#### **jQuery** ―――

<body> **요기 Script 요기** </body> 사이에 넣음!

| <!DOCTYPE html>  <html lang="ko">  <head>  <meta charset="UTF-8">  <title>Document</title>  <script src="<http://code.jquery.com/jquery-latest.min.js>"></script>  <script src="http://code.jquery.com/color/jquery.color.plus-names-2.1.2.min.js" integrity="sha256-Wp3wC/dKYQ/dCOUD7VUXXp4neLI5t0uUEF1pg0dFnAE=" crossorigin="anonymous"></script>  </head>  <body>  html이 와바바봥왕 들어가고 그 아래 스크립트!  <script>  $(document).ready(function(){    });  </script>  </body>  </html> |
| --- |

$(선택자).동작함수();

$기호는 제이쿼리를 의미하며, 제이쿼리에 접근할 수 있게 해 주는 식별자다. $() 함수는 선택된 HTML 요소를 제이쿼리에서 이용할 수 있는 형태로 생성해 주는 역할을 한다. $() 함수의 인수로는 HTML 태그 이름 뿐만 아니라, CSS 선택자를 전달하여 특정 HTML 요소를 선택할 수 있다.

제이쿼리: https://api.jquery.com/category/css/

#### 

#### 

#### **문법**

| **코드 값** | **비고** | | |
| --- | --- | --- | --- |
| css({속성:"속성값"});  태그선택자 | 제이쿼리에서 스타일 직접 입력해서 적용  $("ul>li:nth-child(3) > span").css({color:"blue"})  css("속성", "속성값") 스타일을 한번밖에 선언못함  여러개 선언가능한스타일 => css({속성:속성값, 속성:속성값})  속성값이 px단위가 붙을시 "속성값px"로 선언  css({속성:"속성값"}) | | |
| $("\*") | 전부 | | |
| $("this") | 현재 선택한 HTML 엘리먼트 | | |
| $("#아이디") | id 속성의 값과 일치하는 요소 | | |
| $(".클래스") | class 속성의 값과 일치하는 모든 요소 | | |
| $("n1, n2, …") | 쉼표로 구분해서 여러 요소를 한꺼번에 선택 | | |
| $("p:first") | p태그중에 첫번째 | | |
| $("p.intro") | class="intro" 인 모든 <p> 태그를 선택 | | |
| $("ul li:first") |  | | |
| $("ul li:first-child") | :first-child  부모 요소 안에 있는 첫번째 자식만 선택 | | |

#### **순서&탐색1**

| **코드 값** | **비고** | **예문** | | |
| --- | --- | --- | --- | --- |
| .index() | 일렬번호를 생성하여 선택 | var i = $(".li2").index(); // 1 | | |
| .eq(n) | 몇 번째를 의미로 하여 선택  n번째 요소 | $("ul > li").eq(i).css({  color:"red"  }); | | |
| .first() | 형제태그 중에 처음을 선택  첫번째 요소 | $("ul > li").first().css({  color:"pink"  }); | | |
| .last() | 형제태그 중 마지막을 선택  마지막 요소 | $("ul > li").last().css({  color:"yellow"  }); | | |
| .has() | 어떤 속성이나 요소를 가지고 있는 대상 | $("ul>li").has("span").css({  fontSize: "20px"  }); | | |
| .not() | 어떤 속성이나 요소를 가지지 않은 대상 | $("ul>li").not(".li2").css({  fontFamily: "바탕체"  }); | | |
| even() | 짝수번째 요소  <select>  <option>첫번째</option>  <option>두번째</option>  </select> | $('select option:even')  select 내의 짝수번째 option을 선택한다. | | |
| odd() | 홀수번째 요소 | $('select option:odd') | | |

#### **순서&탐색2**

| **코드 값** | **비고** | | |
| --- | --- | --- | --- |
| children() | 선택한 애의 자식 (자손포함X)  $("ul").children().css({  color: "red"  }); | | |
| find() | 선택한 애의 자손. 조건에 맞는것 (자손포함)  ex) $(".A").find("div")  $("ul > li").eq(2).find("span").css({  color: "gray"  }); | | |
| next() | 자신의 다음에 선언된 태그를 선택  메뉴 슬라이드 다운에서 사용할 수 있음  $(".sm").hide();  $("ul.mn>li>div").click(function(){  $(".sm").hide(); //모든 .sm을 숨깁니다  $(this).next(".sm").slideDown(); //선택한 .sm을 제외한 메뉴만 내림  }); | | |
| nextAll() | 자신의 다음에 선언된 태그 전부를 선택  $("ul > li").eq(2).nextAll().css({  color:"blue"  }); | | |
| parent() | 자신의 부모  $("span").parent().parent().css({  fontSize: "20px"  }); | | |
| parents() | 자신의 부모들  $("span").parents().css({  lineHeight: "20px"  }); | | |
| prev() | next()의 반대. 자신의 이전에 선언된 태그를 선택 | | |
| prevAll() | nextAll()의 반대. 자신의 이전에 선언된 태그 전부를 선택 | | |
| siblings() | 형제 다찾기 (나 자신 포함 X) | | |
| is()  불린 느낌?? | $("선택요소").is("비교할 요소 또는 표현식);  선택한 요소가 is(요소or표현식)과 일치하는지 확인할 때 사용됨  - 선택요소와 비교대상이 동일하면 true 아니면 false를 반환  예제1) p 태그를 가지는 경우 포함한다는 경고창을 보여주고 반대로 없는 경우라면 없다는 문구의 경고창을 출력하는 구문.  <div class="parentEle">  <p>P tag here !!</p>  </div>  위 내용은 html 코드이며 이제 여기에 is()를 사용한 스크립트.  var isChild = $("div").children().is("p");  // isChild 변수에 자식요소인지 검수 후 결과를 저장  if (isChild) {  alert('자식 요소가 맞습니다.');  }  else {  alert('자식 요소가 아닙니다.');  }  p 태그가 맞으므로 '자식 요소가 맞습니다.'를 출력된다.  \*\*만약 해당 클래스가 있는지 없는지 알아보려면?  <script>  var isChild = $("div").is(".test");  if (isChild) {  alert("It has test classname !");  }  </script>  위 코드는 test 라는 클래스명을 가지고 있는지 판단하기 위해서 is() 내부에 .test를 추가하였다. 이처럼 선택자에 클래스를 사용하면 클래스 확인도 가능 | | |

#### **스크롤 이벤트**

| **코드 값** | **비고** | | |
| --- | --- | --- | --- |
| scrollTop  =가변적 | 현재 스크롤의 위치값을 나타낸다.    $(window).scrollTop(); | | |
| offset  =고정적 | 대상의 위치값 (요소의 위치값을 나타내는 코드)  \* 항상 고정적인 것은 아님, 반응형에서 브라우저 크기를 늘리거나 줄일 때 offset의 값도 변할 수 있다.  $('section').offset(); | | |

| **화면의 페이지별 이동 예문 (1219 05.html 참고)** | | |
| --- | --- | --- |
| menu.click 했을 때,  var menu = $("#header ul li");  var content = $("#contents > div");  menu.click(function(){  var tg = $(this);  var i = tg.index();  /\*  menu.removeClass("act");  tg.addClass("act");  \*/  var section = content.eq(i);  var tt = section.offset().top;  //offset은 선택한 것(div)의 기준을 잡아줌  //화면에서 이동할 떄 대상에 타겟을 적용하는데 그 대상을 기준으로 타겟점 설정  $("html, body").stop().animate({scrollTop:tt})  // 화면이동은 애니메이션사용하고 속성으로는 scrollTop(세로방향으로 이동),  // scrollLeft(가로방향으로 이동)  }); | | |
| $(window).scroll(function(){ //스크롤이 발생했을 때 실행할 함수  var sct = $(window).scrollTop(); //현재 화면의 스크롤 top값을 의미  $("#contents > div").each(function(){ //각각의 div들을 순서대로 처리  var tg = $(this); // each에 의해 순서대로 현재 선택된 div  var i = tg.index(); //each에 의해 순서대로 현재 선택된 div 인덱스값  })  }); | | |

| **화면의 스크롤 위치 별 아이콘 반응 (0109 01.html 참고)** | | |
| --- | --- | --- |
| var scene01Num = 1; // 불린형  $(window).scroll(function(){  var scrollTop = $(document).scrollTop();  if(scrollTop > 200 && scene01Num == 1){  scene01Num = 0; // 스크롤은 위아래 개념이 없어서 이게필요  $(".box1-1")  .delay(500)  .stop()  .css({opacity:1})  .animate({top: "800px", opacity: 0}, 500)  $(".box1-2")  .delay(100)  .stop()  .animate({top: "800px"}, 300)  .fadeOut(400);  }  else if(scrollTop < 200 && scene01Num == 0){  scene01Num = 1;  $(".box1-1")  .stop()  .css({opacity:0})  .animate({top: "200px", opacity: 1}, 500);  $(".box1-2")  .stop()  .fadeIn(400)  .animate({top: "500px"}, 300);  } | | |

| **마우스휠\_아이템들 위치 이동 (0109 02.html 참고)** | | |
| --- | --- | --- |
| //마우스휠 화면 사이즈 바꿀때 보조해주는 코드  var ht = $(window).height();  $('#wrap > div').height(ht);  $(window).on('resize', function () {  var ht = $(window).height();  $('#wrap > div').height(ht);  });  //마우스휠  $('#wrap > div').on('mousewheel', function (event, delta) { //delta는 불린개념  if (delta > 0) {  var prev = $(this).prev().offset().top;  $('html, body').stop().animate({ 'scrollTop': prev }, 1000, 'easeOutExpo');  } else if (delta < 0) {  var next = $(this).next().offset().top;  $('html, body').stop().animate({ 'scrollTop': next }, 1000, 'easeOutExpo');  }  });  //이벤트 확인하는 코드  $(window).on("mousewheel", function(e){  var E = e.originalEvent;  console.log(E);  }); | | |

#### **좌우 스크롤 생성**

| <SCRIPT src="./js/jquery.nicescroll.min.js" type="text/javascript"></SCRIPT> |
| --- |

| **마우스휠\_아이템들 위치 이동 (0109 04.html 참고)** | | |
| --- | --- | --- |
| >> CSS <<  body{  margin:0; padding:0;  overflow-y: hidden; /\*오른쪽 스크롤이 없어졌어요\*/  }  >> SCRIPT <<  $(document).ready(function(){  $("#all").niceScroll({  cursorcolor:"#777",  cursorwidth:5,  scrollspeed:100,  horizrailenabled:true //false  //\*\*horizrailenabled, nicescroll은 가로 스크롤을 관리 할 수 ​​있습니다 (기본값 : true)  });  }); | | |

| **코드 값** | **비고** | | |
| --- | --- | --- | --- |
| cursorcolor | 커서 색상을 16 진수로 변경합니다.  기본값은 "# 000000"입니다. | | |
| cursoropacitymin | 불투명도 변경 커서가 비활성 상태 (스크롤바의 "숨김"상태),  범위는 1에서 0까지, 기본값은 0 (숨김)입니다. | | |
| cursoropacitymax | 불투명도 변경 매우 커서가 활성화되어 있습니다  (스크롤바의 "가시"상태).  범위는 1에서 0까지이며, 기본값은 1입니다 (전체 불투명도). | | |
| cursorwidth | 커서의 너비 (픽셀 단위), 기본값은 5입니다  ( "5px"도 쓸 수 있음). | | |
| cursorborder | 커서 테두리의 CSS 정의. 기본값은 "1px solid #fff"입니다. | | |
| cursorborderradius | 커서의 테두리 반경, 기본값은 "4px" | | |
| zindex | 스크롤바 div의 Z- 인덱스 변경, 기본값은 9999 | | |
| scrollspeed | 스크롤 속도, 기본값은 60 | | |
| mousescrollstep | 마우스 휠로 스크롤 속도, 기본값은 40 (픽셀) | | |
| touchbehavior | 데스크톱 컴퓨터의 터치 장치처럼  커서 드래그 스크롤링 사용 (기본값 : false) | | |
| hwacceleration | 지원되는 경우 하드웨어 가속 스크롤 사용 (기본값 : true) | | |
| boxzoom | 상자 내용을 확대 할 수있게합니다 (기본값 : false). | | |
| dblclickzoom | (boxzoom = true 인 경우에만 해당)  상자를 두 번 클릭하면 줌이 활성화됩니다 (기본값 : true). | | |
| gesturezoom | (boxzoom = true 및 터치 장치 인 경우에만)  피치 아웃 / 인 온 상자에서 줌이 활성화됩니다 (기본값 : true). | | |
| -grabcursorenabled | touchbehavior = true로 div의 "grab"아이콘 표시 (기본값 : true)  자동 숨기기 모드, 스크롤 막대의 숨김 방법,  true = 기본값 / "커서"= 커서 만 숨김 / false = 숨기지 않음  배경, 레일 배경에 대한 CSS 변경, 기본값은 "" | | |
|  |  | | |
|  |  | | |
|  |  | | |

iframeautoresize,로드 이벤트에서 iframe 자동 크기 조정 (기본값 : true)

cursorminheight, 픽셀 단위로 최소 커서 높이 설정 (기본값 : 20)

preservenativescrolling을 사용하면 네이티브 스크롤 영역을 마우스로 스크롤하거나 마우스 휠 이벤트를 버블 링 할 수 있습니다 (기본값 : true)

railoffset을 사용하면 레일 위치에 대해 위쪽 / 왼쪽 오프셋을 추가 할 수 있습니다 (기본값 : false).

bouncescroll, 콘텐츠의 끝에서 모바일처럼 (단지 hw accell) 스크롤 바운스 가능 (기본값 : false)

spacebarenabled, 스페이스 바를 눌렀을 때 페이지 아래로 스크롤 가능 (기본값 : true)

레일 파딩, 레일 바용 패딩 설정 (기본값 : {위쪽 : 0, 오른쪽 : 0, 왼쪽 : 0, 아래쪽 : 0})

disableoutline, 크롬 브라우저의 경우 nicescroll을 사용하여 div를 선택할 때 윤곽선을 비활성화합니다 (오렌지색의 hightlight) (기본값 : true).

horizrailenabled, nicescroll은 가로 스크롤을 관리 할 수 ​​있습니다 (기본값 : true).

railalign, 수직 레일 정렬 (defaul : "right")

railvalign, 수평 레일 정렬 (defaul : "bottom")

enabletranslate3d, nicescroll은 내용을 스크롤하기 위해 css translate를 사용할 수 있습니다 (기본값 : true).

enablemousewheel, nicescroll은 마우스 휠 이벤트를 관리 할 수 ​​있습니다 (기본값 : true).

enablekeyboard, nicescroll은 키보드 이벤트를 관리 할 수 ​​있습니다 (기본값 : true).

smoothscroll, 쉽게 움직이는 스크롤 (기본값 : true)

sensitiverail, 레일을 클릭하여 스크롤합니다 (기본값 : true).

마우스 캡션 잠금 API를 사용할 수 있습니다 (개체 드래그에서 동일한 문제) (기본값 : true)

cursorfixedheight, 픽셀 단위의 커서 고정 높이 설정 (기본값 : false)

hidecursordelay, 마이크로 초 단위로 지연을 설정하여 스크롤바를 페이드 아웃 (기본값 : 400)

directionlockdeadzone, 방향 잠금 활성화 (기본값 : 6)의 데드 존 (픽셀)

nativeparentscrolling, 내용의 하단을 감지하고 네이티브 스크롤이 부모처럼 스크롤하도록합니다 (기본값 : true).

선택 스크롤을 활성화하고 선택 텍스트 (기본값 : true) 일 때 내용의 자동 스크롤 사용

#### **마우스 이벤트**

| **코드 값** | **비고** | | |
| --- | --- | --- | --- |
| click | 클릭할때  $("p").click(function(){  alert("텍스트가 선택되었습니다.");  }); | | |
| mousedown,  mouseup (=click) | 클릭할 때  클릭한 걸 떼어낼 때  $("p").mousedown(function(){  $(this).after("<p style='color:red'>마우스버튼 down</p>")  });  $("p").mouseup(function(){  $(this).after("<p style='color:blue'>마우스버튼 up</p>")  }); | | |
| dblclick | 더블 클릭할 때  $("p").dblclick(function(){  alert("텍스트가 선택되었습니다.");  }); | | |
| hover | 마우스 버튼 위에 올라갈 때,  마우스가 버튼에서 벗어날 때  $("p").hover(function(){  $(this).css({background: "#ccc"});  }, function(){  $(this).css({background: "#fff"});  });; | | |
| mouseover,  mouseout  (=hover) | 마우스 포인트가 진입할때  mouseover: 직접 이벤트를 걸지 않은 자식요소에 와도 발생  $("p").mouseover(function(){  $(this).css({background: "#ccc"});  });  $("p").mouseout(function(){ $(this).css({background: "#ccc"});  }); | | |
| mouseenter,  mouseleeve  (=hover) | 마우스 포인트가 진입할때  mouseenter: 오로지 자기자신에게 마우스 포인트가 와야만 발생  $("p").mouseenter(function(){ *//마우스를 버튼 위에 올라갔을 때*  $(this).css({background: "#ccc"});  });  $("p").mouseleeve(function(){ *//마우스를 버튼 위에 올라갔다가 나갈때*  $(this).css({background: "#ccc"});  }); | | |
| focusin()  focusout() | 포커스 인 이벤트는 요소(또는 하위요소)가 포커스를 얻을때 발생  $("div").focusin(function(){  $(this).css("background-color", "#ffffcc");  });  $("div").focusout(function(){  $(this).css("background-color", "#ffffff");  }); | | |
| event.pageX  event.pageY | 마우스의 현재 좌표를 알려준다.  $(document).mousemove(function(e){ $("span").text(event.pageX + "," + event.pageY);  }); | | |

#### **Zoom(화면 확대 / 축소 이벤트)**

| <script src="js/jquery-1.7.2.min.js"></script>  <script src="js/jquery.zoomooz.min.js"></script> |
| --- |

| **코드 값** | **비고** | | |
| --- | --- | --- | --- |
| 클래스 속성값 클릭시  zoomTo({targetsize:0.7})  = 타겟을 70%확대  e.stopPropagation();  이벤트 번짐방지? 느낌 | $(".targetZoom").click(function(e) {  $(this).zoomTo({targetsize:0.7});  e.stopPropagation();  }); | | |
| 원상복귀  zoom 속성 밖의 것을 클릭시 body값에 초점을 맞추며 바깥으로 빠져나옴 | $(window).click(function(e) {  $("body").zoomTo();  e.stopPropagation();  }); | | |

| // 클래스 속성 값이 zoom인 img를 클릭하면  $(".zoom").click(function(e){  // 화면의 줌의 비율을 0.7로 조절하면서 70% 크기까지 확대  $(this).zoomTo({targetsize:0.7});    // click 이벤트 흐름을 차단 이 부분이 없으면  window객체에 설정한 click에도 영향을 미친다.  e.stopPropagation();  });    // 확대된 요소의 바깥쪽을 클릭  $(window).click(function(e){  // body객체를 화면에 가득차게 보여줌으로써 원래 상태로 원복  $("body").zoomTo();  e.stopPropagation();  }); |
| --- |

#### **Tooltip 도움말 효과 (attr활용)**

| **css (220109 06. Tooltip 참고)** | | |
| --- | --- | --- |
| #tip {  position:absolute;  width:300px;  color:#FFF;  padding:5px;  display:none;    background:#05184D;  border-radius: 5px;  -moz-border-radius: 5px;  -webkit-border-radius: 5px;  } | | |

| **html** | | |
| --- | --- | --- |
| <div>  <p>  There are many variations of passages of Lorem Ipsum available,  but the majority have suffered alteration in some form, by injected humour,  or randomised words which don't look even slightly believable.  </p>  <p>  Lorem Ipsum is simply dummy text of the printing and typesetting industry.  Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,  when an unknown printer took a  <a href="#" title="There are many variations of passages of Lorem Ipsum available passages of Lorem Ipsum available passages of Lorem Ipsum available passages of ">  galley  </a>  of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.  </p>  <p>  Contrary to popular belief,  <a href="http://www.lipsum.com/" title="./images/site.jpg" class="img">  Lorem Ipsum  </a>  is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old.  </p>  </div> | | |

| **script** | | |
| --- | --- | --- |
| var tempTitle; //title속성의 내용이 저장되는 변수  var type; // class속성의 내용이 저장되는 변수  $(function(){  $("a").hover(  function(){  tempTitle = $(this).attr("title");  type = $(this).attr("class");    //.attr()은 요소(element)의 속성(attribute)의 값을 가져오거나 속성을 추가.  //console.log(tempTitle)로 검토가능  if(type == 'img'){  //클래스라는 속성이 있다면 tempTitle의 이미지경로를 img태그형식으로 생성  tempImg= "'<img src= '"+ tempTitle +"' width='100px' height:'100px' />"  }  $("body").append("<div id='tip'></div>");  if(type =="img"){  $("#tip").html(tempImg);  $("#tip").css({width:"100px"});  //css("width", "100px");  } else{  $("#tip").text(tempTitle);  }  var pageX = $(this).offset().left-20; //상대값  var pageY = $(this).offset().top-$("#tip").innerHeight();  $("#tip").css({left:pageX+"px", top:pageY+"px"}).fadeIn(500);  },  function(){  $("#tip").remove();  }  )  }); | | |

#### **키보드 이벤트**

| **코드 값** | **비고** | | |
| --- | --- | --- | --- |
| keypress | 꾹 누르고 있는 것  keydown과 비슷하지만 Enter Tab등 특수키에서는 반응하지 않음 | | |
| keydown | 누르기만 하면 되는것 | | |
| keyup | 키 입력 후 발생되는 이벤트 | | |

#### **window 이벤트**

| **코드 값** | **비고** | | |
| --- | --- | --- | --- |
| change |  | | |
| focus | 활성화 | | |
| blur | 활성화 안될때 | | |

#### **form 이벤트**

| **코드 값** | **비고** | | |
| --- | --- | --- | --- |
| submit | 폼 전송을 시도할 때 발생  폼 요소에서만 해당됨 | | |
| preventDefault  \*\*비슷한 함수로  stopPropagetion() 이 있다.  = 부모태그로 이벤트를 전파하는 것을 막아줌 | a 태그나 submit 태그는 누르게 되면 href로 이동하거나, 창이 새로고침하여 실행된다. 이를 막아주는 이벤트.  -1. a태그를 눌렀을 때 href 링크로 이동하지 않음.  $(".thumbs a").click(*function* (e) {  e.preventDefault(); *//a태그 링크 초기화*  });  -2. form안에 submit 역할을 하는 버튼을 눌렀어도 새로 실행하지 않게 하고 싶을 경우 (submit은 작동됨) | | |

#### **사용자정의 이벤트**

| **코드 값** | **비고** | | |
| --- | --- | --- | --- |
| on | 내가 직접 만들 것임 (사용자정의) | | |
|  |  | | |
|  |  | | |

#### **hide / show / toggle / hover**

| .hide()  없다  /숨김 | (selector).hide(speed, easing, callback);  선택한 요소를 사라지게 한다  .css({"display","none"}) 와 동일  $("input").eq(0).click(function(){  $(".box").hide();  }); |
| --- | --- |
| .show()  있다  /보임 | (selector).show(speed, easing, callback);  .선택한 요소를 나타나게 한다.  .css({"display","block"}) 와 동일  $("input").eq(1).click(function(){  $(".box").show(1000);  }); |
| .toggle() | (selector).toggle(speed, easing, callback);  선택한 요소에 .show() 메소드와 .hide() 메소드를 번갈아가며 적용  $("input").eq(2).click(function(){  $(".box").toggle(800);  }); |
| .hover() | hover은 show와 hide가 같이 쓰이므로 function도 2개가 되어야한다.  $("선택자").hover(function(){  $(this).find(".sub").show(1000);  }, function(){  $(".sub").hide();  }); |

| speed | 나타나거나 감추는 속도를 제어. 기본값: 400 (선택 사항)  - slow: 천천히. 200(밀리 초)  - fast: 빠르게. 600(밀리 초)  - 밀리 초: 1000/1(초)를 지정 |
| --- | --- |
| easing | 가속도를 제어. (선택 사항)  - swing: 시작과 끝이 느리게, 중간에는 빠르게 움직임. 기본값  - linear: 일정한 속도로 움직임 |
| complete | 콜백 함수는 작업이 완료된 후에 실행시킬 함수.  (선택 사항) |

#### **fadeIn / fadeOut**

#### 요소의 투명도를 조절하여 요소를 보이거나 숨길 수 있다

#### 

| .fadeOut | .fadeOut(speed, easing, callback)  서서히 투명해지면서 요소를 숨김.  (투명도 조절)  $("input").eq(1).click(function(){  $(".box").fadeOut();  }); |
| --- | --- |
| .fadeIn | .fadeIn(speed, easing, callback)  서서히 불투명해지면서 요소를 보이게 함.  (불투명도 조절)  $("input").eq(0).click(function(){  $(".box").fadeIn();  //애니메이션 기본시간(1000, 1초) 생략가능  }); |
| .fadeToggle | .fadeToggle(speed, easing, callback)  요소를 서서히 불투명하게 또는 투명하게 하여 감추거나 보여줌  speed, easing, callback 매개변수를 그대로 사용 가능. |
| .fadeTo | .fadeTo(speed, opacity, easing, callback)  요소의 투명도를 조절 ( opacity 필수 선택)  $("input").eq(2).click(function(){  $(".box").fadeTo(800, 0.3); //fadeTo(시간, 투명도)  }); |

| speed | 나타나거나 감추는 속도를 제어. 기본값: 400 (선택 사항)  - slow: 천천히. 200(밀리 초)  - fast: 빠르게. 600(밀리 초)  - 밀리 초: 1000/1(초)를 지정 |
| --- | --- |
| easing | 가속도를 제어. (선택 사항)  - swing: 시작과 끝이 느리게, 중간에는 빠르게 움직임. 기본값  - linear: 일정한 속도로 움직임 |
| complete | 콜백 함수는 작업이 완료된 후에 실행시킬 함수.  (선택 사항) |
| opacity | 0.0 ~ 1 사이의 수이며, 0 에 가까울수록 투명해짐 |

#### **slideUP / slideDown**

#### 

| .slideUp() | .slideUp(speed, easing, callback)  슬라이드 효과로 요소를 숨김.  (밑에서 위로 서서히 올라가면서 감춘다.)  $("input").eq(0).click(function(){  $(".box").slideUp(); //애니메이션 기본시간(1000, 1초) 생략가능  }); |
| --- | --- |
| .slideDown() | .slideDown(speed, easing, callback)  슬라이드 효과로 요소를 보이게 함  (위에서 아래로 서서히 내려가면서 보여준다.)  $("input").eq(1).click(function(){  $(".box").slideDown();  }); |
| .slideToggle() | .slideToggle(speed, easing, callback)  요소를 슬라이드로 보여주거나 감추는 메소드  speed, easing, callback 매개변수를 그대로 사용 가능. |

| speed | 나타나거나 감추는 속도를 제어. 기본값: 400 (선택 사항)  - slow: 천천히. 200(밀리 초)  - fast: 빠르게. 600(밀리 초)  - 밀리 초: 1000/1(초)를 지정 |
| --- | --- |
| easing | 가속도를 제어. (선택 사항)  - swing: 시작과 끝이 느리게, 중간에는 빠르게 움직임. 기본값  - linear: 일정한 속도로 움직임 |
| complete | 콜백 함수는 작업이 완료된 후에 실행시킬 함수.  (선택 사항) |

#### **animiate**

#### .animate()는 애니메이션 효과를 만듭니다.

.css({속성:속성값}).animate({{속성:속성값}, 애니메이션 시간})

.delay(대기시간), css(애니메이션을 초기값 스타일로 선언)

.stop() 이전에 작동하고 있던 애니메이션을 멈춘다.

$(".box").css({width:"200px"}).animate({width:"500px"}, 1000);

$(".box").delay(2000).css({width:"200px"}).animate({width:"500px"}, 1000);

| .animate( {properties} , [duration] , [easing] , [complete] ) |
| --- |

| properties | 움직임을 만들어 낼수 있는 CSS 속성들 (필수) |
| --- | --- |
| duration | 애니메이션 효과를 완료할 때까지 걸리는 속도. 기본값: 400  - slow: 천천히. 200(밀리 초)  - fast: 빠르게. 600(밀리 초)  - 밀리 초: 1000/1(초)를 지정 |
| easing | 가속도를 제어. (선택 사항)  - swing: 시작과 끝이 느리게, 중간에는 빠르게 움직임. 기본값  - linear: 일정한 속도로 움직임  \*\*http://easings.net |
| complete | 콜백 함수는 작업이 완료된 후에 실행시킬 함수.  (선택 사항) |

css("속성", "속성값") 스타일을 한번밖에 선언못함

여러개 선언가능한스타일 => css({속성:속성값, 속성:속성값})

속성값이 px단위가 붙을시 "속성값px"로 선언

css({속성:"속성값"})



▶ 이렇게 순차적 애니메이션도 설정할 수도 있다.

#### **setInterval**

#### 자동실행문. 일정 시간마다 반복 실행하는 함수

| .setInterval(function() { ... }, 지연시간); |
| --- |

i. 갤러리 형식의 뷰를 만들어 이미지를 정해진 시간 간격으로 바꾸어 적용하는 경우

ii. 일정 시간 간격으로 배너광고를 바꾸면서 보여줄 경우

iii. 일정 주기로 계속해서 서버와 통신이 필요한 경우

등에 주로 사용됨

| **자동실행 (마우스가 진입하면 멈춤)** |
| --- |
| $("#wrap").on({  mouseover: function(){  clearInterval(setIntervalId);  },  mouseout: function(){  timer();  }  }) |
| function timer(){  setIntervalId = setInterval(function(){  var n = current + 1;  if(n == visual.size()){ //size() => length() => 대상의 개수를 의미  n = 0  }  btn.removeClass("on");  btn.eq(n).addClass("on");  move(n)  }, 3000);  } |

| clearInterval() | setInterval()를 중지시키는 함수 |
| --- | --- |
| setTimeout() | 일정 시간 후에 특정 코드, 함수를 의도적으로 지연한 뒤 실행하고 싶을 때 사용하는 함수로 setTimeout()을 사용  setTimeout(function() { // Code here }, delay); |
| clearTimeout() | setTimeout();의 반대기능 |

#### **.text() .html() .val()**

| $("셀렉터").text() / $("셀렉터").html() / $("셀렉터").val() |
| --- |

| .text() | **셀렉터 하위에 있는 자식 태그의 문자열만 출력한다.**   | <button>1</button> <button>2</button> <button>3</button>  ▶ $("button").text() 의 결과는 1 2 3 | | --- | | <tag> ABC </tag>  ▶ $("tag").text() 의 결과는 ABC |   tag attribute의 value가 아니라(대표적으로 form 요소들) <tag> ABC </tag>에서 ABC 에 해당하는 value를 get한다.  참고 사이트에 의하면 The .text() method cannot be used on form inputs or scripts.  라고 적혀있다. ( form inputs 의 경우 val()를 말하는 거고 scripts 의 경우 html()을 가르킨다. )  *//이미지 클릭했을 때 아래 숨겨진 em의 글을 msg에 담아온다.*  *var* msg = $(*this*).next("em").text();  *//#caption 안에 mgs내용 삽입*  *var* a = $("#largeImg #caption").text(msg) |
| --- | --- | --- | --- |
| .html() | **셀렉터 하위에 있는 자식 태그를 전부(태그 문자열 포함) 출력한다.**  value가 아니라 html code(contents)자체를 get한다. class 네임이 여러 개 매칭되는 경우 제일 처음 매칭되는 class 안에 속하는 html code만 가져온다.  html(value) : htmlString 또는 함수(function(index, oldhtml) 이런 형태)로 넣을 수 있다. 이 역시 Set 하기위해 쓰인다.  class 네임이 여러 개 매칭되는 경우 매칭되는 모든 class 내부에 html code를 삽입.  만약 class 내부에 이미 어떤 code가 삽입되있는 경우, Set 할 html code로 완전히 Replace 해버린다. |
| .val() | **input 태그에 정의된 value 속성 값을 확인할 때 사용된다.**  Form 의 값을 받아오는데 쓰인다. (주로 input 이나 textarea 정도?)  - 주의해야할 점은 Form Element 이외의 값은 받아오질 못한다는 점.  ★ val(value) : value의 경우 string 또는 string의 배열(이 경우 value들의 matching을 잘 시켜야 오류를 피할 수 있다)  또는 함수(function(index, value) 이런 형태)로 넣을 수 있다.  이 함수 역시 Form Element의 Value 값을 Set할 때 주로 쓰인다. |

#### **.append() .prepend()**

| .append() | 선택된 요소의 마지막에 새로운 요소나 컨텐츠 추가  *//append를 이용해서 html을 추가한다.*  $("#largeImg").append("<div id='caption'></div>"); |
| --- | --- |
| .prepend() | 선택된 요소의 첫번째에 새로운 요소나 컨텐츠 추가  *//prepend*  $("ul").prepend("<li style='color:blue'>1-0</li>"); |
| .appendTo() | 선택된 요소를 해당 요소의 마지막에 추가 |
| .prependTo() | 선택된 요소를 해당 요소의 첫번째에 추가 |

#### **.after() .before()**

| .after() | 선택된 요소의 바로 뒤에 새로운 요소나 컨텐츠 추가 |
| --- | --- |
| .before() | 선택된 요소의 바로 앞에 새로운 요소나 컨텐츠 추가 |
| .insertBefore() | 선택된 요소를 해당 요소의 앞에 추가 |
| .insertAfter() | 선택된 요소를 해당 요소의 뒤에 추가 |

https://araikuma.tistory.com/609

#### **.attr()**

속성의 값을 가져올 때 사용. 요소(element)의 속성(attribute)의 값을 쉽게 제어할 수 있다.

제이쿼리에서 태그의 속성를 갖고 오거나 새로이 적용할때 attr()메소드를 사용

속성을 갖고오는 방법은 var 변수 = $(대상).attr(갖고오고자하는속성을 지정);

속성을 적용하는 방법은 $(대상을 지정하고).attr({속성객체:속성값저장한변수});

| - .attr(attributeName) // 속성값 가져오기    - .attr(attributeName, val) // 속성값 제어하기    - .attr({ 속성객체 : 저장한 변수 )} // 속성값 적용하기  *var img = $(this).attr("href"); //src가 아님!!!!!*  *//5-2. #largeImg 안에 img로 그림 변경*  $("#largeImg img").attr({ src: img }); |
| --- |

#### **Banner** ―――

#### 

| - fadeIn / fadeOut    제일 위에 있는 그림을 fadeOut 시킨 후, setInterval로 아래의 것을 무한 반복  1. 두번째 그림을 서서히 사라지게 한다  2. 이미 투명도가 0이 되어있는 제일 첫번째 그림을 따로 떼어낸다. (1번과 동시 진행)  3. 떼어낸 첫번째 그림을 제일 아래로 넣는다.  4. 제일 아래의 그림의 투명도를 100으로 돌려놓는다. |
| --- |
| - slideUp / slideDown    기본적인 구조는 위와 동일  이미지가 위로 올라가며 넘어간다. (height: 0 설정 사용도 가능) |
| - button left  가로로 이어 붙은 배너1~4. 버튼을 누르면 다음 배너가 나오도록. |
| - hover    배너1, 배너2, 배너3, 배너4를 hover 했을 때 스크린1에 띄우기 위한 식  스크린2 위에 스크린 1이 있다.  1. hover 했을 때 현재 hover 된 이미지를 temp1에 담겠다.  2. hover 했을 때 현재 스크린1에 띄워진 이미지를 temp2에 담겠다.  3. 스크린2에 temp2에 저장된 이미지를 담겠다. (이전 이미지를 띄우기 위함)  4. 스크린1에 temp1에 저장된 이미지를 담아서 서서히 투명하게 바꾸겠다. |

#### **kwicks 이벤트**

참고사이트:http://mottie.github.io/Kwicks/

| <script src="./js/jquery.kwicks-1.5.1.pack.js"></script> |
| --- |

| **코드 값** | **비고** | | |
| --- | --- | --- | --- |
| max|min  (필수) | 완전히 확장 된 축소 패널의 너비 또는 높이 (픽셀 단위)입니다. if isVertical가 true이면 크기가 지정된 차원이 높이를 참조.  그렇지 않으면 너비입니다.  max OR min을 지정해야하며 둘 다 지정할 수는 없습니다. | | |
| isVertica  (기본값 : false) | 패널을 세로로 배열할지 어떨지를 나타냅니다. | | |
| sticky  (기본값 : false) | 사실이라면 하나의 패널이 항상 확장됩니다. | | |
| defaultKwick  (기본값 : 0) | 처음에 확장 된 패널의 색인입니다  (고정 점이 true 인 경우에만 해당). | | |
| event  (기본값 : 'mouseover') | 확장 효과를 트리거하는 이벤트입니다.  다른 후보자는 'click'및 'dblclick'입니다. | | |
| duration  (기본값 : 200) | 각 애니메이션을 완료하는 데 필요한 밀리 초 수입니다. | | |
| easing  (기본값 : none) | 애니메이션을위한 커스텀 easing 함수 (easing plugin 필요). | | |
| spacing  (기본값 : 0) | 각 패널을 분리하는 폭 (픽셀 단위)입니다. | | |

| **script** | | | |
| --- | --- | --- | --- |
| $(function(){  $(".imgs").kwicks({  max: 1000,  spacing: 0,  isHorizontal: true,  behavior: "menu",  easing:"easeOutBounce"  });  }); | | | $(function(){  $(".menu").kwicks({  max: 450,  spacing: 3,  isHorizontal: true,  event: "click",  easing:"easeOutBounce"  });  }); |

#### **가로 슬라이딩**

| **html** | | |
| --- | --- | --- |
| <body>  <div id="slide">  <ul class="imgs">  <li><h3>이미지1</h3></li>  <li><h3>이미지2</h3></li>  <li><h3>이미지3</h3></li>  <li><h3>이미지4</h3></li>  <li><h3>이미지5</h3></li>  </ul>  <ul class="btns">  <li><a href="#">1</a></li>  <li><a href="#">2</a></li>  <li><a href="#">3</a></li>  <li><a href="#">4</a></li>  <li><a href="#">5</a></li>  </ul>  </div> | | |

| **제이쿼리** | | |
| --- | --- | --- |
| $(".btns li").click(function(){  var i = $(this).index();  var posx = i\*(-600) + "px"  $("ul.imgs").animate({left:posx}, 600);  }); | | |

#### **버튼식 반응형 슬라이딩**

| **버튼** | | |
| --- | --- | --- |
| btn.on({  click:function(){  var tg = $(this);  var i = tg.index();  //console.log(i)  btn.removeClass("on");  tg.addClass("on");  move(i);  }  }); | | |

| **클릭에 따른 정방향 역방향 강제조정** | | |
| --- | --- | --- |
| $("#prev").on({  click:function(){  var n = current - 1;  if(n == -1){ //size() => length() => 대상의 개수를 의미  n = 2  }  btn.removeClass("on");  btn.eq(n).addClass("on");  move2(n)  }  });  $("#next").on({  click:function(){  var n = current + 1;  if(n == visual.size()){  //size() => length() => 대상의 개수를 의미  n = 0  }  btn.removeClass("on");  btn.eq(n).addClass("on");  move(n)  }  }); | | |

| **이동하는 함수** | | |
| --- | --- | --- |
| //정방향 이동  function move(i){  var currentE1 = visual.eq(current); // 현재 보이는 슬라이드  var nextE1 = visual.eq(i); // 다음에 보이는 슬라이드  currentE1.css({left:"0%"}).animate({left:"-100%"});  nextE1.css({left:"100%"}).animate({left:"0%"});  current = i;  }  //역방향으로 이동하는 함수  function move2(i){  var currentE1 = visual.eq(current); // 현재 보이는 슬라이드  var nextE1 = visual.eq(i); // 다음에 보이는 슬라이드  currentE1.css({left:"0%"}).animate({left:"100%"});  nextE1.css({left:"-100%"}).animate({left:"0%"});  current = i;  } | | |

#### **오퍼시티**

| **html** | | |
| --- | --- | --- |
| <div id="brandVisual">  <ul>  <li class="visual\_0"><a href="#">배너이미지 1</a></li>  <li class="visual\_1"><a href="#">배너이미지 2</a></li>  <li class="visual\_2"><a href="#">배너이미지 3</a></li>  </ul>  </div> | | |

| **css** | | |
| --- | --- | --- |
| #brandVisual .visual\_0 {left:0px; z-index: 300; background:url("img/0.png") 50% 0 no-repeat;}  #brandVisual .visual\_1 {left: 0%; z-index: 200; background:url("img/1.png") 50% 0 no-repeat;}  #brandVisual .visual\_2 {left: 0%; z-index: 100; background:url("img/2.png") 50% 0 no-repeat;} | | |

| **제이쿼리** | | |
| --- | --- | --- |
| function move(i){  var currentE1 = visual.eq(current); // 현재 보이는 슬라이드  var nextE1 = visual.eq(i); // 다음에 보이는 슬라이드  currentE1.css({opacity:1}).animate({opacity:0});  nextE1.css({opacity:0}).animate({opacity:1});  current = i;  } | | |