쉬운 *jQuery*

jQuery

jQuery 시작하기

jQuery 시작하기

jQuery 를 사용하기 앞서 작업을 위한 폴더를 만들어보자

폴더 이름은 jquery-study 라는 폴더를 만들었다.

이제 이 안에 html 문서를 생성하자

index.html 이라는 문서를 생성하고 그 안으로 들어가서 코딩을 시작해 본다.

에디터를 열고 index.html 파일을 열어보자.

*** 참고

기본 코딩에 대한 팁(?)

모든 언어는 HTML 포함 여는 곳이 있으면 닫는 곳 있다.

물론 HTML은 그렇지 않은 태그문(예를 들면 img, br)이 존재하지만 기본적으로는 그렇다.

항상 바깥쪽을 열고 닫기를 먼저 하고 안으로 들어와 열고 닫기를 잘 해줘야 나중에 햇갈리거나 괄호가 꼬이는것을 방지할 수 있다.

jQuery 시작하기

jQuery 를 사용하기 위해서는 기본적으로 jQuery js 문서를 로드하여야 한다.

jQuery 문서를 로드하는 방법은 script 태그로 외부 jQuery js 문서를 로드해야 한다.

1. 외부 사이트에서 직접 불러오는 방법

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

2. 내 사이트 내에 js문서를 저장하여 불러오는 방법

<script type="text/javascript" src="만든 폴더/jquery.min.js"></script>

먼저 jQuery 문서를 다운로드 한다.

jQuery 문서를 다운로드 하는 곳

https://jquery.com/download/

이 사이트에서 보면

Download the compressed, production jQuery 1.11.3

Download the uncompressed, development jQuery 1.11.3

위 두가지중에 하나로 사용한다.(버전은 업그레이드 되었을 시 바뀔 수도 있다.)

jQuery 시작하기

Download the compressed, production jQuery 1.11.3 : 이 js 파일은 내용이 압축된 파일이다. 일반적인 프론트 엔드 개발자 및 퍼블리셔가 많이 사용하는 파일.

Download the uncompressed, development jQuery 1.11.3 : 이 js 파일은 개발자가 변경 가능하도록 압축되지 않은 파일이다.

기본적으로 두가지 파일 모두 내용은 동일하다.

jQuery 파일 또는 jQuery 플러그인 파일은 개발자 버전인지 아니면 압축 파일인지를 알기 쉽도록 파일 이름에 규칙이 있다.

jQuery js

파일의 예)

jquery.min.js 파일 : 압축된 파일

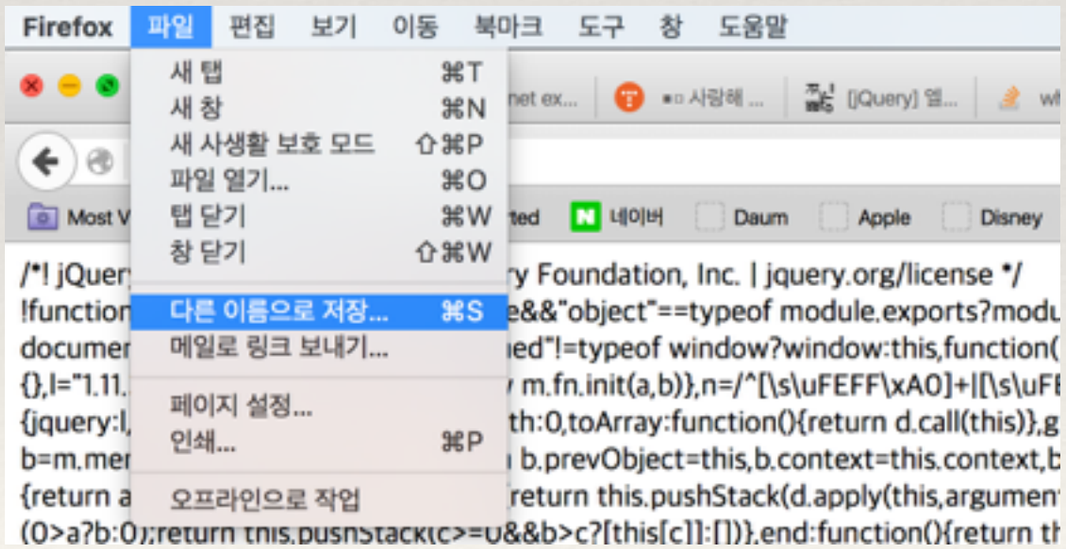
jquery.js 파일 : 압축되지 않은 개발자 버전의 파일

jQuery 시작하기

jQuery 를 개발할 것이 아니므로 Download the compressed, production jQuery 1.11.3 을 다운로드 해보자

https://jquery.com/download/ 에서 Download the compressed, production jQuery 1.11.3 이라는 곳을 클릭하면 query-1.11.3.min.js 파일을 다운로드 할 수 있다

브라우저에 따라 다운 로드 하려는 문서가 다운로드가 되지 않고 웹 페이지처럼 뜨는 경우가 있는데 당황하지 말고 저장 (윈도우 : Ctrl + s, 맥 : Command + s)하면 된다



jQuery 시작하기

이제 다운 받은 jquery 문서를 앞으로 jquery-study 폴더안에 js 라는 폴더를 생성하고 그 안에 다운 받은 jquery js 문서를 넣고 로드 해보자

방식은 외부 사이트에서 직접 로드하는 방식과 동일하다.

<script type=“text/javascript” src=“js/jquery-1.11.3.min.js”></script>

앞서 하는 로드 방식과 다른점이 있을까?

있다.

첫번째 방법으로 로드하는 경우 간혹 가다가 웹 속도가 느린곳에서는 외부 로드가 잘 이루어 지지 않아 사이트가 초반에 단 몇초간이라도 정상작동이 되지 않는 경우가 발생할 수 있다.

jQuery 시작과 끝 1

jQuery 기본 문법

jQuery 선언방식 1

```
(function() {
    실행될 내용
}) (jQuery);
```

jQuery 시작과 끝 2

jQuery 기본 문법

jQuery 선언방식 2

```
jQuery(document).ready(function() {
    실행될 내용
});
```

jQuery 시작과 끝 3

jQuery 기본 문법

jQuery 선언방식 3

```
$(document).ready(function() {
    실행될 내용
});
```

jQuery 시작과 끝 4

jQuery 기본 문법

jQuery 선언방식 4

```
$(function() {
    실행될 내용
});
```

jQuery 시작과 끝

위 4가지 방식 모두 정상적으로 동작한다.

위 jQuery 안에 동작할 내용을 작성한다.

위 내용을 설명하자면 웹 브라우저가 실행되고 브라우저 내에 html 문서가 모두 로드 된 다음에 안의 내용이 실행되는 방식이다.

jQuery는 기본적으로 함수로 시작해서 ($(function() { … 함수가 끝나는 부분 (});)에서 끝이난다.

함수로 감싸는 이유는 한번 사용하고 버릴게 아니라 브라우저 내에서 동작하는 페이지내에 있는 jQuery 를 여러번 재활용해서 사용 할 수 있도록 하기 위해서이다.

.

jQuery 시작과 끝

하 …. 함수는 어려운거 아닌가요?

jQuery 시작과 끝

.

함수의 원리를 알고 나면 간단하다.

3장 또는 4장에서 자세히 다룰 예정이니 미리 겁먹지 말자

일단 지금은 function() {} 또는 사용자 정의 이름() {} 라고 되어 있는것은 함수이구나 라는것만 기억하자.

jQuery 선택자

jQuery 에서는 기존 javascript 와는 다른 독특하고 편리한 방식의 html 태그를 제어 할 수 있는 선택자라는 개념이 도입되었다.

선택자는 html 태그문 자체가 될 수도 있고 해당 태그문의 아이디나 클래스명이 될 수도 있다.

예)

DIV 태그1

클래스 적용한 DIV

아이디 적용한 DIV

jQuery('div').이벤트(finction() {};

또는

$('div').이벤트(finction() {};

클래스 선택자 : .클래스명

아이디 선택자 : #아이디명

jQuery 이벤트

기본적인 jQuery 의 이벤트들

click : 마우스를 클릭했을 때의 이벤트

hover : 마우스 오버와 아웃 됐을 때의 이벤트

mouseover : 마우스 오버시의 이벤트

mouseout : 마우스 아웃시의 이벤트

jQuery 이벤트

jQuery 표현방식

클릭 이벤트

```
(function() {
    jQuery("선택자").click(function() {
        // 클릭했을때 이벤트
    });
    또는
    $("선택자").click(function() {
        // 클릭했을때 이벤트
    });
});
}) (jQuery);
```

jQuery 이벤트

jQuery 클릭이벤트

위에 만들었던 div 세가지를 가지고 적용해 보겠다.

```
(function($) {
    jQuery(“div").click(function() {
        alert(‘DIV를 클릭했습니다.’);
    });
    jQuery(“.class_name”).click(function() {
        alert(‘class_name 이라는 이름의 클래스를 가진 요소를 클릭했습니다.’);
    });
    jQuery(“#id_name”).click(function() {
        alert(‘id_name 이라는 이름의 아이디를 가진 요소를 클릭했습니다.’);
    });
});
}) (jQuery);
```

jQuery 이벤트

jQuery() 는 $() 로 단축 표기 할 수 있다.

특별한 경우가 아니라면 jQuery()라고 표기하지 않아도 된다.

jQuery 함수를 jQuery 레퍼라고 부르는 사람도 있다.

위에서 쓰는 방법인

```
(function($) {
    …
}) (jQuery);
```

이 표기 방법은 여러 외부 jQuery 로 만들어진 플러그인들을 사용할 때 jQuery 레퍼($)를 지역변수로 만들어 내부적으로 실행하고 외부 라이브러리와의 충돌을 막기 위해 사용하는 표기 방법이다.

즉 외부 라이브러리와 별도로 작성한 jQuery는 서로 충돌을 일으키지 않기 위해서 사용하는 방법이다.

jQuery 이벤트

위 내용을 해석해 보면 …

$(document).ready(function() { : Document 즉 웹 페이지가 전체가 불러와진 후에

function() : 함수 즉 계속 사용할 수 있게 해줘~

우엇을?

(…) 안에 내용을 …

}); : jQuery 내용은 여기까지 실행하고 종료

위에서 실행될 { … } 안에 내용 …

```
$("div").click(function() {
    // 이벤트
});
```

div를 클릭하면

안의 내용을 실행해~

몇번을?

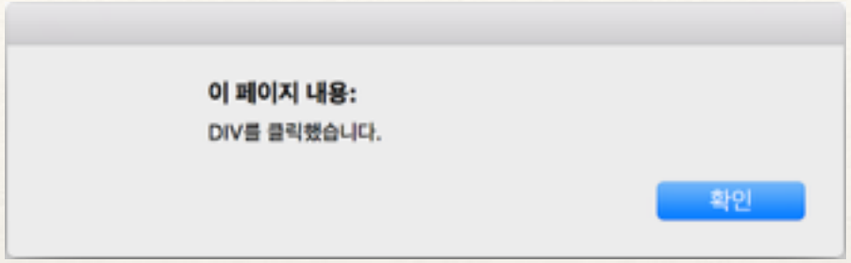
클릭할 때마다 계속

jQuery 이벤트

위 내용을 실행해 보면 어떻게 될까?

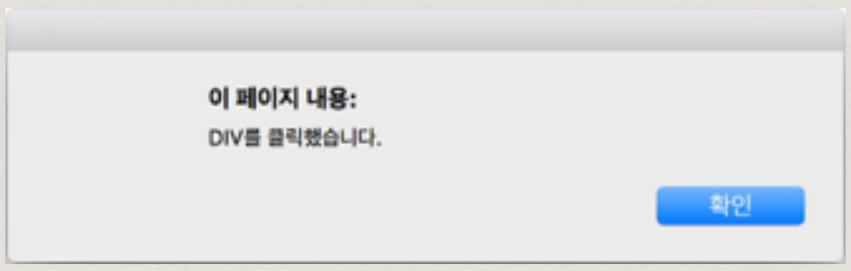
jQuery 이벤트

세가지 div 모두 “DIV를 클릭했습니다.” 라는 내용의 alert 메세지창이 뜨는것을 알 수 있을 것이다.

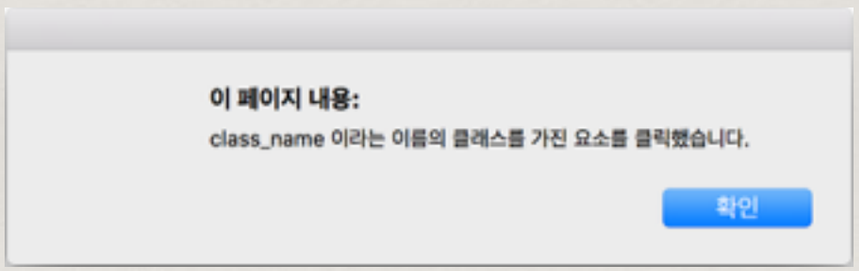


두번째 div 를 클릭해보면 “DIV를 클릭했습니다.” 라는 내용의 alert 메세지창이 뜨고 확인을 누르면 “class_name 이라는 이름의 클래스를 가진 요소를 클릭했습니다.” 라는 alert메세지창이 뜨는것을 볼 수 있을 것이다.

1

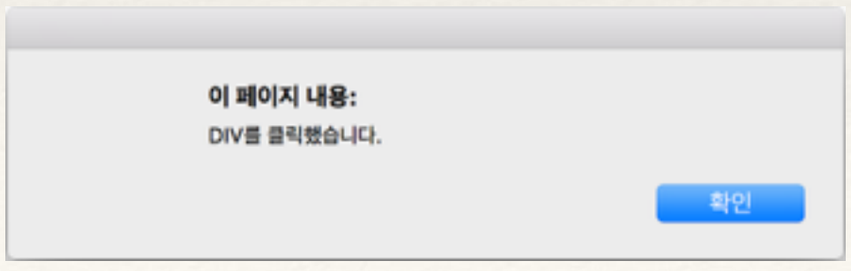


2



세번째 div 를 클릭해보면 “DIV를 클릭했습니다.” 라는 내용의 alert 메세지창이 뜨고 확인을 누르면 “id_name 이라는 이름의 아이디를 가진 요소를 클릭했습니다.” 라는 alert메세지창이 뜨는것을 볼 수 있을 것이다.

1



2