수식 중 ===

Each ()

Eq()

event.preventDefault

hasClass

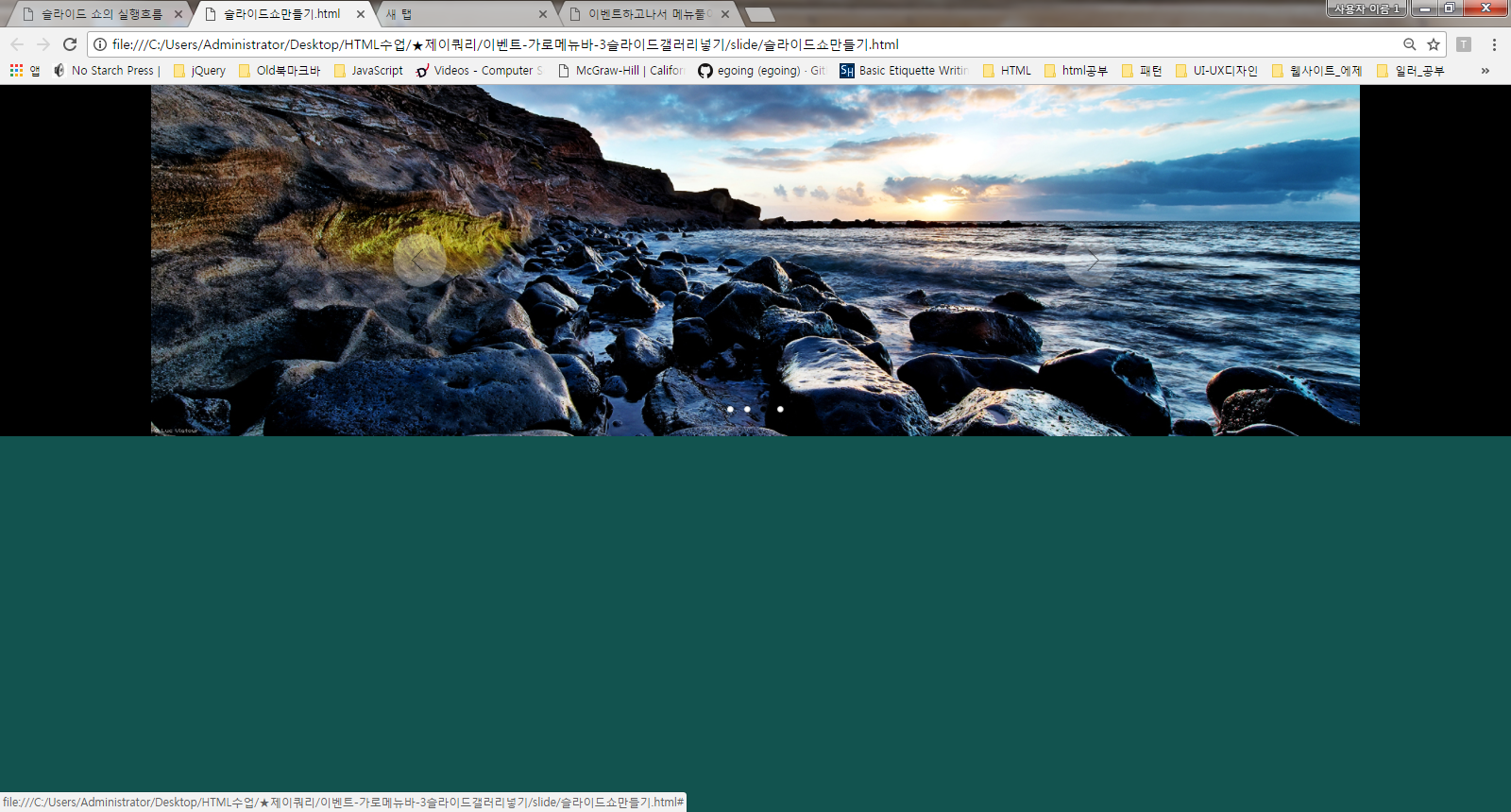
addClass

removeClass

clearInterval

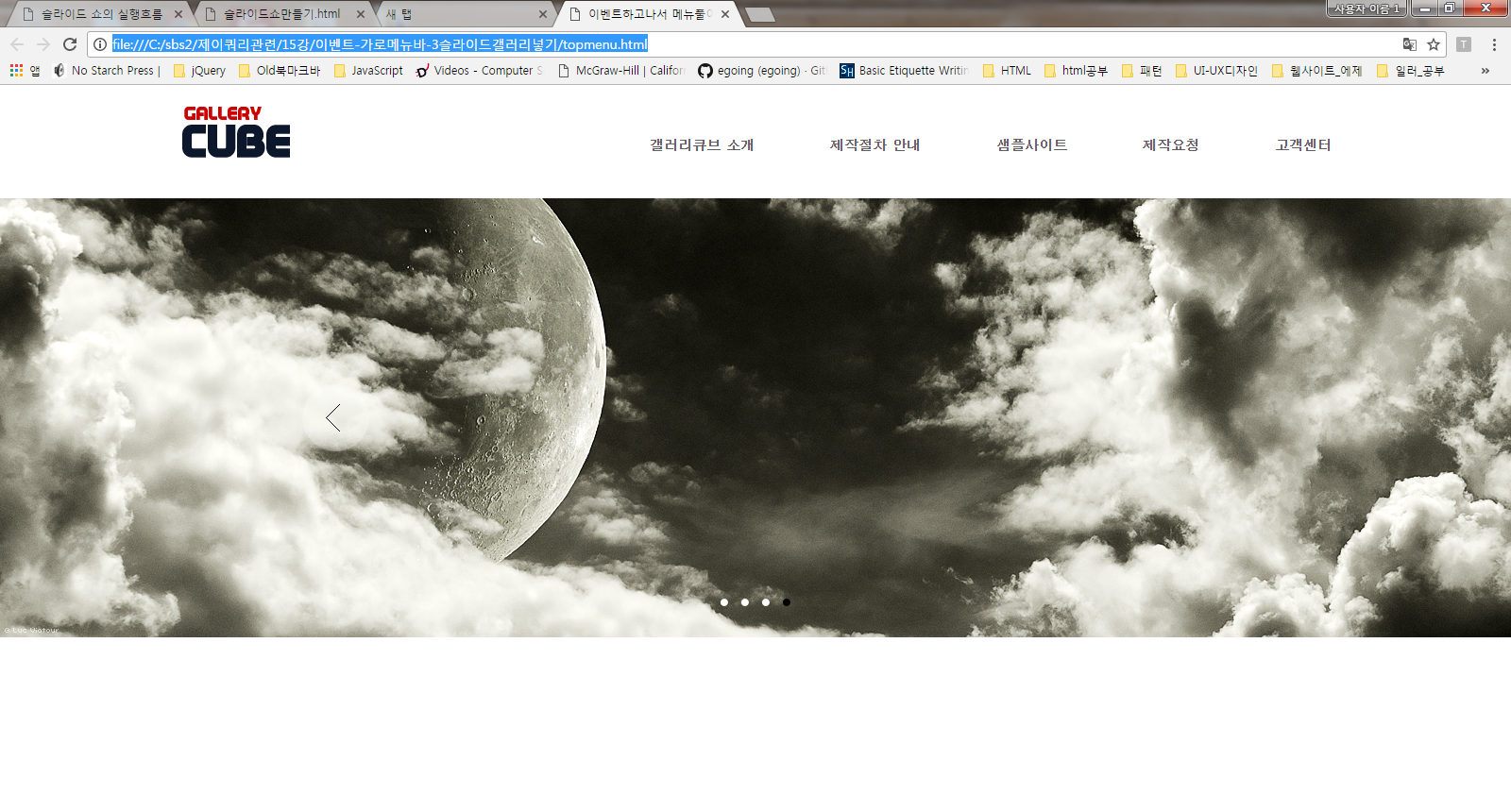
$('img',this) : 표현식 설명필요

**슬라이드쇼 만들기**

‘

|  |
| --- |
| Jquery 내용  $(function () {  /\*  \* Slideshow  \*/  $('.slideshow').each(function () {  // 변수의 준비  // - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  var $container = $(this), // a  $slideGroup = $container.find('.slideshow-slides'), // b  $slides = $slideGroup.find('.slide'), // c  $nav = $container.find('.slideshow-nav'), // d  $indicator = $container.find('.slideshow-indicator'), // e  // 슬라이드 쇼의 각 요소의 jQuery 객체  // a 슬라이드 쇼 전체 컨테이너  // b 모든 슬라이드의 정리 (슬라이드 그룹)  // c 각 슬라이드  // d 네비게이션 (Prev/Next)  // e 인디게이터 (점)  slideCount = $slides.length, // 슬라이드 점수  indicatorHTML = '', // 인디게이터 콘텐트  currentIndex = 0, // 현재 슬라이드의 인덱스  duration = 1000, // 다음 슬라이드에 애니메이션의 소요 시간  easing = 'easeInOutExpo', // 다음 슬라이드에 애니메이션의 이징 종류  interval = 4500, // 자동으로 다음 슬라이드로 옮길 때까지의 시간  timer; // 타이머  // HTML 요소의 배치 생성 삽입  // - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  // 각 슬라이드의 위치를 결정하고,  // 해당 인디게이터의 앵커를 생성  $slides.each(function (i) {  $(this).css({ left: 100 \* i + '%' });  indicatorHTML += '<a href="#">' + (i + 1) + '</a>';  });  // 인디게이터에 콘텐트를 삽입  $indicator.html(indicatorHTML);  // 함수의 정의  // - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  // 모든 슬라이드를 표시하는 함수  function goToSlide (index) {  // 슬라이드 그룹을 대상 위치에 맞게 이동  $slideGroup.animate({ left: - 100 \* index + '%' }, duration, easing);  // 현재 슬라이드의 인덱스를 덮어쓰기  currentIndex = index;  // 탐색 및 표시 상태를 업데이트  updateNav();  }  // 슬라이드의 상태에 따라 탐색 및 표시를 업데이트하는 함수  function updateNav () {  var $navPrev = $nav.find('.prev'), // Prev (뒤로) 링크  $navNext = $nav.find('.next'); // Next (앞으로) 링크  // 만약 첫 번째 슬라이드이라면 Prev 탐색을 해제  if (currentIndex === 0) {  $navPrev.addClass('disabled');  } else {  $navPrev.removeClass('disabled');  }  // 만약 마지막 슬라이드이라면 Next 탐색을 해제  if (currentIndex === slideCount - 1) {  $navNext.addClass('disabled');  } else {  $navNext.removeClass('disabled');  }  // 현재 슬라이드의 표시를 해제  $indicator.find('a').removeClass('active')  .eq(currentIndex).addClass('active');  }  // 타이머를 시작하는 함수  function startTimer () {  // 변수 interval에서 설정 한 시간이 경과 할 때마다 작업을 수행  timer = setInterval(function () {  // 현재 슬라이드의 인덱스에 따라 다음 표시 할 슬라이드의 결정  // 만약 마지막 슬라이드이라면 첫 번째 슬라이드에  var nextIndex = (currentIndex + 1) % slideCount;  goToSlide(nextIndex);  }, interval);  }  // 타이머를 중지있는 함수  function stopTimer () {  clearInterval(timer);  }  // 인벤토리 등록  // - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  // 네비게이션 링크를 클릭하면 해당 슬라이드를 표시  $nav.on('click', 'a', function (event) {  event.preventDefault();  if ($(this).hasClass('prev')) {  goToSlide(currentIndex - 1);  } else {  goToSlide(currentIndex + 1);  }  });  // 인디게이터의 링크를 클릭하면 해당 슬라이드를 표시  $indicator.on('click', 'a', function (event) {  event.preventDefault();  if (!$(this).hasClass('active')) {  goToSlide($(this).index());  }  });  // 마우스오버를 하면 타이머를 정지 그렇지 않으면 시작  $container.on({  mouseenter: stopTimer,  mouseleave: startTimer  });  // 슬라이드 쇼 시작  // - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -  // 첫 번째 슬라이드를 표시  goToSlide(currentIndex);  // 타이머를 시작  startTimer();  });  });  Body 에서  <body>  <!--전체슬라이드쇼의 영역-->  <div class="slideshow">  <div class="slideshow-slides">  <a href="#" class="slide">  <img src="slide-1.jpg" /></a>  <a href="#" class="slide">  <img src="slide-2.jpg" /></a>  <a href="#" class="slide">  <img src="slide-3.jpg" /></a>  <a href="#" class="slide">  <img src="slide-4.jpg" /></a>  </div>  <div class="slideshow-nav">  <a href="#" class="prev">Prev</a>  <a href="#" class="next">Next</a>  </div>  <div class="slideshow-indicator"></div>  </div>  </body>  Css 에서  <style type="text/css">  \* {  margin: 0;  padding: 0;  }  body {  background-color: #12534f;  min-width:1200px;  }  .slideshow {  background-color: rgb(0, 0, 0);  height: 465px;  min-width: 960px;  overflow: hidden;  position: relative;  }  .slideshow-slides {  height: 100%;  position: absolute;  width: 100%;  /\* left JavaScript \*/  }  .slideshow-slides .slide {  height: 100%;  overflow: hidden;  position: absolute;  width: 100%;  /\* left \*/  }  .slideshow-slides .slide img {  left: 50%;  margin-left: -800px;  position: absolute;  }  /\*내비게이션스타일\*/  .slideshow-nav a,  .slideshow-indicator a {  overflow: hidden;  }  /\*하단작은버튼\*/  .slideshow-nav a:before,  .slideshow-indicator a:before {  content: url(sprites.png);  display: inline-block;  font-size: 0;  line-height: 0;  }  .slideshow-nav a {  position: absolute;  top: 50%;  left: 50%;  width: 72px;  height: 72px;  margin-top: -36px;  }  /\*표시되는 슬라이드가 첫번째 슬라이드인 경우에는 Prev버튼을 숨김\*/  .slideshow-nav a.prev {  margin-left: -480px;  }  .slideshow-nav a.prev:before {  margin-top: -20px;  }  /\*표시되는 슬라이드가 마지막 슬라이드인 경우에는 Next버튼을 숨김\*/  .slideshow-nav a.next {  margin-left: 408px;  }  .slideshow-nav a.next:before {  margin-left: -80px;  margin-top: -20px;  }  .slideshow-nav a.disabled {  display: none;  }  /\*버튼을 표시하거나 숨기는 효과는 표시되는 슬라이드의 순서가 결정되었을때 자바스크립트로 각버튼의 .disabled를 클릭하거나 제거하는것으로 구현.\*/  .slideshow-indicator {  bottom: 30px;  height: 16px;  left: 0;  position: absolute;  right: 0;  text-align: center;  }  .slideshow-indicator a {  display: inline-block;  width: 16px;  height: 16px;  margin-left: 3px;  margin-right: 3px;  }  .slideshow-indicator a.active {  cursor: default;  }  .slideshow-indicator a:before {  margin-left: -110px;  }  .slideshow-indicator a.active:before {  margin-left: -130px;  }  </style> |

슬라이드쇼 만들고 메뉴가 있는 웹페이지 집어넣기



|  |
| --- |
| <!DOCTYPE html>  <html lang="en" xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta charset="utf-8" />  <title>이벤트하고나서 메뉴풀이</title>  <style>  \* {  margin: 0;  padding: 0;  }  li {  list-style: none;  }  a {  text-decoration: none;  color:#5e5e5e;  }  a:hover {  color: black;  }  body {  font-family: Dotum;  font-size: 12px;  min-width:1200px;  }  /\*구조\*/    /\*주메뉴 각각 공통클래스\*/  #top {  width: 1400px;  margin: 0 auto;  position:relative;  height:120px;  }  .menu {  font-size:14px;  text-align:center;  font-weight:900;  line-height:49px;  position:relative;  float:left;  overflow:hidden;  padding-left:20px;  padding-right:20px;  }  .menu a {  display: block;  padding-left:20px;  padding-right:20px;  }  /\*서브메뉴 각각공통클래스\*/  .sub\_menu li {  height: 29px;  font-size:12px;  text-align:center;  line-height:29px;  padding: 10px;    }  .sub\_menu {  display: none;  }  .sub\_menu a {  font-size: 12px;  display: block;  font-weight:100;  }  .sub\_menu a:hover {  color:white;  background-color:black;  }      .logo {  float: left;  width:200px;  text-align:center;  line-height:100px;  position:absolute;  left:50px;  background-image:url(toplogo.png);  background-repeat:no-repeat;  background-position:center;  text-indent:-9999px;  }  .logo a {  display: block;  }  #nav {  overflow:hidden;  position:absolute;  right:50px;  top:40px;  }  .mainBody {  width: 100%;  height: 465px;  background-color: #0c172c;  color:white;  overflow:hidden;  }  .mainBody h2 {  width: 400px;  font-family: Tahoma;  margin: 100px auto;  font-size:40px;  text-align:center;  background-color:blue;  }  .mainBody p {  width: 500px;  height: 100px;  margin: 0px auto;  font-size:15px;  font-weight:900;  text-align:center;  line-height:25px;  }  iframe {  width: 100%;  height: 465px;  border: none;  min-width:1200px;    }  </style>  <script src="jquery-1.11.3.min.js"></script>  <script>  $(document).ready(function () {  /\*주메뉴에 올렸을때 서브메뉴보이도록\*/    $('.menu').hover(function () {  $(this).css('background', '#fff');  $('.sub\_menu', this).css('display', 'block'); //this 여기 menu영역에 마우스를 올렸을때  }, function () {  $(this).css('background', '#fff');  $('.sub\_menu').css('display', 'none');  });    });  </script>  </head>  <body>  <div id="top">  <h1 class="logo"><a href="#">logo</a></h1>  <ul id="nav">  <li class="menu"><a href="#">갤러리큐브 소개</a>  <ul class="sub\_menu">  <li><a href="#">서브01</a></li>  <li><a href="#">서브02</a></li>  <li><a href="#">서브03</a></li>  </ul>  </li>  <li class="menu"><a href="#">제작절차 안내</a>  <ul class="sub\_menu">  <li><a href="#">서브01</a></li>  <li><a href="#">서브02</a></li>  <li><a href="#">서브03</a></li>  </ul>  </li>  <li class="menu"><a href="#">샘플사이트</a>  <ul class="sub\_menu">  <li><a href="#">서브01</a></li>  <li><a href="#">서브02</a></li>  <li><a href="#">서브03</a></li>  </ul>  </li>  <li class="menu"><a href="#">제작요청</a>  <ul class="sub\_menu">  <li><a href="#">서브01</a></li>  <li><a href="#">서브02</a></li>  <li><a href="#">서브03</a></li>  </ul>  </li>  <li class="menu"><a href="#">고객센터</a>  <ul class="sub\_menu">  <li><a href="#">서브01</a></li>  <li><a href="#">서브02</a></li>  <li><a href="#">서브03</a></li>  </ul>  </li>  </ul>  </div>  <!--mainBody-->  <div class="mainBody">  <iframe src="slide/slide-show.html" scrolling="no"></iframe>  </div>    </body>  </html> |

jQuery 2. 요소 선택 방법 - 셀렉터

셀렉터를 이용해서 HTML 태그 요소를 선택하는겁니다.

1. 태그 셀렉터

$("h1").

html 문서내의 <h1>...</h1> 을 선택합니다.

<h1>헤드라인</h1>

2. id 셀렉터

$("#title").

html 태그요소중에 id="title" 인 요소를 선택합니다.

<h1 id="title">헤드라인</h1>

만약 h1 태그중에 id가 title 인 요소를 선택하려면, 아래와 같이 셀렉터를 작성할 수 있습니다.

$("h1#title").

3. class 셀렉터

$(".title").

html 태그요소중에 class="title" 인 요소를 선택합니다.

<h1 class="title">헤드라인</h1>

만약 h1 태그중에 class가 title 인 요소를 선택하려면, 아래와 같이 셀렉터를 작성할 수 있습니다.

$("h1.title").

4. 자손 셀렉터

자손이라고 하는 것은 특정 태그 내에 포함되어 있는 모든 태그 요소를 의미합니다.

<ul id="grand">

    <li>아들 1</li>

    <li>딸 1</li>

    <li>

            <ul id="parent">

                  <li>손자1</li>

                  <li>손녀1</li>

            <ul>

    </li>

</ul>

위 구조에서 아들1, 딸1, 손자1, 손녀1 모두 최상위 ul의 자손인거죠..ㅡ.ㅡ

위의 모든 li 요소들을 선택하기 위한 셀렉터는 아래와 같이 작성할 수 있습니다.

$("ul#grand li").

아이디가 grand 인 ul 의 하위에 있는 모든 li 요소를 선택한다는 의미입니다.

5. 자식 셀렉터

자식은 바로 아래에 있는 요소들을 의미합니다.

$("ul#parent > li").

이렇게 셀렉터를 작성하면, 바로 아래 단계의 손자1, 손녀1만 선택할 수 있습니다.

이건 예가 그닥 좋지는 않군요..ㅡ.ㅡ

5. 그룹 셀렉터

그룹은 여러개를 묶어놓으면 그룹인거죠.

쉽게 콤마(,)를 이용해서 나열해 주는 방법으로 선택할 수 있습니다.

$("h1, h2").

7. 유니버설 셀렉터

특정 요소 하위의 모든 요소를 선택할 수 있는 방법입니다.

$("ul#grand \*").

8. 인접 셀렉터

인접은 바로 옆이라는 의미죠. 가장 가까이 있는 요소를 말합니다.

$("#first + li").

위와 같이 작성하면, id가 first 인 요소 다음에 가장 먼저 오는 li 요소를 선택할 수 있습니다.

9. first-child 셀렉터

말그대로 첫번째 자식을 선택할 수 있습니다.

$("li:first-child").

li는 ul의 자식요소인데, li가 여러개 있을 경우, 첫번째 li 요소를 선택합니다.

10. 간접셀렉터

특정 요소 뒤에 나타나는 모든 요소를 선택할 수 있습니다.

$("#second ~ li").

11. only-child 유사클래스 셀렉터

$("li span:only-child")

예)

<li><span>제이쿼리</span></li>

li 요소 안에 span 요소가 온리 하나뿐이면 선택됩니다. 자식이라곤 span 밖에 없는 li요소..

12. nth-child 유사클래스 셀렉터

$("li:nth-child(2)").

ul 하위 li들 중에 2번째 li를 선택합니다.

$("li:nth-child(even)").

ul 하위 li들 중에 짝수째 li를 선택합니다.

$("li:nth-child(odd)").

ul 하위 li들 중에 홀수째 li를 선택합니다.

$("li:nth-child(3n)").

ul 하위 li들 중에 3의 배수번째 li들을 선택합니다.

13. last-child 유사클래스 셀렉터

마지막 자식, 막내를 선택합니다.

$("li:last-child").

14. not 유사클래스 셀렉터

$("li:not(:first-child)").

not 으로 지정한 특성을 제외한 li요소들을 선택합니다.

15. empty 유사클래스 셀렉터

$("li:empty").

<li></li> 처럼 속 내용이 빈 요소를 선택합니다.

속성은 html 태그에 더해진 id, class 등을 의미합니다.

<li id="first" class="number" >블라블라</li>

id도 속성이고 해당 value는 first, class도 속성이고 해당 value 는 number 인거죠.

$("[속성]").

$("[id]").

html 태그 요소중에 id 라는 속성을 가진 요소들을 선택합니다.

$("[속성='값']").

$("[id='first']").

html 태그 요소중에 id값이 first 인 요소들을 선택합니다.

$("[속성!='값']").

$("[id=!'first']").

html 태그 요소중에 id값이 first 가 아닌 요소들을 선택합니다...만, 위와 같이 사용할 수 없고 반드시 해당 html태그를 명시해줘야합니다.

$("li[id=!'first']").

위 예는 li 요소중에 id가 first가 아닌 것들을 선택합니다.

$("[속성^='값']").

속성이 특정 문자열로 시작되는 요소들을 선택합니다.

$("[title^='f']").

태그중에 title 속성의 값이 f로 시작하는 태그 요소들을 선택합니다.

<li title='first'>블라블라</li>

$("[속성$='값']").

속성이 특정 문자열로 끝나는 요소들을 선택합니다.

$("[title$='d']").

태그중에 title 속성의 값이 d로 끝나는 태그 요소들을 선택합니다.

<li title='second'>블라블라</li>

$("[속성\*='값']").

속성이 특정 문자열을 포함하는 요소들을 선택합니다.

$("[title$='u']").

태그중에 title 속성의 값이 u를 포함하는 태그 요소들을 선택합니다.

<li title='fourth'>블라블라</li>

위 속성 셀렉터를 중복으로 사용할 수 있습니다.

title속성이 f 로 시작하고, u를 포함하는 요소를 선택하고 싶으면 아래와 같이 할 수 있습니다.

$("[title^='f'][title$='u']").

여기까지 내용은 CSS 에서 요소를 선택하는 방법과 같습니다. CSS 작성할 때 위와 같은 방법으로 요소들을 지정해 줄수 있습니다. 아! CSS3까지 포함된 내용입니다.

jQuery 자체 필터

그리고, jquery 자체 필터라는게 있습니다. jquery 가 지원하는 셀렉팅 방법이지요.

- first / last  필터

$("li:first") / $("li:last")

위와 같이 첫번째 li 요소와 마지막 li 요소를 선택할 수 있습니다.

first-child, last-child 와 의미가 비슷하죠? 하지만, 같은 방법을 굳이 만들었을리가..ㅡ.ㅡ

first-child와 last-child는 공통된 부모 요소 내에서 첫번째, 마지막을 골라내주는 것이고,

first, last 필터는 부모와 관계없이 페이지내에 존재하는 모든 태그들 중에서 첫번째, 마지막을 골라내주는 것입니다.

- even, odd 필터

$("li:even") / $("li:odd")

nth-child에서 even, odd 를 사용하는 것과 비슷하지만, 이 필터 역시 공통 부모의 의미가 없이 전체 태그들을 대상으로해서 짝수번째, 홀수번째를 골라내줍니다.

- eq, gt, lt 필터

eq 는 같다, gt는 뭐보다 크다, lt는 뭐보다 작다...의미입니다.

같은 종류의 태그가 여러개 있을 경우,

eq 는 몇번째 태그를 지정해주거나,

gt 는 몇번째 태그보다 큰것들을 모두 골라내거나,

lt 는 몇번째 태그보다 작은것들을 모두 골라내줍니다.

크고 작음을 어떻게 구분할까요?

같은 태그가 여러개 있을 경우, 첫번째 태그는 0번, 그 이후 1,2,3... 등으로 인덱스가 붙습니다. 인덱스가 0번부터 시작함에 주의해야합니다.

이 필터를 사용할 때는 이 인덱스 번호를 이용합니다.

$("li:eq(2)")

여러개의 li 요소중에 3번째 li요소를 선택합니다.

$("li:gt(2)")

여러개의 li 요소중에 인덱스 번호가 2보다 큰 li요소를 선택합니다. 4번째, 5번째...등등 골라내주는거죠.

$("li:lt(2)")

여러개의 li 요소중에 인덱스 번호가 2보다 작은 li요소를 선택합니다. 1번째, 2번째 li요소를 골라내주는거죠.

이 세가지 필터 모두 역시 공통 부모와 무관하게 전체 태그요소들을 대상으로 동작합니다.

- header  필터

$(":header") 로 사용해서 h1~h6 태그를 선택할 수 있습니다.

- contains / has 필터

contains 는 특정 문자열을 포함하고 있는지를 확인해서 선택하고, has 는 특정 태그를 포함하고 있는지를 확인해서 선택합니다.

<li>제이쿼리</li>의 경우, $("li:contains("제이")") 로 선택할 수 있습니다.

<li><strong>제이쿼리</strong></li> 는 strong이라는 태그를 포함하고 있으니까, 태그를 기준으로 선택하려면, 아래와 같이 has를 사용할 수 있습니다.

$("li:has(strong)").

- parent 필터

값을 가지고 있는 자식태그를 선택해 줍니다.

$("li:parent") 라고 작성하면, li 들 중에, 값을 가지고 있고, 어떤 태그의 자식 태그인 요소를 선택해줍니다.

li는 무조건 ul 태그의 자식 태그니까, 자식 태그라는 요건은 만족하고, 값을 가지고 있느냐를 더 확인을 하겠지요.

<ul>

    <li>아들 1</li>

    <li>딸 1</li>

    <li></li>

</ul>

위 예에서는 아들1을 포함한 li 태그와 딸1을 포함한 li태그만이 선택됩니다

## 여러 개의 [셀렉터](http://nyaongnyaong.com/974) 선택

여러 가지를 한꺼번에 선택해야 하는 경우가 간혹 발생한다.

<span id="str">TEST</span>

<span class="str">TEST</span>

<span>TEST</span>

위와 같은 상황에서 TEST라는 텍스트를 모두 bold를 적용해야 하는 경우가 있다.  
이런 경우 Javascript로 할 경우 코드가 좀 길어진다.  
id로 갖고 오고 class로 갖고 오고 tag로 갖고.  
jQuery의 경우는 한 줄로 모두 선택이 가능하다.

<script type="text/javascript">

$(document).ready(function() {

$("#str, .str, span").css("font-weight", "bold");

});

</script>

위에서 보는 것처럼 ,(comma)로 구분지어서 선택이 가능하다.  
  
문서 내의 모든 요소를 선택하려면 **\*** 셀렉터를 사용합니다.

--- ----------------------

$(셀렉터 , this) 는 이런 표현이다.

var newimg = $('img',this).attr('src') == 'tick.png'?'cross.png':'tick.png';

var newimg = $(this).find('img').attr('src')=='tick.png'?'cross.png':'tick.png';

예제보기

|  |
| --- |
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">  <head>  <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.min.js"></script>  <script type="text/javascript">  $(function()  {  //handle the click for each of 'li'  $('ul#services li').click(function()  {  //toggle the image    var newimg = $(this).find('img').attr('src')=='tick.png'?'cross.png':'tick.png';  // var newimg = $('img',this).attr('src') == 'tick.png'?'cross.png':'tick.png';  // jQuery("img", this) 는 jQuery(this).find("img") 같은 뜻  //notice the second parameter is the context. 'this' is the clicked 'li'  $('img',this).attr('src',newimg);  });  });  </script>  <style type="text/css">  body{ font-family:Verdana, Arial, serif; font-size:14px;}  ul#services li{padding:5px; list-style-type: none; cursor:default}  ul#services li:hover{ color:red;}  </style>  </head>  <body>  <h2>Select Services Required:</h2>  <ul id='services'>  <li><img src='tick.png'> Build a Website</li>  <li><img src='tick.png'> Page design</li>  <li><img src='tick.png'> Flash banners</li>  <li><img src='tick.png'> SEO</li></ul>  </ul>  </body>  </html> |