

ToDoMVC(ToDoList) Clone Project

이정희

프로젝트 설명 및 목적

- ToDoMVC(<http://todomvc.com>) 똑같이 개발하기
 - 기본적인 toDoList 기능 구현
- 하나는 Javascript로 하나는 AngularJS로 개발
 - Javascript & AngularJS 학습

결과

ORIGINAL



CLONE



구현 기능 목록

1. ToDo 등록, 삭제, 수정, 체크
2. 완료가 안 된 ToDo들 개수 표시하기
3. 완료된 ToDo들 한번에 삭제하기
4. 한번에 ToDo 완료 check 또는 uncheck(Toggle)
5. All, Completed, Active 버튼을 누르면 해당 상태의 toDo들만 보여주기

등록 Process

1. Enter를 누르면 입력한 text 받기 - `getInputText` 함수
2. `todoObject` 생성(`status`, `text`)
3. object를 바탕으로 `todo`를 보여주는 `div Element(todoElement)` 생성
- `createToDoElement` 함수
4. `todoList` `div`에 append - `appendToDoElement` 함수



등록 코드

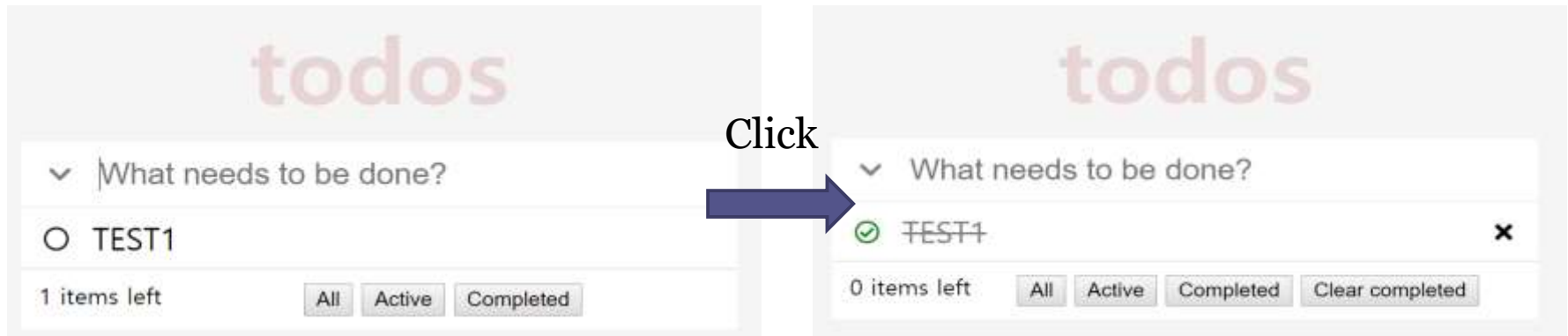
```
var addNewToDo = function(event){
  if(event.keyCode !== 13) return; // enter를 누를 시 toDo 등록
  var text = getInputText();
  if(!text) return;
  addToDo(text);
  updateFooterInfo();
};

var addToDo = function(text){
  var toDoObject = createToDoObject("active",text); // toDoObject 생성
  var toDoElement = createToDoElement(toDoObject); // toDo를 표시하는 Element생성
  appendToDoElement(toDoElement); // Element붙이기
};

//inputBox에서 text 입력받기
var getInputText = function(){
  var inputText = document.getElementById('todo_input').value;
  if(inputText.replace(/^\s+/g, '').length === 0) return;
  if(isToDoExist(inputText)) return;
  document.getElementById('todo_input').value = "";
  return inputText;
};
```

Check & unCheck Process

1. click 이벤트가 발생한 toDoObject의 status 수정(active ↔ completed)
 - changeToDoObjectStatus함수
2. 바뀐 toDoObject 정보를 바탕으로 새로운 toDoElement 생성
 - createToDoElement함수 : status에 따라 버튼 class가 다르게 설정
3. 기존 toDoElement와 새로운 toDoElement replace
 - changeToDoElement함수



Check & unCheck 코드

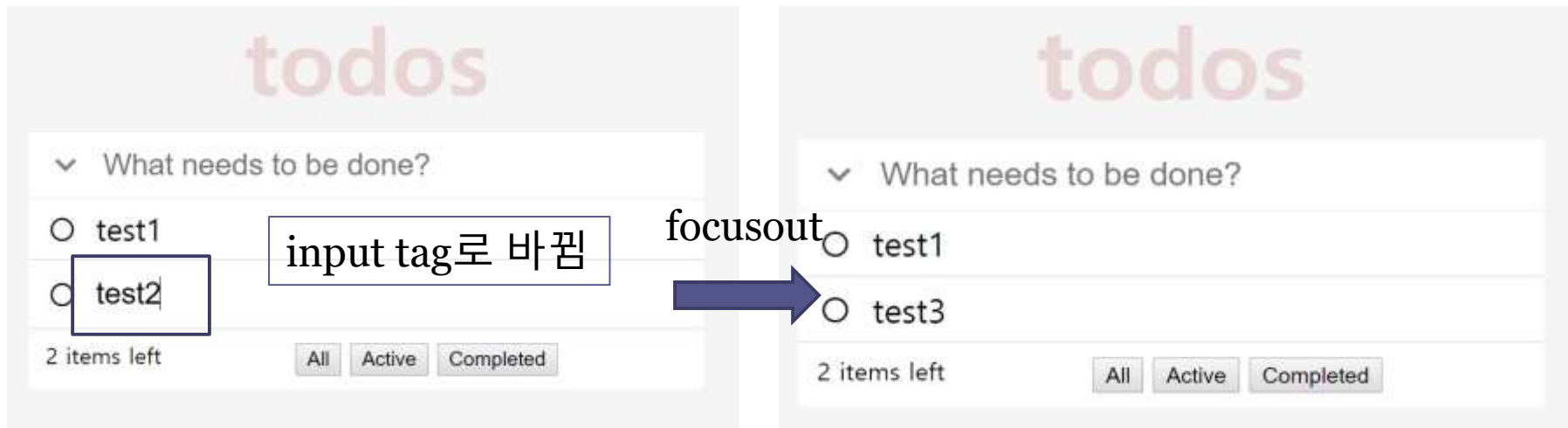
```
//changeToDoStatus
var changeToDoObjectStatus = function(todoElement){
    var text = getToDoElementText(todoElement);
    var todoObject = getToDoObjectByText(text);
    todoObject.status = (todoObject.status === "active")? "completed" : "active";
    return todoObject;
};

// 변경된 todoObject를 바탕으로 새로운 element생성 후 targetToDoElement와 replace
var changeToDoElement = function(targetToDoElement, changedToDoObject){
    var changedToDoElement = createToDoElement(changedToDoObject);
    document.getElementById("todoList").replaceChild(changedToDoElement, targetToDoElement);
};

// click event 발생 시 (check, uncheck)
var changeToDoStatus = function(event){
    // 이벤트 발생한 todoElement 가져오기
    var targetToDoElement = getTargetToDoElement(event);
    // todoObject의 status 변경
    var changedToDoObject = changeToDoObjectStatus(targetToDoElement);
    changeToDoElement(targetToDoElement, changedToDoObject);
    updateFooterInfo(); // active todo count update, clearCompletedBtn 보여주거나 숨기기
};
```


수정 Process

1. dblclick 이벤트가 발생한 toDoElement의 text를 input tag로 변경
2. focusout 이벤트 발생 시 수정 완료
3. 수정된 text값으로 toDoObject text 변경
4. 원래 toDoElement와 변경된 object 정보로 만든 toDoElement replace



수정 코드

```
//edit - dblclick시
var editToDo = function(event){
    var targetToDoElement = getTargetToDoElement(event); //event가 발생한 toDoElement가져오기
    var toDoObject = getToDoObjectByElement(targetToDoElement); //event가 발생한 toDoObject가져오기
    startEdit(targetToDoElement);
    endEdit(targetToDoElement,toDoObject); // focusout 시 수정 완료
};

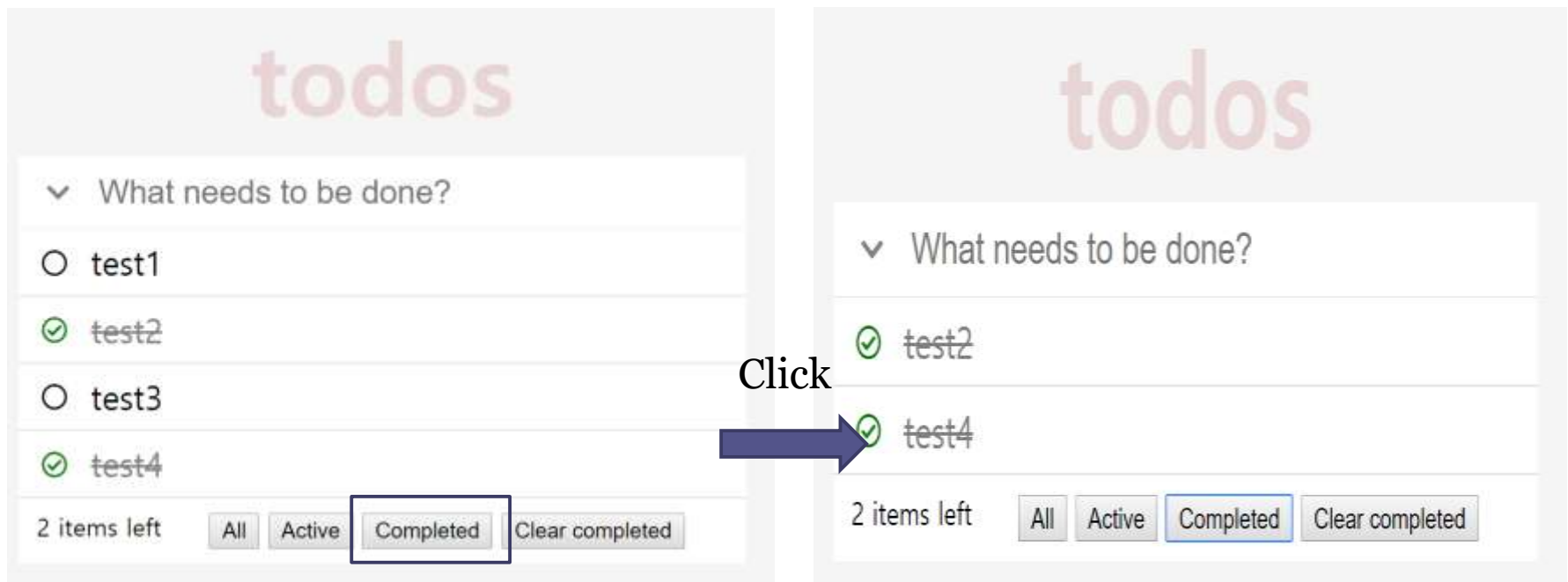
var endEdit = function(toDoElement,toDoObject){
    $(".toDo_text > input").on("focusout", function(){
        editToDoText(toDoElement,toDoObject); // input tag를 변경한 text로 바꿔 수정 완료
    });
};

var startEdit = function(toDoElement){
    replaceTextToInput(toDoElement); // text를 input tag로 바꾸기
    offEvents(toDoElement); // 삭제 버튼을 보여주기 위한 이벤트 off
};

var editToDoText = function(toDoElement,toDoObject){
    var text = $(toDoElement).find("input").val(); // input value 가져오기
    if(isToDoExist(text)) return; // 이미 존재하면 수정 x
    toDoObject.text = text; // toDoObject의 text 바꾸기
    changeToDoElement(toDoElement,toDoObject); |
};
```

Completed만 보여주는 Process

1. toDoList를 전부 지우기
2. toDoArr를 루프 돌면서 status가 completed인 toDoObject 찾기
3. toDoElement를 생성 후 append



Completed만 보여주는 코드

```
//show completed toDo
$("#completed").on("click", function(){
    drawToDoListByStatus("completed");
});

var drawAllToDoList = function(){
    removeAllToDoElement();
    drawAllToDoObjects();
};

//status가 parameter에 들어온 값과 일치하는 toDoObject를 찾으면 toDoElement 생성후 append
var drawToDoObjectsByStatus = function(status){
    for(var index in toDoArr){
        if(status !== toDoArr[index].status) continue;
        var toDoObject = createToDoElement(toDoArr[index]);
        document.getElementById("toDoList").appendChild(toDoObject);
    }
};

// toDoList 내용 지우기
var removeAllToDoElement = function(){
    $("#toDoList").empty();
};
```

전체 Code

1. Javascript version

<https://github.com/jjoylee/JavascriptAndAngularStudy/blob/master/Javascript/toDoList-version2.js> - js file

<https://github.com/jjoylee/JavascriptAndAngularStudy/blob/master/Javascript/toDoList.html> - html file

<https://github.com/jjoylee/JavascriptAndAngularStudy/tree/master/Javascript/css> - css files

2. AngularJS version

<https://github.com/jjoylee/JavascriptAndAngularStudy/tree/master/Angularjs/jeongheeToDo>