



섹션 4: 리액트 State 및 이벤트 다루기

⌚ 작성일시	@February 22, 2023 2:47 PM
📅 강의날짜	@2023/02/22
⌚ 편집일시	@February 24, 2023 9:03 AM
▼ 분야	React
▼ 공부유형	강의
☑ 복습	<input type="checkbox"/>
☰ 태그	

50. 이벤트 리스닝 및 이벤트 핸들러 수행하기

```
// ExpenseItem.js
import ExpenseDate from '../ExpenseDate';
import Card from '../UI/Card';
import './ExpenseItem.css';

const ExpenseItem = props => {
  // [2] 함수 clickHandler 구현
  const clickHandler = () => {
    console.log('Clicked!!!');
  };

  return (
    <Card className="expense-item">
      <ExpenseDate date={props.date} />
      <div className="expense-item__description">
        <h2>{props.title}</h2>
        <div className="expense-item__price">${props.amount}</div>
      </div>
      // [1] onClick={clickHandler} 함수 이름 지정
      <button onClick={clickHandler}>Change Title</button>
    </Card>
  );
};
```

```
export default ExpenseItem;
```

52. “State”와 함께 일하기

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

import ExpenseDate from './ExpenseDate';
import Card from '../UI/Card';
import './ExpenseItem.css';

const ExpenseItem = props => {
  // [2] props.title = 초기값 / useState에 인자로 전달 /
  const [title, setTitle] = useState(props.title);

  const clickHandler = () => {
    setTitle('Updated!');
    console.log(title);
  };

  return (
    <Card className="expense-item">
      <ExpenseDate date={props.date} />
      <div className="expense-item__description">
        <h2>{title}</h2>
        <div className="expense-item__price">${props.amount}</div>
      </div>
      <button onClick={clickHandler}>Change Title</button>
    </Card>
  );
};

export default ExpenseItem;
```

53. "useState" 혹 자세히 살펴보기

useState

- 몇몇 state를 등록
- 컴포넌트별 인스턴스를 기반으로 해서 독립적인 state를 갖음
- 항상 두 개의 값을 얻음 : 현재 상태값과 업데이트하는 함수
- state가 변할 때마다 업데이트 함수를 호출

- 상태값을 사용하고 싶을 때마다 컴포넌트형 함수를 다시 실행하고 JSX코드를 다시 평가
- 응용프로그램에게 반응성 추가

57. 여러 State 다루기

```
// ExpenseForm.js
const ExpenseForm = () => {
  // [3] useState('') 사용하기 <- 문자열로 저장
  const [enteredTitle, setEnteredTitle] = useState('');
  // [2] const titleChangeHandler 함수 작성
  const titleChangeHandler = event => {
    setEnteredTitle(event.target.value);
  };

  return (
    <form>
      <div className="new-expense__controls">
        <div className="new-expense__control">
          <label>Title</label>
          // [1] onChange={titleChangeHandler} 작성
          <input type="text" onChange={titleChangeHandler} />
        </div>
      </div>
    </form>
  );
};
```

58. State 대신 사용하기(그리고 더 나은 방법)

```
// ExpenseForm.js
// State를 한번만 이용하며 같이 관리하는 법
// title만 바뀌어도 다른 Amount나 Date의 input을 같이 전송해야함
// ...userInput, => spread 연산자 사용
const ExpenseForm = () => {
  // const [enteredTitle, setEnteredTitle] = useState('');
  // const [enteredAmount, setEnteredAmount] = useState('');
  // const [enteredDate, setEnteredDate] = useState('');
  // [1] useState 사용해서 하나로 묶기
  const [userInput, setUserInput] = useState({
    enteredTitle: '',
    enteredAmount: '',
    enteredDate: '',
  });

  const titleChangeHandler = event => {
    // setEnteredTitle(event.target.value);
  };
  // [2] 각각의 Handler 안에 다른 input 저장하는 함수
  setUserInput({
    ...userInput, // spread 연산자 사용해서 다른 input 저장
    enteredTitle: event.target.value,
  });
};
```

```

    });
  };

  const amountChangeHandler = event => {
    // setEnteredAmount(event.target.value);
    setUserInput({
      ...userInput,
      enteredAmountL: event.target.value,
    });
  };

  const dateChangeHandler = event => {
    // setEnteredDate(event.target.value);
    setUserInput({
      ...userInput,
      enteredDate: event.target.value,
    });
  };
};

```

60. 양식 제출 처리

```

import React, { useState } from 'react';

import './ExpenseForm.css';

const ExpenseForm = () => {
  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);
  };

  // [2] submitHandle 함수
  const submitHandler = event => {
    event.preventDefault();
  };
  // [3] 합체
  const expenseData = {
    title: enteredTitle,
    amount: enteredAmount,
    date: new Date(enteredDate),
  };

  console.log(expenseData);
};

```

```

    };

    return (
      // [1] form에 대한 제출함수
      <form onSubmit={handleSubmit}>
        <div className="new-expense__controls">
          <div className="new-expense__control">
            <label>Title</label>
            <input type="text" onChange={titleChangeHandler} />
          </div>
          <div className="new-expense__control">
            <label>Amount</label>
            <input
              type="number"
              onChange={amountChangeHandler}
              min="0.01"
              step="0.01"
            />
          </div>
          <div className="new-expense__control">
            <label>Date</label>
            <input
              type="date"
              onChange={dateChangeHandler}
              min="2019-01-01"
              max="2022-12-31"
            />
          </div>
        </div>
        <div className="new-expense__actions">
          <button type="submit">Add Expense</button>
        </div>
      </form>
    );
  };

  export default ExpenseForm;

```

61. 양방향 바인딩 추가

- 상태를 업데이트하기 위해 입력에서 변경사항을 수신하는 것뿐만 아니라 입력에 상태를 다시 보내주기도 함
- 상태를 변경하면 입력도 변함

```

import React, { useState } from 'react';

import './ExpenseForm.css';

const ExpenseForm = () => {
  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),
  };

  console.log(expenseData);

// [2] 빈 칸으로 되돌려주기
  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}
// [1] onChange 로 양방향 바인딩
          onChange={titleChangeHandler}
        />
      </div>
      <div className="new-expense__control">
        <label>Amount</label>
        <input
          type="number"
          value={enteredAmount}
// [1] onChange 로 양방향 바인딩
          onChange={amountChangeHandler}
          min="0.01"
          step="0.01"
        />
      </div>
      <div className="new-expense__control">
        <label>Date</label>
        <input
          type="date"
          value={enteredDate}

```

```
// [1] onChange 로 양방향 바인딩
    onChange={dateChangeHandler}
    min="2019-01-01"
    max="2022-12-31"
  />
</div>
</div>
<div className="new-expense__actions">
  <button type="submit">Add Expense</button>
</div>
</form>
);
};

export default ExpenseForm;
```

62. 자식 대 부모 컴포넌트 통신 (상향식)

1. 부모 컴포넌트로부터 자식 컴포넌트로 함수를 전달
2. 자식 컴포넌트에서 그 함수를 호출
3. 함수를 호출했을 때 그 함수에 매개변수로 데이터 전달

```
// App.js => 조부모 컴포넌트
import React from 'react';

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

const App = () => {
  const 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),
    },
  ],
  ];
```

```

// [5] addExpensehandler 함수 추가
const addExpenseHandler = expense => {
  console.log('In App.js');
  console.log(expense);
};

return (
  <div>
// [6] onAddExpense로 addExpensehandler 전달
    <NewExpense onAddExpense={addExpenseHandler} />
    <Expenses items={expenses} />
  </div>
);
};

export default App;

// NewExpense.js => 부모 컴포넌트
import React from 'react';

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

// [7] props
const NewExpense = props => {

// [2] 상수 saveExpenseDataHandler 추가
// 중요한 것 : 매개변수로 enteredExpenseData를 설정
  const saveExpenseDataHandler = enteredExpenseData => {
    const expenseData = {
      ...enteredExpenseData,
      id: Math.random().toString(),
    };
  };
//[8] props expenseData보냄
  props.onAddExpense(expenseData);
};

  return (
    <div className="new-expense">
// [1] onSaveExpenseData
// 이 속성에 대한 값은 함수여야 하고 컴포넌트 내부에서 어떤 일이 벌어졌을 때 작동되는 함수
// 사용자가 입력된 ExpenseData를 저장할 때인데 폼이 마지막에 제출될 때를 의미함
      <ExpenseForm onSaveExpenseData={saveExpenseDataHandler} />
    </div>
  );
};

export default NewExpense;

// ExpenseForm.js => 자식 컴포넌트
import React, { useState } from 'react';

import './ExpenseForm.css';

// [3] props
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),
  };

  // [4] props에 onSaveExpensedData실행
  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}
        />
      </div>
    </div>
  </form>
);

```

```

        onChange={dateChangeHandler}
        min="2019-01-01"
        max="2022-12-31"
      />
    </div>
  </div>
  <div className="new-expense__actions">
    <button type="submit">Add Expense</button>
  </div>
</form>
);
};

export default ExpenseForm;

```

65. 연습하기 : 이벤트 및 State 작업하기 (정답)

```

// Expenses.js => 부모 컴포넌트
import React, { useState } from 'react';

import './Expenses.css';
import Card from '../UI/Card';
import ExpenseItem from './ExpenseItem';
// [1] 자식 컴포넌트 import
import ExpensesFilter from './ExpensesFilter';

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

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

  return (
    <div>
      <Card className="expenses">
        // [2] 자식 컴포넌트 ExpensesFilter 배치
        <ExpensesFilter
          // [7] 양방향binding selected={filteredYear}
          selected={filteredYear}
          onChangeFilter={filterChangeHandler}
        />
        <ExpenseItem
          title={props.items[0].title}
          amount={props.items[0].amount}
          date={props.items[0].date}
        />
        <ExpenseItem
          title={props.items[1].title}
          amount={props.items[1].amount}
          date={props.items[1].date}
        />
        <ExpenseItem

```

```

        title={props.items[2].title}
        amount={props.items[2].amount}
        date={props.items[2].date}
      />
      <ExpenseItem
        title={props.items[3].title}
        amount={props.items[3].amount}
        date={props.items[3].date}
      />
    </Card>
  </div>
);
};

export default Expenses;

// ExpensesFilter.js => 자식 컴포넌트
import React from 'react';

import './ExpensesFilter.css';

// [3] props
const ExpensesFilter = props => {
  // [5] dropdownChangeHandler
  const dropdownChangeHandler = event => {
    console.log(event.target.value);
  };
  // [6] event.target.value props
  props.onChangeFilter(event.target.value);
};

return (
  <div className="expenses-filter">
    <div className="expenses-filter__control">
      <label>Filter by year</label>
      // [4] onChange 함수
      // [8] 양방향 binding value={props.selected}
      <select value={props.selected} onChange={dropdownChangeHandler}>
        <option value="2022">2022</option>
        <option value="2021">2021</option>
        <option value="2020">2020</option>
        <option value="2019">2019</option>
      </select>
    </div>
  </div>
);
};

export default ExpensesFilter;

```