

jQuery



강사 소개



- 문성훈
- [주] ATGLab. 개발 / 컨설팅 / 직무교육
Gamification 전문기업.
Sencha Inc. 국제 공인 Trainer.
W3C(W3DevCampus Online Trainer.)
- 고려대학교 컴퓨터 학과 Ph.D.
- 삼성전자 / LGCNS / KT 직무교육
한국IBM / SK C&C 직무교육
LGCNS 신입사원교육 컨설팅
고려대학교 / 한국기술교육대학교 강의
- Email : moon9342@atglab.co.kr

과정 목적

- 목적 (1)

- JavaScript의 가장 대중적인 라이브러리인 jQuery를 통해서 웹 페이지의 동적 처리가 얼마나 손쉬워지는지에 대해서 학습.

표지	Title	ISBN	Price	Author	상세보기	삭제
	IT EXPERT, 모바일 자바 프로그래밍	89-7914-206-4	23000	권기경, 박용우	상세보기	삭제

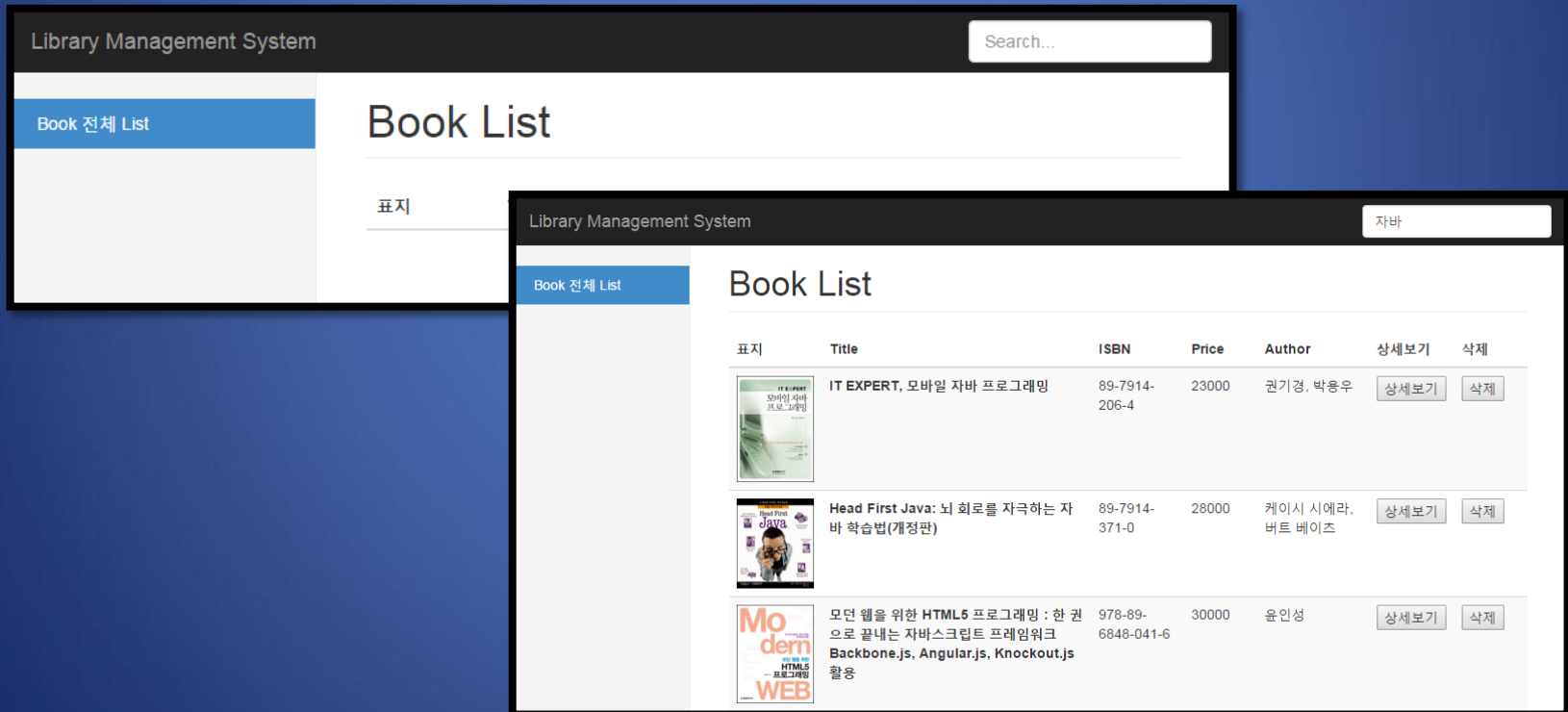


	<div>IT EXPERT, 모바일 자바 프로그래밍 출판일 : 2002년 09월 페이지수 : 528 출판사 : 한빛미디어(주) 부록 : CD 1장</div>	89-7914-206-4	23000	권기경, 박용우	상세보기	삭제
---	---	---------------	-------	----------	----------------------	--------------------

과정 목적

- 목적 (2)

➤ Server와의 AJAX통신처리가 얼마나 손쉬워지는지에 대해서 학습.



Library Management System

Search...

Book 전체 List

Book List




표지

Library Management System

자바

Book 전체 List

Book List

표지	Title	ISBN	Price	Author	상세보기	삭제
	IT EXPERT, 모바일 자바 프로그래밍	89-7914-206-4	23000	권기경, 박응우	상세보기	삭제
	Head First Java: 뇌 회로를 자극하는 자바 학습법(개정판)	89-7914-371-0	28000	케이스 시에라, 버트 베이즈	상세보기	삭제
	모던 웹을 위한 HTML5 프로그래밍 : 한 권으로 끝내는 자바스크립트 프레임워크 Backbone.js, Angular.js, Knockout.js 활용	978-89-6848-041-6	30000	윤인성	상세보기	삭제

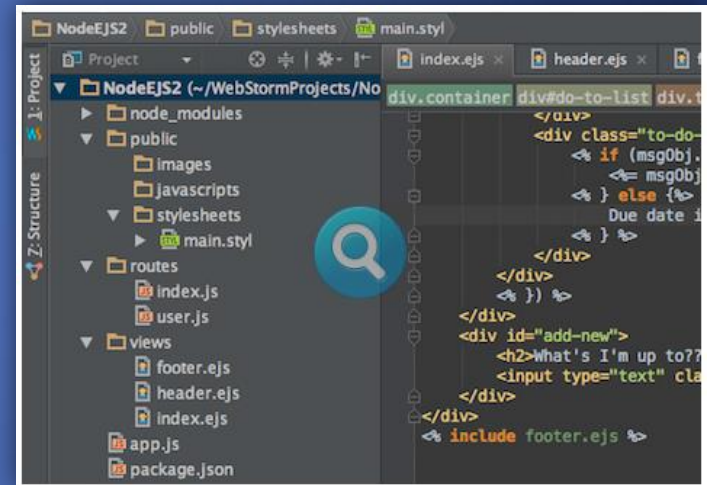
환경 설정

- 개발 IDE download

- WebStorm  <http://www.jetbrains.com/webstorm/>
- Eclipse  <http://www.eclipse.org>

- Google Chrome

- <http://www.google.co.kr/chrome/browser/desktop/>



Contents



JavaScript Basic



jQuery Selector



jQuery Traveling & Manipulation



jQuery Event



jQuery AJAX

Contents

I JavaScript Basic

II jQuery Selector

III jQuery Traveling & Manipulation

IV jQuery Event

V jQuery AJAX

JavaScript 개요

- JavaScript

- 웹 브라우저에서 사용하는 프로그래밍 언어
- 유연하지만 상당히 loose한 언어
- The world's most misunderstood programming language
- ECMAScript 표준을 따른다 (현재 버전 ECMAScript 6)



- JavaScript 코드 위치

- 웹 페이지에 <script> tag를 이용하여 삽입
- 별도의 파일을 만들어서 html 페이지에서 링크

JavaScript 개요

- JavaScript에 대한 평가

- 웹 페이지의 동적 처리를 담당
- DOM에 접근하는 방법만 알면 됨
- 객체지향적 프로그래밍?
- Copy & Paste
- 보조적인 언어라는 인식
- Document의 부재

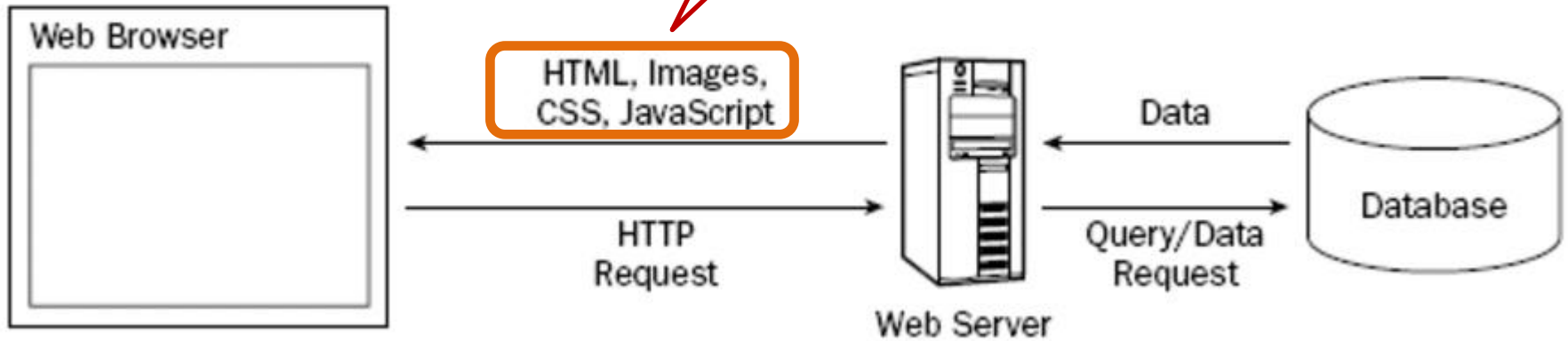


JavaScript 개요

- 전통적인 Web Application Model (Round Trip 방식)

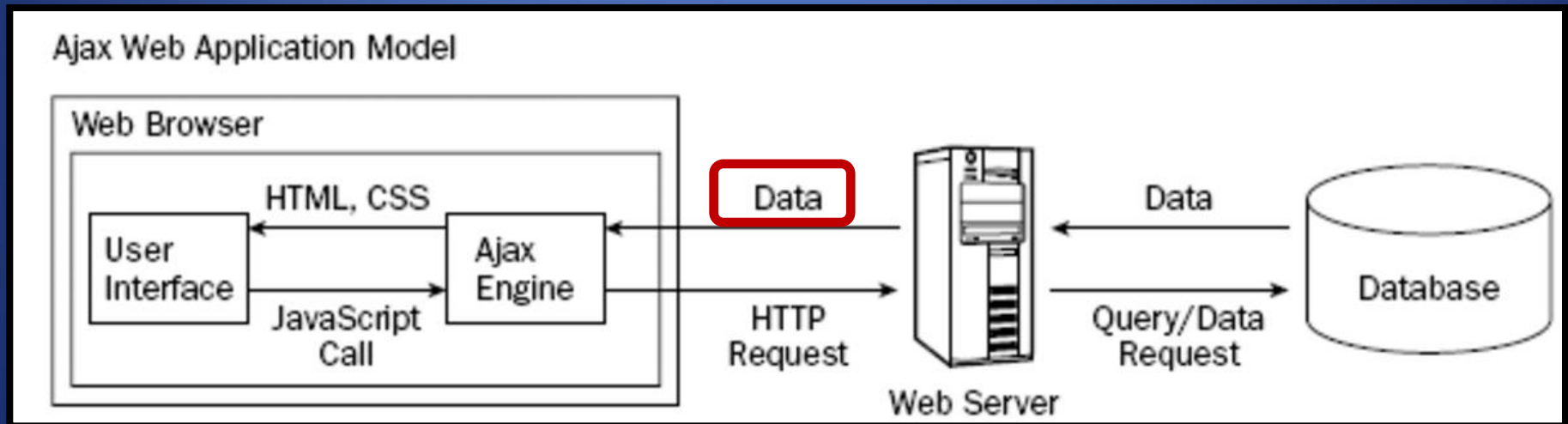
화면 Refresh로 인한
성능상의 문제

Traditional Web Application Model



JavaScript 개요

- AJAX Web Application Model (Single Page Application 방식)



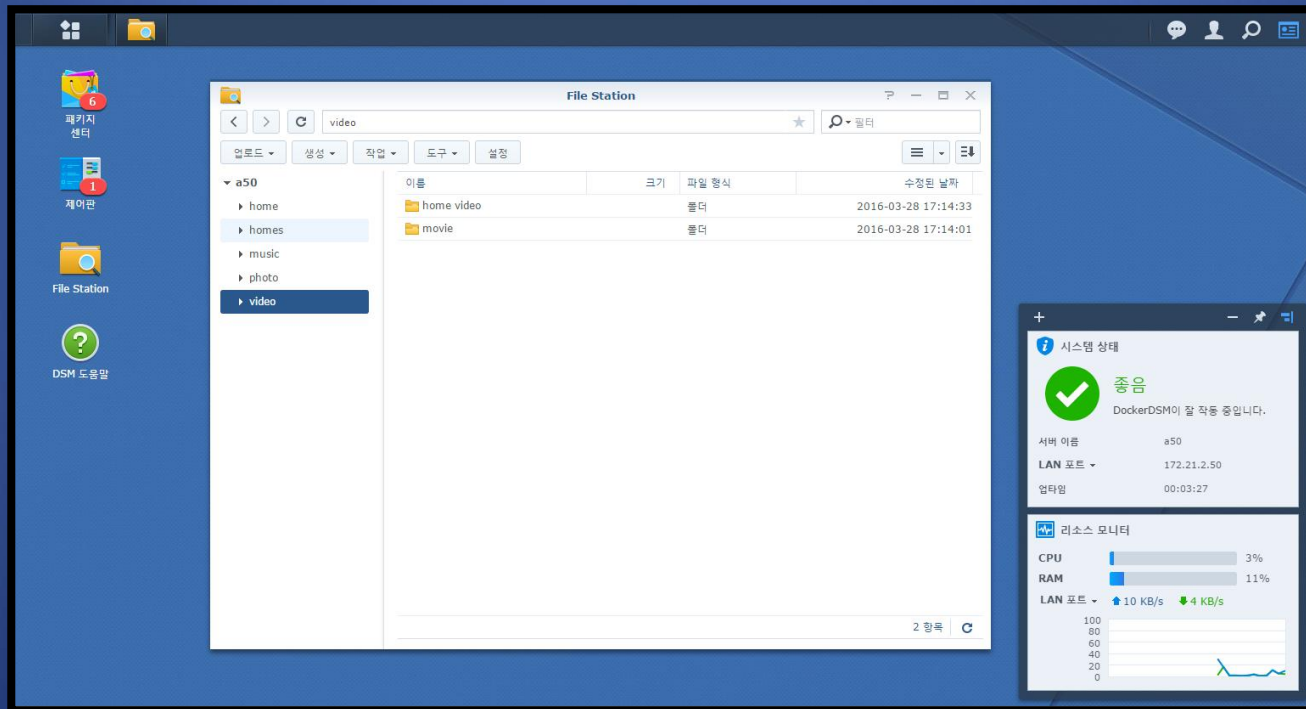
JavaScript 개요

- Web개발 환경의 변화
 - Front End Web Application 등장
 - AJAX (서버와의 통신을 위한)
 - Design Pattern (application의 유지보수와 재활용을 위한)
 - 다양한 Framework, Library의 활용
- AngularJS, React.js, Backbone.js, Ext JS, jQuery, jQuery Mobile, etc.

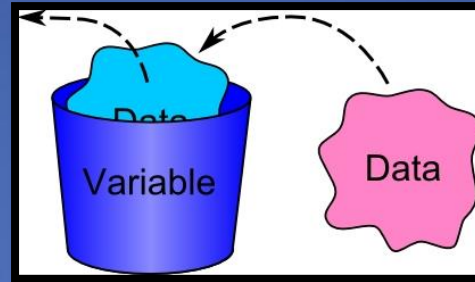
JavaScript 개요

- Ext JS를 이용한 Web Application (Synology NAS DSM)

➤ https://www.synology.com/ko-kr/dsm/live_demo



JavaScript Variable



- var keyword
- var 변수에 어떤 type의 값이라도 할당 가능(weakly typed language)
- var keyword를 사용하지 않고 변수선언 가능. 단, var로 변수를 선언했을 때와는 차이가 있다.

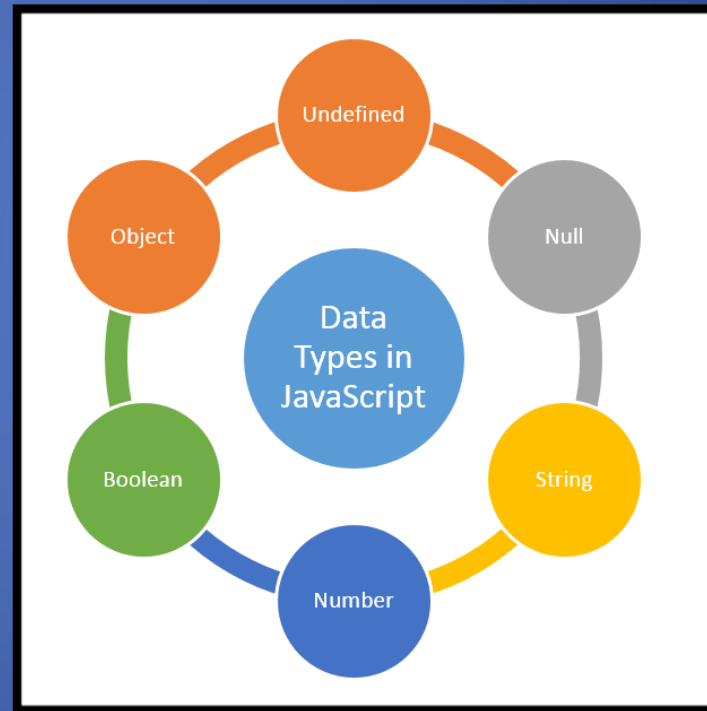
```
var tmp1 = "hello";  
tmp2 = "hi";
```

JavaScript Data Type

- JavaScript의 Data Type은 크게 2가지로 구분

- Primitive Type (5가지)

- Object Type



JavaScript Literal

- 각 type의 literal 확인 (typeof 연산자를 이용해서 확인)

```
// 각 type에 대한 literal 표현
var tmp1 = "Sample";           // string literal
var tmp2 = 10;                 // number literal
var tmp3 = true;              // boolean literal
var tmp4 = [ 1, 2, 3, 4 ];    // array literal

// object literal
var tmp5 = { name : "홍길동", age : 25 };

// function literal
var tmp6 = function(x,y) {
    return x + y;
};
```


JavaScript Execution

- 프로그램의 실행 순서

- 프로그램 실행



- 프로그램 전역 level에서의 parsing

- 함수에 포함되지 않는 변수를 찾아서 "undefined"로 초기화
- 함수에 포함되지 않는 named function에 대해 함수명과 동일한 이름의 변수를 생성한 후 함수 실행 코드에 대한 참조로 초기화
(함수 변수)

- Parsing이 끝나면 프로그램 실행. 실행 중 함수 호출을 만나면 해당 함수 level에서 parsing 반복
(지역 변수와 함수 변수를 정의)한 후 함수 코드 실행

JavaScript 실습

- 생각해보자

```
console.log(square(4));  
  
var square = 0;  
  
function square(input) {  
  return input * input;  
}  
  
console.log(square);
```



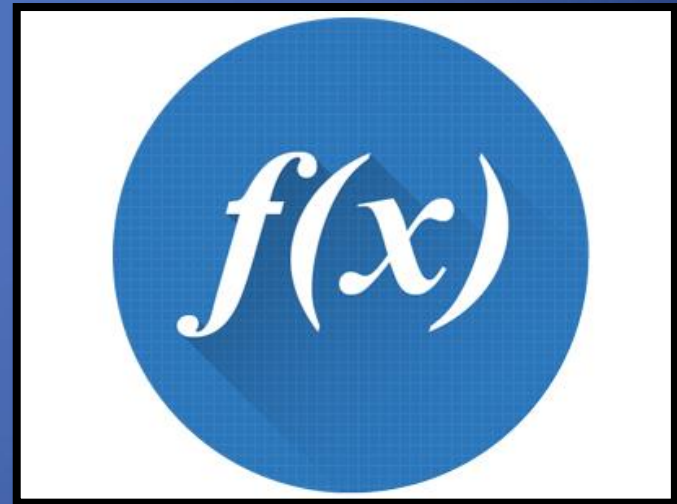
- 실행결과는?



JavaScript Function

- class vs. function

➤ JavaScript는 class가 존재하지 않는다. function으로 모든 것을 처리.



JavaScript Function

- JavaScript 함수 정의 방법

- 선언적 함수

```
function add(x,y) {  
    return x+y;  
}
```

- 람다 함수
(익명 함수)

```
var add = function(x,y) { return x+y; };
```

JavaScript Function

- 익명함수 (람다함수)

- 함수 literal을 이용하여 정의된 함수
- 이름이 없기 때문에 변수에 할당하거나 함수 인자의 값, return값으로 사용

```
// 변수에 함수 저장
var result = function(x,y) { return x+y; };

// 함수를 다른 함수의 인자로 활용
obj.myFunc(function(x,y) { return x+y; });

// 함수를 정의하고 바로 호출
var sum = (function(x,y) { return x+y; })(1,2);
```

JavaScript Function

- JavaScript 함수의 역할

- JavaScript의 함수와 객체지향 언어의 method와는 전혀 다른 개념.
- JavaScript의 함수는 몇 가지 역할을 수행할 수 있는 요소.
 - 호출 가능한 routine으로의 함수 (일반적인 함수)
 - 값의 역할을 하는 함수 (일급함수 – First Class Functions)
(인자로 함수 전달, 함수를 변수에 저장, 다른 함수의 return 값으로 활용)
 - instance를 생성할 수 있는 요소. 즉, 객체 타입으로서의 함수
(타 언어의 class의 개념으로의 함수)



JavaScript Function 호출

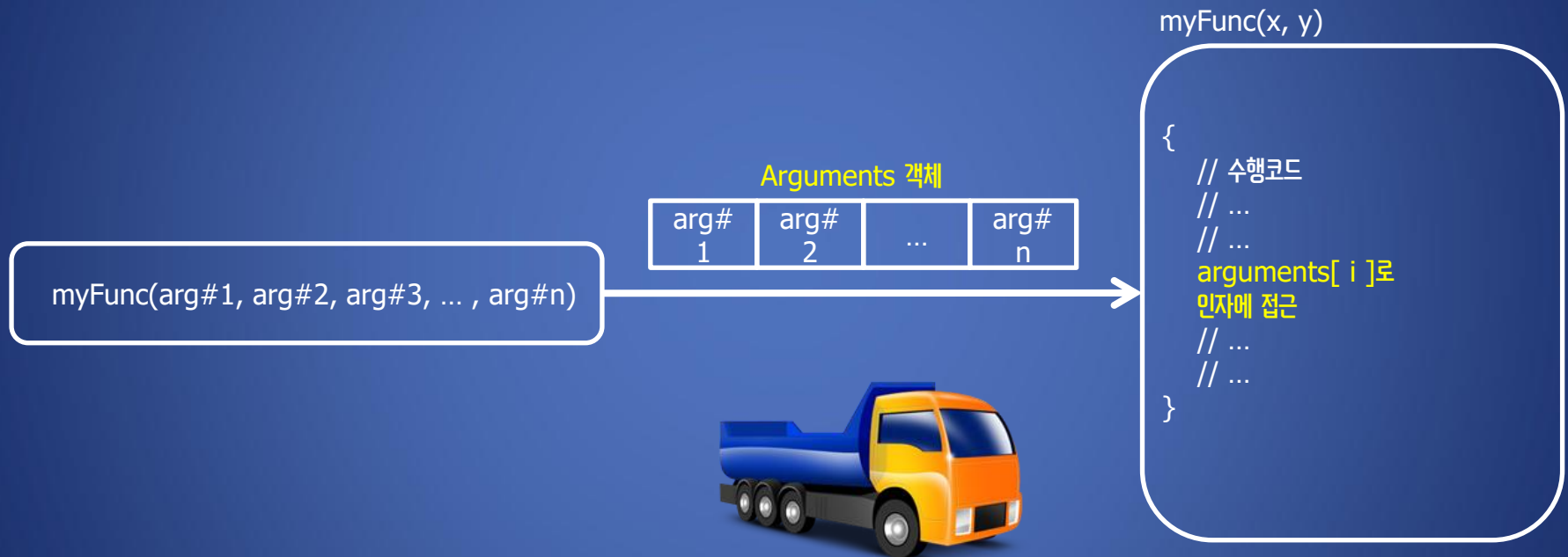
- JavaScript 함수 인자



- JavaScript는 매개변수(parameter)와 인자(argument)의 개수가 달라도 호출이 가능
- Arguments 타입의 객체 이용.
 - 함수가 호출되면 JavaScript가 Arguments 타입의 객체를 생성한 후 인자를 이 객체에 배열처럼 넣어서 함수에 전달. (단, Arguments는 배열이 아님. 단지 배열처럼 사용할 수 있도록 length와 [] 연산자를 지원)
 - Arguments 타입의 객체에 접근할 때는 arguments[] 이용


JavaScript Function 호출

- JavaScript 함수 인자



JavaScript Function 호출

- JavaScript 함수 인자




```
function myFunc(x,y) {  
    var sum = x + y;  
    var total = 0;  
    for(var i=0; i<arguments.length; i++) {  
        total+= arguments[i];  
    }  
    console.log("sum : " + sum);  
    console.log("totla : " + total);  
}
```

```
myFunc();  
myFunc(1);  
myFunc(1,2);  
myFunc(1,2,3);
```

JavaScript Object

- JavaScript 객체 형태

➤ key와 value의 쌍으로 property와 method를 표현



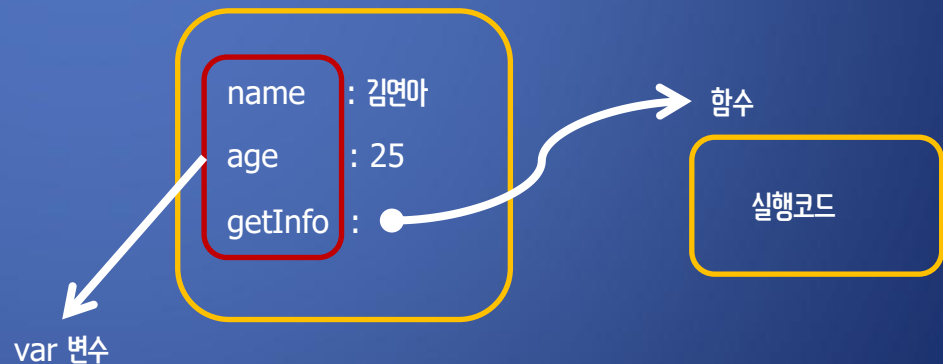
```
var obj = {  
  pName : "YF Sonata",  
  pPrice : 1000,  
  pColor : "White",  
  pCC : 2000,  
  pInfo : function() {  
    console.log("제품의 가격은 :",this.pPrice);  
  }  
}  
  
console.log("제품의 이름은 :",obj.pName);
```

JavaScript Object

- JavaScript 객체 형태

- JavaScript 객체의 member는 모두 var 변수.
- Property와 Method라는 용어를 사용하지만 실제 JavaScript engine은 Property, Method, 변수를 구분하지 않는다. (모두 var 변수)

```
var obj = {  
  name : "김연아",  
  age : 25,  
  getInfo : function() {  
    return this.name;  
  }  
};
```



JavaScript Object

- JavaScript 객체의 key
 - 대부분 개발자가 식별자를 키로 사용
 - 식별자로 사용할 수 없는 단어를 키로 사용할 때는 문자열 사용
(이 경우 대괄호 사용해야 객체 요소 접근 가능)

```
var myObj = {  
  myName : "홍길동",  
  "myAge" : 20,  
  "my Address" : "서울"  
}  
  
console.log("이름 :", myObj.myName);  
console.log("나이 :", myObj.myAge);  
console.log("주소 :", myObj["my Address"]);
```



JavaScript Object

- JavaScript 객체의 동적 제어

➤ 처음 객체를 생성하는 시점 이후에 **객체의 속성을 추가하거나 삭제가능**

```
var myObj = {};  
  
myObj.myName = "홍길동";  
myObj.myAge = 30;  
myObj["myAddress"] = "서울";  
  
myObj.myInfo = function() {  
    console.log("나이 :", myObj.myAge);  
}  
  
delete myObj.myName;
```



JavaScript BOM

- BOM(Browser Object Model)

- 웹 브라우저와 관련된 객체의 집합

- window, location, navigator, history, screen, document 객체

Client-side JavaScript

Core
language

BOM

DOM

JavaScript DOM

- 문서 객체 모델 (DOM : Document Object Model)

- 넓은 의미 : 웹 브라우저가 HTML 페이지 인식하는 방식
- 좁은 의미 : document 객체와 관련된 객체 집합
- HTML 페이지에 태그를 추가, 수정, 제거할 수 있음
- **문서 객체** : HTML 페이지 안의 Element(태그)를 자바스크립트에서 제어할 목적으로 객체화 시킨 것
- **문서객체의 생성, 수정, 삭제와 같은 DOM제어 관련 사항은 jQuery에서 진행.**

JavaScript 실습

- window 객체

- Browser를 제어하기 위한 객체
- open(), close() method
- Timer method
 - setTimeout() , setInterval()
 - clearTimeout(), clearInterval()
- 10초간 document에 현재시간을 찍어주고 그 이후에는 중지하는 스크립트를 작성

JavaScript Event

- Event

- 사용자가 하는 모든 행동 : 키 입력 or 마우스 클릭 etc.

- Event관련 용어

- **Event Name** (이벤트 명) : 처리하고자 하는 Event Type
 - **Event Source** : Event가 발생한 객체
 - **Event 속성** : on + Event Name (onclick)
 - **Event Handler** : Event 속성에 할당된 JavaScript 처리함수
 - **Event 객체** : Event가 발생한 후 자동 생성되는 객체로 Event의 세부 정보를 가지고 있다

JavaScript Basic Q&A

- Question & Review



Contents



JavaScript Basic



jQuery Selector



jQuery Traveling & Manipulation



jQuery Event



jQuery AJAX

jQuery

- jQuery

- 모든 브라우저에서 동작하는 클라이언트 자바스크립트 라이브러리

- 무료로 사용 가능한 오픈 소스 라이브러리

- jQuery 사용 목적

- DOM과 관련된 처리 쉽게 구현
 - 일관된 이벤트 연결 쉽게 구현
 - 시각적 효과 쉽게 구현
 - Ajax 애플리케이션 쉽게 개발



jQuery 사용방법

- jQuery 다운로드

- <http://www.jquery.com>

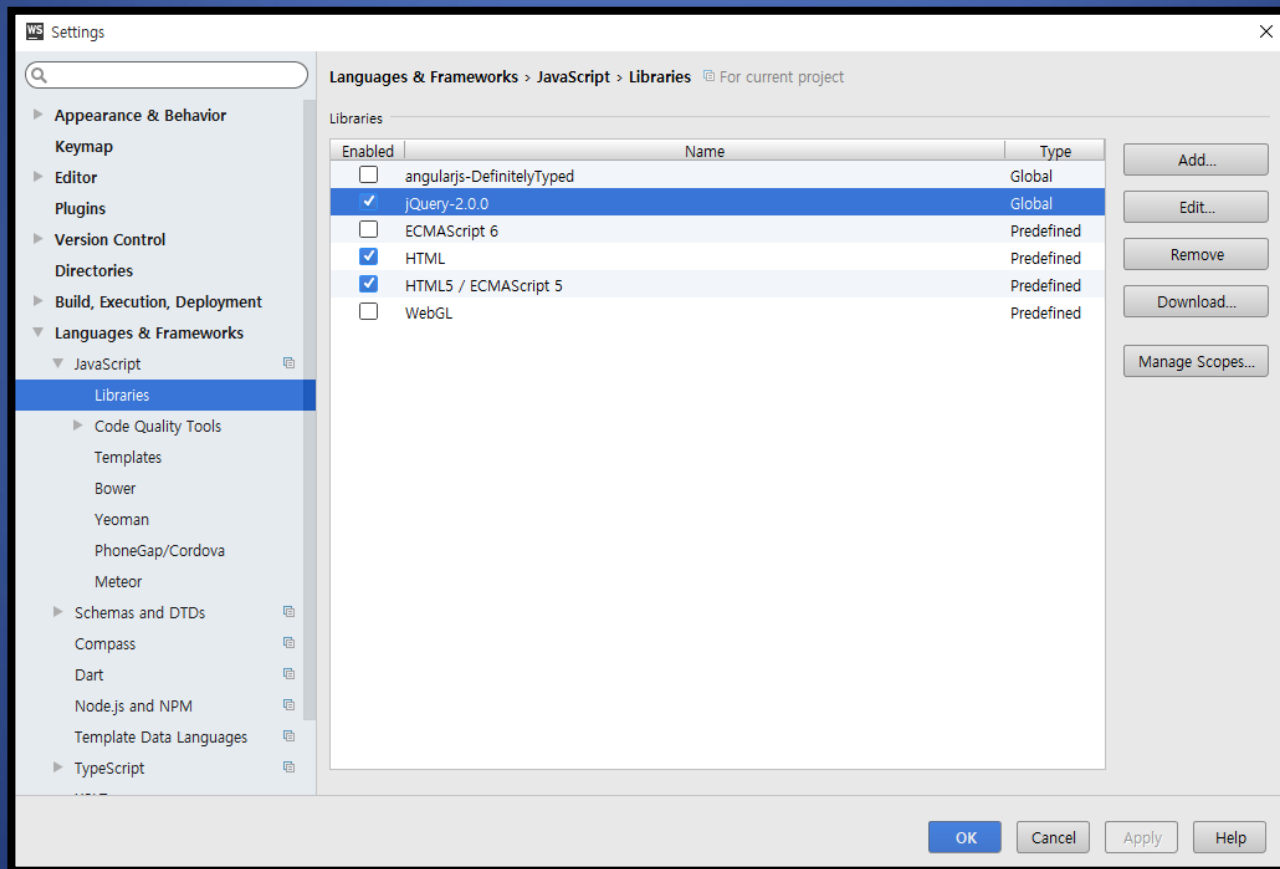
- CDN 방식 (Content Delivery Network)

- 사용자에게 간편하게 Contents를 제공하는 방식
 - Google, MS, jQuery측에서 jQuery를 사용하기 편하게 URL 제공

- 일반적으로, 개발 시점에는 download 방식, 서비스 시점에는 CDN방식을 이용.

jQuery IDE 환경설정

- WebStorm 환경설정 (jQuery highlighting and auto complete)



jQuery Selector 종류

- 전체 선택자 (Universal Selector)
 - 가장 기본적인 선택자
 - 하위에 있는 모든 문서객체를 선택하는 선택자
 - 예제 (모든 문서객체의 color 스타일 속성을 "red"로 변경)

jQuery Selector 종류

- 태그 선택자

- 태그 선택자는 특정한 태그만 선택하는 선택자
- 하나 이상의 태그 선택자를 동시에 사용하고 싶을 때 콤마(,)로 선택자 구분
- 예제

jQuery Selector 종류

- 아이디 선택자 (ID Selector)

- HTML 페이지 내에 존재하는 id속성의 값은 각각 다르다. (unique)
- id 속성의 값을 이용하여 문서 객체를 선택하는 선택자
- 유일한 문서객체를 선택하게 되며 속도 면에서 가장 빠르고 편하다.
- 예제

jQuery Selector 종류

- 클래스 선택자 (Class Selector)
 - class 속성의 값을 이용하여 문서 객체를 선택하는 선택자
 - 예제

jQuery Selector 종류

- 자식 선택자와 후손 선택자
(Child combinator & Descendant combinator)
 - 기본 선택자의 뒤에 붙여 사용하며 기본 선택자의 범위 제한
 - 자식 선택자는 " > " 기호로 표시
 - 후손 선택자는 " " 공백으로 표시
 - 예제

jQuery Selector 종류

- 동위 선택자
(Adjacent sibling combinator & General sibling combinator)
 - element A + element B : element A 바로 다음에 위치하는 element B를 선택
 - element A ~ element B : element A 다음에 위치하는 모든 element B를 선택
 - 예제

jQuery Selector 종류

- 속성 선택자 (Attribute selectors)

- element[attribute] : 특정 속성을 가지고 있는 문서객체 선택
- element[attribute=value] : 속성값이 value값과 같은 문서객체 선택
- element[attribute~=value] : 속성값이 value값을 단어로서 포함하는 문서객체 선택
- element[attribute^=value] : 속성값이 value값으로 시작하는 문서객체 선택
- element[attribute\$=value] : 속성값이 value값으로 끝나는 문서객체 선택
- element[attribute*=value] : 속성값이 value값을 포함하는 문서객체 선택
- 예제

jQuery Selector 실습

● 연습문제

➤ console 출력결과

사과
파인애플
참외
uId
고양이
호랑이
강아지

```
<div>
  <ul>
    <li id="apple">사과</li>
    <li id="pineapple">파인애플</li>
    <li class="myList">참외</li>
  </ul>
  <form action="#" method="post">
    <input type="text" id="uId" size="20">
  </form>
  <ol>
    <li class="myList">고양이</li>
    <li class="myList">호랑이</li>
    <li class="myList">강아지</li>
  </ol>
</div>
```

jQuery Selector 종류

- 상태 선택자 (The UI element states pseudo-classes)

- :checked : 체크된 입력양식을 선택
- :disabled : 비활성화된 입력양식을 선택
- :enabled : 활성화된 입력양식을 선택
- :focus : 현재 포커스가 잡혀있는 입력 양식 선택
- :selected : option객체 중 선택된 객체를 선택
- 예제

jQuery Selector 종류

- 위치 필터 선택자

- :odd : 홀수 번째 위치한 문서 객체를 선택
- :even : 짝수 번째 위치한 문서 객체를 선택
- :first : 첫 번째 위치한 문서 객체를 선택
- :last : 마지막에 위치한 문서 객체를 선택
- 예제

jQuery Selector 종류

- 함수 필터 선택자

- `:contains(string)` : 특정 문자열을 포함하는 문서객체 선택
- `:eq(n)` : n번째 위치하는 문서객체 선택
- `:gtn(n)` : n번째를 초과하는 위치에 있는 문서객체 선택
- `:lt(n)` : n번째 미만의 위치에 있는 문서객체 선택
- `:not(선택자)` : 선택자와 일치하는 않는 문서객체 선택
- `:nth-child(2n+1)` : $2n+1$ 번째에 위치하는 문서객체 선택
- 예제

jQuery Selector Q&A

- Question & Review



Contents



JavaScript Basic



jQuery Selector



jQuery Traveling & Manipulation



jQuery Event



jQuery AJAX

jQuery 반복처리 method

- jQuery의 배열처리

- each() method 사용

- \$.each(object, function(index,item) { })

- \$(selector).each(function(index,item) { })

- 예제



jQuery Method 종류

- jQuery method

- css() method
- addClass() method
- attr() method
- removeAttr() method
- html() method
- text() method
- 예제

jQuery Method 종류

- jQuery method
 - remove(), empty() method
 - \$()
 - \$(A).append(B) , \$(A).prepend(B)
 - \$(A).after(B) , \$(A).before(B)
 - clone() method
 - 예제

jQuery Method Q&A

- Question & Review



Contents



JavaScript Basic



jQuery Selector



jQuery Traveling & Manipulation



jQuery Event



jQuery AJAX

jQuery Event 개요

- jQuery Event

- jQuery 1.7이후부터는 on() method 사용
- `$(selector).on(event, function(event) { ... })`
- Event handler에서의 this는 event source 문서객체를 지칭
- 예제

jQuery Event

- `$(document).ready()`
 - 문서 준비가 완료되면 매개 변수로 전달된 함수를 실행하라는 의미
 - first-class function (일급함수)
 - `window.onload` 비슷한 역할 (실제로는 차이가 있다.)
 - 예제

jQuery Event

- jQuery 단축 Event

- `$(selector).method(function(event) { ... });`
- 사용가능 method
 - blur, focus, focusin, focusout, load
 - resize, scroll, unload, click, dbclick
 - mousedown, mouseup, mousemove, mouseover, mouseout
 - mouseenter, mouseleave, change, select, submit
 - keydown, keypress, keyup, error, ready
- touch이벤트는 이 방식으로 사용불가

jQuery Event

- jQuery Event 제거
 - `off()` : 모든 이벤트 핸들러를 제거
 - `off("eventName")` : 특정 이벤트 핸들러를 제거
 - `one()` : 이벤트를 한 번만 연결

jQuery Event

- jQuery Event 강제 발생
 - `trigger("eventName")`
 - jQuery에서 Event를 강제로 발생시키기 위해 사용
 - 예제

jQuery Event

- jQuery Keyboard Event

- 주의할 것은 keypress event는 한글처리가 까다롭다는 점.
- keydown : keyboard가 눌러질 때 발생
- keypress : 글자가 입력될 때 발생
- keyup : keyboard가 떼어질 때 발생
- 동적으로 글자 수 세기 예제 작성
keydown 이벤트가 발생한 순간에는 글자가 입력돼 있지 않음.
입력한 글자 수를 표시해야 하므로 keyup 이벤트 사용

jQuery Event

- jQuery 입력양식 Event

- change
- focus, focusin, focusout
- blur
- select
- submit, reset

jQuery 최종 실습

- jQuery 연습문제

- 지금까지의 내용을 복습하기 위해 문제 제공
- HTML은 제공하며 HTML에서 요구하는 내용을 jQuery로 작성
- HTML의 내용은 수정하지 않음.

jQuery Event Q&A

- Question & Review



Contents



JavaScript Basic



jQuery Selector



jQuery Traveling & Manipulation



jQuery Event



jQuery AJAX

jQuery AJAX 개요

- \$.ajax() method의 형태

- \$.ajax(options);

- \$.ajax(url, options);

- Ajax가 성공했을 때 자동으로 success 실행

- success handler의 첫 번째 매개 변수는 Ajax가 성공했을 때 서버로 부터 받은 데이터

- 예제

jQuery AJAX 사용 Option

- \$.ajax() method 옵션

- async : 동기, 비동기 지정 (boolean)
- complete : Ajax 완료 event handler 지정 (function)
- data : 요청 매개 변수 지정 (일반적으로 JSON)
- error : Ajax 실패 event handler 지정 (function)
- success : Ajax 성공 event handler 지정 (function)
- timeout : 만료시간 지정 (number)
- type : "GET", "POST" 지정 (string)
- url : 호출 대상 URL 지정 (string)
- dataType : 결과 데이터가 json 혹은 jsonp일 경우 설정 (string)
- jsonp : JSONP방식일 경우 서버로 보내질 hash값에 대한 Key (string)



jQuery AJAX JSON 실습

- 예제

- 서버 프로그램에 문자 전달 (한글처리 포함)
- 서버로부터 JSON 데이터 받아서 처리
 - dataType : "json"

jQuery AJAX JSONP

- AJAX Cross Domain 통신

- XMLHttpRequest 객체는 보안상 이유로 JavaScript source를 가져온 서버하고만 통신할 수 있음.
- 불특정 다수의 도메인 서버에 접근 : cross domain 통신
- 가장 쉬운 방법은 JSONP를 이용하는 방식.
(단, JSON을 이용하는 경우에만 가능)
- 예제

jQuery AJAX 실습환경 세팅

- jQuery AJAX 연습문제를 위한 setting
 - MySQL설치 및 설정
 - 서버 프로그램 설치 (Servlet 프로그램 제공)
 - 책 제목에 대한 keyword를 입력 받아서 DB로부터 해당 책들의 세부정보를 JSON형식으로 return하는 프로그램
 - 프로그램 설치 후 실행결과 확인

jQuery AJAX 실습





- jQuery AJAX Book Search Screenshot

Library Management System

자바

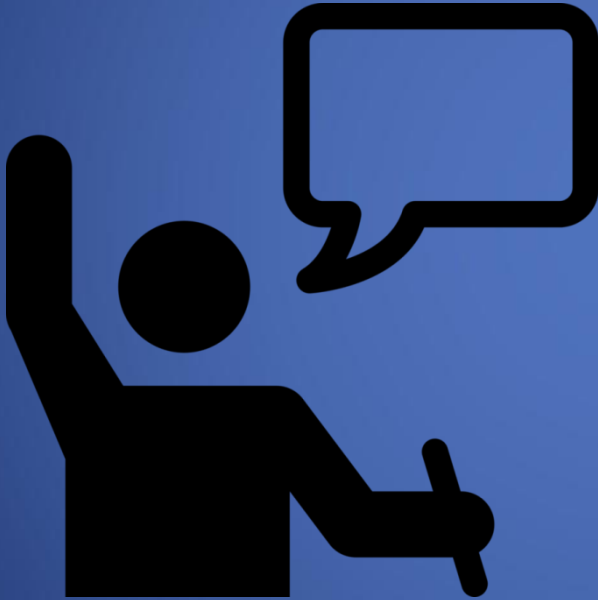
Book 전체 List

Book List

표지	Title	ISBN	Price	Author	상세보기	삭제
	IT EXPERT, 모바일 자바 프로그래밍 출판일 : 2002년 09월 페이지수 : 528 출판사 : 한빛미디어(주) 부록 : CD 1장	89-7914-206-4	23000	권기경, 박용우	상세보기	삭제
	Head First Java: 뇌 회로를 자극하는 자바 학습법(개정판) 출판일 : 2002년 09월 페이지수 : 528 출판사 : 한빛미디어(주) 부록 : CD 1장	89-7914-371-0	28000	케이지 시에라, 버트 베이츠	상세보기	삭제
	모던 웹을 위한 HTML5 프로그래밍 : 한 권으로 끝내는 자바스크립트 프레임워크 Backbone.js, Angular.js, Knockout.js 활용 출판일 : 2013년 08월 페이지수 : 720 출판사 : 한빛미디어(주) 부록 : 없음	978-89-6848-041-6	30000	윤인성	상세보기	삭제
	모던 웹을 위한 JavaScript + jQuery 입문(개정판) : 자바스크립트에서 제이쿼리, 제이쿼리 모바일까지 한 권으로 끝낸다 출판일 : 2013년 08월 페이지수 : 720 출판사 : 한빛미디어(주) 부록 : 없음	978-89-6848-042-3	32000	윤인성	상세보기	삭제

jQuery AJAX

- Question & Review



**Thanks for
listening**

