

---

# 웹 접근성을 고려한 jQuery 유저 인터페이스 효과 플러그인

이선웅\* · 문일영\*

\*한국기술교육대학교

## User Interface Effect Plugin using jQuery for Web Accessibility

Seonung Lee\* · Il-Young Moon\*

\*Korea University of Technology and Education

### 요 약

본 논문에서는 웹 접근성을 고려한 유저 인터페이스 효과를 jQuery로 구현하는 방법을 제안하였다. jQuery는 세계에서 가장 많이 사용하고 확장하기 편한 자바스크립트 라이브러리이다. jQuery는 DOM(Document Object Model)과 이벤트, Ajax의 제어에 적합하지만 고급 효과를 나타내기에는 부족한 부분이 있다. 이를 보완하고 UI(User Interface) 효과를 쉽게 적용할 수 있는 방법을 제시한다.

### ABSTRACT

In this paper, we propose an implementation of user interaction effect of web application using jQuery. jQuery is a very popular and extensible JavaScript library. It is strong in controlling DOM(Document Object Model), events and Ajax. But, it is poor at advanced effect of web user UI(User Interface). Implementing advanced effects need to extend jQuery or some skills. So, our proposal is a good way to implement UI effects.

### 키워드

Web Accessibility, jQuery, Web UI

### 1. 서 론

웹 기술이 플랫폼으로 진화되어 가면서 HTML 5는 크로스플랫폼(Cross-platform)을 지원하며 video와 canvas와 같은 몇 가지 새로운 기능들을 추가하였다 [1]. HTML5는 넓게 본다면 세 가지 주요 기술들을 포함한다. 이는 문서의 구조를 나타내는 HTML5 자체, 문서양식과 각 엘리먼트들의 모양새를 지정하는 CSS3, 그리고 웹 페이지에서의 동작을 기술하는 JavaScript이다[2].

현대의 웹사이트들은 JavaScript를 사용하며, 특히 Ajax(Asynchronous JavaScript and XML)와 JSON(JavaScript Object Notation)을 많이 사용한다. 이 기능들은 웹사이트들을 좀 더 동적이며 사용자측(Client-side)의 JavaScript 코드를 줄이게 되어 초기 로딩시간을 단축시킨다. 현재 세계의 웹사이트들 중 약 절반 정도가 가장 우수한

JavaScript 라이브러린 jQuery를 사용하고 있다 [3]. jQuery는 사용이 간단하고 기타 라이브러리들에 비하여 코딩이 짧아 웹 개발자들의 구현 속도가 빠르다[4]. 그러나 jQuery는 간단하고 쉬운 반면 배경색을 부드럽게 바꾸거나 연결된 효과를 표현하는 것과 같은 고급 효과를 나타내기에는 부족하다. jQuery UI와 같이 UI에 특화된 jQuery 플러그인이 있지만 상당히 무겁고, 위젯 위주의 기능들을 제공하고 있어 각 엘리먼트 및 위젯의 모양새를 수정할 필요가 있다.

웹 기술이 크로스플랫폼을 지원함으로써 디바이스 및 브라우저와 사용자의 간격을 줄어 들었다. 이러한 웹의 개방과 공유에 대한 정신이 장애를 가진 사용자들의 웹 응용 접근성으로 연결되고 있다[6]. 이전까지 개발된 많은 웹 응용들을 모두 웹 접근성을 지원하려 한다면 큰 일이 되지만, 새로이 개발되거나 보수되는 웹 응용들이 이러한

웹 접근성을 고려하여 개발이 된다면 모두가 차별 없이 웹 서핑을 할 수 있을 것이라 보여진다.

따라서 본 논문에서는 이러한 웹 접근성을 고려하여 유저 인터페이스 효과를 jQuery 플러그인으로 구현하는 방법을 기술한다.

## II. jQuery 기본

jQuery의 철학은 “적게 쓰고, 더 많이 하자”이다[4]. 이러한 철학은 크게 세 가지 개념으로 나뉜다. 첫째, 엘리먼트들을 선택하고 jQuery 메소드를 실행한다. 둘째, 선택된 하나의 엘리먼트 집합에 대해서 여러 개의 jQuery 메소드들을 연결시킨다. 마지막은 jQuery 랩퍼의 사용과 암묵적인 반복이다[5].

```
$("#div").hide();
```

위 코드는 jQuery의 기본 예제이다. \$("#div")가 문서 내의 모든 div 엘리먼트를 선택하여 집합으로 만든다. 그리고 집합 안의 모든 엘리먼트에게 hide()라는 메소드를 실행시킨다. 위의 짧은 코드만으로 jQuery의 첫째와 셋째 개념을 이해할 수 있다. jQuery의 메소드 연결은 아래와 같이 실행한다.

```
$("#div").hide().slideDown();
```

jQuery는 show, hide, fadeIn, fadeOut, animate, delay, stop, slideUp, slideDown과 같은 효과 메소드들이 있다. 이 메소드들은 암묵적으로 각 엘리먼트에 순차적으로 쌓이고 이전 효과가 끝나면 실행되도록 설계되어 있다. 또한 효과가 진행 중인 엘리먼트는 ‘animated’라는 선택자를 통하여 선택할 수 있다. 그러나 이러한 방식의 효과처리는 외부의 메소드나 다른 작업으로 인하여 인터럽트가 발생할 수 있다. 또한 효과가 진행 중인 엘리먼트를 스크린리더가 제대로 확인할 수 없는 문제가 있다. 이러한 문제를 해결하기 위해 각 효과에 대해 우선순위를 매기고 이를 이용하여 웹 접근성도 제공할 수 있게 한다.

## III. 웹 접근성의 고려

앞장에서는 jQuery의 기본과 효과와 관련된 문제를 설명하였다. 이를 해결하기 위하여 우리는 효과에 대하여 우선순위를 매기고 이를 표현하여 웹 접근성을 제공하려고 한다.

웹 접근성을 제공하기 위해서는 WAI-ARIA (Web Accessibility Initiative - Accessible Rich Internet Application)을 알아야 한다. WAI-ARIA는 장애를 가진 사람들이 웹 콘텐츠와 웹 응용에 좀 더 접근이 용이한 방법을 정의한다. 이는 웹

작성자들이 작성한 위젯들의 역할과 속성들을 설명하도록 한다[6].

WAI-ARIA에 의하면, ‘aria-’ 접두어를 붙인 속성을 사용한다. 예를 들어 메뉴의 버튼이 비활성된 상태라면 아래의 코드와 같이 ‘aria-disabled’ 속성을 ‘true’로 설정한다.

```
<li role="menuitem" aria-disabled="true">
menu1</li>
```

우리는 이러한 웹 접근성을 고려한 HTML 작성을 위해 ‘aria-animate’ 속성의 사용을 요청한다. 여기에 적합한 값은 효과의 우선순위를 나타내는 정수형 값을 갖도록 한다. 이 속성은 WAI-ARIA 문서에는 명시되어 있지 않다. 하지만 이런 효과의 우선순위를 제시한다면 웹 접근성뿐만 아니라 Ajax 라이브 영역과 같은 외부와의 인터랙션에 도움이 될 것이다.

## IV. jQuery 확장

jQuery를 확장하여 플러그인을 만들기 위해서는 고급 자바스크립트 프로그래밍과 jQuery 구조에 대해서 알고 있어야 한다. jQuery의 핵심 부분에 들어가는 ‘jQuery.prototype’을 확장하여야 플러그인 형태로 사용할 수 있다. ‘jQuery.prototype’는 ‘jQuery.fn’으로 연결되어 있다. 아래의 표 1에서 jQuery 플러그인의 기본 패턴을 표현한다.

표 1. jQuery 플러그인 기본 패턴

```
new function(document, $) {
    var privateMethod = function() {
    };
    $.fn.myPlugin = function() {
    };
    $.myPlugin = function() {
    };
} (document, jQuery);
```

표 1에서 privateMethod 처럼 내부에서 효과의 우선순위를 저장할 변수를 만든다. 그리고 myPlugin 부분을 플러그인 이름으로 치환한다. 함수의 내부에 변수를 만들기 때문에 외부에서 우선순위를 변경하여 인터럽트가 발생할 수 없다. 또한 jQuery의 기본 효과 함수를 확장하여 우리가 만든 플러그인을 거쳐서 실행이 되어야 하기 때문에 ‘jQuery.fn.extend’를 사용하여 기존의 효과 관련 메소드를 새로이 작성한다. 마지막으로 추가 확장으로 만든 플러그인들의 마지막에는 ‘return this’를 붙임으로서 jQuery 메소드 연결을

지원하도록 한다.

## V. 결 론

본 논문에서는 웹 접근성을 고려한 jQuery 유저 인터페이스 효과 플러그인을 구현하는 방법은 제안하였다. 이는 jQuery를 사용하여 고급 효과를 제공하려는 웹 개발자들에게 도움이 될 것이다. 또, 장애를 가진 사람의 웹 접근성에 더 나은 모델이 될 수 있으며 다른 프레임워크들의 접근성도 향상되어 연결된 효과의 인터럽트를 방지할 수 있다. 앞으로 효과의 우선순위를 매기는 수의 범위를 제한하는 것도 고려해야 할 부분이며, 스크린리더에서의 표현방법 및 기타 프레임워크에서의 지원방법도 지켜봐야할 가치있는 분야로 보고 있다.

## 참고문헌

- [1] Mark Pilgrim, "HTML5 : Upand Running - Preface", O'Reilly, 2010, pp. ix
- [2] David Flanagan, "JavaScript: The Definitive Guide: Activate Your Web Pages", O'Reilly, 2011, pp.1-3
- [3] <http://trends.builtwith.com/JavaScript/>
- [4] <http://www.jquery.com/>
- [5] jQuery Community Experts, "jQuery Cookbook", O'Reilly, 2010, pp.1-51
- [6] <http://www.w3.org/WAI/intro/aria>
- [7] <http://www.splashnology.com/article/the-structure-of-jquery-dive-into-source-code/>