

12. 동적인 스타일링

- 동적인 인라인 스타일링
- CSS 클래스를 동적으로 조작하기

practice/3-11

1. 동적인 인라인 스타일링

빈 input으로 값을 입력하려 하면 배경색을 빨간색으로 바꾸기

동적 : 때에 따라서 상황에 따라서 달라진다.

```
const CourseInput = (props) => {
  const [enteredValue, setEnteredValue] = useState("");
  const [isValid, setIsValid] = useState(true);

  const goalInputChangeHandler = (event) => {
    if(event.target.value.trim().length > 0){
      setIsValid(true);
    }
    setEnteredValue(event.target.value);
  }

  const formSubmitHandler = (event) => {
    event.preventDefault();
    //값이 ''경우, 배경을 빨간색으로 해 주어야 합니다.
    if(enteredValue.trim().length === 0){
      setIsValid(false);
      setEnteredValue("");
      // return 꼭 해줘야 한다.
      return;
    }
    props.onAddGoal(enteredValue);
  };
}
```

```
return (
  <form onSubmit={formSubmitHandler}>
    <div>
      //isValid가 아닐때 (빈 string을 입력했을 때)는 invalid 클래스 명이 들어가고
      //아닐때는 빈 string이 되어서 invalid가 되지 않도록 하는 조건절
      className={`form-control ${!isValid ? 'invalid' : ''}`}>
      <label>목표</label>
      <input type="text" onChange={goalInputChangeHandler}
        style={{backgroundColor: isValid? "transparent" : "salmon",
          borderColor: isValid? "#ccc" : "red"}}
      />
    </div>
  </form>
)
```

2. CSS 클래스를 동적으로 조작하기

인라인 스타일링을 CSS클래스를 사용해서 바꿔주기

```
.form-control {  
  margin: 0.5rem 0;  
}  
  
.form-control label{  
  font-weight: bold;  
  display: block;  
  margin-bottom: 0.5rem;  
}  
  
.form-control input{  
  display: block;  
  width: 100%;  
  border: 1px solid gray;  
  font: inherit;  
  line-height: 1.5rem;  
  padding: 0 0.25rem;  
}  
  
.form-control input:focus{  
  outline: none;  
  border-color: blue;  
}
```

```
.form-control.invalid input{  
  background-color: salmon;  
  border-color: red;  
}
```