



jQuery



jQuery

자바스크립트 언어를 간편하게 사용할 수 있도록 단순화시킨 오픈 소스 기반의 **자바스크립트 라이브러리**

라이브러리란?

자주 사용하는 기능들을 모아두었다가 필요할 때 가져다 쓸 수 있도록 한 자원들의 묶음

jQuery 장점

- 주요 웹 브라우저의 구버전을 포함한 대부분의 브라우저에서 지원 가능
- HTML DOM를 손쉽게 조작 가능하며, CSS 스타일도 간단히 적용 가능
- 같은 동작을 하더라도 더욱 짧게 구현 가능
- 오픈 라이선스를 적용해 누구나 자유롭게 이용 가능

jQuery의 현재?



▲ 구글 트렌드(Google Trends)로 본 제이쿼리(jQuery) 키워드 검색량 변동 추이(2015~2020년)

jQuery 기초

```
$ ( 선택자 ) . 동작함수 ( ) ;
```

jQuery 메소드

값 가져오기 & 설정하기

- `$(선택자).val();`
 - Input 과 같은 value 값을 가져올 수 있다.

```
var value = $("input").val();
```

- `$(선택자).val("설정할 값");`
 - Input 과 같은 곳에 value 값을 설정할 수 있다.

```
$("input").val("설정완료!");
```

값 가져오기 & 설정하기

- `$(선택자).val();`
 - Input 과 같은 value 값을 가져올 수 있다.

```
var value = $("input").val();
```

→ `let var = document.querySelector("input").value;`

- `$(선택자).val("설정할 값");`
 - Input 과 같은 곳에 value 값을 설정할 수 있다.

```
$("input").val("설정완료!");
```

→ `document.querySelector("input").value = "원하는 값";`

Text 변경하기

- `$(선택자).text(“글자”);`

```
$("span").text("jQuery로 바꿨습니다.");
```

```
var span = document.querySelector("span");  
span.innerText = "js로 바꿨습니다."
```

HTML 변경하기

- \$(선택자).html(html);

```
$("#p").html("<h1>jQuery</h1>");
```

```
var p = document.querySelector("p");  
p.innerHTML = "<h3>javascript</h3>"
```

속성 변경하기

- `$(선택자).attr(“속성”, “속성값”);`

```
$("#li2").attr("style", "color: red");
```

→ `document.querySelector(“선택자”).setAttribute(“속성”, “값”);`

CSS 변경하기

- `$(선택자).css(“속성” , “속성값”);`

```
$("span").css("font-size", "28px");  
$("span").css("color", "blue");
```

```
var span = document.querySelector("span");  
span.style = "font-size: 10px; color: red";
```

요소 추가

- `$(선택자).append(요소);`
 - 선택된 요소의 자식 요소 제일 마지막 요소로 추가
- `$(선택자).prepend(요소);`
 - 선택된 요소의 자식 요소 중 첫번째로 추가
- `$(선택자).before(요소);`
 - 선택된 요소의 이전 형제 요소로 추가
- `$(선택자).after(요소);`
 - 선택된 요소의 다음 형제 요소로 추가

요소 삭제

- `$(선택자).remove();`
 - 선택한 요소를 DOM 트리에서 삭제
- `$(선택자).empty();`
 - 선택한 요소의 자식 요소를 모두 삭제
 - 선택된 요소는 남아 있고 그 안이 다 비워진다
 - JS 에는 없는 기능
 - 단, `(요소).innerHTML = ""` / `(요소).textContent = ""`; 로 동일하게 처리 가능

요소 탐색

- `$(선택자).parent()`
 - 바로 위에 존재하는 하나의 부모 요소 선택
 - JS에서는 `(요소).parentNode`
- `$(선택자).parents()`
 - 바로 위 부모뿐만 아니라 부모의 부모, 모든 조상을 선택
 - JS에서는 없음, `closest("선택자")` 로 가까운 조상 찾기는 가능
- `$(선택자).children()`
 - 선택된 요소의 자식 요소 모두 선택
 - JS에서는 `(요소).childNodes` → 비 노드 요소 포함(
 - JS에서는 `(요소).children` → 비 노드 요소 제외

요소 탐색

- `$(선택자).next()`
 - 선택된 요소 바로 다음에 위치한 형제 요소 선택
 - JS에서는 (요소).nextSibling
- `$(선택자).prev()`
 - 선택된 요소 바로 이전에 위치한 형제 요소 선택
 - JS에서는 (요소).previousSibling

클래스 추가, 삭제, 확인

- `$(선택자).addClass("클래스명")`
 - 선택된 요소에 클래스 추가
- `$(선택자).removeClass("클래스명")`
 - 선택된 요소에 클래스 삭제
- `$(선택자).hasClass("클래스명")`
 - 선택된 요소에 클래스가 있는지 여부 확인
 - Return 값은 true 와 false 로 나온다.

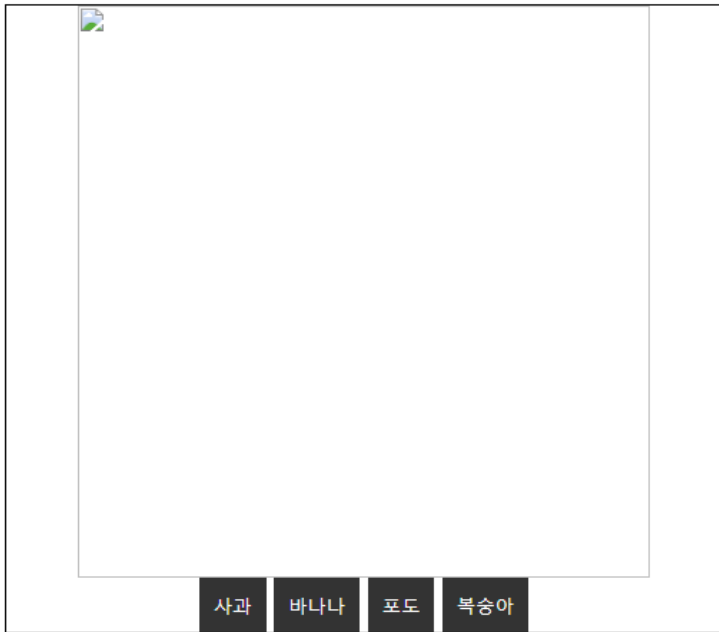
jQuery 사용

1. 파일 다운로드
2. CDN(Content Delivery Network) 사용

<https://releases.jquery.com/>



실습 이미지 바꾸기



JQuery를 이용해 과일 버튼을 누르면
해당하는 과일 이미지 나오게 하기