



11. Llistes i Keys



Llistes i Keys

```
import React from 'react';
import { useParams } from 'react-router-dom';

function Refugis(){
  const nombres = [1, 2, 3, 4, 5];
  const doble = nombres.map((nombre) => nombre * 2);
  console.log(doble);
  const { id } = useParams();
  return (
    <>
    <h1>Llista de refugis</h1>
    <ul>
      <li><a href="/refugis/1">Refugi 1</a></li>
      <li><a href="/refugis/2">Refugi 2</a></li>
    </ul>
    <h2>El refugi seleccionat és { id }</h2>
    </>
  )
}

export default Refugis;
```

Primer anem a revisar com transformar llistes en Javascript i com administrarles en React.



Llistes i Keys: Renderitzat de múltiples components

```
import React from 'react';
import { useParams } from 'react-router-dom';

function Refugis(){
  const nombres = [1, 2, 3, 4, 5];
  const listRefus = nombres.map((nombre) =>
    <li><a href={`/refugis/${nombre}`}>Refugi {`${nombre}`}</a></li>
  );
  const { id } = useParams();
  return (
    <>

    <h1>Llista de refugis</h1>
    <ul>
      {listRefus}
    </ul>
    <h2>El refugi seleccionat és { id }</h2>
    </>
  )
}
```

`export default Refugis;`

Podeu fer col·leccions d'elements i incloure'ls a JSX usant claus `{}`.

A sota, recorrerem l'array numbers usant la funció `map()` de Javascript. Tornem un element `` per cada ítem . Finalment, assignem l'array d'elements a `listItems`



Llistes i Keys: Coneixent les Keys

```
import React from 'react';
import { useParams } from 'react-router-dom';
```

```
function Refugis(){
  const nombres = [1, 2, 3, 4, 5];
  const listRefus = nombres.map((nombre) =>
    <li key={nombre.toString()}><a href={`/${refugis}/${nombre}`}>Refugi {`${nombre}`}</a></li>
  );
  const { id } = useParams();
  return (
    <>
      <h1>Llista de refugis</h1>
      <ul>
        {listRefus}
      </ul>
      <h2>El refugi seleccionat és { id }</h2>
    </>
  )
}
```

Les keys ajuden a React a identificar quins ítems han canviat, són agregats, o són eliminats. Les keys han de ser donades als elements dins de l'array per donar als elements una identitat estable. També podem emprar un camp de l'objecte com `key={refugi.id}` o emprar l'index del ítem dins l'array (com a últim recurs):

```
const todosItems = todos.map((todo, index) =>
  <li key={index}>
    {todo.text}
  </li>
```



Llistes i Keys: Coneixent les Keys

```
import React from 'react';
import { useParams } from 'react-router-dom';

function Refugis(){
  const ListRefus = refugios.map((refugio) =>
    <li key={refugio.refugio_id}><a href={`/${refugis}/${refugio.refugio_id}`}>`${refugio.name}`</a></li>
  );
  const { id } = useParams();
  return (
    <>

    <h1>Llista de refugis</h1>
    <ul>
      {ListRefus}
    </ul>
    <h2>El refugi seleccionat és { id }</h2>
    </>

  )
}
```

export default Refugis;

Adaptem les llistes i Keys el projecte de refugis.



Llistes i Keys: Coneixent les Keys

```
import { useState, useEffect } from 'react';
```

```
function Rutes() {
```

```
  const [articles, setArticulos] = useState([])
```

```
  useEffect(() => {
```

```
    fetch('https://scratchya.com.ar/react/datos.php')
```

```
    .then((response) => {
```

```
      return response.json()
```

```
    })
```

```
    .then((articles) => {
```

```
      setArticulos(articles)
```

```
    })
```

```
  }, [])
```

Ara realitzarem una prova connectant amb una API Rest.

Recordem que un dels objectius dels Hook d'efectes són les peticions de dades a un servidor.

Definim una variable d'estat per emmagatzemar un acord amb tots els productes recuperats del servidor.



Llistes i Keys: Coneixent les Keys

```
return (  
  <div>  
    <table border="1">  
      <thead>  
        <tr>  
          <th>Código</th>  
          <th>Descripción</th>  
          <th>Precio</th>  
        </tr>  
      </thead>  
      <tbody>  
        {articles.map(art => {  
          return (  
            <tr key={art.codigo}>  
              <td>{art.codigo}</td>  
              <td>{art.descripcion}</td>  
              <td>{art.precio}</td>  
            </tr>  
          );  
        })}  
      </tbody>  
    </table>  
  </div>  
);
```

El bloc JSX es mostra la capçalera de la taula HTML i mitjançant el recorregut de la variable d'estat es mostren els articles recuperats del servidor