



32 jQuery 플러그인 만들기

모던 웹을 위한 Javascript jQuery 입문

❖ FancyBox 플러그인의 전역 메서드

- 별도의 선택자를 사용하지 않음
- 간단하게 jQuery 객체에 메서드를 추가하면 완료

코드 32-1 선택자를 사용하지 않는 플러그인 생성

```
<script>
    jQuery.reload = function () {

    };
</script>
```



❖ 예제 코드 32-2]

- reload 플러그인 - 현재 페이지를 새로고침하는 플러그인

코드 32-2 플러그인의 사용

```
<script>
    jQuery.reload = function () {
        // 페이지를 새로고침합니다.
        location = location;
    };

    $(document).ready(function () {
        $.reload();
    });
</script>
```



❖ 선택자를 사용하는 플러그인

- 예제 코드 32-3]
- jQuery 객체의 fn 속성에 메서드 추가

코드 32-3 jQuery 객체의 fn 속성

```
<script>  
    jQuery.fn.myPluginMethod = function (options) { };  
</script>
```



❖ 선택자를 사용하는 플러그인

- 플러그인 메서드 내부에서 `this` 키워드는 선택자로 선택한 대상 의미
- 예제 코드 32-4] 스타일을 변경하는 플러그인

코드 32-4 플러그인 내의 `this` 키워드

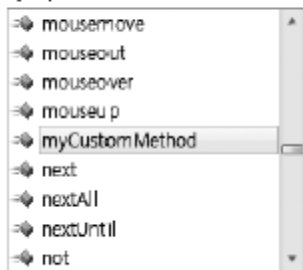
```
<script>
  jQuery.fn.myPluginMethod = function (options) {
    $(this).css('color', 'White');
    $(this).css('background', 'Black');
  };

  $(document).ready(function () {

  });
</script>
```

그림 32-1 추가된 `myCustomMethod` 메서드

`$('.h1').mycu|`



❖ 선택자를 사용하는 플러그인

- myCustomMethod() 메서드 - 객체의 스타일 변경하는 플러그인

코드 32-5 사용자 플러그인 사용

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-1.7.js"></script>
  <script>
    // 플러그인을 만듭니다.
    jQuery.fn.myCustomMethod = function (options) {
      $(this).css('color', 'White');
      $(this).css('background', 'Black');
    };

    $(document).ready(function () {
      // 플러그인을 사용합니다.
      $('h1').myCustomMethod();
    });
  </script>
</head>
<body>
  <h1>Header A</h1>
  <h1>Header B</h1>
</body>
</html>
```

그림 32-2 플러그인 적용 결과

Header A

Header B



32.3 피벗 슬라이드 플러그인 만들기 (1)

❖ 피벗 슬라이드란?

- 마우스나 터치 이벤트를 사용해 화면의 내용 넘기는 기능
 - 모바일 네이버와 모바일 다음의 메인 화면에 사용

그림 32-3 네이버와 다음의 피벗 슬라이드



32.3 피벗 슬라이드 플러그인 만들기 (1)

❖ 플러그인을 만들 때 고려해야 할 사항

- CSS 파일을 제공할 것인가?
- 플러그인을 적용할 대상 태그는 무엇인가?
- 어떠한 메서드를 제공할 것인가?
- 어떠한 옵션을 제공할 것인가?
- 별도의 문서로 만들어야 배포 시 다른 개발자가 쉽게 이용 가능



32.3 피벗 슬라이드 플러그인 만들기 (1)

❖ Pivot 플러그인 제작

■ 예제 코드 32-6]

- 외부의 div 태그에 Pivot 플러그인 적용
- 내부에 있는 div 태그를 각각 하나의 페이지로 피벗

코드 32-6 적용 대상

```
<div id="pivot_slider">  
  <div>/div>  
  <div>/div>  
  
  <div>/div>  
  <div>/div>  
</div>
```



32.3 피벗 슬라이드 플러그인 만들기 (1)

❖ Pivot 플러그인 제작

■ Pivot 플러그인의 핵심 메서드

표 32-1 Pivot 플러그인의 메서드

메서드 이름	설명
<code>pivot()</code>	피벗 플러그인을 적용합니다.

표 32-2 Pivot 플러그인의 옵션 속성

옵션 속성 이름	설명
<code>width</code>	각 페이지의 너비를 지정합니다.
<code>height</code>	각 페이지의 높이를 지정합니다.



32.3 피벗 슬라이드 플러그인 만들기 (1)

❖ 예제 HTML 페이지 코드

코드 32-7 HTML 페이지 구성

```
<!DOCTYPE html>
<html>
<head>
  <script src="http://code.jquery.com/jquery-1.7.js"></script>
  <script>
    jQuery.fn.pivot = function (options) {

      };
    </script>
  <script>
    $(document).ready(function () {
      $('#pivot_slider').pivot({
        width: 700,
        height: 450
      });
    });
  </script>
</head>
<body>
  <div id="pivot_slider">
  </div>
</body>
</html>
```



32.3 피벗 슬라이드 플러그인 만들기 (1)

❖ 예제 HTML 페이지 코드 (2)

```
</script>
</head>
<body>
  <div id="pivot_slider">
    <div>
      <h1>Lorem ipsum dolor sit amet</h1>
      
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    </div>
    <div>
      <h1>Fusce eleifend</h1>
      
      <p>Fusce eleifend, massa at elementum ultrices.</p>
    </div>
    <div>
      <h1>Donec venenatis mattis elit.</h1>
      
      <p>Donec venenatis mattis elit. Etiam suscipit.</p>
    </div>
    <div>
      <h1>Phasellus sagittis quam mi.</h1>
      
      <p>Phasellus sagittis quam mi. Ut non neque a tellus laoreet.</p>
    </div>
  </div>
</body>
</html>
```



32.4 피벗 슬라이드 플러그인 만들기 (2)

❖ 플러그인 제작

- CSS 파일을 제공하지 않으므로 스타일 내부에서 지정
- 옵션 객체 제어하는 방법 활용



32.4 피벗 슬라이드 플러그인 만들기 (2)

❖ 예제 코드 32-8] 스타일과 옵션 객체 지정 (1)

코드 32-8 변수 선언 및 스타일 지정

```
<script>
    jQuery.fn.pivot = function (options) {
        // 변수를 선언합니다.
        var $target = $(this);
        var $items = $target.children();
        var $container = $target.wrap('<div>/div').parent();
        var option = { width: 500, height: 450 };

        // 옵션을 처리합니다.
        $.extend(option, options);

        // 스타일을 지정합니다.
        $target.css({
            width: $items.length * option.width,
            height: option.height,
            position: 'absolute'
        });
        $items.css({
            float: 'left',
            width: option.width, height: option.height
        });
        $container.css({
            overflow: 'hidden', position: 'relative',
            width: option.width, height: option.height
        });
    };
</script>
```



32.4 피벗 슬라이드 플러그인 만들기 (2)

❖ 예제 코드 32-8] 스타일과 옵션 객체 지정 (2)

- wrap() 메서드와 parent() 메서드의 사용
 - Pivot 플러그인 - 드래그해 페이지 넘긴다는 것 외에는 이미지 슬라이더와 비슷
 - 이미지 슬라이더를 만들 때 캔버스 생성
 - 이미지를 담은 div 태그를 만들어 div 태그를 이동해 화면 전환 효과
 - 사용자가 쉽게 사용할 수 있도록 캔버스는 플러그인에서 자체 생성



32.4 피벗 슬라이드 플러그인 만들기 (2)

❖ 예제 코드 32-8] 스타일과 옵션 객체 지정 (3)

- wrap() 메서드를 사용해 그림 32-4처럼 문서 객체 구성
- 외부의 div 태그를 parent() 메서드를 사용해 변수 \$container에 넣음

그림 32-4 플러그인을 사용한 문서 객체 구성

```
▼ <body>
  ▼ <div>
    ▼ <div id="pivot_slider">
      ▶ <div>...</div>
      ▶ <div>...</div>
      ▶ <div>...</div>
      ▶ <div>...</div>
    </div>
  </div>
</body>
```



32.4 피벗 슬라이드 플러그인 만들기 (2)

❖ 예제 코드 32-8] 스타일과 옵션 객체 지정 (4)

■ 변수 option과 매개 변수 options

- Option은 기본 옵션 객체
- options는 사용자로부터 입력 받는 옵션 객체
- width 속성이나 height를 입력해도 되고 입력하지 않아도 되는 옵션
- 옵션 객체를 합치는 방법은 \$.extend() 메서드 사용

```
var option = { width: 500, height: 450 };
```

```
// 옵션을 처리합니다.
```

```
$.extend(option, options);
```



32.5 피벗 슬라이드 플러그인 만들기 (3)

❖ Pivot 플러그인 구현 – 예제 코드 32-9]

코드 32-9 이벤트 연결

```
<script>
  jQuery.fn.pivot = function (options) {
    // 변수를 선언합니다.
    /* 생략 */

    // 옵션을 처리합니다.
    /* 생략 */

    // 스타일을 지정합니다.
    /* 생략 */

    // 이벤트를 연결합니다.
    var originalLeft = 0;
    var oldLeft = 0;
    var nowPosition = 0;
    var isDown = false;

    $target.bind('mousedown', function () { });
    $target.bind('mousemove', function () { });
    $target.bind('mouseup', function () { });

  };
</script>
```



32.5 피벗 슬라이드 플러그인 만들기 (3)

❖ 예제 코드 32-9]

■ isDown

- 사용자가 마우스를 누르는지 확인하는 변수
- mousedown 이벤트가 발생하면 true (누르면 true)
- mouseup 이벤트가 발생하면 false

■ nowPosition

- 현재 페이지를 나타내는 변수
- 초깃값은 0
- 사용자가 페이지를 오른쪽으로 넘기면 1 추가
- 왼쪽으로 페이지를 넘기면 1 감소
- 이를 이용해 이미지 슬라이더처럼 animate() 메서드 적용



32.5 피벗 슬라이드 플러그인 만들기 (3)

❖ 예제 코드 32-9]

- 변수 originalLeft와 oldLeft
 - 이벤트 발생 순서와 함께 이해

그림 32-5 mousedown 이벤트

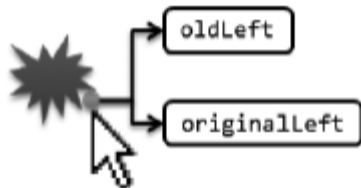


그림 32-6 mousemove 이벤트 (1)

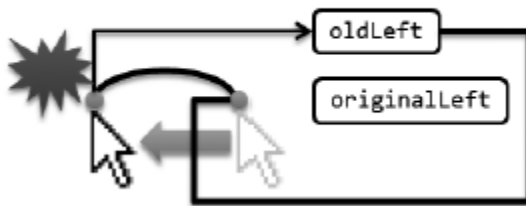


그림 32-7 mousemove 이벤트 (2)

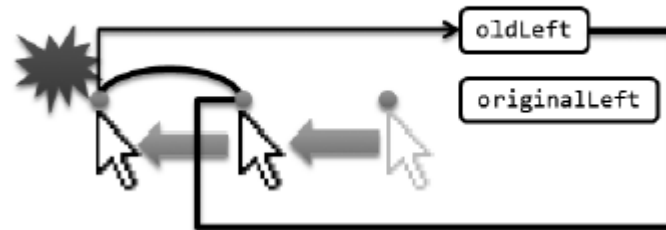
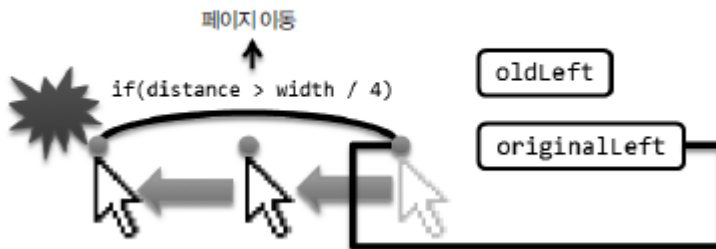


그림 32-8 mouseup 이벤트



32.5 피벗 슬라이드 플러그인 만들기 (3)

❖ 예제 코드 32-9]

■ 변수 originalLeft와 oldLeft (2)

- mousedown 이벤트가 발생하면 그림 32-5처럼 현재 위치 두 변수에 저장
- 마우스 커서를 이동할 때마다 mousemove 이벤트 발생
 - 변수 oldLeft를 이용
 - » 이동한 마우스 커서와의 거리를 구해 페이지를 조금 이동
 - » 페이지 이동 후 현재 위치를 다시 변수 oldLeft에 넣음
 - 페이지를 조금씩 드래그해 이동할 때는 변수 oldLeft만 사용
- mouseup 이벤트가 일어날 때는 변수 originalLeft만 사용
 - 변수 originalLeft는 최종적으로 mouseup 이벤트가 발생시
 - mouseup 이벤트가 발생한 위치와 시작 지점의 거리를 구하는 데 사용
 - 최종 이동 거리가 페이지 width 속성의 1/4이면 페이지 이동



32.5 피벗 슬라이드 플러그인 만들기 (3)

❖ 예제 코드 32-10]

- mousedown 이벤트가 발생할 때, 변수 oldLeft와 originalLeft 지정
- mousedown 이벤트에서 isDown 속성 true로 변경
- mouseup 이벤트에서 isDown 속성을 false

코드 32-10 변수 선언과 이벤트 구현

```
// 이벤트를 연결합니다.
var originalLeft = 0;
var oldLeft = 0;
var nowPosition = 0;
var isDown = false;

$target.bind('mousedown', function (event) {
    oldLeft = originalLeft = event.clientX;
    isDown = true;
    event.preventDefault();
});

$target.bind('mousemove', function () {
    if (isDown) {

    }
    event.preventDefault();
});

$target.bind('mouseup', function () {
    isDown = false;
    event.preventDefault();
});
```



32.5 피벗 슬라이드 플러그인 만들기 (3)

❖ 예제 코드 32-11]

- 움직인 거리를 구하고 animate() 메서드 사용해 드래그한 만큼 이동
 - 페이지 부드럽게 이동 가능한 장점

코드 32-11 mousemove 이벤트

```
$target.bind('mousemove', function (event) {  
    if (isDown) {  
        // 변수를 선언합니다.  
        var distance = oldLeft - event.clientX;  
        oldLeft = event.clientX;  
  
        // 움직입니다.  
        $target.animate({ left: '-' + distance }, 0);  
        $target.stop(true);  
    }  
    event.preventDefault();  
});
```



32.5 피벗 슬라이드 플러그인 만들기 (3)

❖ mouseup 이벤트

- 사용자가 width 옵션 속성의 1/4 이상 왼쪽이나 오른쪽으로 드래그
 - 변수 nowPosition 변화
 - **animate() 메서드 사용해 nowPosition 위치로 이동**

코드 32-12 mouseup 이벤트

```
$target.bind('mouseup', function (event) {  
    // 내부 함수를 선언합니다.  
    function movePosition(direction) {  
        // 위치를 설정합니다.  
        var changePosition = nowPosition + direction;  
        if (0 <= changePosition && changePosition < $items.length) {  
            nowPosition = changePosition  
        }  
    }  
  
    // 요소의 1/4 이상 드래그를 했을 경우 피벗합니다.  
    if (originalLeft - event.clientX > option.width / 4) {  
        movePosition(+1);  
    } else if (originalLeft - event.clientX < -option.width / 4) {  
        movePosition(-1);  
    }  
  
    // 이동합니다.  
    $target.animate({ 'left': -nowPosition * option.width }, 'fast');  
    isDown = false;  
    event.preventDefault();  
});
```



32.5 피벗 슬라이드 플러그인 만들기 (3)

❖ 전체 예제 코드

코드 32-13 전체 코드

```
<!DOCTYPE html>
<html>
<head>
  <style>
    img { width:100%; }
    body{ width:700px; margin:0px auto; }
  </style>
  <script src="http://code.jquery.com/jquery-1.7.js"></script>
  <script>
    jQuery.fn.pivot = function (options) {
      // 변수를 선언합니다.
      var $target = $(this);
      var $items = $target.children();
      var $container = $target.wrap('<div></div>').parent();
      var option = { width: 500, height: 450 };

      // 옵션을 처리합니다.
      $.extend(option, options);

      // 스타일을 지정합니다.
      $target.css({
        width: $items.length * option.width,
        height: option.height,
        position: 'absolute'
      });
      $items.css({
        float: 'left',
        width: option.width, height: option.height
      });
    };
  </script>
</head>
</html>
```



32.5 피벗 슬라이드 플러그인 만들기 (3)

❖ 전체 예제 코드 (2)

```
$container.css({
    overflow: 'hidden', position: 'relative',
    width: option.width, height: option.height
});

// 이벤트를 연결합니다.
var originalLeft = 0;
var oldLeft = 0;
var nowPosition = 0;
var isDown = false;

$target.bind('mousedown', function (event) {
    oldLeft = originalLeft = event.clientX;
    isDown = true;
    event.preventDefault();
});

$target.bind('mousemove', function (event) {
    if (isDown) {
        // 변수를 선언합니다.
        var distance = oldLeft - event.clientX;
        oldLeft = event.clientX;

        // 움직입니다.
        $target.animate({ left: '-=' + distance }, 0);
        $target.stop(true);
    }
    event.preventDefault();
});
```



32.5 피벗 슬라이드 플러그인 만들기 (3)

❖ 전체 예제 코드 (3)

```
$target.bind('mouseup', function (event) {  
    // 내부 함수를 선언합니다.  
    function movePosition(direction) {  
        // 위치를 설정합니다.  
        var changePosition = nowPosition + direction;  
        if (0 <= changePosition && changePosition < $items.length) {  
            nowPosition = changePosition  
        }  
    }  
  
    // 요소의 1/4 이상 드래그했을 경우 피벗합니다.  
    if (originalLeft - event.clientX > option.width / 4) {  
        movePosition(+1);  
    } else if (originalLeft - event.clientX < -option.width / 4) {  
        movePosition(-1);  
    }  
  
    // 이동합니다.  
    $target.animate({ 'left': -nowPosition * option.width }, 'fast');  
  
    isDown = false;  
    event.preventDefault();  
});  
};  
</script>
```



32.5 피벗 슬라이드 플러그인 만들기 (3)

❖ 전체 예제 코드 (4)

```
    }

    // 이동합니다.
    $target.animate({ 'left': -nowPosition * option.width }, 'fast');

    isDown = false;
    event.preventDefault();
  });
};
</script>
<script>
$(document).ready(function () {
  $('#pivot_slider').pivot({
    width: 700,
    height: 800
  });
});
</script>
</head>
<body>
<div id="pivot_slider">
  <div>
    <h1>Lorem ipsum dolor sit amet</h1>
    
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  </div>
```



32.5 피벗 슬라이드 플러그인 만들기 (3)

❖ 전체 예제 코드 (5)

```
<div>
  <h1>Fusce eleifend</h1>
  
  <p>Fusce eleifend, massa at elementum ultrices.</p>
</div>
<div>
  <h1>Donec venenatis mattis elit.</h1>
  
  <p>Donec venenatis mattis elit. Etiam suscipit.</p>
</div>
<div>
  <h1>Phasellus sagittis quam mi.</h1>
  
  <p>Phasellus sagittis quam mi. Ut non neque a tellus laoreet.</p>
</div>
</div>
</body>
</html>
```



32.5 피벗 슬라이드 플러그인 만들기 (3)

❖ 예제 실행 결과

그림 32-9 완성한 Pivot 플러그인





Thank You !

모던 웹을 위한 Javascript jQuery 입문