j-Query

jQuery

<http://jquery.com/>

> jquery.js 다운받기

<!-- 로컬에서 js파일 찾기 -->

<script src="../script/jquery-3.4.1.js" type="text/javascript"></script>

<!-- 인터넷에서 js파일 찾기 -->

<script src="https://code.jquery.com/jquery-3.4.1.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(){

//

}); -> 이런 식으로 생략 가능

$(function() {

$(”셀렉터”).명령어

.명령어;

$(”셀렉터”).명령어;

}); ->이런식으로 생략가능

jQuery 셀렉터

|  |  |  |
| --- | --- | --- |
| 명칭 | 서식 | 지정대상 |
| css에서 자주 쓰이는 셀렉터(“”또는 ‘’로 감싸줘야함) | | |
| 태그 셀렉터 1 | $(“tag명”) | 해당 태그 전부 |
| Id 셀렉터 2 | $(“#id명”) | 해당 아이디 전부 |
| 클래스 셀렉터 3 | $(“.클래스명”) | 해당 클래스 전부 |
| 자손 셀렉터 4 | $(“태그1 태그2”) | 태그1 태그2에 해당하는 것 |
| 유니버설 셀렉터 5 | $(“\*”) | 전체를 부름 |
| 그룹 셀렉터 6 | $(“셀렉터1, 셀렉터2”) | 해당 그룹의 해당 그룹 |
| css2에서 자주 쓰이는 셀렉터 | | |
| 자식셀렉터 7 | $(“부모태그명 > 자식태그명”) | 바로 아래 태그 |
| 인접셀렉터 8 | $(“태그1 + 태그2”) |  |
| first-child 유사 클래스 9 | $(“태그: first-child”) | 태그의 첫번째 자식 모두 |
| Css3셀렉터 | | |
| 간접 셀렉터 10 | $(“태그1 ~ 태그2”) | 밑에있는 형제태그모두! |
| 부정유사 셀렉터 11 | $(“태그1: not(태그2)”) | 1태그중에 2태그가 아닌애 |
| empty 셀렉터 12 | $(“태그:empty”) |  |
| nth-child 셀렉터 13 | $(“태그:nth-child(번호)”) | n번째 있는 값(1부터시작) |
| last-child 셀렉터 14 | $(“태그:last-child”) |  |
| only-child 셀렉터 15 | $(“태그:only-child”) |  |
| css의 속성 셀렉터 | | |
| [attribute] 16 | $(“[속성명]”) | Id란 속성을 가지고있는애모두 |
| [attribute=”value”] 17 | $(“[속성명]=’값’”) | 속성에 값을가지고있는애모두 |
| [attribute! =”value”] 18 | $(“태그명[속성명!=’값’]”) | 속성에 그값을 안가지고있는애 |
| [attribute^=”value”] 19 | $(“[속성명^=’값’]”) | F로 시작하는애 |
| [attribute$=”value”] 20 | $(“[속성명$=’값’]”) | F로 끝나는애 |
| [attribute\*=”value”] 21 | $(“[속성명\*=’값]”) | F를 포함하는애 |
| [attributeFilter1] 22  [attributeFilter2] | $(“[태그셀렉터1]  [태그셀렉터2]”) | 둘다 포함하는애 |
| J-Query필터 | | |
| first 필터 23 | $(“태그:first”) | 태그의 첫번쨰 |
| last 필터 23 | $(“태그:last”) | 태그의 마지막 |
| even 필터 24 | $(“태그:even”) | 0부터시작해서짝수번째 |
| odd 필터 24 | $(“태그:odd”) | 0부터 시작해서 홀수번째 |
| eq 필터 25 | $(“태그:eq(번호)”) | 2번에 있는애 |
| gt 필터 25 | $(“태그:gt(번호)”) | 2번보다 뒤에있는애들 |
| lt 필터 25 | $(“태그:lt(번호)”) | 2번보다 앞에있는애들 |
| header 필터 26 | $(“:header”) | h태그 모두 모아옴 |
| contain 필터 27 | $(“태그:contains(문자열)”) | 이러한 문자열을 포함한 태그 |
| has 필터 27 | $(“태그:has(태그2)”) | 태그2를 자식으로 가지고있는 |
| parent 필터 28 | $(“태그:parent”) | 자식태그가 있는애 무도 |

jquery명령어

|  |  |
| --- | --- |
| 명령 | 의미 |
| text변경과 가져오기 | |
| text(…) 챕터2 02~1 | p태그중에 아이디가 세컨드인애 문자열을 변경 |
| text() 2 | //문자열을 ()안에애가 가지고있는 값으로 변경 |
| HTML 변경과 취득 | |
| html(…) 3 | 문자열말고 효과를 ()안에내용으로변경 |
| html() 4 | 문자열말고 효과를 ()안에 애로 변경 |
| HTML삽입 | |
| prepend(…) 5 | 앞부분에 (“”) 내용 삽입 |
| append(…) 6 | 뒷부분에 (“”)내용 삽입 |
| before(…) 7 | 태그 앞에 삽입 |
| after(….) 8 | 태그 뒤에 삽입 |
| HTML이동 | |
| prependTo(…) 9 | 해당태그 앞부분에 “”태그 내용삽입 |
| appendTo(…) 10 | 해당태그 뒷부분에 “”태그 내용삽입 |
| insertBefore(…) 11 | H1태그를 p태그 앞으로 이동 |
| insertAfter(…) 12 | H1태그를 p태그 뒤로 이동 |
| 다른 태그로 묶음 | |
| wrap(…) 13 | 대상요소를 감쌈 |
| wrapAll(…) 14 | 대상요소를 감쌈  (사이에 다른 요소가 있으면 밑으로 뺌) |
| wrapInner(…) 15 | 대상요소를 감쌈  (사이에 다른 요소가 있으면 같이 감쌈) |
| 태그 변경 | |
| replaceWith() 16 | ( )안의 내용으로 값을 변경 |
| 태그 제거 | |
| remove() 17 | 해당 태그를 제거함 |
| 속성값 변경과 취득 | |
| attr(… , …) 18 | 속성의 이름을 앞에 적고 해당속성으로 뒤에애를 바꿔줌 |
| attr(…) 19 | 해당속성을 가져오겠다 |
| removeAttr(…) 20 | 선택한 요소의 특정 속성을 제거함 |
| class속성 추가와 제거 | |
| addClass(…) 21 | $("p").addClass("red");  p태그 붙으면 red로바꾸겠다 |
| removeClass(…) 22 | $("p").removeClass("red");  .red라는 클래스안의 속성 color를 지움 |
| css | |
| css(… , …) 23 | $("p").css("color","red");  p태그안의 속성 color에 red를 넣어줌 |
| css(…) 24 |  |
| val() 24 |  |
| val(…) |  |
| prop() |  |
| prop(… , …) |  |

jquery 이벤트

$(function(){

$("셀렉터").이벤트(function(){

이벤트가 발생한 시점에 실행할 문장

});

});

one()

$(function(){

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

이벤트가 발생한 시점에 실행할 문장

});

});

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

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 시점에 실행할 문장

});

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

alert('클릭!');

}); -\*/

$(function(){

$("셀렉터"). unbind(“이벤트 발생 조건” );

});

|  |  |
| --- | --- |
| 명령 | 의미 |
| click() 챕터2 03-1~6 | 클릭됐을 때 |
| dblclick() 7,8 | 더블클릭했을 때 |
| mouseup() 10 |  |
| mousedown() 9 |  |
| mouseover() 11 | 마우스를 올릴 때 |
| mouseout() 11 | 마우스를 뗄 때 |
| mousemove() 12 | 마우스가 움직일때마다 |
| one() 13 | 클릭이벤트가 한번만 실행됨 |
| unbind() 14 | unbind(a) a이벤트를 취소함 |
| on() 15,16 | 재탐색해서 새로 추가된 필드한테도 이벤트반영 |
| focus() 챕터2 04 3,4 | 커서에 넣으면 이벤트 발생 |
| blur() 5,9 | 포커스를 잃을 때 |
| submit() 7,8 | submit버튼이 눌렸을 때(onsubmit과 같은기능) |
| change() 6 | 이벤트 체인지 |

jquery 폼 필터

|  |  |  |  |
| --- | --- | --- | --- |
| 태그 | 폼 필터 | css셀렉터 | 쓰임새 |
| input  textarea  select  button | :input | input  textarea  select  button | input[name='name'  (input 태그의 name의 속성값이 name인 애) |
| 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’] |  |
| 체트 상태인 태그 | :checked | - |  |
| 선택되어있는 태그 | :selected | - |  |
|  | each() |  | 여러개의 태그 집합에서 하나씩 꺼내서 처리(반복문) |

jquery 애니메이션 효과

$(셀렉터).명령어([ “속도” ], [콜백함수] ); //속도, 콜백함수 생략가능

<http://localhost/JunWeb/day0625/06/building.jpg>

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(…) | 보여줌 |
| hide(…) | 숨김 |
| toggle(..) |  |
| sildeDown(…) | Display가 none인 상태에서 블록이나 인라인으로바꿈 h가 길어짐 |
| sildeUp(…) | H가 0으로되고 w도 0으로됨 |
| slideToggle(…) |  |
| fadeIn(…) | 투명->1 |
| fadeOut(…) | 1->투명 w,h도 0으로 display도 none |
| fadeTo(…) |  |
| animate(…) |  |

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

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();