사용준비

2022년 10월 8일 토요일 오후 6:10

1. 다운 로드 받은 파일 활용
   코드를 저장하고 제이쿼리 사용 준비

JQuery.min.js란 파일이 있을 때

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JQuery Basic</title>
<script src = "JQuery.min.js"></script>
<script>
  // 여기에 코드 입력
</script>
</head>
<body>
</body>
</html>
```

화면 캡처: 2022-10-16 오후 4:06

2. CDN 사용
   - 사용자에게 간편하게 콘텐츠를 제공하는 방식

기타 CDN

다음 CDN은 또한 jQuery 릴리스의 압축 및 비압축 버전을 호스팅합니다. jQuery 1.9부터 소스맵 파일을 호스트할 수도 있습니다 . 사이트의 문서를 확인하십시오.

jQuery 릴리스와 해당 릴리스 사이에 지연이 있을 수 있습니다. 블로그 게시물이 공개되는 동시에 파일을 받게 됩니다. 베타 및 릴리스 후보는 이러한 CDN에서 호스팅되지 않습니다.

- 구글 CDN
- 마이크로소프트 CDN
- CDNJS CDN
- jsDelivr CDN

제이쿼리

3.x 스니펫:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
```

2.x 스니펫:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
```

1.x 스니펫:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js">
</script>
```

화면 캡처: 2022-10-16 오후 4:10

위 주소 활용해서 태그 만들어 넣으면 된다

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JQuery Basic</title>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
</head>
<body>
</body>
</html>
```

화면 캡처: 2022-10-16 오후 4:12

JQuery 객체

2022년 10월 16일 일요일 오후 4:12

- 제이쿼리 라이브러리는 $ 함수를 활용한다
- 보통 $ 함수의 매개변수에는 문서 객체, css 형식, HTML형식의 문자열 삽입
- [$(매개_변수).메소드(매개_변수, 매개_변수)]
- 위와 같은 기본 형태
- [window.jQuery = window.$ = jQuery;]
- jQuery 이름의 함수를 너무 많이 사용하기 때문에 $ 식별자로 간단하게 입력할 수 있게 만든 것

```
// 일반 문서 객체로 jQuery 객체를 생성한다
$(document)

// css 선택자로 jQuery 객체를 생성한다
$('h1')

// HTML 문자열로 jQuery 객체를 생성한다
$('<h1></h1>')
```

- Css 선택자를 사용하면 '기존의 문서 객체 선택'
- HTML 문자열을 사용하면 '새로운 문서 객체 생성'
- 문서 객체 사용하면 '기존의 문서 객체 선택' 과 '새로운 문서 객체 생성' 모두 가능

- 이전에는 문서 객체 모델을 다룰 때 window 객체의 load 이벤트를 사용해야 화면이 구성된 이후부터 문서 객체 조작이 가능 했지만
- jQuery 에서는 doucment 객체의 ready 이벤트를 활용한다

화면 캡처: 2022-10-16 오후 4:24

시작 코드

문서 객체 선택

2022년 10월 16일 일요일 오후 4:25

- Css 선택자를 사용해 문서 객체를 선택하는 경우
- $() 함수를 사용해 다음 형태로 문서 객체를 선택한다

```
// h1 태그를 선택
$('h1')

// h1 태그 중에서 class 속성으로 logo를 가진 태그를 선택
$('h1.logo')

//id 속성이 head인 태그를 선택한다
$('#head')

// input 태그 중에서 type 속성이 sumit 인 태그를 선택
$('input[type=submit]')
```

화면 캡처: 2022-10-16 오후 4:29

- 객체 탐색 메소드

메소드	설명
Parent()	부모 태그 선택
Find()	후손 태그 찾음

- 예를 들어 h1 태그에 parent() 메소드와 find()메소드를 사용하는 경우

```
// h1 태그의 부모 태그 선택
$('h1').parent()

// h1 태그 내부에 있는 i 태그 선택
$('h1').find(i)
```

화면 캡처: 2022-10-16 오후 4:33

- H1 태그가 여러 개 선택 된 상태로 parent() 메소드와 find() 메소드를 사용하면
- 첫 번째 h1 태그를 기준으로 부모와 후손을 찾는다

문서 객체 개별 조작

2022년 10월 16일 일요일 오후 4:34

- $() 함수를 사용하면 여러 개의 문서 객체를 선택 할 수 있다
- 이러한 문서 객체가 몇 개 선택 되었는지 확인 할 때 length 속성 사용
- 선택된 문서 객체의 수

속성	설명
Length	선택된 문서 객체의 수 구한다

- 베열처럼 대괄호를 사용해서 요소 하나를 꺼낼 수 도 있고 get() 메소드 활용 가능
- 선택된 문서 객체 추출

메소드	설명
Get()	선택한 문서 객체 중 하나를 선택

- 단, get() 메소드를 사용해 추출한 요소는 jQuery 문서 객체가 아니라 일반 문서 객체이다
- jQuery 객체로 다시 만들고 싶다면 $() 함수의 매개 변수로 넣어야 한다

```
<meta charset="UTF-8">
<title>JQuery Basic</title>
  <script src="jquery-3.6.1.min.js"></script>
  <script>
    // 웹 페이지를 모두 불러오면
    $(document).ready(function () {
      //h1태그를 모두 추출하고
      const $headers = $('h1');
      //h1 태그의 개수만큼 반복을 돌려서
      for(let i = 0; i <$headers.length; i++){
        //특정 위치에서
        if(i % 2 ==1){
          //i 번째 요소 추출
          const domElement = $headers.get(i);
          // 배경 색상 적용
          $(domElement).css('backgroundColor','red');
        }
      }
});
  </script>
</head>
<body>
<h1>HeaderA</h1>
<h1>HeaderB</h1>
<h1>HeaderC</h1>
<h1>HeaderD</h1>
```

- 변수 이름 const$headers=$('h1'); 자바스크립트에서는 변수 이름으로 _와 $ 사용 가능
- jQuery를 이러한 형태로 활용 하다보면 jQuery 문서 객체와 일반 문서 객체가 섞인다
- 이때 두 가지를 구분하려고 jQuery 문서 객체가 들어있는 변수는 앞에 $를 붙여 구분한다
- 개발자 관례임

- 선택된 문서 객체 반복 적용

메소드	설명
Each()	선택한 문서 객체에 반복을 적용

- Each() 메소드를 사용 할 때 기본적인 자바스크립트에서 제공하는 Array 객체의 forEach() 메소드와 인덱스, 요소 순서가 다르다
- Each() 메소드의 콜백함수

Array 객체의 forEach() 메소드	jQuery의 each() 메소드
[ ].forEach(function (item, index) {	$('h1').each(function (index, item) {
});	});

- Each() 메소드의 콜백 함수 내부에서 this 는 문서객체를 나타낸다
- Each()메소드를 사용해 문서 객체 반복

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JQuery Event</title>
  <script src="jquery-3.6.1.min.js"></script>
  <script>
    // 웹 페이지를 모두 불러오면
    $(document).ready(function () {
      //h1태그를 모두 추출하고 반복을 돌린다
      $('h1').each(function (index, element) {
        //특정 위치에서
        if(index % 2 ==1 ){
          // 배경 색상 적용
          $(this).css('backgroundColor','red');
        }
      });
});
  </script>
</head>
<body>
<h1>HeaderA</h1>
<h1>HeaderB</h1>
<h1>HeaderC</h1>
<h1>HeaderD</h1>

</body>
</html>
```

화면 캡처: 2022-10-16 오후 4:58

HeaderA

HeaderB

HeaderC

HeaderD

화면 캡처: 2022-10-16 오후 4:58

[odd 선택자와 even 선택자]

- 홀수 번째 또는 짝수 번째 요소를 선택해서 무언가를 적용하는 코드는 자주 사용한다
- 따라서 쉽게 사용할 수 있는 형태를 제공해준다
- Odd선택자(홀수)와 even 선택자(짝수)
- $() 함수 의 매개변수로 넣은 선택자 뒤에 :odd 또는 :even 을 넣어 사용할 수 있다

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JQuery Event</title>
  <script src="jquery-3.6.1.min.js"></script>
  <script>
    // 웹 페이지를 모두 불러오면
    $(document).ready(function () {
    // 짝수 위치에 h1 태그에 배경 색상을 적용
          $('h1:even').css('backgroundColor','red');
        });

  </script>
```

화면 캡처: 2022-10-16 오후 5:07

문서 객체 조작

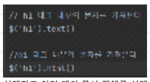
2022년 10월 16일 일요일 오후 5:07

1. 글자 조작

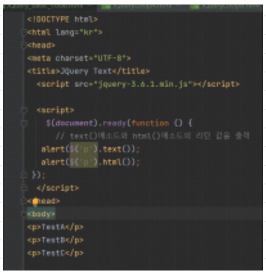
- 글자 조작 메소드

메소드	설명
text()	html태그 내부의 문자를 조작
html()	html태그 내부의 문자를 조작(html태그 인식)

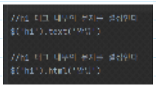
- jQuery 객체가 가진 대부분의 메소드는 하나의 이름으로 get(추출, 가져오기)와 Set(설정, 할당)을 수행한다



- 선택자로 여러 개의 문서 객체를 선택 할 때 text() 메소드는 모든 문서 객체 내부의 문자를 출력
- Html() 메소드는 첫 번째 문서 객체 내부의 문자를 출력



- Text() 메소드는 모든 p 태그 내부의 문자를
- Html()메소드는 첫번째로 선택된 p 태그 문자를 출력한다
- 실제로 get 형태로 사용할 때는 하나의 객체만 선택한 상태로 사용할 때가 많다
- 문서 객체 내부의 문자를 변경 할 때



화면 캡처: 2022-10-16 오후 5:28

- Text() 메소드와 html()메소드 모두 여러 개의 문서 객체를 선택하고 set 형태의 메소드를 사용하면 모든 객체 내부에 문자가 들어간다
- 기능(메소드) 이름이 다르므로 당연히 차이가 있다

화면 캡처: 2022-10-16 오후 5:32

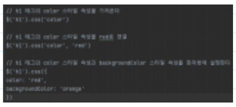
Get 형태에서 text() 메소드는 모두 가져오고
Html()메소드는 하나만 가져오므로 set 형태도 동일 할 것이라 생각하지만
Set 형태는 두 메소드 모두 선택한 문서 객체에 적용한다

2. 스타일 조작

- 스타일 조작 메소드

메소드	설명
Css()	스타일 조작 한다

- 스타일도 GET형태와 SET 형태를 나누어 사용한다
- GET 형태는 선택자로 선택된 문서 객체 중 첫 번째 문서 객체의 스타일을 가져온다



화면 캡처: 2022-10-17 오후 7:28

스타일 조작

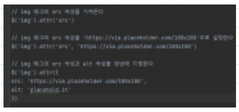
그라데이션 효과

3. 속성 조작

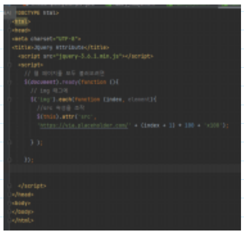
- 속성 조작 메소드

메소드	서렴ㅇ
Attr()	속성을 조작

- 속성도 GET과 SET 형태 나ㅣ눠서 사용
- Get 형태는 선택자로 선택된 문서 객체중 첫 번째 문서 객체의 속성을 가져온다



화면 캡처: 2022-10-17 오후 7:37



화면 캡처: 2022-10-17 오후 7:43

문서 객체 생성

2022년 10월 17일 월요일 오후 7:44

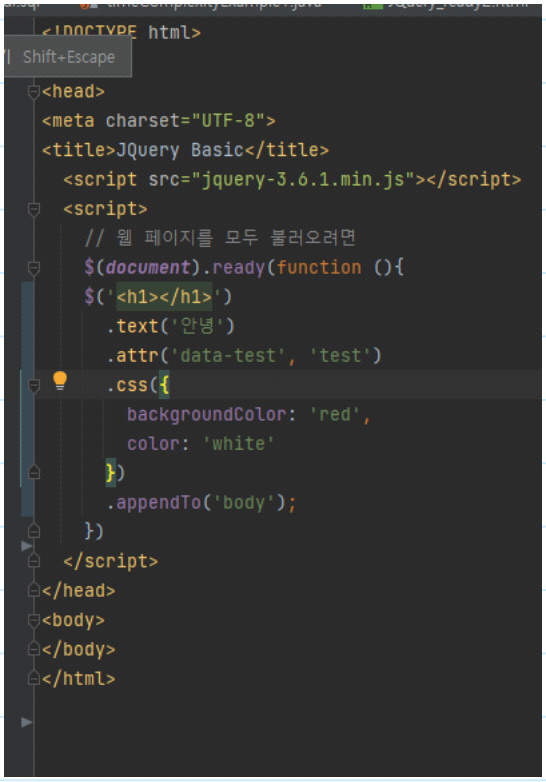
- 문서 객체 생성할 때 $() 함수의 매개 변수에 'HTML' 형식의 문자열을 입력한다
- [$('<h1></h1>')]

```
$('<h1></h1>')
  .text('안녕')
  .attr('data-test', 'test')
  .css({
    backgroundColor: 'red',
    color: 'white'
  });
```

- 문서 객체 추가 메소드

메소드	설명
$(A).prependTo(B)	A 를 B 안쪽 앞에 추가한다
$(A).appendTo(B)	A 를 B 안쪽 뒤에 추가한다
$(A).insertBefore(B)	A 를 B 앞에 추가한다
$(A).insertAfter(B)	A 를 B 뒤에 추가한다

```
<h1> insertBefore</h1>
    <div class = "target">
      <h1>prependTo</h1>
      <h1>prependTo</h1>
    </div>
<h1>insertAfter</h1>
```



화면 캡처: 2022-10-18 오전 9:37

이벤트

2022년 10월 18일 화요일 오전 9:37

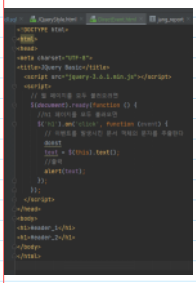
- jQuery의 이벤트 메소드

메소드	설명
On()	이벤트 연결
Off()	이벤트 제거

1. 이벤트 직접 연결
   - 특정 태그에 이벤트를 연결하고, 특정 태그를 눌렀을 때 이벤트가 발생 하게 하는 것을 의미
   - jQuery로 이벤트를 직접 연결 할 때는 on() 메소드를 사용하며 기본 형태는
   - $(선택자).on(이벤트 이름, 콜백함수)
   - 이벤트를 연결 할 때 콜백함수의 매개 변수로 이벤트 객체가 전달된다
   - 이벤트 객체를 사용하면 이벤트 정보를 알아 낼 수 있다.
   - 경로 확인

jQuerty -event object
https://api.jquery.com/category/event/event-object/

   - 콜백함수 내부에서 this 키워드는 이벤트를 발생시키는 객체
   - This 키워드는 이벤트를 발생 시키는 객체이다
   - This 키워드를 활용해 서 이벤트를 발생시킨 객체의 스타일을 변경할 수 있다.



화면 캡처: 2022-10-18 오후 5:44

- 키보드 이벤트

메소드	설명
Keydown()	키보드 키를 눌렀을 때
Keypress()	키가 입력 되었을 때
Keyup()	키보드 키를 떼었을 때

- 마우스 이벤트

메소드	설명
Click()	마우스를 클릭 했을 때
Dbclick()	마우스를 더블 클릭 했을 때
mousedown()	마우스 버튼을 눌렀을 때
mousenter()	마우스 커서가 해당 태그로 들어갔을 때
mouseleave()	마우스 커서가 해당 태그에서 나갔을 때
Mousemove()	마우스가 움직일 때
Mouseup()	마우스 버튼을 뗄 때

- 입력 양식 이벤트

메소드	설명
Blur()	입력 양식에 값 입력을 종료할 때
Change()	입력 양식의 값이 변경됨 때
Focus()	입력 양식에 값 입력을 시작할 때
select()	Type 속성이 select 인 입력 양식의 목록에서 값을 선택했을 때
Submit()	Type 속성이 submit 인 입력 양식을 클릭 했을 때

- 웹 브라우저 이벤트

메소드	설명
Resize()	웹 브라우저의 크기를 변경할 때
Scroll?()	웹 브라우저를 스크롤 할 때

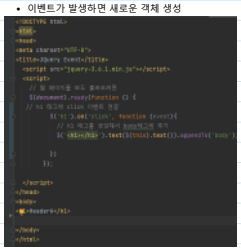
$(document).ready(function(){}) 의 ready도 이벤트이다

- 기본 이벤트 제거
  - Event 객체의 preventDefault() 메소드를 사용한다

```
//submit 버튼의 기본 이벤트를 모두 제거한다.
$('input[type=submit]').submit(function (event){
  event.preventDefault();
});
```

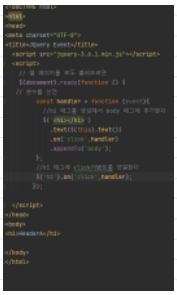
화면 캡처: 2022-10-18 오후 6:13

- 이벤트가 발생하면 새로운 객체 생성



- 코드를 실행하고 첫 번째 h1 태그를 클릭하면 h1 태그를 생성해서 body 태그 아래에 추가한다
- 하지만 새로 생성된 h1 태그에는 이벤트가 연결되어 있지 않다
- 이벤트 직접 연결은 연결 하는 시점에 존재하는 문서 객체에만 이벤트를 연결한다.
- 문서 객체를 생성 했을 때 이벤트를 따로 연결해서 해결 할 수 있다.

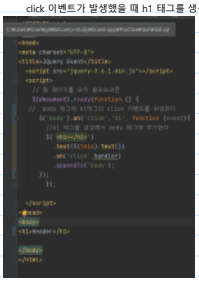
- 이벤트 추가연결



화면 캡처: 2022-10-18 오후 6:24

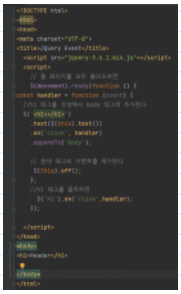
- 자주사용해야 하는 코드인데 형태가 복잡함
- 그래서 이벤트 간접 연결 기능을 제공한다

2. 이벤트 간접 연결
   - 부모에게 이벤트를 위임 해서 부모가 이벤트를 처리하게 하는것
   - 예제 : 이벤트 간접 연결, body 태그 내부에 있는 h1 태그에서 click 이벤트가 발생했을 때 h1 태그를 생성해 추가한다



   - Body 태그 내부에서 h1 태그를 클릭 했을 때 구현할 수 있다.

3. 이벤트 제거
- off() 메소드 사용
  - H1 태그를 클릭하면 새로 h1 태그를 생성해 추가하고 기존 태그에서는 이벤트를 제거한다



- 이벤트를 한 번만 연결하는 메소드

메소드	설명
One()	이벤트를 한 번만 연결한다

애니메이션

2022년 10월 18일 화요일 오후 6:41

- 애니메이션 기능은 대부분 외부 라이브러리를 사용한다
- 애니메이션 메소드

메소드	설명
Animate()	애니메이션을 적용한다

- [$(선택자).animate(속성, 시간, 콜백함수)]

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JQuery Event</title>
 <style>
   .box {
     /* 크기*/
     width: 100px; height: 100px;
     /* 색상*/
     background-color: red;
     /* 위치*/
     position: absolute;
     left: 10px; top: 10px;
   }

 </style>
  <script src="jquery-3.6.1.min.js"></script>
  <script>
    // 웹 페이지를 모두 불러오려면
    $(document).ready(function () {
      //box 태그를 클릭해서
      $('.box').click(function () {
        //1초 동안 left를 1000픽셀로 변경한다
        $(this).animate({
          left: 1000
        }, 1000);
      })
    });

  </script>
</head>
<body>
<dib class="box">박스</dib>

</body>
</html>
```

화면 캡처: 2022-10-18 오후 6:48