

## 5. Form Input

- Form Input
- 여러 개의 상태 다루기

### Form Input

- `<form>` `<input>` tag를 통해 값을 입력할 수 있는 요소를 추가
- 데이터를 유저로부터 받을 수 있는 중요한 요소 중 하나
- 값을 입력 받을 땐 `onChange` 이벤트 리스너를 통해 변화를 감지

`user agent`

- 종류

button, checkbox, color, date, file, number, text, password, radio, submit, range, 등등

### 여러개의 상태 다루기

- 각각을 하나씩 state로 다루기

```
const [state, setState] = useState(initState1);
const [state, setState] = useState(initState2);
const [state, setState] = useState(initState3);
```

- 여러 state를 하나의 객체에 다루기

```
const [state, setState] = useState(initState);
```

- 만약 객체인 `state` 를 다룰 때 `setState` 에서 `state` 값을 불러와서 쓰고 싶다면?

```
setState(prevState => {...prevState});
```

### 리액트 상태 part4

- starter : [github.com/dev-owen/react-fundamental/tree/practice 3-4](https://github.com/dev-owen/react-fundamental/tree/practice%203-4)  
이름, 금액, 날짜를 입력해서 그 값을 `console.log` 로 찍어보기

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

const Payment = () => {
  const [name, setName] = useState('');
  const [price, setPrice] = useState(0);
  const [today, setToday] = useState(null);
}
return (
  <input type='text' value={name} />
)
```

`input` 에 각각 `value` 선언하기

=====> 초기 값 설정 완료

```
<input type='text' value={name} onChange={} />
```

`onChange` 를 통해 `input` 값이 바뀌면 바뀐 것이 이벤트로 발생하는 것

그 이벤트가 발생하면, 그 이벤트를 감지해서 핸들러를 만들어 놓으면, 그 핸들러가 실행이 되고

이벤트가 바뀔때마다 로직이 수행이 된다.

😞 이벤트 핸들러는 어떤 역할을 하면 좋을까?

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

const Payment = () => {
  const [name, setName] = useState('');
  const [price, setPrice] = useState(0);
  const [today, setToday] = useState(null);

  //이벤트 핸들러 생성
  const inputTypeTextChangeHandler = (event)=> {
    setName(event.target.value);
  };
  const inputTypeNumberChangeHandler = (event) => {
    setPrice(event.target.value);
  };
  const inputTypeDateChageHandler = (event) => {
    setToday(event.target.value);
  };
  const buttonSubmitHandler = () => {
    console.log("name", name);
    console.log("price", price);
    console.log("today", today);
  }

}
return (
  <input type='text'
    value={name}
    onChange={inputTypeTextChangeHandler}
  />
)
```

- HTMLElement : input

```
<div className="new-payment__actions">
  <button type="button" onClick={buttonSubmitHandler} >결제 추가</button>
</div>
```

```
const Payment = () => {
  const [objectState, setObjectState] = useState({
    name: '',
    price: 0,
    today: null
  });
  const inputTypeTextChangeListener = (event) => {
    setObjectState(prevState => ({...prevState, name: event.target.value}));
  };
  const inputTypeNumberChangeHandler = (event) => {
    setObjectState(prevState => ({...prevState, price: event.target.value}));
  };
  const inputTypeDateChangeHandler = (event) => {
    setObjectState(prevState => ({...prevState, today: event.target.value}));
  };
  const buttonSubmitHandler = () => {
    console.log("name", objectState.name);
    console.log("price", objectState.price);
    console.log("today", objectState.today);
  }

  return (
    <input type="text"
      value={objectState.name}
      onChange={inputTypeTextChangeListener}
    />
  )
}
```

- form input은 `<form>` `<input>` 태그를 통해 값을 입력할 수 있는 요소를 추가
- form input에 값을 입력하는 변화는 onChange 메서드를 사용
- 여러 개의 상태 처리 시 각각 처리해 주어도 되고, 객체로 하나로 만들어서 처리해 주는 것도 가능

- `<input>` : The Input (Form Input) element