

`$(document).ready()` : 문서가 준비되면 매개변수로 넣은 콜백함수를 실행하라.

= `$(function () {});`

`window.onload = function () {};` 는 한번에 하나의 이벤트만 연결할 수 있지만,

`jQuery`는 여러개 연결 가능.

`$`은 `window.jQuery = window.$ = jQuery`로 바꿔 쓸 수 있다.

기본선택자 : 문서객체를 다룰 때 사용.

`$('h1').css('color', 'red');`

선택자

메서드

↳ 전체선택자 : HTML 페이지에 있는 모든 문서객체를 선택

`$(document).ready(function () {});`

`$('*').css('color', 'red');` ↗ `h1, head, html, head, body, script` 등  
모든 태그에 적용

{};

↳ 태그 선택자: 특정한 태그를 선택하는 선택자.

`$('h1').css('color', 'orange');`

`$('h1, p').css('color', 'red')`

↳ 여러개의 태그를 선택할 때, ,로 구분

↗ 유일한 값.

↳ 아이디 선택자: 특정한 id 속성이 있는 문서 객체 선택

`<h1 id='target'> Header </h1>`

↗ `h1` 태그 중 `id`가 `target`

`$('#target').css('color', 'red');`

`$('#target').css('color', 'red');`

↳ 클래스 선택자: 특정한 class 속성이 있는 문서 객체 선택.

`<h1 class='item'> Header </h1>`

↗ `h1` 태그이면서 `item` 클래스

`$('.item').css('color', 'red');`

`$('.item').css('color', 'red');`

`$('.item.select').css('color', 'red');`

↳ item 클래스와 select 클래스 동시에

ex) `<h1 class='item select'>`

자손선택자와 후손선택자 : 기본선택자의 앞에 붙여 사용하며 기본선택자의 범위를 제한한다.

↳ 바로 아래      ↳ 아래에 있는 모든 것.  
\$(‘body > \*’)      \$(‘body \*’)

속성선택자 : 입력 양식과 관련된 태그를 선택할 때 사용

요소 [속성 = 값] : 속성과 값이 같은 문서 객체

요소 [속성 != 값] : 속성 안의 값이 특정 값과 같은 문서 객체

요소 [속성 ~= 값] : 속성 안의 값이 특정 단어로 시작하는 문서 객체

요소 [속성 ^= 값] : 속성 안의 값이 특정 값으로 시작하는 문서 객체

요소 [속성 \$= 값] : 속성 안의 값이 특정 값으로 끝나는 문서 객체

요소 [속성 \*= 값] : 속성 안의 값이 특정 값을 포함하는 문서 객체

<input type = "text">    \$\(`input[type="text"]`).val('a');

↳ 속성을 변경

val() : 속성을 검사.

\$(`input:text`)

필터 선택자 ': ' 기호를 포함.

요소:button : input 태그 중 type 속성이 button인 문서 객체와 button 태그

:checkbox , :file, :image, :password, :radio, :reset

:submit , :text

요소:checked , :disabled, :enabled

:focus , :input , :selected

ex) var value = \$\(`select > option:selected`).val();

↳ :selected 자주 씀!

위치필터 선택자

요소:odd : 홀수번째 위치한 문서 객체 선택.

요소:even : 짝수 "      \$\(`tr:even`).css('color', 'red');

요소:first : 첫번째 "      \$\(`tr:first`).css('color', 'red');

요소: last : 마지막 "

### 함수필터 선택자

요소: contains(문자열) : 특정 문자열을 포함하는 문서객체 선택.

요소: eq(n) : n번째 위치하는 문서객체 선택

요소: gt(n) : n번째 초과에 위치하는 문서객체 선택

요소: has(h1) : h1 태그가 있는 문서 객체 선택

요소: lt(n) : n번째 미만에 있는 문서객체 선택

요소: not(선택자) : 선택자와 일치하지 않는 문서객체 선택

요소: nth-child(3n+1) :  $3n + 1$  번째 위치하는 문서객체 선택 (1,4,7)

### 배열관리

Each() 메서드: 매개변수로 입력한 함수로 for in 반복문처럼 객체나 배열의 요소 검사

- \$each(object, function(index, item));
- (selector).each(function(index, item));

var array = [

{name: 'abc', link: 'naver'},

{name: 'song', link: 'whdgdp'}

];

↓ 배열

값

\$each(array, function(index, item))

↳ 배열의 index

Output += item.name

↳ array 배열을 돌면서

Output += item.link

name, link에 접근!

`addClass()` : 문서 객체에 class 속성을 추가

`removeClass()` : 문서 객체의 class 속성을 제거

`$('.h1').each(function(index, item){`

`$(this).addClass('high-light-' + index))`

`});`  $\xrightarrow{\text{↳ } 0, 1, 2..}$

## 객체 확장 (`$.extend`)

`var object = {};`

`object.name = 'JONG';`

`object.gender = 'Male';`

`object.part = 'Second';`

`var object = {name: 'JONG'}`

`$.extend(object, {`

`gender: 'Male',`

`part: 'Second'`

`});`

$\hookrightarrow$  object 객체에 합쳐짐.

`$.noConflict()` : jQuery 식별자 \$를 사용하지 않겠다.

`$.noConflict();`

`$(document).ready();`  $\rightarrow X$

`jQuery(document).ready();`  $\rightarrow O$

`var j = jQuery`

`j(document).ready`

`$.extend()`로 옵션 객체 보안

optionX 값이 있으면 값.

`function test(options){`

$\rightarrow$  없으면 초기화된 값

```
|  
options.valueA = options.valueA || 10;      (10, 20, 30)  
options.valueB = options.valueB || 20;  
options.valueC = options.valueC || 30;  
?  
  
↓  
test({ valueA: 52,  
       valueB: 213 })
```

```
function test(options){  
    options = $.extend({  
        valueA: 10,      ↳ $extend(샘플, options);  
        valueB: 20,  
        valueC: 30  
    }, options);  
}
```