

Class . 제이쿼리

01

▶ 제이쿼리(jQuery)란 :

- 장점: 자바스크립트의 DOM호환성에 대한 문제를 해결함.
- 단점: 필요없는 소스까지 같이 불러와 속도가 느림.

▶ 버전 :

- 1x버전: 구형 브라우저에 적합. 익스 6~8 버전
- 우리나라 공공기관은 아직 구형브라우저가 이용되는 곳이 많아 1x버전대도 많이 쓰임.
- 2x버전: 익스 9이상 지원되며, 1x의 간소화 버전으로 기능이 제외된 버전임.
- 3x버전: 익스 8이상 지원되는 최신 버전임.
- uncompressed: 압축하지 않은 버전으로 소스 확인 가능
- minified: 압축버전으로 위의 버전보다 가벼움. 소스 확인 불가
- slim: 경량버전으로 아작스, 애니메이션등의 기능이 제거됨. 소스 확인 가능
- slim minified: 경량버전의 압축 버전임. 소스 확인 불가
- jquery.js : 소스코드에 들여쓰기, 줄 바꿈, 주석까지 정리된 비압축 파일.
- jquery.min.js : 소스코드에 들여쓰기, 줄 바꿈, 주석이 무시된 압축 파일.
- 버전이 다른 제이쿼리를 여러 개 적용할 경우 충돌이 되어 원하는 결과와 다른 결과가 나올 수 있다. 따라서 되도록 버전은 한 버전만을 연결하여 이용할 것.
- jquery.js나 jquery.min.js 등으로 기술된 파일은 한 버전만 이용할 것.
- 단, 기존 버전에서 삭제된 이전 버전의 기능을 이용할 때 다운 받는 Migrate plugin버전을 다운 받는 경우는 같이 사용 가능하다.

▶ 형식 :

\$기호 => jquery를 지칭하는 예약어(키워드)임.

jquery("선택자") 또는 \$("선택자")로 표기

▶ 직접 선택자

- `$("*") or jquery("*")` : 전체 선택자 , 모든 요소를 선택.
- `("#아이디명") or jquery("#아이디명")` : 해당 id속성값을 가진 요소를 선택.
- `(“.클래스명”) or jquery(“.클래스명”)` : 해당 class속성값을 가진 요소를 선택.
- `(“요소명”) or jquery(“요소명”)` : 해당 요소(=엘리먼트=태그)를 선택
- `(“선택자1, 선택자2, ... , 선택자n”) or jquery(“선택자1, 선택자2, ... , 선택자n”)` : 해당 나열된 선택자에 해당되는 요소들을 선택.
- `(“요소명.클래스명”) 또는 $("요소명#아이디명”)` : 해당 요소 중 해당 class속성값을 가진 요소를 선택.
또는 해당 요소 중 해당 id속성값을 가진 요소를 선택.

▶ 인접 선택자

- `($("선택자").parent()` : 부모 선택자, 해당 요소의 부모 요소만 선택함.
- `($("선택자").parents()` : 상위 선택자 , 해당 요소의 조상 요소(부모 포함)를 선택함.
- `($("선택자").closest("div")` : 해당 요소의 부모 요소들 중 가장 가까운 div요소를 선택함.
- `($("요소 선택자"))` : 후손 선택자. (자식 포함) 한 칸 공백으로 표현.
- `($("요소 > 선택자"))` : 자식 선택자. 자식 요소만 선택함.

형 선택자 (선택자 제외) :

- `($("선택자").children()` : 선택한 요소의 모든 자식 요소를 선택함.
- `($("선택자").prev()` : 바로 이전 요소 선택자. 바로 앞 요소 선택자.
- `($("선택자").prevAll()` : 바로 이전 요소들 모두 선택. 바로 앞 요소들 모두 선택.
- `($("선택자").prevUtil("요소명"))` : 선택자 요소부터 지정한 요소 바로 전까지 그 안에 있는 요소 모두 선택.

▶ 인접 선택자

동생 선택자 (선택자 제외) :

- \$("선택자").next() or \$("선택자 + 다음 선택자") : 선택한 요소의 바로 다음 요소를 선택함.
- \$("선택자").nextAll() : 바로 다음 요소들 모두 선택. 바로 다음 요소들 모두 선택.
- \$("선택자").nextUntil("요소명") : 선택자 요소부터 지정한 요소 이전까지 안에 있는 요소 모두 선택.

형제 선택자 (선택자 제외) :

- \$("선택자").siblings() : 선택한 요소의 형제 요소 전체를 선택함.

▶ 탐색 선택자

- \$("요소명:first") or \$("요소명").first() : 해당 요소명을 가진 전체 요소들 중 가장 첫 번째 요소만 선택.
- \$("요소명:last") or \$("요소명").last() : 해당 요소명을 가진 전체 요소들 중 가장 마지막 번째 요소만 선택.
- \$("요소명").eq(인덱스번호) : 해당 요소들 중 인덱스번호(0부터 시작)에 해당되는 요소만 선택
- \$("요소명").lt(인덱스번호) : 해당 요소들 중 인덱스번호보다 작은 요소들만 선택
- \$("요소명").gt(인덱스번호) : 해당 요소들 중 인덱스번호보다 큰 요소들만 선택

02

▶ 이벤트 등록 함수란:

- 이벤트란: 사용자가 실행하는 모든 행위(클릭, 마우스 오버, 키보드 입력 등등)를 말함
- 이벤트 핸들러란 : 이벤트가 발생할 때 실행되는 소스 코드를 말함.
- 이벤트 등록 함수란: 사용자가 지정한 요소에서 어떠한 이벤트가 발생했을 때 저장된 코드를 실행 할 수 있도록 이벤트와 이벤트 핸들러를 이용하여 만든 함수를 말함.
- 이벤트를 하나만 등록할 수 있는 단독 이벤트 등록 함수와 2개 이상 등록할 수 있는 그룹 이벤트 등록 함수가 있다.

▶ 이벤트 등록 함수의 종류

1. 로딩 이벤트 :

- load() : 선택한 이미지 또는 프레임 요소에 연동된 소스의 로딩이 완료된 후 이벤트 발생
- ready() : 지정한 html문서 객체의 로딩이 완료된 후 이벤트 발생
- error() : 이벤트 대상 요소에서 오류가 발생하면 이벤트 발생

2. 키보드 이벤트:

- keypress() : 선택한 요소에서 키보드를 눌렀을 때 이벤트가 발생. 문자키를 제외한 키 코드 값을 반환함.
- keydown() : 선택한 요소에서 키보드를 눌렀을 때 이벤트가 발생. 모든 키 코드 값을 반환함.
- keyup() : 선택한 요소에서 키보드를 뗐을 때 이벤트가 발생함.

3. 포커스 이벤트:

- focus() : 선택한 요소에 포커스가 생성되었을 때 이벤트가 발생하거나 선택한 요소에 강제로 포커스를 생성한다.
- change() : 이벤트 대상의 입력 요소의 값이 변경되고, 포커스가 이동하면 이벤트 발생. 또는 강제로 변경 이벤트 발생.

▶ 이벤트 등록 함수의 종류

4. 마우스 이벤트 :

- click() : 선택한 요소를 클릭했을때 이벤트 발생
- dblclick() : 선택한 요소를 더블 클릭했을때 이벤트 발생
- hover() : 선택한 요소안에 마우스 포인터가 들어왔을 때, 벗어났을 때 이벤트 발생
- mousemove() : 선택한 요소안에서 마우스 포인터가 움직였을 때 이벤트 발생

▶ 단독 이벤트 등록 함수의 형식(Syntax)

```
$(“선택자”).이벤트등록함수(콜백함수);
```

예시:

```
$(“button”).click( function(){  
    alert(“버튼을 클릭하였습니다.”);  
});
```

▶ 그룹 이벤트 등록 함수의 형식(Syntax)

1. \$(“선택자”).on(“이벤트1 이벤트2 이벤트3 ... 이벤트n” , 콜백함수);
2. \$(“선택자”).on({“이벤트1 이벤트2 이벤트3 ... 이벤트n” : 콜백함수 });
3. \$(“선택자”).on({
 “이벤트1” : 콜백함수,
 “이벤트2” : 콜백함수,
});

▶ 그룹 이벤트 등록 함수의 형식(Syntax)

예시:

```
$("input[name=user_id]").on("keyup focus", function(){
    console.log("키보드를 눌렀다 떼었을 때 포커스가 생성이 되었을 때 이벤트 발생");
});

$("input[name=user_id]").on({
    "keyup" : function(){
        console.log("키보드를 눌렀다 떼었을 때 이벤트 발생");
    },
    "focus" : function(){
        console.log("포커스가 생성이 되었을 때 이벤트 발생");
    }
});
```