

Jquery

- 자바스크립트
- 1. javaScript vs Jquery
- 2. Jquery사용법

1. JavaScript vs JQuery

-JavaScript

- 자바스크립트는 웹 페이지를 동적으로 표현해 주는 언어이다.
 - ex) 경고창 & 확인창 & drop다운 기능 & 탭 기능 등
-

-JQuery

- 자바스크립트를 더 간편하게 사용하게 해주는 자바스크립트 라이브러리 입니다.
- Jquery를 사용하면 순수 자바스크립트로 코딩하는 것보다 높은 생산성을 기대할 수 있습니다.

Jquery의 장점

- 태그를 선택자로 한번에 선택하는 강력한 방법을 제공
- 선택자로 선택한 태그를 제어하는 강력한 기능을 제공

2. Jquery의 사용방법

직접 사용하는 경우

1. <http://jquery.com/download/>에서 제이쿼리 소스코드를 다운로드 하는 방법
2. Download the compressed, production jQuery x.x.x 로 처리된 것을 우클릭 다른이름저장
3. 다운로드 후 프로젝트 안에 파일을 넣어준다

링크를 사용하는 방법

1. 사용할 페이지에서 아래와 같은 링크를 걸어 사용한다
2. `<script type="text/javascript" src="//code.jquery.com/jquery-3.4.0.min.js"></script>`
3. 또는
4. `<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.min.js "></script>`

3. 제이쿼리 함수

함수

	함수 명	함수 설명
엘리먼트 조작	each(Function)	선택된 엘리먼트가 다수일 경우에 each 함수를 사용하여 차례대로 엘리먼트를 선택한다.
에트리뷰트 조작	attr(name, value)	선택된 엘리먼트의 name 에트리뷰트의 값을 value로 설정
	attr(name)	선택된 엘리먼트의 name 에트리뷰트의 값을 얻어옴.
	attr(Attributes)	선택된 엘리먼트를 프로퍼티(ison)형태로 설정할 수 있음
	val()	엘리먼트의 value 에트리뷰트 값을 얻어옴. attr("value")와 동일함.
	val(content)	엘리먼트의 value 에트리뷰트 값을 content로 설정함.
에트리뷰트 제거	removeAttr(name)	해당 어트리뷰트의 값이 초기화 된다.
스타일 변경	addClass(names)	선택된 엘리먼트에 CSS Class를 적용함. 만약에 다수의 Class를 적용하려면 공백으로 구분하여 할당하면 됨.
	removeClass(names)	선택된 엘리먼트들을 제거함.
	toggleClass(names)	특정 Class를 적용하지 않은 상태면 적용하고, 적용한 상태면 제거함.
스타일 얻고 설정	css(name, value)	선택된 엘리먼트의 name 에트리뷰트 값을 value로 설정함.
	css(properties)	{ "name1": "value1", "name2": "value2", } 와 같은 형태로 설정함.
	css(name)	특정 name의 프로퍼티의 스타일 값을 얻을 수 있음.
	width(value)	선택된 엘리먼트의 width 값을 설정함.
	height(value)	선택된 엘리먼트의 height 값을 설정함.
	width()	선택된 엘리먼트의 width 값을 얻어옴.
	height()	선택된 엘리먼트의 height 값을 얻어옴.
	offset()	선택된 엘리먼트의 left 값과 top 값을 E.offset().left, E.offset().top과 같은 방법으로 얻을 수 있음.
엘리먼트 내용 설정	html()	선택된 엘리먼트 태그 내용을 얻을 수 있음.
	html(content)	선택된 엘리먼트의 태그 내용을 content로 설정함.
	text()	선택된 엘리먼트의 태그 내용 중 텍스트 값만 얻을 수 있음.
	text(content)	선택된 엘리먼트의 태그 내용을 content로 설정함.
엘리먼트 복사&이동	append(content)	선택된 엘리먼트의 내용 마지막에 새로운 content를 추가함.
	appendTo(target)	선택된 엘리먼트가 단일이면 target으로 이동시키고 다수라면 복사됨.
	prepend(content)	append와 달리 앞으로 추가함
	prependTo(target)	appendTo와 달리 앞으로 복사 또는 이동함
	before(), insertBefore()	엘리먼트를 대상의 첫 번째 자식으로 삽입하는 대신에 대상의 바로 앞 형제로 추가함.
	after(), insertAfter()	엘리먼트를 대상의 마지막 자식으로 삽입하는 대신에 대상의 바로 뒤 형제로 추가함
엘리먼트 감싸기	wrap(wrapper)	매개 변수로는 String과 엘리먼트 타입이 올 수 있으며, "<div class='hello'></div>"형태로 매개 변수 값을 넘기면 됨.
	wrapAll(wrapper)	선택된 모든 엘리먼트를 wrapper로 감쌈
엘리먼트 제거	remove()	페이지 DOM에서 확장 집합의 모든 엘리먼트를 삭제함.
	empty()	일치하는 집합의 모든 엘리먼트의 Content를 제거함.
엘리먼트 복사	clone(copyHandlers)	일반적으로 엘리먼트를 복사한 확장 집합을 만들면 이들은 DOM 어딘가에 덧붙일 수 있음

주된사용방법은

\$("선택자").함수 이며

제이쿼리는 단순히 함수 암기 싸움입니다

1. 제이쿼리 선택자

\$(선택자)

\$("#태그 아이디")
 \$(".태그 클래스")
 \$("요소[속성=값]")
 \$(this)

예시
\$("a[href=#]")

2. 값 확인 val()

\$(선택자).val()

```

<input type="text" id="apple" value="홍길자">
<input type="text" id="melon" value="홍길숙">
<input type="text" id="grape" value="홍길동">

<button onclick="check()">확인</button>

<script type="text/javascript" src="//code.jquery.com/jquery-3.4.0.min.js"></script>
<script type="text/javascript">

```

```
function check() {
```

```
var a = $("#melon").val();
```

```
alert(a);
```

```
}
```

```
</script>
```

스크립트 함수

태그 속성제어

자주 사용하는 속성 제어 (필수)함수

val() – 태그의 값을 확인

val(변경값) – 태그의 값 변경

attr(속성, 변경값) – 태그의 내부 속성을 변경

attr(속성) – 태그 속성 여부 확인

removeAttr() – 특정 속성 제거

css() – css속성 여부 확인

css(속성, 변경값) – css속성 변경

html() – html 형태로 값을 얻음

html(값) – html 형태로 문자 변경

text(값) – 순수문자 형태로 삽입

addClass(값) – 클래스 추가

removeClass(값) – 클래스 삭제

toggleClass(값) – 토글 클래스

focus() – 태그의 마우스 커서 위치시킴

submit() – 태그 서브밋

부트스트랩전용 함수

modal("show") – 모달 띄우기

modal("hide") – 모달 숨기기

3. 제이쿼리 문서 객체 조작 attr()

`$(선택자).attr()`

`$("#선택자").attr("src")`

- 특정 값을 확인

`$("#선택자").attr("src", 변경값)`

- 특정 값을 변경

`$("#선택자").attr({.....})`

- 여러값을 변경

자주 사용하는 속성 제어 (필수)함수

val() – 태그의 값을 확인

val(변경값) – 태그의 값 변경

attr(속성, 변경값) – 태그의 내부 속성을 변경

attr(속성) – 태그 속성 여부 확인

removeAttr() – 특정 속성 제거

css() – css속성 여부 확인

css(속성, 변경값) – css속성 변경

html() – html 형태로 값을 얻음

html(값) – html 형태로 문자 변경

text(값) – 순수문자 형태로 삽입

addClass(값) – 클래스 추가

removeClass(값) – 클래스 삭제

toggleClass(값) – 토글 클래스

focus() – 태그의 마우스 커서 위치시킴

submit() – 태그 서브밋

부트스트랩전용 함수

modal("show") – 모달 띄우기

modal("hide") – 모달 숨기기

3. 제이쿼리 문서 객체 조작 css()

\$(선택자).css()

\$("선택자").css("backgroundColor")

- 특정 값을 확인

\$("선택자").css("backgroundColor", 변경값)

- 특정 값을 변경

\$("선택자").css({.....})

- 여러값을 변경

자주 사용하는 속성 제어 (필수)함수

val() – 태그의 값을 확인

val(변경값) – 태그의 값 변경

attr(속성, 변경값) – 태그의 내부 속성을 변경

attr(속성) – 태그 속성 여부 확인

removeAttr() – 특정 속성 제거

css() – css속성 여부 확인

css(속성, 변경값) – css속성 변경

html() – html 형태로 값을 얻음

html(값) – html 형태로 문자 변경

text(값) – 순수문자 형태로 삽입

addClass(값) – 클래스 추가

removeClass(값) – 클래스 삭제

toggleClass(값) – 토글 클래스

focus() – 태그의 마우스 커서 위치시킴

submit() – 태그 서브밋

부트스트랩전용 함수

modal("show") – 모달 띄우기

modal("hide") – 모달 숨기기

3. 제이쿼리 문자 조작 html()

\$(선택자).html()

\$("#선택자").html()

- 선택자의 값을 확인

\$("#선택자").html("홍길동")

- 선택자의 값을 변경

\$("#선택자").text("<h2>텍스트</h2>")

- 선택자의 값을 순수텍스트로 변경

자주 사용하는 속성 제어 (필수)함수

val() – 태그의 값을 확인

val(변경값) – 태그의 값 변경

attr(속성, 변경값) – 태그의 내부 속성을 변경

attr(속성) – 태그 속성 여부 확인

removeAttr() – 특정 속성 제거

css() – css속성 여부 확인

css(속성, 변경값) – css속성 변경

html() – html 형태로 값을 얻음

html(값) – html 형태로 문자 변경

text(값) – 순수문자 형태로 삽입

addClass(값) – 클래스 추가

removeClass(값) – 클래스 삭제

toggleClass(값) – 토글 클래스

focus() – 태그의 마우스 커서 위치시킴

submit() – 태그 서브밋

부트스트랩전용 함수

modal("show") – 모달 띄우기

modal("hide") – 모달 숨기기

3. 제이쿼리 클래스 조작 addClass(), removeClass(), toggleClass()

\$(선택자).클래스조작()

\$("#선택자").addClass("aaa")

- 선택자에 클래스 추가

\$("#선택자").removeClass("aaa")

- 선택자에 클래스 삭제

\$("#선택자").toggleClass("aaa")

- 선택자에 aaa를 토글형식으로 전환

자주 사용하는 속성 제어 (필수)함수

val() – 태그의 값을 확인

val(변경값) – 태그의 값 변경

attr(속성, 변경값) – 태그의 내부 속성을 변경

attr(속성) – 태그 속성 여부 확인

removeAttr() – 특정 속성 제거

css() – css속성 여부 확인

css(속성, 변경값) – css속성 변경

html() – html 형태로 값을 얻음

html(값) – html 형태로 문자 변경

text(값) – 순수문자 형태로 삽입

addClass(값) – 클래스 추가

removeClass(값) – 클래스 삭제

toggleClass(값) – 토글 클래스

focus() – 태그의 마우스 커서 위치시킴

submit() – 태그 서브밋

부트스트랩전용 함수

modal("show") – 모달 띄우기

modal("hide") – 모달 숨기기

4. 제이쿼리 이웃, 형제 노드 조작

`$(선택자).노드조작()`

`$("#선택자").closest("td")`

- 선택자의 최근접 단일부모 탐색 (td탐색)

`$("#선택자").children()`

- 선택자의 자식요소 탐색

`$("#선택자").first()`

- 선택자의 첫번째 자식요소

`$("#선택자").find("td")`

- 선택자의 특정 자식요소 탐색 (td탐색)

`$("#선택자").next()`

- 선택자의 다음 형제노드

`$("#선택자").prev()`

- 선택자의 이전 형제노드

이웃, 형제 노드 조작

`closest(선택자)` - 근접 단일부모요소 탐색

`children()` - 자식요소 탐색

`first()` - 자식요소중 첫번째

`find(선택자)` - 특정 자식 요소 탐색

`next()` - 다음형제노드

`prev()` - 이전형제노드

3. 반드시 알아야 하는 제이쿼리 이벤트 함수

페이지 로딩 시 스크립트를 실행

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

```
});
```

```
</body>
```

```
<input type="text" id="apple" value="홍길자">
<input type="text" id="melon" value="홍길숙">
<input type="text" id="grape" value="홍길동">
```

```
<button onclick="check()">확인</button>
```

```
<script type="text/javascript" src="//code.jquery.com/jquery-3.4.0.min.js"></script>
<script type="text/javascript">
```

```
$(document).ready(function(){
    alert('페이지 진입하자 마자 실행!');
});
```

```
</script>
```

```
</body>
```



`document.ready`는 `window.onload`와 상당히 유사합니다.
단,
JS의 `window.onload`는 단 한번 사용 가능하지만,
Jquery의 `document.ready`는 여러 번 사용 가능합니다.

3. 반드시 알아야 하는 제이쿼리 이벤트 함수

클릭시 소괄호 안에 익명 함수 실행

```
$("#선택자").click(function(){  
}
```



```
$("#replyRegist").click(function(){  
    add();  
})
```

이벤트 위임함수

이벤트 위임이란, 부모태그에 이벤트를 등록하고, 자식태그에 이벤트를 넘겨서 사용하는 방식입니다

```
$("#선택자 ").on("이벤트", "선택자", function() {  
  
}
```



```
<div id="replyList">  
    <a href='#' >  
    <a href='#' >  
    <a href='#' >
```

```
</div>
```

```
<script>
```

```
$("#replyList").on("click", "a", function(event) {  
    event.preventDefault(); //막는다
```

```
</script>
```

3. 반드시 알아야 하는 제이쿼리 이벤트 함수

비동기 통신 Ajax

ajax란?

- 자바스크립트 이용해서 비동기식으로 서버와 통신
- 비동기란 여러 일이 동시에 발생한다는 뜻.
- 쉽게 생각해서 **화면이동 없이 서버와의 통신을 처리함**
- 서버와 통신할때 json이라는 형식을 사용한다

Ajax사용 준비사항 (json라이브러리 다운)

```
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
  <version>2.9.8</version>
</dependency>
```

ajax문법

```
$.ajax({
  속성
})
```



주요 속성

data : 서버에 전송할 데이터 key/value 형식
dataType : 서버가 리턴하는 데이터 타입
type: 서버로 전송하는 데이터타입(Post,Get)
url: 전송할 URL
success: 통신에 성공했을 때 호출할 스크립트 함수
error: 통신 실패시 호출할 스크립트 함수

3. 반드시 알아야 하는 제이쿼리 이벤트 함수

비동기 통신 Ajax

```
<body>
<input type="text" id="apple" value="홍길자">
<input type="text" id="melon" value="홍길숙">
<input type="text" id="grape" value="홍길동">

<button onclick="check()">확인</button>

<script type="text/javascript" src="//code.jquery.com/jquery-3.4.0.min.js"></script>
<script type="text/javascript">
    function check() {
        $.ajax({
            type : "post",
            url : "checkId",
            data : {"id" : "kim"},
            success : function(result) {
                alert("비동기 성공!");
            }
        })
    }
</script>

</body>
```



3. 반드시 알아야 하는 제이쿼리 이벤트 함수

비동기 통신 \$.getJSON

```
$.getJSON(  
    "../reply/getReply/" + bno,  
    function(data) { // 응답 함수  
    }  
)
```

요청 보내는 주소
제이슨형식 응답받을 익명함수