j-Query

jQuery

<http://jquery.com/>

> jquery.js 다운받기

<!-- 로컬에서 js파일 찾기 -->

<script src="../script/jquery-3.6.4.js" type="text/javascript"></script>

<!-- 인터넷에서 js파일 찾기 -->

<script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>

<script src="http://code.jquery.com/jquery-latest.js"></script> (추천X)

<script type="text/javascript">

**$**(document).ready(function() {

//jQuery Script

//TODO

});

</script>

**ready(): 준비가 된 상태에서 실행하겠다**.

페이지 로딩이 끝난 후 실행 onload

$(function(){

// ready 뒤에 쓸 익명 함수 적기

}); -> 이런 식으로 생략 가능

$(function() {

$(”셀렉터”).명령어

.명령어;

$(”셀렉터”).명령어;

}); ->이런식으로 생략가능

jQuery 셀렉터

|  |  |  |
| --- | --- | --- |
| 명칭 | 서식 | 지정대상 |
| css에서 자주 쓰이는 셀렉터 | | |
| 태그 셀렉터 | $(“tag명”) | 해당 tag를 가지고 있는 대상 |
| Id 셀렉터 | $(“#id명”) | 해당 id를 가지고 있는 대상 |
| 클래스 셀렉터 | $(“.클래스명”) | 해당 class를 가지고 있는 대상 |
| 자손 셀렉터 | $(“태그1 태그2”) |  |
| 유니버설 셀렉터 | $(“\*”) |  |
| 그룹 셀렉터 | $(“셀렉터1, 셀렉터2”) |  |
| css2에서 자주 쓰이는 셀렉터 | | |
| 자식셀렉터 | $(“부모태그명 > 자식태그명”) |  |
| 인접셀렉터 | $(“태그1 + 태그2”) |  |
| first-child 유사 클래스 | $(“태그: first-child”) | 같은 부모 태그 중 첫 번째 자식 |
| Css3셀렉터 | | |
| 간접 셀렉터 | $(“태그1 ~ 태그2”) | 모든 형제 태그 |
| 부정유사 셀렉터 | $(“태그1: not(태그2)”) | Not 제외 |
| empty 셀렉터 | $(“태그:empty”) | 자식 태그와 노드가 없음 |
| nth-child 셀렉터 | $(“태그:nth-child(번호)”) | 번호대로 |
| last-child 셀렉터 | $(“태그:last-child”) | 마지막 자식 |
| only-child 셀렉터 | $(“태그:only-child”) | 유일한 자식 |
| css의 속성 셀렉터 | | |
| [attribute] | $(“[속성명]”) |  |
| [attribute=”value”] | $(“[속성명]=’값’”) | 해당 속성이 값을 가지고 있음 |
| [attribute! =”value”] | $(“태그명[속성명!=’값’]”) |  |
| [attribute^=”value”] | $(“[속성명^=’값’]”) |  |
| [attribute$=”value”] | $(“[속성명$=’값’]”) |  |
| [attribute\*=”value”] | $(“[속성명\*=’값]”) |  |
| [attributeFilter1][attributeFilter2] | $(“[태그셀렉터1][태그셀렉터2]”) | 두 가지 속성을 모두 가짐 |
| J-Query필터 | | |
| first 필터 | $(“태그:first”) |  |
| last 필터 | $(“태그:last”) |  |
| even 필터 | $(“태그:even”) |  |
| odd 필터 | $(“태그:odd”) |  |
| eq 필터 | $(“태그:eq(번호)”) |  |
| gt 필터 | $(“태그:gt(번호)”) |  |
| lt 필터 | $(“태그:lt(번호)”) |  |
| header 필터 | $(“:header”) |  |
| contain 필터 | $(“태그:contains(문자열)”) | 태그가 가지고 있는 문자열 내에서 검사(태그와 태그 사이의 txt). |
| has 필터 | $(“태그:has(태그2)”) | 어떤 태그를 포함하고 있는지 검사(자식 태그 검사) |
| parent 필터 | $(“태그:parent”) | 해당 태그가 부모일 때만 가져옴. |

jquery명령어

|  |  |
| --- | --- |
| 명령 | 의미 |
| text변경과 가져오기 | |
| text(…) | (텍스트 한정)괄호 안의 값으로 변경 |
| text() | (텍스트 한정)안의 텍스트 내용 취득 |
| HTML 변경과 취득 | |
| html(…) | (태그 포함)괄호 안의 값으로 변경 |
| html() | (태그 포함)안의 HTML 내용 취득 |
| HTML삽입 – 없던 내용 추가. | |
| **prepend(…)** | 해당 태그의 내용 앞쪽에 삽입된다. |
| append(…) |  |
| before(…) |  |
| **after(….)** | 해당 태그의 내용 뒤쪽에 삽입된다. |
| HTML이동 | |
| prependTo(…) | 해당하는 태그가 값 태그의 자식 태그가 된다.  셀렉터가 여러 개라면 해당 태그를 가진 모든 객체에 내용이 추가된다, |
| appendTo(…) |  |
| insertBefore(…) |  |
| insertAfter(…) | 해당하는 태그가 값 태그의 형제 태그가 된다. |
| 다른 태그로 묶음 | |
| wrap(…) |  |
| wrapAll(…) |  |
| wrapInner(…) |  |
| 태그 변경 | |
| replaceWith() |  |
| 태그 제거 | |
| remove() | 태그와 그 안의 내용을 전부 제거한다.  $(“대상태그 삭제대상태그”).remove(); |
| 속성값 변경과 취득 | |
| attr(… , …) | 해당 속성의 값을 괄호 안의 값으로 변경 |
| attr(…) | (이름만 입력) tag 속성이 가지고 있는 기본값으로 변경. |
| removeAttr(…) |  |
| class속성 추가와 제거 | |
| addClass(…) |  |
| removeClass(…) |  |
| css | |
| css(… , …) |  |
| css(…) |  |
| val() |  |
| val(…) |  |
| prop(… , …) |  |
| prop(…) |  |

jquery 이벤트

**!! 이벤트를 진행할 때 반드시 콜백함수를 사용해야 한다.**

$(function(){

$("셀렉터").이벤트(

function(){

// 이벤트가 발생한 시점에 실행할 문장

});

});

one()

$(function(){

$("셀렉터").one(“이벤트 발생 조건” , function(){

이벤트가 발생한 시점에 실행할 문장 => 다음부터는 실행되지 않음

});

});

이벤트 발생 조건 : click, dbclick, mouseup, mousedown, mouseover, mouseout, mousemove 등

$(function(){

$("**셀렉터**"). unbind(“**이벤트 발생 조건**” );

});

on(): 기본적으로 jQuery 객체를 리턴하지만 메서드의 매개변수로 추가적인 선택자를

최초 요소의 명시하면 하위를 재탐색하여 리턴한다.

이벤트가 발생한 노드가 추가선택자와 일치하면 핸들러가 작동하는 방식

$("셀렉터").on("이벤트 발생 조건", "추가선택자" ,function(){

// **이벤트가 발생한 시점**에 실행할 문장

});

/\*( $(document).on('click','a',function(){

alert('클릭!');

}); -\*/

|  |  |
| --- | --- |
| 명령 | 의미 |
| click() |  |
| dblclick() |  |
| mouseup() |  |
| mousedown() |  |
| mouseover() |  |
| mouseout() |  |
| mousemove() |  |
| one() |  |
| unbind() |  |
| on() |  |
| focus() | 해당 태그를 선택해서 커서가 주어진 상태 |
| blur() | 커서가 빠져 나왔을 때 |
| submit() |  |
| change() |  |

jquery 폼 필터

|  |  |  |
| --- | --- | --- |
| 태그 | 폼 필터 | css셀렉터 |
| input  textarea  select  button | :input | input  textarea  select  button |
| 한줄 텍스트 입력 폼 | :text | input[type=’text’] |
| 패스워드 입력 폼 | :password | input[type=’password’] |
| 라디오버튼 | :radio | input[type=’radio’] |
| 체크박스 | :checkbox | input[type=checkbox’’] |
| 전송버튼 | :submit | input[type=’submit’] |
| 이미지 버튼 | :image | input[type=’image’] |
| 리셋버튼 | :reset | input[type=’reset’] |
| 버튼 | :button | input[type=’button’] |
| 파일선택 폼 | :file | input[type=’file’] |
| 체크 상태인 태그 | **:checked** | - |
| 선택되어있는 태그 | **:selected** | - |

jquery 애니메이션 효과

$(셀렉터).명령어(“속도”[, 콜백함수(생략가능)] );

animate() :: 독자적인 에니메이션 효과 작성

$(셀렉터).animate(값을 변경하고 싶은 css속성:값, 속도, 움직임, 콜백함수 );

단 속성은 수식(px,em,%)으로 입력할 수 있는 것만 가능.

움직임 : swing(점점 빠르게) , linear(일정한 속도)

|  |  |
| --- | --- |
| 명령 | 의미 |
| show(…) | Display 상태를 지정한 시간 동안 일정 비율로 키워 가며 none에서 block으로 변경한다. |
| hide(…) | Display 상태를 지정한 시간 동안 일정 비율로 줄여 가며 block에서 none으로 변경한다. |
| toggle(..) | 현재 상태에 맞추어 show와 hide를 번갈아 진행한다. |
| sildeDown(…) | Display 상태를 정해진 시간 동안 Height를 일정 비율로 늘려 가며 block 상태로 변경한다. |
| sildeUp(…) | Display 상태를 정해진 시간 동안 Height를 일정 비율로 줄여 가며 none 상태로 변경한다. |
| slideToggle(…) | 현재 상태에 맞추어 slideDown과 slideUp 상태를 번갈아 진행한다. |
| fadeIn(…) | visibility 상태를 정해진 시간 동안 투명도를 일정 비율로 늘려 가며 보이게 만든다. |
| fadeOut(…) | visibility 상태를 정해진 시간 동안 투명도를 일정 비율로 줄여 가며 보이지 않게 만든다. |
| fadeTo(…) | Visibility 상태를 정해진 시간 동안 정해진 투명도로 변경한다. |
| animate(…) | 해당 객체에 애니메이션을 부여한다. |

filter()

::: 셀렉터로 가져온 태그에 대해 조건을 붙이는 명령어

ex) $(this).filter(".number")

지금 이벤트가 발생한 태그중에서 class속성에 number라는 값을 가지고 있는 태그

ajax

: javascript와 xml을 이용한 비동기 통신 처리 (현재는 jason을 더 많이 사용한다.)

java script로 웹페이지 전체를 읽어들이지 않고 서버와 데이터를 주고 받는다.

java script를 이용해 서버에서 데이터를 가져와 refresh없이 특정 부분만을 변경하는 것이 가능

-웹페이지에 텍스트를 삽입

$(셀렉터).load(“파일명”)

$(셀렉터).load(“파일명 태그명”)

-XML데이터를 가져와서 변환하여 표시

$.ajax({ url : ‘가져올 xml파일’

dataType : ‘xml’

success : function(data){

가져온 파일에 대한 처리

xml파일을 html로 변환

} });

plugin만들기

파일명 : jquery.기능명.js

$.fn.메서드명 = function() {

실행할 처리;

return this;

}

호출

$(셀렉터).메서드명();

each() : 여러 개의 태그집합에서 하나씩 꺼내서 처리(반복문)

꺼내온 태그는 $(this)로 다룬다.

$(:checkbox).each(function(){

$(this).명령어;

});