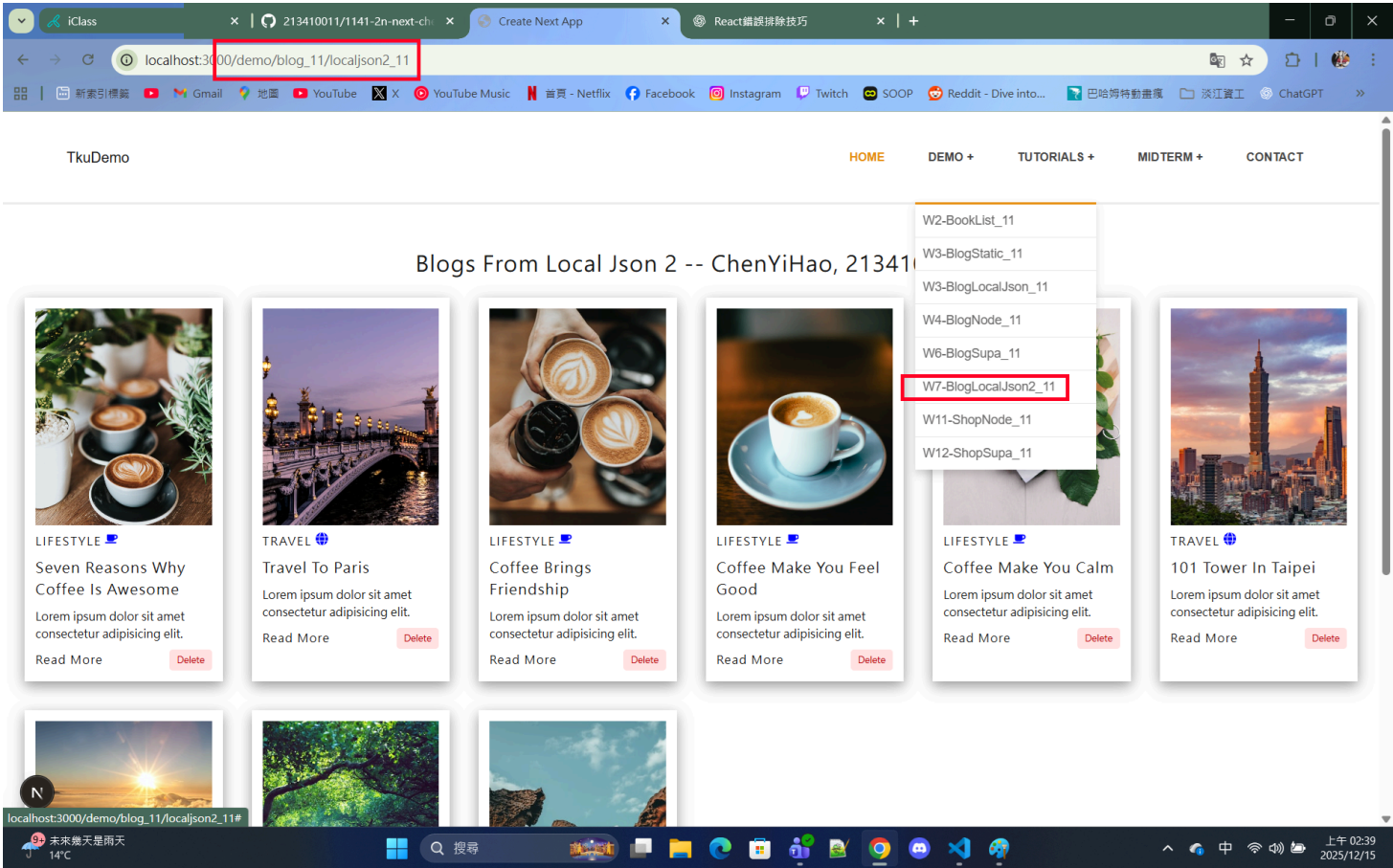
2963e66 Hao Yi Chen

Mon Dec 15 01:53:04 2025 +0800

W13-P2: Context API demo for W13-P1

W13-P3: Move W7 BlogJson2 page to the route /demo/blog_xx/localjson2_xx

=> Chrome, show the result



=> relevant code

```
src > app > (root) > demo > blog_11 > localjson2_11 > page.jsx > BlogLocalJsonPage2_11
1 'use client'
2 import { useState } from 'react';
3
4 import blogData_11 from '../_assets/data/blogData.json';
5 import Blog2_11 from '../_components/Blog2_11';
6 import Wrapper from '../_assets/wrapper/Blog2_11';
7 import Alert_11 from '../_components/Alert_11';
8
9 const BlogLocalJsonPage2_11 = () => {
10   const [name, setName] = useState('ChenYiHao');
11   const [id, setid] = useState(213410011);
12   const [blogs_11, setBlogs_11] = useState(blogData_11);
13   const [alert, setAlert] = useState({
14     show: false,
15     msg: '',
16     type: '',
17   });
18
19   const showAlert = (show = false, msg = '', type = '') => {
20     setAlert({ show, msg, type });
21   };
22
23   const removeAllBlogs = (id) => {
24     showAlert(true, 'blog removed', 'danger');
25     setBlogs_11(blogs_11.filter((blog) => blog.id !== id));
26   };
27
28   const clearAllBlogs = () => {
29     showAlert(true, 'clear all blogs', 'danger');
30     setBlogs_11([]);
31   };
32
33   const loadAllBlogs = () => {
34     showAlert(true, 'load all blogs', 'success');
35     setBlogs_11([]);
36     setBlogs_11(blogData_11);
37   };
38
39   return (
40     <wrapper>
41       {alert.show && <Alert_11 alert={alert} showAlert={showAlert} />}
42       <section className='blogs'>
43         <div className='section-title'>
44           <h2>
45             blogs from local json 2 -- {name}, {id}{' '}
46           </h2>
47         </div>
48         <div className='blogs-center'>
49           {blogs_11.map((item) => {
50             const { id, title, descrip, category, img } = item;
51             return (
52               <Blog2_11
53                 key={id} ...
54                 removeItem={removeItem}
55               />
56             );
57           })}
58         </div>
59         <div className='flex justify-center items-center gap-8 mt-8'>
60           <button
61             type='button'
62             className='text-red-700 bg-red-200 hover:bg-red-300'
63             onClick={clearAllBlogs}
64           >
65             clear all blogs
66           </button>
67           <button
68             type='button'
69             className='text-blue-700 bg-blue-200 hover:bg-blue-300'
70             onClick={loadAllBlogs}
71           >
72             load all blogs
73           </button>
74         </div>
75       </section>
76     </wrapper>
77   );
78 }
79
80 export default BlogLocalJsonPage2_11;
```

cdcd162 Hao Yi Chen Mon Dec 15 02:43:34 2025 +0800 W13-P3: Move W7 BlogJson2 page to the r