

# jQuery 플러그인 작성



## ■ jQuery 기반 확장 라이브러리

- 이미 배포된 jQuery 플러그인을 활용해서 애플리케이션을 개발하는 것은 생산성을 높일 수 있는 좋은 방법
  - 팀내에서 사용할 플러그인을 직접 개발해 사용하거나 기존 플러그인에 새로운 기능을 추가해 사용 -> 생산성 제고

## ■ 플러그인 작성법?

- 플러그인의 형태를 파악하는데 도움이 됨.
- 배포된 플러그인을 활용하는데 도움이 됨.
- 모든 플러그인을 학습할 수는 없기 때문에...

# 플러그인의 유형(1)



## ■ 이제까지 사용해본 jQuery 메서드의 형태 2가지

- 정적 메서드 형태 : 예) \$.ajax()
- jQuery 객체 메서드 형태 : 예) \$("#a").css("color", "yellow")

## ■ 정적 메서드?

- jQuery는 함수. 함수는 객체.
- 객체에 속성을 추가하는 것이 자유로움. 속성에 함수를 할당한 것이 메서드임

```
> $.ajax
< function ( url, options ) {

    // If url is an object, simulate pre-1.5 signature
    if ( typeof url === "object" ) {
        options = url;
        url = undefined;
    }

    // Force options to be an object
    opt...
```

## 플러그인의 유형(2)



### ■ jQuery 객체 메서드

- jQuery 함수(\$함수)가 호출되고 나면 jQuery 객체가 리턴됨
- 이 객체를 통해서 제공되는 메서드

```
> $.prototype.css
< function ( name, value ) {
    return access( this, function( elem, name, value ) {
        var styles, len,
            map = {},
            i = 0;

        if ( jQuery.isArray( name ) ) {
            styles = getStyles( elem );
            l...
```

- ### ■ jQuery 플러그인은 이 두가지 메서드 형태를 추가해서 사용할 수 있는 것!

# 정적 메서드 형식의 플러그인



## ■ 주로 화면 UI와 관계없는 유틸리티 성격의 함수인 경우가 많음

- 예) \$.ajax(), \$.extend()
- 작성법 단순함.

[예제 17-02]

```
01: <script type="text/javascript" src="https://code.jquery.com/jquery-3.1.0.js"></script>
02: <script type="text/javascript">
03: jQuery.sortArray = function(arr) {
04:     for (var i=0; i < arr.length; i++) {
05:         for (var j=0; j < arr.length-1; j++) {
06:             if (arr[j] > arr[j+1]) {
07:                 var temp = arr[j];
08:                 arr[j] = arr[j+1];
09:                 arr[j+1] = temp;
10:             }
11:         }
12:     }
13: };
14: </script>
```

```
15: <script type="text/javascript">
16: $(document).ready(function() {
17:     var arr = [10, 20,11,25,36,90,1,54,22];
18:     $.sortArray(arr);
19:     console.log(arr);
20: });
21: </script>
```

# jQuery 객체 메서드 형식의 플러그인(1)



## ■ jQuery 함수의 prototype에 메서드를 추가

- prototype이라는 용어에 익숙하지 않은 사람을 위해 fn이라는 별칭을 제공함.

```
■ jQuery.fn.abc = function() { }  
■ jQuery.prototype.abc = function() { }
```

### ■ 예제 17-03

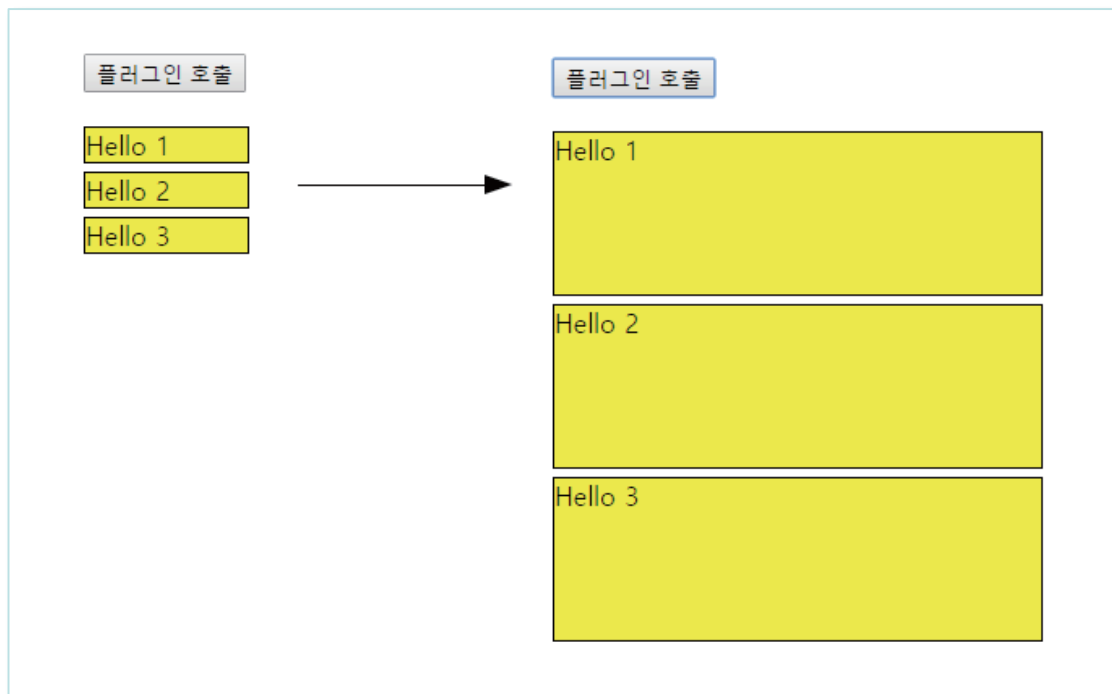
```
11: <script type="text/javascript">  
12: jQuery.fn.setSize = function(width, height) {  
13:     this.css("width", width+"px");  
14:     this.css("height", height+"px");  
15: };  
16: </script>
```

```
19:     $("#call_plugin").click(function() {  
20:         $("#div.test").setSize(300,100);  
21:     });
```

## jQuery 객체 메서드 형식의 플러그인(2)



- 이 플러그인 작성시 주의할 점은 this의 처리
  - 플러그인 메서드로 전달되는 this는 jQuery 객체이다.
  - 13~14행의 this는 바로 20행의 \$("div.test")가 리턴하는 jQuery 객체
- 예제 17-03 실행 결과



# 복잡한 기능의 플러그인 작성(1)



## ■ 단순한 플러그인은 하나의 함수만으로 구성이 가능하지만...

- 복잡한 플러그인은 하나의 함수로 개발이 힘들. 여러개의 함수를 결합해 만들게 됨.
- 여러개의 함수를 사용하는 경우 이름 충돌을 피하기 위해 즉시실행함수를 이용할 수 있음
- 전역에 util1, util2 함수가 만들어져 있다면 이름 충돌 발생

[예제 17-04: 즉시실행 함수를 이용하지 않은 경우 test.js]

```
var state = { width:100, height:200 };
var util1 = function(p1,p2) {
    //재사용 기능1
};
var util2 = function(p1) {
    //재사용 기능2
};
jQuery.fn.abc = function(width, height) {
    ...
    util1(100,200);
    ...
    util2(a);
```

```
...
util1(200,400);
...
};
```

## 복잡한 기능의 플러그인 작성(2)



- 즉시 실행 함수를 사용하면?
  - util1, util2 함수는 즉시 실행 함수가 호출되어 만들어지는 호출 객체 내부에 생성됨
  - 전역에 util1, util2 함수가 존재한다하더라도 이름 충돌은 발생하지 않음
- 대부분의 jQuery 플러그인은 이와 같이 즉시실행함수를 이용해 만듦

[예제 17-04: 즉시 실행 함수를 적용한 경우]

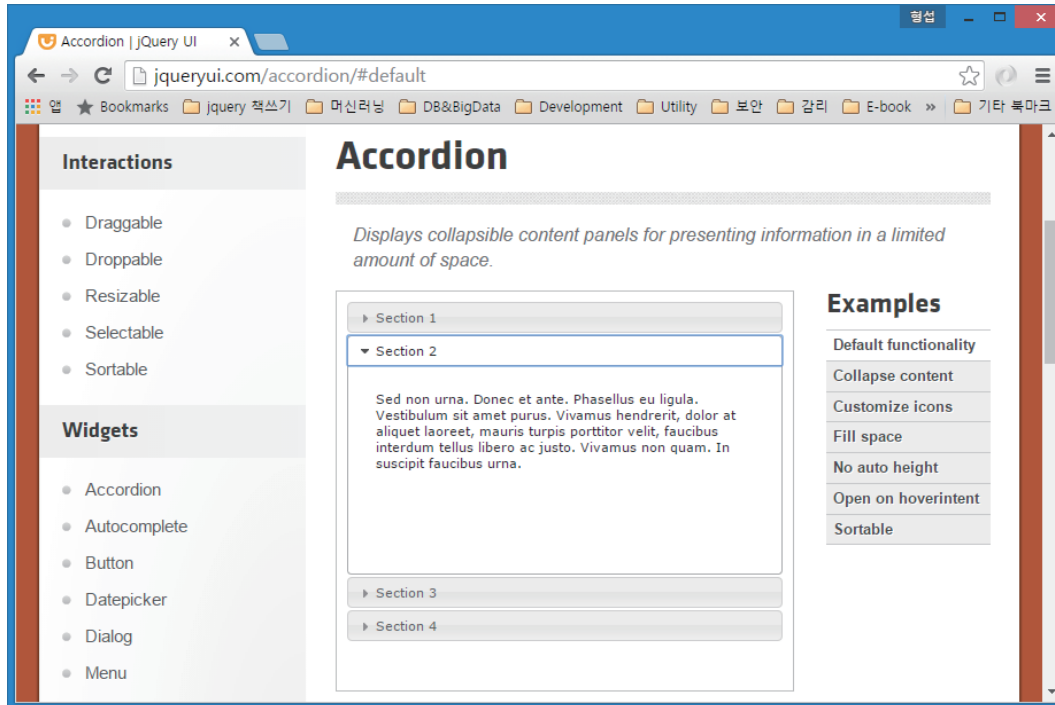
```
(function($) {  
    var state = { width:100, height:200 };  
    var util1 = function(p1,p2) {  
        //재사용 가능1  
    };  
    var util2 = function(p1) {  
        //재사용 가능2  
    };  
  
    $.fn.abc = function(width, height) {  
        ...  
        util1(100,200);  
        ...  
        util2(a);  
        ...  
        util1(200,400);  
        ...  
    };  
})(jQuery);
```



# 어코디언 플러그인 작성(1)



## jQuery UI의 어코디언 플러그인



- 비교적 간단한 플러그인이어서 플러그인의 작성형태를 이해하는데 도움이 됨.

# 어코디언 플러그인 작성(2)



## ■ 테스트 페이지 작성

- 테스트 페이지를 작성해 기능 테스트를 한 후 플러그인으로 전환할 수 있음
- 예제 17-05

```
06: <style>
07:   div.accordion { width:350px; font-size:9pt; }
08:   div.header { border:solid 1px black;
09:     text-align: center; padding:2px 2px 2px 2px;
10:     margin-bottom: 2px; background-color: purple;
11:     color:yellow; font-weight: bold; cursor:pointer; }
12:   div.content { border:solid 1px #DDDDDD; margin:0px 0px 0px 0px;
13:     padding:5px 5px 5px 5px; display:none; }
14: </style>
18:   $("#acc1").find("div.header").click(function() {
19:     if ($(this).next().is(":visible") == false) {
20:       $(this).parent().find("div.content:visible").slideUp(300);
21:       $(this).next().slideDown(300);
22:     }
23:   });
24:
25:   $("#acc2").find("div.header").click(function() {
26:     $(this).next().slideToggle(300);
27:   });
```

```
32:   <h2>어코디언 UI</h2>
33:   <div id="acc1" class="accordion">
34:     <div class="header">제목1</div>
35:     <div class="content">내용1<br/><br/><br/><br/></div>
36:     <div class="header">제목2</div>
37:     <div class="content">내용2<br/><br/><br/><br/></div>
38:     <div class="header">제목3</div>
39:     <div class="content">내용3<br/><br/><br/><br/></div>
40:     <div class="header">제목4</div>
41:     <div class="content">내용4<br/><br/><br/><br/></div>
42:   </div>
43:   <hr />
44:   <div id="acc2" class="accordion">
45:     <div class="header">title 1</div>
46:     <div class="content">content 1<br/><br/><br/><br/></div>
47:     <div class="header">title 2</div>
48:     <div class="content">content 2<br/><br/><br/><br/></div>
49:     <div class="header">title 3</div>
50:     <div class="content">content 3<br/><br/><br/><br/></div>
51:     <div class="header">title 4</div>
52:     <div class="content">content 4<br/><br/><br/><br/></div>
53:   </div>
```

## 어코디언 플러그인 작성(3)



### ■ 테스트 페이지 작성(이어서)

- 18~23행 : 한번에 하나씩만 펼쳐지는 어코디언 UI
- 25~27행 : 토글모드의 어코디언 UI.
  - 한번 클릭하면 펼치고 또 클릭하면 접힌다.
- 이렇게 테스트 페이지를 작성해 충분히 기능 테스트를 한 다음 플러그인으로 전환하는 것이 오류를 줄일 수 있는 좋은 방법이다.

# 어코디언 플러그인 작성(4)



## ■ 어코디언 플러그인 사용 형태 정의

### ■ jQuery 객체 메서드

아래와 같은 두 가지 방법을 지원한다. 옵션을 전달하지 않으면 기본값이 사용된다.

```
$(selector).myaccordion( )
```

```
$(selector).myaccordion({ width:300, duration:300, mode:"single" })
```

전달 가능한 옵션 값

width : 어코디언의 가로 폭. 기본값:300

duration : slide 효과가 일어나는 시간(ms). 기본값:300

mode : 어코디언의 모드. single, toggle의 두 가지 값을 적용할 수 있음. 기본값:single

### ■ 정적 메서드

기본 옵션값을 설정하는 메서드이다.

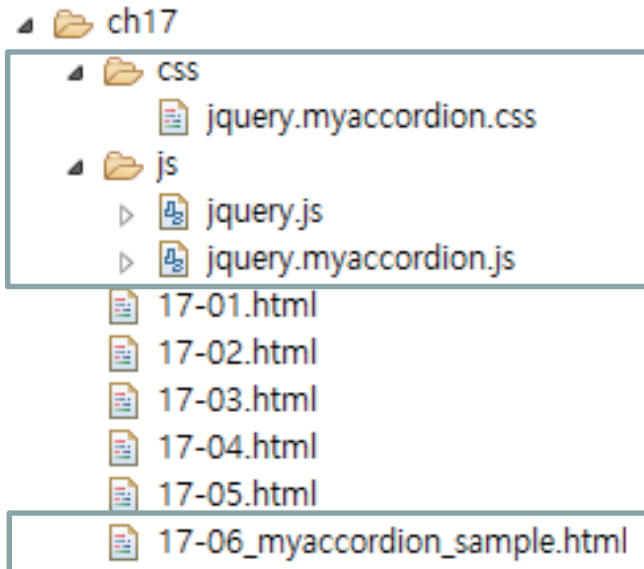
```
$.myaccordion.setDefaults({ width:400, duration:500, mode:"toggle" })
```

# 어코디언 플러그인 작성(5)



## ■ 디렉토리와 파일 생성

- 예제 디렉토리에 샘플 페이지와 js, css 파일 미리 생성
- 샘플 페이지 미리 작성 : 17-06\_myaccordion\_sample.html
- js 디렉토리에 jquery.js 파일도 미리 포함



# 어코디언 플러그인 작성(6)



## ■ 샘플 페이지의 작성

- 예제 17-06
- 전체적인 틀을 잡고 시작한다는 측면에서 개발에 오히려 도움이 됨.
  - HTML 마크업은 테스트 페이지를 그대로 옮김
  - css 파일과 js 파일 단순 참조

```
06: <link rel="stylesheet" href="css/jquery.myaccordion.css" />
07: <style>
08:
09: </style>
10: <script type="text/javascript" src="js/jquery.js"></script>
11: <script type="text/javascript" src="js/jquery.myaccordion.js"></script>
12: <script type="text/javascript">
13: $(document).ready(function() {
14:     $.myaccordion.setDefaults({ duration:200 });
15:     $("#acc1").myaccordion();
16:
17:     $("#acc2").myaccordion({ mode:"toggle", width:500, duration:1000 });
18: });
```

# 어코디언 플러그인 작성(7)



## ■ 샘플 페이지의 작성(이어서)

```
22: <h2>어코디언 샘플 페이지</h2>
23: <div id="acc1" class="accordion">
24:     <div class="header">제목1</div>
25:     <div class="content">내용1<br/><br/><br/><br/></div>
26:     <div class="header">제목2</div>
27:     <div class="content">내용2<br/><br/><br/><br/></div>
28:     <div class="header">제목3</div>
29:     <div class="content">내용3<br/><br/><br/><br/></div>
30:     <div class="header">제목4</div>
31:     <div class="content">내용4<br/><br/><br/><br/></div>
32: </div>
33: <hr />
34: <div id="acc2" class="accordion">
35:     <div class="header">title 1</div>
36:     <div class="content">content 1<br/><br/><br/><br/></div>
37:     <div class="header">title 2</div>
38:     <div class="content">content 2<br/><br/><br/><br/></div>
39:     <div class="header">title 3</div>
40:     <div class="content">content 3<br/><br/><br/><br/></div>
41:     <div class="header">title 4</div>
42:     <div class="content">content 4<br/><br/><br/><br/></div>
43: </div>
```

# 어코디언 플러그인 작성(8)



## ■ CSS 파일 작성

- 테스트 페이지에서 작성한 스타일을 그대로 옮김

[ 예제 17-07 : jquery.myaccordion.css ]

```
div.accordion { width:350px; font-size:9pt; }  
div.header { border:solid 1px black;  
    text-align: center; padding:2px 2px 2px 2px;  
    margin-bottom: 2px; background-color: purple;  
    color:yellow; font-weight: bold; cursor:pointer; }  
div.content { border:solid 1px #DDDDDD; margin:0px 0px 0px 0px;  
    padding:5px 5px 5px 5px; display:none; }
```

## ■ js 파일 작성

- 간단하지 않음. 즉시실행함수 형태를 적용해야 하기 때문에...



# 어코디언 플러그인 작성(9)



## ■ 1단계 : 즉시 실행 함수 틀

- 즉시 실행 함수 내부에 유지할 상태 정보를 저장할 자유 변수 정의
  - 기본 옵션 값을 저장하도록 함
- setDefaults 메서드 : 기본 옵션을 변경하기 위한 유틸리티 메서드
- myaccordion 메서드 : jQuery 객체의 메서드 형태로 지원되는 플러그인

[ 예제 17-08 : jquery.myaccordion.js ]

```
(function($) {  
    var defaultOptions = { width:300, duration:300, mode:"single" };  
  
    $.myaccordion.setDefaults = function(options) {  
  
    };  
  
    $.prototype.myaccordion = function(options) {  
  
    };  
})(jQuery);
```

# 어코디언 플러그인 작성(10)



## ■ 2단계 : 예외처리 & 내부 코드 구현

[ 예제 17-09 : jquery.myaccordion.js의 완성 ]

```
01: (function($) {  
02:     var defaultOptions = { width:300, duration:300, mode:"single" };  
03:  
04:     if (typeof($.myaccordion) != "undefined" && typeof($.myaccordion) != "object") {  
05:         throw new Error("이미 $.myaccordion 이름이 사용되고 있습니다.");  
06:     } else {  
07:         $.myaccordion = {};  
08:     }  
09:  
10:     if (typeof($.myaccordion.setDefaults) != "undefined") {  
11:         throw new Error("이미 $.myaccordion.setDefaults 이름이 사용되고 있습니다.");  
12:     }  
13:  
14:     if (typeof($.prototype.myaccordion) != "undefined") {  
15:         throw new Error("$(selector).myaccordion 메서드가 이미 존재합니다.");  
16:     }  
17:  
18:     $.myaccordion.setDefaults = function(options) {  
19:         $.extend(defaultOptions, options);  
20:     };  
21:
```

# 어코디언 플러그인 작성(11)



## ■ 2단계 : 예외처리 & 내부 코드 구현(이어서)

```
22:     $.prototype.myaccordion = function(options) {
23:         var temp = {};
24:         $.extend(temp, defaultOptions);
25:         $.extend(temp, options);
26:
27:         var acc = this;
28:         acc.css("width", temp.width+"px");
29:         acc.find("div.header").click(function() {
30:             if (temp.mode == "toggle") {
31:                 $(this).next().slideToggle(temp.duration);
32:             } else {
33:                 if ($(this).next().is(":visible") == false) {
34:                     acc.find("div.content:visible").slideUp(temp.duration);
35:                     $(this).next().slideDown(300);
36:                 }
37:             }
38:         });
39:     };
40: })(jQuery);
```

# 어코디언 플러그인 작성(12)



## ■ 2단계 : 예외처리 & 내부 코드 구현(이어서)

- 18행의 \$.myaccordion.setDefaults 메서드에서 기본 옵션값을 설정하기 위해 \$.extend() 메서드 사용
- \$.extend() 메서드는 두 개의 객체를 파라미터로 전달하면 두 번째 객체의 속성을 첫 번째 객체의 속성에 overwrite
- 23~25행에도 \$.extend() 메서드 사용
  - 빈 temp 객체 생성후 defaultOption 병합
  - temp 객체에 사용자가 전달한 옵션 병합
  - > 원본 옵션을 유지하기 위해서

```
> var a = { name:"홍길동", age:20 }  
⏏ undefined  
> var b = { age:30, email:"gdhong@opensg.net" }  
⏏ undefined  
> $.extend(a,b)  
⏏ Object {name: "홍길동", age: 30, email: "gdhong@opensg.net"}  
> a  
⏏ Object {name: "홍길동", age: 30, email: "gdhong@opensg.net"}
```

# 어코디언 플러그인 작성(13)

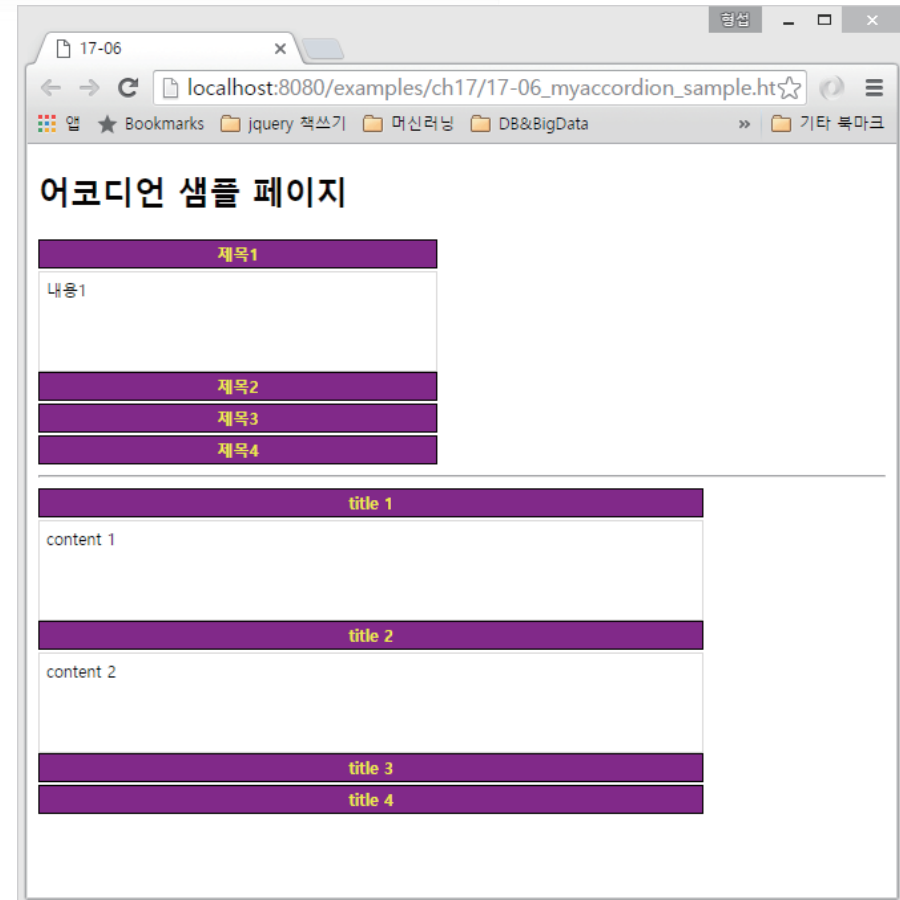


## ■ 샘플 페이지 실행 결과

- js, css 폴더와 샘플 페이지를 묶어서 압축하여 배포 가능

## ■ 훌륭한 플러그인을 만드는 것도 중요하지만...

- 다른 개발자가 만든 플러그인을 적절히 활용하여 애플리케이션을 개발하는 것도 좋은 방법



# jQuery UI 플러그인(1)



## ■ jQuery UI란?

- jQuery UI는 jQuery 기반에서 작동하는 라이브러리로서 UI 상호작용(interaction), 위젯(widget), 효과(effect), 테마(theme), 유틸리티(utility) 등이 조합된 플러그인의 집합체
- 이미 16장에서 다이얼로그 플러그인 사용해본 적 있음

## ■ jQuery UI만으로도 광범위한 내용

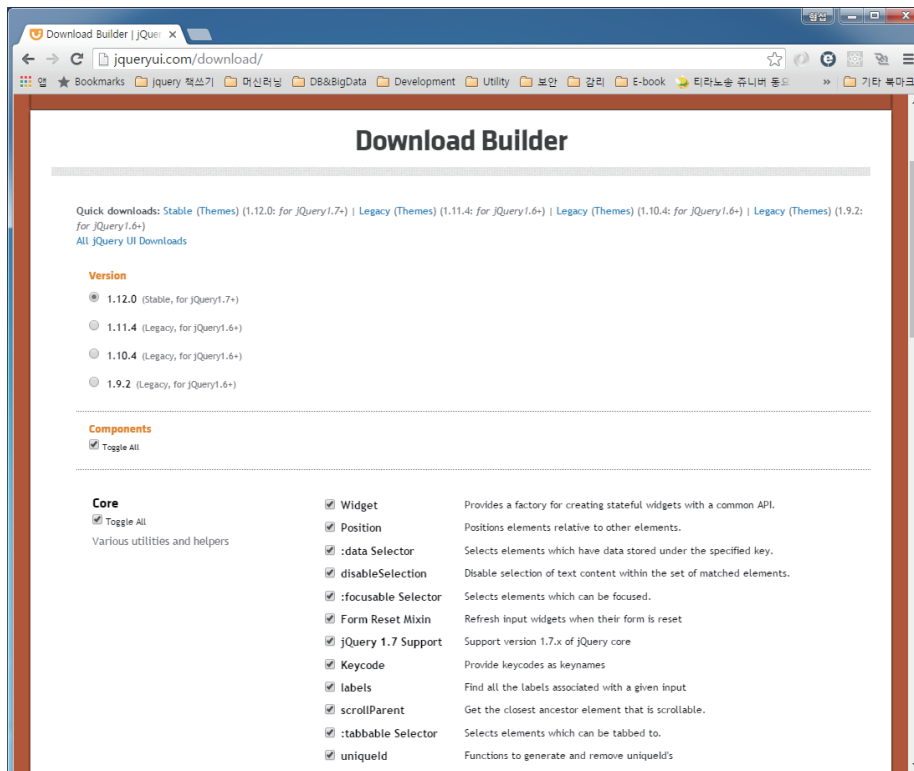
- 모두 다루어보기는 힘들.
- jQuery UI의 API 사용 방법은 일관성있게 작성되어 있어 하나의 플러그인 사용 방법을 익히면 나머지에도 적용할 수 있다.

# jQuery UI 플러그인(2)



## ■ jQuery UI 다운로드

- <http://jqueryui.com/download/> 에서 다운로드 빌더 기능 이용
  - 원하는 플러그인만 선택하여 다운로드할 수 있음

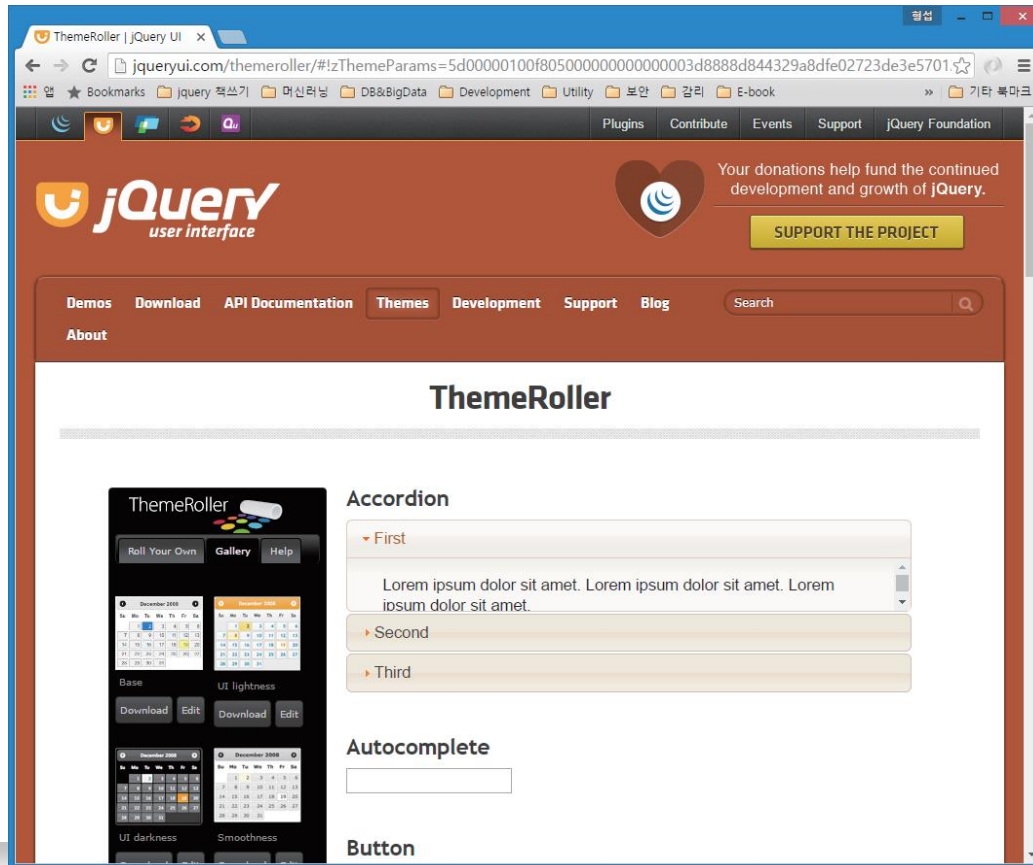


# jQuery UI 플러그인(3)



## ■ jQuery UI 테마

- 테마 선택 페이지에서 설정하여 다운로드할 수 있음
- 익숙하지 않다면 갤러리에서 선택

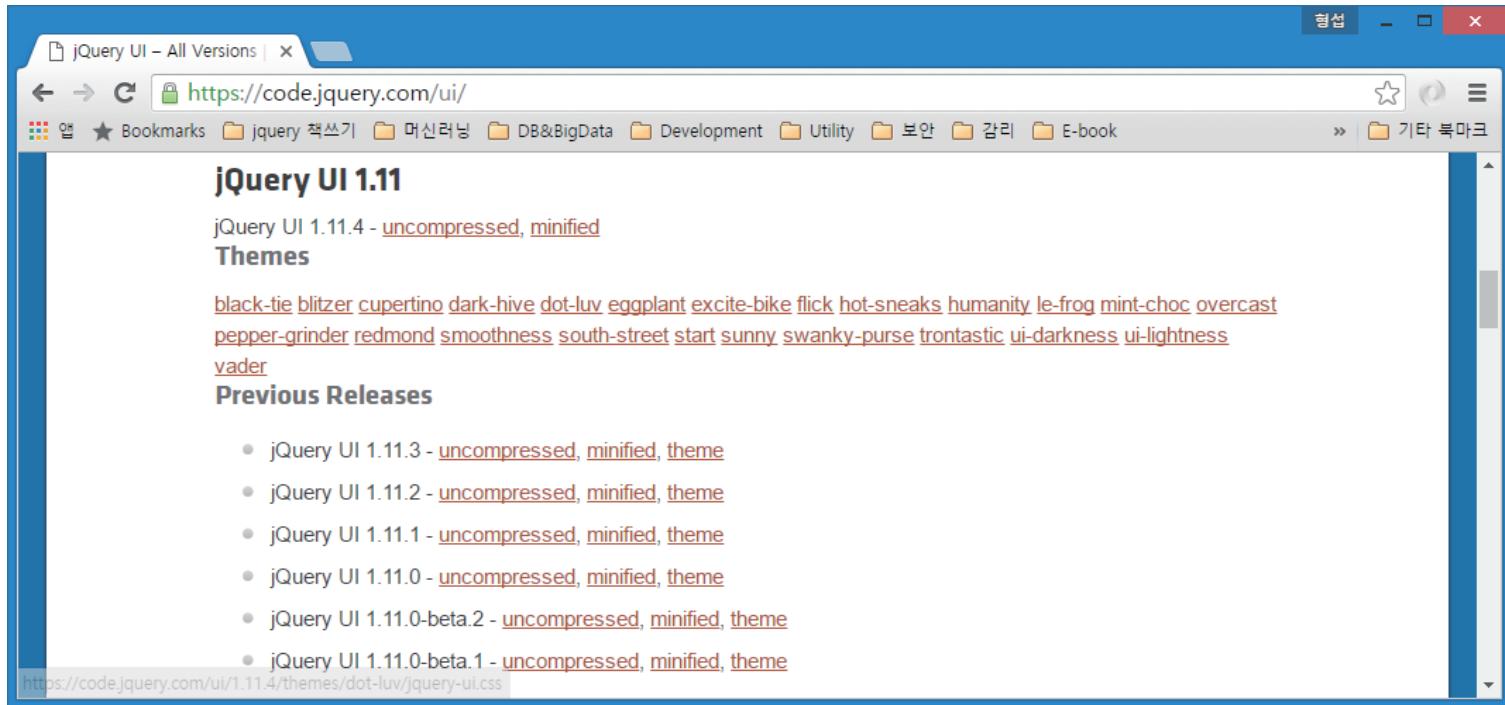




# jQuery UI 플러그인(4)



- 직접 내려받을 수도 있지만 CDN으로 원격서버의 주소를 참조할 수 있음.



# jQuery UI 플러그인(5)



## ■ jQuery UI 용어

- 자바스크립트의 용어와 조금 다르다.
- API 문서도 이 용어에 기준을 두고 기술하고 있음

옵션 Option

플러그인 Plugin

이벤트 Event

메서드 Method

```
$("#dialogtest").dialog({  
  modal : true  
  autoOpen : true  
  resize : function(e, ui){  
    console.log(ui);  
  }  
});  
  
$("#dialogtest").dialog("open");
```

The screenshot shows the jQuery UI Dialog Widget API documentation. The page title is "Dialog Widget | jQuery UI". The description is "Open content in an interactive overlay." The page includes a "QuickNav" section with tabs for "Options", "Methods", and "Events". The "Options" tab is selected, showing a list of options: appendTo, autoOpen, buttons, closeOnEscape, closeText, dialogClass, draggable, height, hide, maxHeight, maxWidth, minHeight, minWidth, modal, position, resizable, show, title, and width. The "Methods" tab shows a list of methods: close, destroy, instance, isOpen, moveToTop, open, option, and widget. The "Events" tab shows a list of events: beforeClose, close, create, drag, dragStart, dragStop, focus, open, resize, resizeStart, and resizeStop. The "Extension Points" section shows a list of extension points: \_allowInteraction.

# jQuery UI 플러그인(6)



## jQuery UI를 위한 기본 템플릿

### ■ 예제 17-10

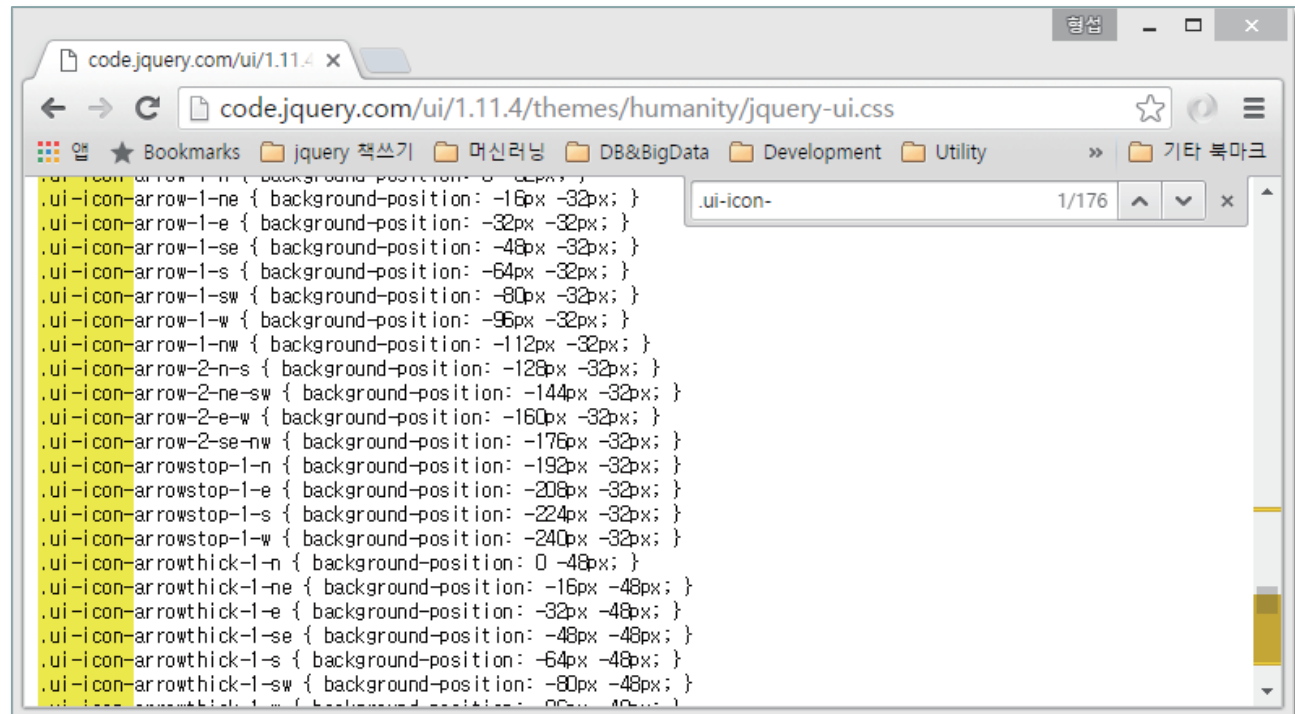
[예제 17-10]

```
01: <!DOCTYPE html>
02: <html lang="">
03: <head>
04: <meta charset="utf-8">
05: <title>17-07</title>
06: <link rel="stylesheet"
07:   href="http://code.jquery.com/ui/1.12.0/themes/humanity/jquery-ui.css" />
08: <style>
09:   body { font-size: 70%; }
10: </style>
11: <script type="text/javascript" src="http://code.jquery.com/jquery-3.1.0.js"></script>
12: <script type="text/javascript" src="http://code.jquery.com/ui/1.12.0/jquery-ui.js"></script>
13: <script type="text/javascript">
14: $(document).ready(function() {
15:
16: });
17: </script>
18: </head>
19: <body>
20:
21: </body>
22: </html>
```

# jQuery UI 플러그인(7)



- css 파일을 참조하면 이미지 아이콘도 사용할 수 있음
  - jQuery UI를 다운로드하여 압축을 풀고 images 디렉토리를 확인



# jQuery UI 플러그인 - 위젯(1)



## ■ 위젯?

- 위젯은 데스크톱(Desktop) 화면과 유사한 웹 애플리케이션을 만들 때 사용하는 플러그인 유형

## ■ Button 플러그인

- 가장 간단한 플러그인

```
$(selector).button();
```

[ 예제 17-11 : 버튼 플러그인 적용 ]

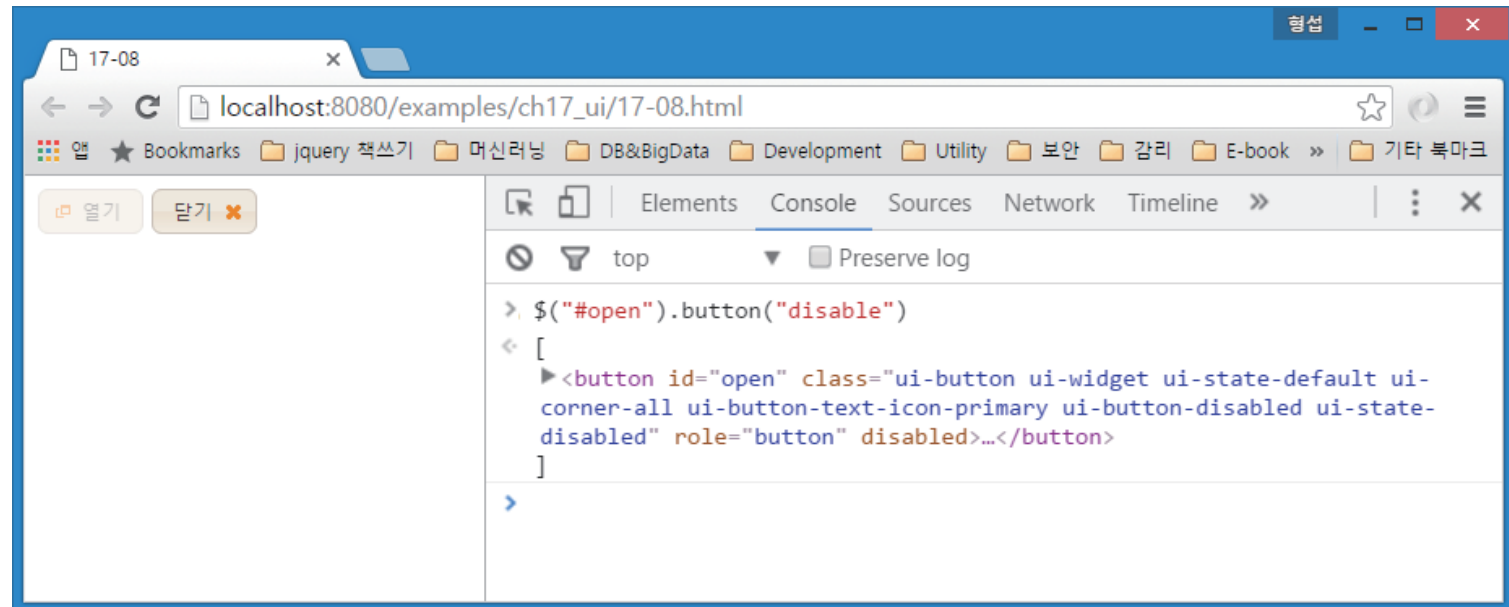
```
01: .....
02: <script type="text/javascript">
03: $(document).ready(function() {
04:     $("#open").button({
05:         icons : { primary : "ui-icon-newwin" }
06:     });
07:
08:     $("#close").button({
09:         icons : { secondary : "ui-icon-closethick" }
10:     });
11: });
12: </script>
13: </head>
14: <body>
15:     <div id="container">
16:         <button id="open">열기</button>
17:         <button id="close">닫기</button>
18:     </div>
19: </body>
20: .....
```

# jQuery UI 플러그인 - 위젯(2)



## ■ Button 플러그인(이어서)

- 옵션
  - icons : 버튼의 아이콘을 지정함.
- 메서드 테스트
  - disable, enable



# jQuery UI 플러그인 - 위젯(3)



## ■ Dialog 플러그인

- 더 다양한 옵션, 메서드, 이벤트
- 예제 17-12

```
10: $("#dialog-confirm").dialog({
11:     autoOpen: false,
12:     modal: true,
13:     resizable: false,
14:     position: { my: "center", at: "center", of: "#container" },
15:     closeText: "창닫기",
16:     show: {
17:         effect: "slide",
18:         duration: 200
19:     },
20:     hide: {
21:         effect: "explode",
22:         duration: 200
23:     },
24:     buttons: {
25:         "전송" : function() {
26:             $(this).dialog("close");
27:             alert("전송 완료");
28:         },
29:         "취소" : function() {
30:             $(this).dialog("close");
31:         }
32:     },
```

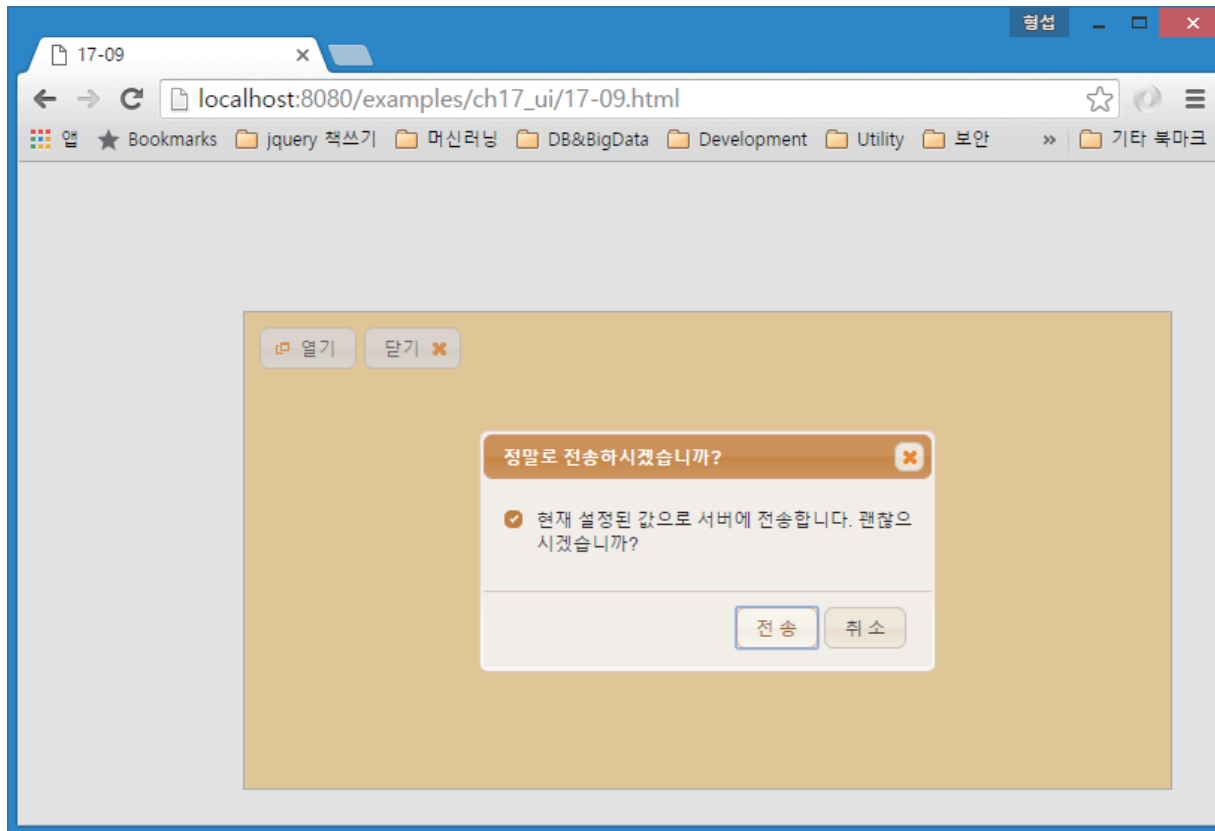
```
33:     dragStart : function(e, ui) {
34:         console.log(ui);
35:     },
36:     dragStop : function(e, ui) {
37:         console.log(ui);
38:     }
39: });
40:
41: $("#open").button({
42:     icons : { primary : "ui-icon-newwin" }
43: }).click(function() {
44:     $("#dialog-confirm").dialog("open");
45: });
```

```
56: <div id="dialog-confirm" title="정말로 전송하시겠습니까?">
57:     <p>
58:         <span class="ui-icon ui-icon-circle-check"
59:             style="float:left; margin:0 7px 20px 0;"></span>
60:         현재 설정된 값으로 서버에 전송합니다. 괜찮으시겠습니까?
61:     </p>
62: </div>
```

# jQuery UI 플러그인 - 위젯(4)



## ■ 예제 17-12 실행 결과

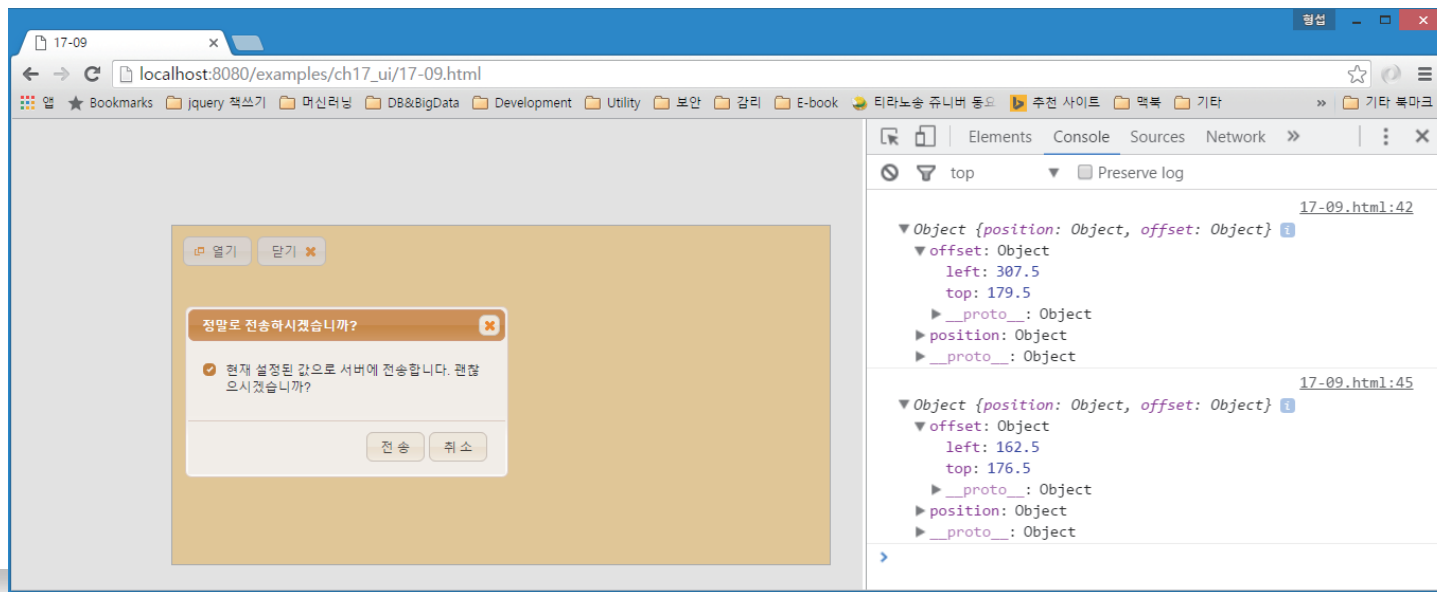




# jQuery UI 플러그인 - 위젯(5)



- autoOpen 옵션: false로 지정하여 자동으로 다이얼로그 창이 나타나는 것 방지
- modal 옵션 : 다이얼로그 창을 닫기전까지 메인 화면의 기능 이용(X)
- position 옵션: 다이얼로그 창의 위치를 지정하는 방법을 제공
- show, hide 옵션 : 화면에 보여지거나 사라질 때의 효과 지정
- dragStart, dragStop 이벤트



# jQuery UI 플러그인 - 위젯(6)



- 다이얼로그 위젯의 API 문서

**Dialog Widget**version added: 1.0

**Description:** *Open content in an interactive overlay.*

**QuickNav**

**Options**

[appendTo](#)  
[autoOpen](#)  
[buttons](#)  
[closeOnEscape](#)  
[closeText](#)  
[dialogClass](#)  
[draggable](#)  
[height](#)  
[hide](#)  
[maxHeight](#)  
[maxWidth](#)  
[minHeight](#)  
[minWidth](#)  
[modal](#)  
[position](#)  
[resizable](#)  
[show](#)  
[title](#)  
[width](#)

**Methods**

[close](#)  
[destroy](#)  
[instance](#)  
[isOpen](#)  
[moveToTop](#)  
[open](#)  
[option](#)  
[widget](#)

**Extension Points**

[\\_allowInteraction](#)

**Events**

[beforeClose](#)  
[close](#)  
[create](#)  
[drag](#)  
[dragStart](#)  
[dragStop](#)  
[focus](#)  
[open](#)  
[resize](#)  
[resizeStart](#)  
[resizeStop](#)

[Examples](#)

# jQuery UI 플러그인 - 상호작용(1)



## ■ 상호작용?

- 상호 작용 플러그인은 정적인 요소(Element)에 동적인 생동감을 불어넣어 줄 수 있는 훌륭한 플러그인
- 예제 17-03

```
02: <style>
03: body { font-size: 70%; }
04: div#outer { width:400px; height:300px; border: solid 1px black;
05:     position:absolute; left:100px; top:50px; padding:10px 10px 10px 10px;
06:     background-color:aqua; }
07: div#popup { width:100px; height:100px; background-color: #ccFF88;
08:     border:solid 1px gray; }
09: </style>
```

```
13:     $("#popup").resizable().draggable();
14:     $("#outer").draggable();
```

```
19:     <div id="outer">
20:         <h1 id="title" class="ui-widget-header">Draggable??</h1>
21:         <div id="popup"></div>
22:     </div>
```

# jQuery UI 플러그인 - 상호작용(2)

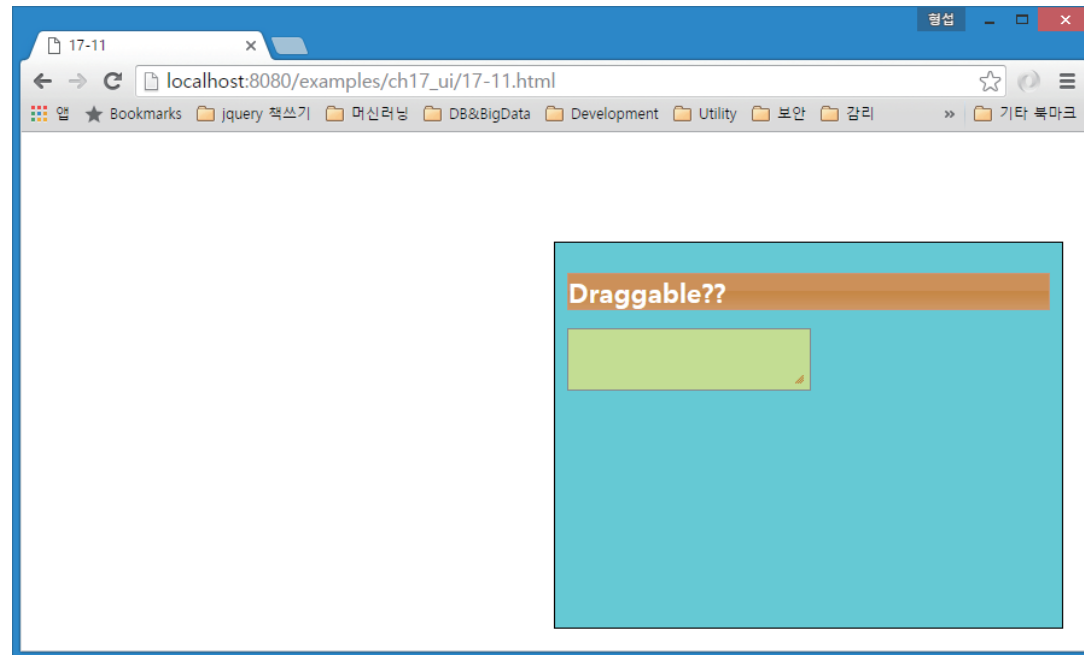


## ■ 예제 17-14

- 예제 17-13을 수정
- 4~5행 : resizable 플러그인의 크기 조정 제한값 지정
- 7행 : revert 옵션 true로 지정. 드래킹을 중단했을 때 원래 위치로 돌아오는 기능
- 11~15행 : #outer 요소의 위치를 변경하기 위해 #outer 내부의 h1#title을 드래킹함.

```
03: $("#popup").resizable({
04:     maxWidth : 200, maxHeight:200,
05:     minWidth : 50, minHeight: 50
06: }).draggable({
07:     revert : true,
08:     containment: "#outer"
09: });

11: $("#outer").draggable({
12:     handle : "h1#title",
13:     cursor : "pointer",
14:     scroll : false
15: });
```



# 정리



■ jQuery 플러그인의 종류가 대단히 많으므로 모든 것을 학습할 수는 없음

- 따라서 플러그인의 구조를 이해하여 어떤 플러그인을 만나더라도 API 문서를 읽으면서 적용할 수 있는 능력을 키우는 것이 중요함