**배열관리**

jQuery로 배열을 관리할 때는 each() 메서드를 사용한다.

each() 메서드는 매개변수로 입력한 함수로 for in 반복문처럼 객체나 배열의 요소를 검사하는 메서드로 다음 두가지 형태로 사용한다

1. $.each(object, function(index, item) {})

2. $(selector).each(function(index, item) {})

**1) 자바스크립트 배열 관리**

$.each() 형태 예제



**forEach() 메서드와 차이점**

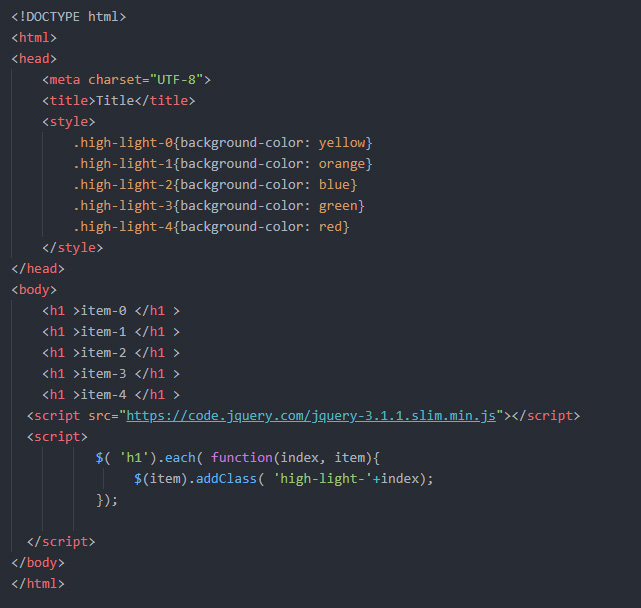
: EXMAScript5에서 추가된 forEach() 메서드와 매개변수의 순서가 다르므로 주의!

|  |  |
| --- | --- |
| forEach() 메서드 | each() 메서드 |
| [].forEach(function (item, index) {}) | $.each(function (index, item) {}) |

**2) jQuery 배열관리**

jQuery의 배열 객체는 따로 만드는 것이 아니라, 선택자로 여러 개의 문서 객체를 선택할 때 생성된다.

[예제](ex2.html)



$() 메서드로 h1 객체를 선택하면 현재 body 태그안에 h1객체가 5개이므로 5개의 문서 객체를 가져옵니다. 각각의 객체에다 다른 설정을 하고싶을 때 each() 메서드를 사용한다.

매개변수로 입력한 함수는 각각의 index와 item을 가진다.

**객체확장**

많은 수의 속성을 추가할 때 매우 귀찮고 코드가 지저분해진다. 이런 문제를 해결하고자 만들어진 메서드가 $.extend() 메서드다.

[예제](ex3.html)



$.extend() 메서드의 첫 번째 매개변수로 입력한 객체에 두 번째 매개변수 이후의 객체를 합친다.

**jQuery 충돌방지**

jQuery 이외에도 여러 프레임워크가 있다. 여러 프레임워크를 사용할 경우 충돌이 발생할 수 있다. 이를 방지할 때 사용하는 메스트는 $.noConflict() 메서드다.

