섹션 5: 렌더링 리스트 및 조건부 Content

작성일시	@February 24, 2023 9:30 AM
강의날짜	@2023/02/24
편집일시	@February 24, 2023 12:48 PM
분야	React
공부유형	강의
복습	□
태그	

69. 데이터의 렌더링 목록

```
// Expenses.js
import React, { useState } from 'react';

import './Expenses.css';
import Card from '../UI/Card';
import ExpenseItem from './ExpenseItem';
import ExpensesFilter from './ExpensesFilter';

const Expenses = props => {
  const [filteredYear, setFilteredYear] = useState('2020');

  const filterChangeHandler = selectedYear => {
    setFilteredYear(selectedYear);
  };

  return (
    <div>
      <Card className="expenses">
// [1] map method 이용
        {props.items.map(expense => (
          <ExpenseItem
            title={expense.title}
            amount={expense.amount}
            date={expense.date}
          />
        ))}
      </Card>
    </div>
  );
};

export default Expenses;
```

70. State 저장 목록

```
// App.js
// [1] useState 추가
import React, { useState } from 'react';

import Expenses from './components/Expenses/Expenses';
import NewExpense from './components/NewExpense/NewExpense';

// [2] DUMMY_EXPENSES 밖으로 빼기
const DUMMY_EXPENSES = [
  {
    id: 'e1',
    title: 'Toilet Paper',
    amount: 94.12,
    date: new Date(2020, 7, 14),
  },
  { id: 'e2', title: 'New TV', amount: 799.49, date: new Date(2021, 2, 12) },
  {
    id: 'e3',
    title: 'Car Insurance',
    amount: 294.67,
    date: new Date(2021, 2, 28),
  },
  {
    id: 'e4',
    title: 'New Desk (Wooden)',
    amount: 450,
    date: new Date(2021, 5, 12),
  },
];

const App = () => {
// [3] useState
  const [expenses, setExpenses] = useState(DUMMY_EXPENSES);

// [4] addExpenseHandle 함수 수정
  const addExpenseHandler = expense => {
    setExpenses(prevExpenses => {
      return [expense, ...prevExpenses];
    });
  };
  return (
    <div>
      <NewExpense onAddExpense={addExpenseHandler} />
      <Expenses items={expenses} />
    </div>
  );
};

export default App;
```

71. “keys” 이해하기

```
// Expenses.js
import React, { useState } from 'react';

import './Expenses.css';
import Card from '../UI/Card';
import ExpenseItem from './ExpenseItem';
import ExpensesFilter from './ExpensesFilter';

const Expenses = props => {
  const [filteredYear, setFilteredYear] = useState('2020');

  const filterChangeHandler = selectedYear => {
    setFilteredYear(selectedYear);
  };

  return (
    <div>
      <Card className="expenses">
        <ExpensesFilter
          selected={filteredYear}
          onChangeFilter={filterChangeHandler}
        />
        {props.items.map(expense => (
          <ExpenseItem
// [1] 고유값으로 key={expense.id} 추가하기
            key={expense.id}
            title={expense.title}
            amount={expense.amount}
            date={expense.date}
          />
        ))}
      </Card>
    </div>
  );
};

export default Expenses;
```

73. 연습하기 : Lists 다루기 (정답)

```
// Expenses.js
import React, { useState } from 'react';

import './Expenses.css';
import Card from '../UI/Card';
import ExpenseItem from './ExpenseItem';
import ExpensesFilter from './ExpensesFilter';

const Expenses = props => {
  const [filteredYear, setFilteredYear] = useState('2020');

  const filterChangeHandler = selectedYear => {
    setFilteredYear(selectedYear);
  };

// [1] filter() method 사용
  const filteredExpenses = props.items.filter(expense => {
// [2] getFullYear() => 4자리수 연도 가져옴
    return expense.date.getFullYear().toString() === filteredYear;
  });

  return (
    <div>
      <Card className="expenses">
        <ExpensesFilter
          selected={filteredYear}
          onChangeFilter={filterChangeHandler}
        />
// [3] props.items -> filteredExpenses 로 바꾸기
        {filteredExpenses.map(expense => (
          <ExpenseItem
            key={expense.id}
            title={expense.title}
            amount={expense.amount}
            date={expense.date}
          />
        ))}
      </Card>
    </div>
  );
};

export default Expenses;
```

74. 조건부 내용 출력하기

```
// Expenses.js
import React, { useState } from 'react';

import './Expenses.css';
import Card from '../UI/Card';
import ExpenseItem from './ExpenseItem';
import ExpensesFilter from './ExpensesFilter';

const Expenses = props => {
  const [filteredYear, setFilteredYear] = useState('2020');

  const filterChangeHandler = selectedYear => {
    setFilteredYear(selectedYear);
  };

  const filteredExpenses = props.items.filter(expense => {
    return expense.date.getFullYear().toString() === filteredYear;
  });

  return (
    <div>
      <Card className="expenses">
        <ExpensesFilter
          selected={filteredYear}
          onChangeFilter={filterChangeHandler}
        />
// [2] 두번째 방법
{filteredExpenses.length === 0 && <p>No expenses found.</p>}
        {filteredExpenses.length > 0 &&
          filteredExpenses.map(expense => (
            <ExpenseItem
              key={expense.id}
              title={expense.title}
              amount={expense.amount}
              date={expense.date}
            />
          ))}


// [1] 첫번째 방법
        {filteredExpenses.length === 0 ? (
          <p>No expenses found.</p>
        ) : (
          filteredExpenses.map(expense => (
            <ExpenseItem
              key={expense.id}
              title={expense.title}
              amount={expense.amount}
              date={expense.date}
            />
          ))
        )}
      </Card>
    </div>
  );
};

export default Expenses;


// [3] 세번째 방법 : JSX 로직 따로, 가장 깔끔한 로직
import React, { useState } from 'react';

import './Expenses.css';
import Card from '../UI/Card';
import ExpenseItem from './ExpenseItem';
import ExpensesFilter from './ExpensesFilter';

const Expenses = props => {
  const [filteredYear, setFilteredYear] = useState('2020');

  const filterChangeHandler = selectedYear => {
    setFilteredYear(selectedYear);
  };

  const filteredExpenses = props.items.filter(expense => {
    return expense.date.getFullYear().toString() === filteredYear;
  });

// expensesContent 따로
  let expensesContent = <p>No expenses found.</p>;

  if (filteredExpenses.length > 0) {
    expensesContent = filteredExpenses.map(expense => (
      <ExpenseItem
        key={expense.id}
        title={expense.title}
        amount={expense.amount}
        date={expense.date}
      />
    ));
  }

  return (
    <div>
      <Card className="expenses">
        <ExpensesFilter
          selected={filteredYear}
          onChangeFilter={filterChangeHandler}
        />
// {expensesContent} 추가
        {expensesContent}
      </Card>
    </div>
  );
};

export default Expenses;
```

76. 연습하기 : 조건부 내용 (정답)

```
// NewExpenses.js
// [1] useState 추가
import React, { useState } from 'react';

import ExpenseForm from './ExpenseForm';
import './NewExpense.css';
```

```
const NewExpense = props => {

// [3] useState로 상태 관리 / 초깃값 : false
  const [isEditing, setIsEditing] = useState(false);

  const saveExpenseDataHandler = enteredExpenseData => {
    const expenseData = {
      ...enteredExpenseData,
      id: Math.random().toString(),
    };
    props.onAddExpense(expenseData);
    setIsEditing(false);
  };

// [4] startEditingHandler 함수 추가
  const startEditingHandler = () => {
    setIsEditing(true);
  };

// [8] stopEditingHandle 함수 추가
  const stopEditingHandler = () => {
    setIsEditing(false);
  };

  return (
    <div className="new-expense">
// [2] button 추가
// [4] onClick 속성 추가
// [5] 시작부분에 !, &&사용 !isEditing일 때는 button
      {!isEditing && (
        <button onClick={startEditingHandler}>Add New Expense</button>
      )}
// [6] isEditing일 때는 ExpenseForm 보여주기
      {isEditing && (
        <ExpenseForm
          onSaveExpenseData={saveExpenseDataHandler}
          onCancel={stopEditingHandler}
        />
      )}
    </div>
  );
};

export default NewExpense;


// ExpenseForm.js
import React, { useState } from 'react';

import './ExpenseForm.css';

const ExpenseForm = props => {
  const [enteredTitle, setEnteredTitle] = useState('');
  const [enteredAmount, setEnteredAmount] = useState('');
  const [enteredDate, setEnteredDate] = useState('');

  const titleChangeHandler = event => {
    setEnteredTitle(event.target.value);
  };

  const amountChangeHandler = event => {
    setEnteredAmount(event.target.value);
  };

  const dateChangeHandler = event => {
    setEnteredDate(event.target.value);
  };

  const submitHandler = event => {
    event.preventDefault();

    const expenseData = {
      title: enteredTitle,
      amount: enteredAmount,
      date: new Date(enteredDate),
    };

    props.onSaveExpenseData(expenseData);
    setEnteredTitle('');
    setEnteredAmount('');
    setEnteredDate('');
  };

  return (
    <form onSubmit={submitHandler}>
      <div className="new-expense__controls">
        <div className="new-expense__control">
          <label>Title</label>
          <input
            type="text"
            value={enteredTitle}
            onChange={titleChangeHandler}
          />
        </div>
        <div className="new-expense__control">
          <label>Amount</label>
          <input
            type="number"
            value={enteredAmount}
            onChange={amountChangeHandler}
            min="0.01"
            step="0.01"
          />
        </div>
        <div className="new-expense__control">
          <label>Date</label>
          <input
            type="date"
            value={enteredDate}
            onChange={dateChangeHandler}
            min="2019-01-01"
            max="2022-12-31"
          />
        </div>
      </div>
      <div className="new-expense__actions">
// [7] Cancel 버튼 추가, onClick 속성 추가
        <button type="button" onClick={props.onCancel}>
          Cancel
        </button>
        <button type="submit">Add Expense</button>
      </div>
    </form>
  );
};

export default ExpenseForm;
```

78. 데모 앱: 차트 추가하기

```
// Chart.js
import React from 'react';

import ChartBar from './ChartBar';
import './Chart.css';

const Chart = props => {
  return (
    <div className="chart">
      {props.dataPoints.map(dataPoint => (
        <ChartBar
          key={dataPoint.label}
          value={dataPoint.value}
          maxValue={null}
          label={dataPoint.label}
        />
      ))}
    </div>
  );
};

export default Chart;


// ChartBar.js
import React from 'react';

import './CharBar.css';

const ChartBar = props => {
  let barFillHeight = '0%';

  if (props.max > 0) {
    barFillHeight = Math.round((props.value / props.maxValue) * 100) + '%';
  }

  return (
    <div className="chart-bar">
```

```
      <div className="chart-bar__inner">
        <div
          className="chart-bar__fill"
          style={{ height: barFillHeight }}
        ></div>
      </div>
      <div className="chart-bar__label"> {props.label} </div>
    </div>
  );
};

export default ChartBar;
```