jQuery Study - 7주차

장영석

플러그인 만들기!! - 시작하기

- 플러그인 선언하기!

```
jQuery.fn.플러그인명   혹은    $.fn.플러그인명
```

- 라이브러리 겹침 방지를 위한 조치

```
;(function($){

        jQuery.fn.플러그인명 혹은 $.fn.플러그인명

        예 ) $.fn.jys = function(){
                //작업내용
        };

})(jQuery);
```

끝!!!!!!!!

플러그인 만들기!! - extend

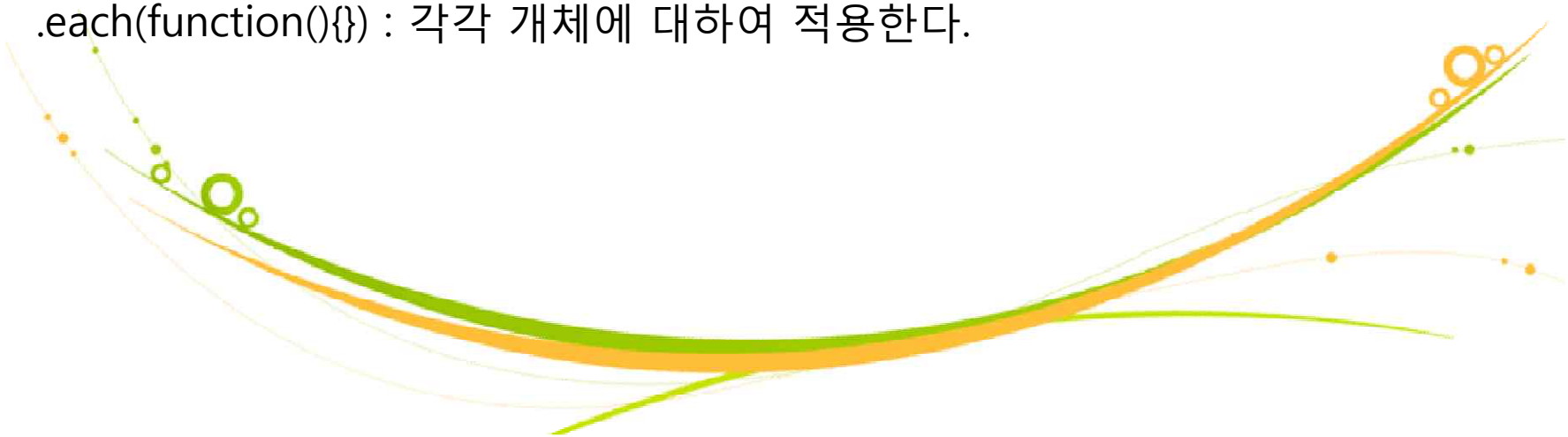
- 개체 확장하기 (.extend)
  - $.extend() : 첫번째 매개변수로 지정된 개체와 그 이후에 지정된 개체를 합친 개체의 참조를 반환, 만약 동일한 속성이 이미 존재한다면 후자의 것으로 덮어쓰게 됨

- .extend 활용하기

```
;(function($){
        $.fn.jys = function(options){
                var options = $.extend({
                        //속성명 : 속성값
                },options);
        };
})(jQuery);
```

플러그인 만들기!! - return this?!

- return this.each()?!?!
  - html에서 스크립트를 호출하게되면, 호출하는 쪽에서 'jQuery 개체집합에 플러그인을 호출 하였다.'
  - 즉, this는 호출한 jquery 개체집합을 가리키는 것이며, 개체집합에 대해 일괄 적용을 위해서 each()문을 사용합니다.
- return : 플러그인을 적용한 후 다른 메소드를 계속 이어갈 수 있도록 한다.
- this : 플러그인을 호출한 jQuery개체집합을 가리킨다.
- .each(function(){}) : 각각 개체에 대하여 적용한다.



플러그인 만들기!! - return this?!

```
<script type="text/javascript">
        $('li').jqueryText();
</script>
;(function($){
        $.fn. jqueryText = function(options){
                var opt = $.extend({
                        text : "즐겁고 신나는 jQuery 수업"
                },options);
                return this.each(function(){
                        $(this).text(opt.text);
                });
        };
})(jQuery);
```

사용자 정의 반복기

```
jQuery.slowEach = function(array,interval, callback){ //반복기
        if (!array.length)return;
            var i = 0;
            next();
            function next(){
            if(callback.call(array[i], i , array[i]) !== false)
            if(++i < array.length)
            setTimeout(next, interval);
        }
        return array;
}
jQuery.fn.slowEach = function(interval, callback){
        return jQuery.slowEach(this, interval, callback);
}
```

신나고 즐거운 실습시간~♬