

10-5

플러그인 직접 제작하기

앞에서는 다른 개발자가 만든 플러그인을 사용해 보았습니다. 이번에는 플러그인을 직접 제작해 보겠습니다.

플러그인 제작을 위한 \$.fn.extend() 메서드

\$.fn.extend() 메서드는 개발자가 직접 제작한 함수를 jQuery에 확장시켜 주는 역할을 합니다. 다음은 \$.fn.extend() 메서드의 기본형입니다. \$.fn.extend() 메서드는 두 가지 방법으로 사용할 수 있습니다.

기본형

```
1 $.fn.extend({
  확장 함수명: function(매개변수1, 매개변수2,...) {
    $(this); // $(this)는 플러그인을 적용한 요소를 가리킵니다.
    자바스크립트 코드;
  }
});
$("요소 선택"). 확장 함수명(인자값1, 인자값2); // 플러그인 적용

2 $.fn.확장 함수명 = function(매개변수1, 매개변수2, ...) {
  ...
}
```

간단한 플러그인 제작하기

다음은 이벤트와 경고 창을 나타내는 open 플러그인을 제작하여 버튼 태그에 적용한 예제입니다.

```

06: <script>
07: $(function() {
08:     $.fn.open = function(eventType, message) {
09:         var ts = $(this);
10:         $.each(ts, function(i, o){
11:             $(o).on(eventType, function() {
12:                 alert(message);
13:             });
14:         });
15:     }
16:     $(".btn1").open("mouseover", "welcome!");
17:     $(".btn2").open("click", "hello!");
18: });
19: </script>
20: </head>
21: <body>
22:     <p><button class="btn1">버튼1</button></p>
23:     <p><button class="btn2">버튼2</button></p>
24:     <p><button class="btn3">버튼3</button></p>
25: </body>

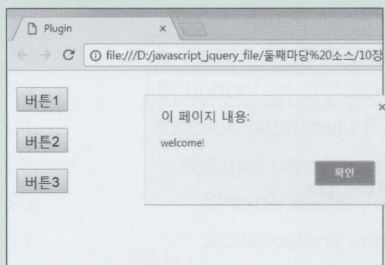
```

선택한 요소에 open 플러그인을 적용했습니다. 이벤트 타입(eventType)과 메시지(message)를 매개변수로 전달하면 해당 이벤트(eventType)가 발생할 때 지정한 메시지(message)를 경고 창으로 나타낼 수 있습니다.

플러그인 적용

[버튼1]에 마우스 포인터를 올리면 'welcome!'이라고 경고 창을 나타내고, [버튼2]를 클릭하면 'hello!'라고 경고 창을 나타냅니다.

:: 결과 화면



자주 묻는 질문 플러그인 제작하기

이번에는 자주 묻는 질문(FAQ)에 사용할 수 있는 플러그인을 만들어 보겠습니다.

1. 먼저 자주 묻는 질문을 2개 준비합니다. 그런 다음 플러그인을 제작하여 자주 묻는 질문에 적용해 보겠습니다.

코딩해 보세요!

• 실습 파일 jquery_plugin_6_test.html • 완성 파일 jquery_plugin_6.html

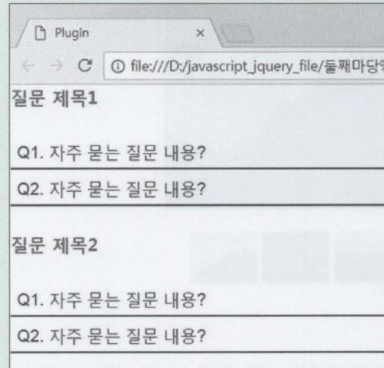
```
12: <link rel="stylesheet" href="css/faq.css">
13: <script src="js/jquery.js"></script>
14: <script>
15:     //자바스크립트 코드를 여기에 작성합니다.
16: </script>
17: </head>
18: <body>
19:     <h1>질문 제목1</h1>
20:     <div class="faq">
21:         <ul>
22:             <li>
23:                 <p><button>Q1. 자주 묻는 질문 내용?</button></p>
24:                 <div>
25:                     <div class="cont">답변 내용1</div>
26:                 </div>
27:             </li>
28:             <li>
29:                 <p><button>Q2. 자주 묻는 질문 내용?</button></p>
30:                 <div>
31:                     <div class="cont">답변 내용2</div>
32:                 </div>
33:             </li>
34:         </ul>
35:     </div>
36:     <h1>질문 제목2</h1>
37:     <div class="faq">
38:         <ul>
39:             <li>
40:                 <p><button>Q1. 자주 묻는 질문 내용?</button></p>
41:                 <div>
42:                     <div class="cont">답변 내용1</div>
43:                 </div>
44:             </li>
45:             <li>
46:                 <p><button>Q2. 자주 묻는 질문 내용?</button></p>
47:                 <div>
48:                     <div class="cont">답변 내용2</div>
```

```

49:         </div>
50:     </li>
51: </ul>
52: </div>
53: </body>

```

:: 결과 화면



2. `$.fn.extend()` 메서드로 자주 묻는 질문 플러그인을 제작합니다. 플러그인 이름은 `faq`입니다.

코딩해 보세요!

• 실습 파일 `jquery_plugin_6_test.html` • 완성 파일 `jquery_plugin_6.html`

```

14: <script>
15: $(function() {
16:     $.fn.extend({
17:         faq: function() {
18:             var ts = $(this);
19:             $.each(ts, function(i, o){
20:                 $("button", o).on("click", tabmenu);
21:                 function tabmenu() {
22:                     if ($(this).parent().next().is(":hidden")) {
23:                         $("li>div:visible", o).hide();
24:                         $(this).parent().next().show();
25:                     } else {
26:                         $("li>div:visible", o).hide();
27:                     }
28:                 }
29:             });
30:         }

```

faq 플러그인을 적용한 요소의 개수만큼 함수를 실행합니다.

faq 플러그인을 적용한 요소에 포함되어 있는 버튼에 클릭 이벤트를 등록합니다.

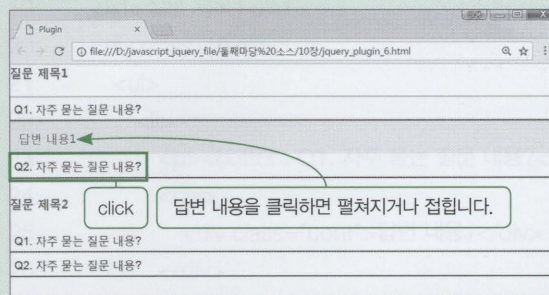

```

31: });
32:
33: $(".faq").tabmenu();
34: });
35: </script>

```

[자주 묻는 질문 내용?]을 클릭하면 답변 요소(<div>)가 열립니다. 그리고 답변 내용을 클릭하면 답변 요소가 닫힙니다.

:: 결과 화면



전문가의 조언

웹 개발에 유용한 플러그인 다섯 가지를 소개합니다

순번	종류	사이트	설명
1	① TweenMax / ② TweenLight	https://greensock.com/tweenmax https://greensock.com/tweenlite	복잡한 애니메이션을 쉽게 구현해 주는 플러그인입니다. ① 풀 패키지 ② 기본 패키지
2	lazyload	http://appelsiini.net/projects/lazyload/	페이지에 전체 이미지를 한 번에 로딩하면 로딩 속도가 느려지는데, lazyload 플러그인은 스크롤바가 이미지 영역에 내려오면 이미지를 불러옵니다.
3	SVG Convert	https://github.com/madebyshape/svg-convert	이미지를 svg 벡터 이미지로 변환하는 플러그인입니다.
4	spin.js	http://spin.js.org/	스핀 로딩 바를 생성하는 플러그인입니다.
5	youtube-background	http://florian-chapon.fr/dev/youtube-background/	유튜브 백그라운드 플러그인입니다.

| 총정리 실습! | bxSlider 이용해 섬네일 페이지 슬라이드 갤러리 구현하기

bxSlider를 이용하여 다음과 같은 슬라이드 갤러리를 만들어 보겠습니다.



슬라이드 갤러리 문단 태그를 작성한 다음 bxSlider() 플러그인을 적용합니다. 섬네일 목록()의 class 값("slider-pager")과 <a> 태그의 'data-slide-index' 속성값을 자세히 살펴보고 코드를 입력해 보세요.

코딩해 보세요!

• 실습 파일 bxslider_thumb_test.html • 완성 파일 bxslider_thumb.html

```
17: <script>
18: $(function() {
19:   $('.slide_gallery').bxSlider({
20:     auto: true,
21:     autoControls: false,
22:     pagerCustom: 'slider-pager'
23:   });
24: });
25: </script>
26: </head>
27: <body>
28:   <div id="gallery_wrap">
29:     <ul class="slide_gallery">
30:       <li></li>
31:       <li></li>
```

섬네일 pager의 class 값입니다.


```

32:     <li></li>
33:     <li></li>
34:     <li></li>
35: </ul>
36: <ul class="slider-pager">
37:     <li><a href="#" data-slide-index="0">
38:         </a>
39:     </li>
40:     <li><a href="#" data-slide-index="1">
41:         </a>
42:     </li>
43:     <li><a href="#" data-slide-index="2">
44:         </a>
45:     </li>
46:     <li><a href="#" data-slide-index="3">
47:         </a>
48:     </li>
49:     <li><a href="#" data-slide-index="4">
50:         </a>
51:     </li>
52: </ul>
53: </div>
54: </body>

```

섬네일 pager 영역입니다.