

웹 표준에 맞는 HTML5 프로그래밍 강의 노트

제 24회차
jQuery 플러그인 개발

■ 학습목표

- jQuery 플러그인을 설명할 수 있다.
- 플러그인을 직접 개발 할 수 있다.

■ 학습내용

- 기본 플러그인 작성하기
- 예제 플러그인 개발

1. 기본 플러그인 작성하기

1) 플러그인

- 플러그인의 정의

- jQuery 기능을 확장하는 방법으로 제공되는 메커니즘
- jQuery의 다양한 기능을 쉽게 사용할 수 있도록 제공해주는 라이브러리

- 플러그인 사용방법 : jQuery를 사용하기 위한 코드 → 플러그인을 사용하기 위한 코드 → 플러그인 생성 및 확장

```
<script src=../js/jquery.js" type="text/javascript"></script>
<script src="media/js/jquery.dataTables.js" type="text/javascript"></script>
<script type="text/javascript">
$(function(){
    $('#example').dataTable();
});
```

2) 플러그인 개발

- 플러그인 유형 : 유틸리티 함수, 확장 집합 메소드

- 유틸리티 함수

- \$에 정의되는 유틸리티 함수 형태는 일종의 정적(전역) 메소드

```
$.MethodName()
```

- 확장 집합 메소드

- jQuery 확장 집합에서 동작하는 메소드로 객체 인스턴스의 메소드

```
$("#id").MethodName()
```

- 플러그인 파일명 규칙

- 고유한 명칭을 중간이름으로 사용함
ex) JQuery.dataTable-1.0.js

1. 기본 플러그인 작성하기

2) 플러그인 개발

- jQuery에서의 별칭 사용

- 코드 작성시 편리함
- \$로 작성된 코드가 읽기 편함

- > 플러그인에서의 별칭 사용

```
jquery.say = function(what) { alert('I say ' + what); }
```

2. 예제 플러그인 개발

1) 플러그인 제작

- jQuery 네임스페이스의 함수

- 일반 전역함수와의 충돌 가능성 없음
- jQuery 메소드와 이름이 충돌하지 않도록 주의함

- jQuery 라이브러리에서 제공하는 전역함수

`$.each()`, `$.map()`, `$.grep()`

- 배열을 받아 합계를 리턴하는 함수와 평균을 리턴하는 함수 만들기

```
(function ($) {
    $.sum = function(array) {
        var total = 0;
        $.each(array, function(index, value) {
            value = $.trim(value);
            value = parseFloat(value) || 0;

            total += value;
        });
        return total;
    };
})(jQuery);

(function ($) {
    $.average = function(array) {
        if($.isArray(array)) {
            return $.sum(array) / array.length;
        }
        return "";
    };
})(jQuery);
```

2. 예제 플러그인 개발

1) 플러그인 제작

- 전역 함수의 캡슐화

- jQuery(제이쿼리) 네임 스페이스 내에서 같은 이름을 갖는 함수가 존재할 경우 충돌 가능성이 있음

```
(function ($) {
    $.mathUtils = {
        sum: function(array) {
            var total = 0;
            $.each(array, function(index, value) {
                value = $.trim(value);
                value = parseFloat(value) || 0;

                total += value;
            });
            return total;
        },
        average: function(array) {
            if ($.isArray(array)) {
                return $.sum(array) / array.length;
            }
            return "";
        }
    };
})(jQuery);
```

- 객체 메소드

```
(function($) {
    $.fn.swapClass = function(class1, class2) {
        if (this.hasClass(class1)) {
            this.removeClass(class1).addClass(class2);
        }
        else if (this.hasClass(class2)) {
            this.removeClass(class2).addClass(class1);
        }
    };
})(jQuery);
```

- 묵시적 반복

- 플러그인 설계 시 jQuery 선택자가 복수인 경우를 감안하여 .each() 메소드를 이용함

2. 예제 플러그인 개발

1) 플러그인 제작

- 메소드 체인

- 간결한 코드
- 기존 jQuery 메소드들과 일관성 유지
- 개발자들이 쉽게 이해하고 사용할 수 있음

```
(function($) {
    $.fn.swapClass = function(class1, class2) {
        return this.each(function() {
            var $element = $(this);
            if ($element.hasClass(class1)) {
                $element.removeClass(class1).addClass(class2);
            }
            else if ($element.hasClass(class2)) {
                $element.removeClass(class2).addClass(class1);
            }
        });
    };
})(jQuery);
```

2) jQuery UI 플러그인

- jQuery UI 플러그인의 특징

- 서로 잘 동작하고 사용하기 쉬움
- 접근이 용이하고 확장이 가능함
- 테마의 적용이 가능함
- 풍부한 웹 인터페이스와 리치 인터넷 응용 프로그램 제작에 용이함

- jQuery UI의 기능

- 상호작용 : Draggable, Droppable, Resizable, Selectable, Sortable
- 위젯 : Accordion, Button, Dialog, Progressbar, Slider, Tabs
- 유틸리티 : Position
- 효과 : Blind, bounce, clip, fold, highlight, shake, Color 애니메이션 등

2. 예제 플러그인 개발

2) jQuery UI 플러그인

- Drag & Drop

- Draggable : 선택한 모든 것을 마우스로 드래그
- Droppable : 드래그 가능한 아이템의 목적지에 대한 엘리먼트에 적용

```
<body>
<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>



</body>
```

- Drag & Drop 실습 예제

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Drag and Drop</title>
<link rel="stylesheet"
href="//code.jquery.com/ui/1.11.1/themes/smoothness/jquery-ui.css">
<script src="//code.jquery.com/jquery-1.10.2.js"></script>
<script src="//code.jquery.com/ui/1.11.1/jquery-ui.js"></script>
<link rel="stylesheet" href="/resources/demos/style.css">
<style>
#draggable { width: 100px; height: 100px; padding: 0.5em; float: left;
margin: 10px 10px 10px 0; }
#droppable { width: 150px; height: 150px; padding: 0.5em; float: left;
margin: 10px; }
</style>
<script>
$(function() {
    $( "#draggable" ).draggable();
    $( "#droppable" ).droppable({
        drop: function( event, ui ) {
            $( this )
                .addClass( "ui-state-highlight" )
                .find( "p" )
                .html( "Dropped!" );
        }
    });
});
</script>
</head>
```


2. 예제 플러그인 개발

2) jQuery UI 플러그인

- DatePicker

- 날짜를 선택해야 할 때 사용
ex) 회원 가입, 비행기 예매
- 적용방법 : Input 태그에 DatePicker()를 호출

```
<script>
    $(function() {
        $( "#datepicker" ).datepicker();
    });
</script>
<input type="text" id="datepicker" >
```

- DatePicker 실습 예제

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>DatePicker</title>
<link rel="stylesheet"
href="//code.jquery.com/ui/1.11.1/themes/smoothness/jquery-ui.css">
<script src="//code.jquery.com/jquery-1.10.2.js"></script>
<script src="//code.jquery.com/ui/1.11.1/jquery-ui.js"></script>
<link rel="stylesheet" href="/resources/demos/style.css">
<script>
                $(function() {
                    $( "#datepicker" ).datepicker();
                });
</script>
</head>
<body>

                <p>Date: <input type="text" id="datepicker"></p>

</body>
</html>
```

■ 정리하기

1. 기본 플러그인 작성하기

- 플러그인의 개념 : **jQuery 기능을 확장하는 방법으로 제공되는 메커니즘**
- jQuery 기능을 확장하는 플러그인의 유형
 - **\$에 정의되는 유틸리티 함수 형태**
 - **jQuery 확장 집합에서 동작하는 메소드 형태**
- 플러그인을 제작하기 위해서 몇 가지 준수해야 하는 규칙이 있음
- jQuery를 사용하는 개발자라면 대부분 '**\$**' 별칭을 사용함

2. 예제 플러그인 개발

- jQuery 전역함수 : 실제로 **jQuery 객체의 메소드**지만, 현실적으로 보면 **jQuery 네임스페이스의 함수**임
- 대부분의 jQuery 내장 기능은 **객체 메소드로 제공됨**
- jQuery UI에 있는 플러그인들은 서로 잘 동작하고 사용하기 쉬우며, 접근이 용이하고 확장이 가능하며 테마의 적용도 가능함
- jQuery UI의 기능 : **상호작용, 위젯, 효과, 유틸리티**