

## 6강 활용 빈도가 높은 addClass() 명령어

0601.html :

```
<script>
$(function(){
    // $("img").eq(0).css({"background-color":"#a1deff"});
    $("img").eq(0).addClass("type3");

    $("img").click(function(){
        var n=Math.ceil(Math.random()*5);
        // console.log("n : "+n);
        var existClass=$(this).attr("class");
        // console.log("existClass : "+existClass);

        if(!existClass){
            // console.log("정해진 클래스가 없습니다.");
        }else{
            // $(this).removeClass(existClass);
            $(this).removeAttr("class");
        }
        $(this).addClass("type"+n);
    });
});
</script>
```

line0050 : `$("img").eq(0).css({"background-color":"#a1deff"});`

css() 메서드로 적용된 스타일은 inline 스타일로 요소에 적용되는 단점이 있습니다.

line0051: `$("img").eq(0).addClass("type3");`

이를 보다 세련되게 작성하는 방법은 CSS에서 미리 만들어 놓은 클래스를 추가하는 방식입니다.

line0055 : `console.log("n : "+n);`

1 ~ 5까지의 n 변수 값을 확인할 수 있습니다.

line0057 : `console.log("existClass : "+existClass);`

기존 클래스 값을 확인할 수 있습니다.

line0062 : `$(this).removeClass(existClass);`

기존 클래스를 해당 요소에서 제거합니다.

line0063 : `$(this).removeAttr("class");`

기존 클래스를 해당 요소에서 제거합니다.

0602.html :

```
<script>
$(function(){
    $("p:last").addClass("selected highlight");
});
</script>
```

line0041 : `$("p:last").addClass("selected highlight");`

이중으로 클래스 추가가 가능합니다.

0603.html :

```
<script>
$(function(){
    $("p").text("The p elements is based on #555.");

    $("div").addClass(
        function(i, className){
            className+="green";
            return className;
        }
    );
});
</script>
```

line0045 : `$("p").text("The p elements is based on #555.");`

빈 p 요소에 텍스트를 작성합니다.

line0049 : `className+="green";`

기존 클래스에 'green'을 추가합니다.