chapter05

JavaScript
jQuery
Ajax

1. **JavaScript**
2. jQuery
3. Ajax, json

■브라우저 내에 내장된 자바스크립트 실행엔진 (해석기)를 통해 실행되어지고 브라우저 화면에 반영된다.

■본문 어디에도 삽입되어 실행될 수 있다 실습 ex00.html

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">

    <script type="text/javascript">
        document.write( "header" );
    </script>

<title>Mysite</title>
</head>
...
```

```
...
<body>

    <script type="text/javascript">
        document.write( “body" );
    </script

</body>

    <script type="text/javascript">
        document.write( "body이후" );
    </script>

</html>
```

■기본데이터타입

- Number, String, Boolean
- 변수선언시 자료형을 선언할 필요가 없다

■null 과 undefined

- null : 아무런 값도 나타내지 않는 특수한 값이다.
- undefined: 선언은 되었지만 값이 할당되지 않은 경우다.

```
<script type="text/javascript">

    var myVar, myVar2 = null;
    alert( myVar + ", " + myVar2 );
    alert( myVar == myVar2 );
    alert( myVar === myVar2 );

</script>
```

■변수이름 규칙

- 자바스크립트의 예약어(키워드)는 사용할 수 없습니다(if, true, false, break, null 등)
- 영문자 혹은 밑줄(_)로 시작해야 하며, 숫자로는 시작할 수 없음
- 문자의 대문자(A~Z), 소문자(a~z), 숫자(0~9), 밑줄만 사용 가능
- 특수 문자는 _와 $만 허용되고, 공백 문자를 포함할 수 없음.

■선언없이 대입시 변수 타입이 결정된다.

```
<script type="text/javascript">

    var v01 = "This is Test";
    alert( typeof v01 );

    var v02 = 20;
    alert( typeof v02 );

</script>
```

■변수의 범위 var 를 붙여서 사용하는 것이 좋다.

```
<script type="text/javascript">

    var name = "global";    // 전역 변수를 선언

    function checkscope(){
            var name = "local"; // 지역 변수를 선언
            console.log(name);  // 전역 변수가 아닌 지역 변수를 사용
    }
    checkscope();           // 출력 결과: "local"

</script>
```

```
var name = "global";    // 전역 변수를 선언
    function checkscope(){
            name = "local";     // 전역 변수를 변경
            name2 = "local";    // 암묵적으로 새 전역 변수가 선언됨
    }
checkscope();
console.log(name);      // output: "local"
console.log(name2);     // output: "local"
```

- name 에 var를 붙이면

■문장의 끝 세미콜론

•반드시 붙이지 않아도 되지만 붙여 쓴다.

■Camel 표기법이 기본

```
var guestbookNo ;
function getGuestbookList(){} ;
```

■주석

•한줄주석 //내용

•여러줄주석

```
/*
   내용1
   내용2
*/
```

■if, if~slse 등은 다른 언어와 유사

■switch 다른 언어와 유사

■for, while, do~while 다른 언어와 유사

■실습 00.html
자바스크립트로 다음과 같이 출력되도록 작성하시요.

2 * 1 = 2 2 * 2 = 4 2 * 3 = 6 2 * 4 = 8 2 * 5 = 10 2 * 6 = 12 2 * 7 = 14 2 * 8 = 16 2 * 9 = 18
3 * 1 = 3 3 * 2 = 6 3 * 3 = 9 3 * 4 = 12 3 * 5 = 15 3 * 6 = 18 3 * 7 = 21 3 * 8 = 24 3 * 9 = 27
4 * 1 = 4 4 * 2 = 8 4 * 3 = 12 4 * 4 = 16 4 * 5 = 20 4 * 6 = 24 4 * 7 = 28 4 * 8 = 32 4 * 9 = 36
5 * 1 = 5 5 * 2 = 10 5 * 3 = 15 5 * 4 = 20 5 * 5 = 25 5 * 6 = 30 5 * 7 = 35 5 * 8 = 40 5 * 9 = 45
6 * 1 = 6 6 * 2 = 12 6 * 3 = 18 6 * 4 = 24 6 * 5 = 30 6 * 6 = 36 6 * 7 = 42 6 * 8 = 48 6 * 9 = 54
7 * 1 = 7 7 * 2 = 14 7 * 3 = 21 7 * 4 = 28 7 * 5 = 35 7 * 6 = 42 7 * 7 = 49 7 * 8 = 56 7 * 9 = 63
8 * 1 = 8 8 * 2 = 16 8 * 3 = 24 8 * 4 = 32 8 * 5 = 40 8 * 6 = 48 8 * 7 = 56 8 * 8 = 64 8 * 9 = 72
9 * 1 = 9 9 * 2 = 18 9 * 3 = 27 9 * 4 = 36 9 * 5 = 45 9 * 6 = 54 9 * 7 = 63 9 * 8 = 72 9 * 9 = 81

■함수

•기명 함수표현식

```
<script type="text/javascript">

    function company() {
      console.log("기명함수 실행");
    };

    company();
</script>
```

•익명 함수표현식

```
<script type="text/javascript">

    var company = function() {
          console.log("익명함수 실행");
    };

    company();
</script>
```

■배열

```
<script type="text/javascript">

    var array = [273, 32, 103, 57, 52];

</script>
```

```
<script type="text/javascript">

    var array = [ 273, "String", true, function(){}, {}, [273,103] ];
    console.log(array);
    alert(array);

    console.log(array[1]);
    alert(array[1]);

</script>
```

■객체

```
<script type="text/javascript">
	/*
	var guestbookVo = {};
	 */

	var guestbookVo = {
			no : 1,
			name : "황일영",
			password : "1234",
			content : "안녕하세요 첫번째 방문"
	};

	 console.log(guestbookVo.content);
</script>
```

■이벤트

```
<div>
    <input id="btn" type="button" VALUE="js테스트버튼" />
    <h1 id="display">이름출력영역</h1>
</div>
```

```
<script type="text/javascript">
    window.onload = function(){
        var testButton = document.getElementById("btn");
        var testArea = document.getElementById("display");

        console.log(testArea);
        testButton.onclick = function(){
            testArea.innerHTML = "<font color='red'>홍길동<font>";

        };
    };

</script>
```

chapter05

JavaScript
jQuery
Ajax

1. JavaScript
2. **jQuery**
3. Ajax, json

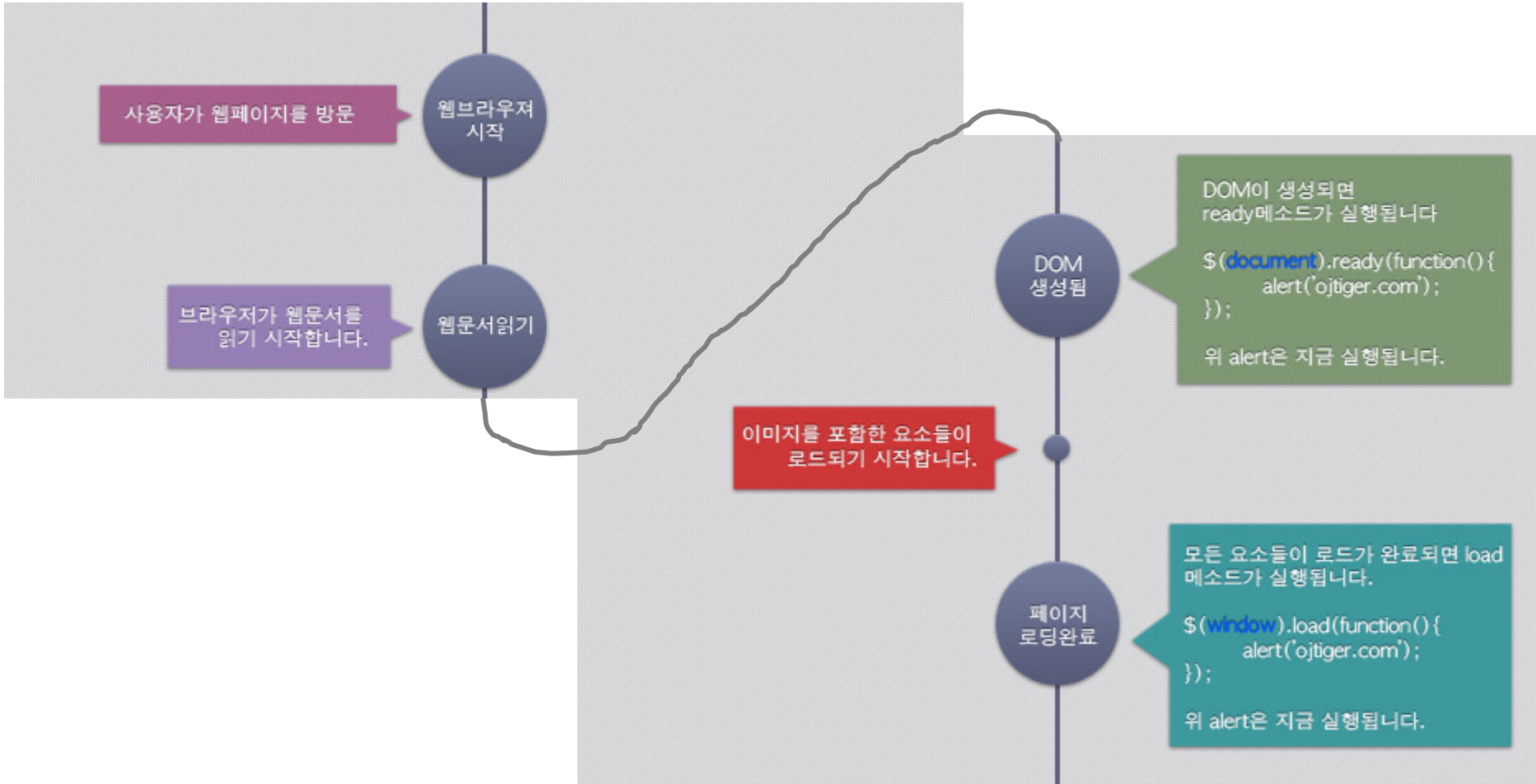
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<script type="text/javascript" src="/jquery/jquery/jquery-1.12.4.js"></script>
<title>Insert title here</title>
</head>
<body>

<div>
    <input id="btn" type="button" VALUE="js테스트버튼" />
    <h1 id="display">이름출력영역</h1>
</div>

</body>

<script type="text/javascript">
    $("#btn").on("click",function(){
        $("#display").html("<font color='red'>홍길동<font>");
    });
</script>

</html>
```


사용자가 웹페이지를 방문
웹브라우저 시작
브라우저가 웹문서를 읽기 시작합니다.
웹문서읽기
DOM 생성됨
DOM이 생성되면 ready메소드가 실행됩니다
$(document).ready(function(){
alert('ojtiger.com');
});
위 alert은 지금 실행됩니다.
이미지를 포함한 요소들이 로드되기 시작합니다.
페이지 로딩완료
모든 요소들이 로드가 완료되면 load 메소드가 실행됩니다.
$(window).load(function(){
alert('ojtiger.com');
});
위 alert은 지금 실행됩니다.

■선택자(selector)

•HTML Element를 선택하는 역할을 한다.

```
var object = $( “selector” )
```

•문서에서 Element 를 가져온 후, 반환된 객체 함수를 사용하여 Element를 조작하게 된다.

■선택자(selector) – 태그

- 특정 html 태그를 컨트롤 하기 위해 사용

■실습 ex02

- css() 함수를 이용하여
  li 엘리먼트의 색을 빨간색으로 변경하세요

- css() 함수를 이용하여
  p 엘리먼트의 색을 파란색으로 변경하세요

■선택자(selector) – #ID

- •특정 id 속성을 가진 html태그를 컨트롤하기 위해 사용
- •ID 값에 # 을 붙인다.
- •ID는 한문서에 1개(유일)만 있어야 한다.

■실습 ex03

- •css() 함수를 이용하여
  id 가 second 인 li엘리먼트의 색을 빨간색으로 변경하세요

- •ID값을 변경해서 테스트 하세요

- •같은 ID가 있는 경우 결과를 확인하세요

■선택자(selector) – .클래스

- 특정 class 속성을 가진 html태그를 컨트롤하기 위해 사용
- .(dot)에 class 속성갑을 지정하여 선택
- class는 한문서에 여러 개 있을 수 있다

■실습 ex04

- css() 함수를 이용하여
  class 가 red 인 li엘리먼트의 색을 빨간색으로 변경하세요

- css() 함수를 이용하여
  class 가 blue 인 li엘리먼트의 색을 파란색으로 변경하세요

- 한 개의 선택자로
  class 가 red 와 blue 인 li엘리먼트의 색을 파란색으로 변경하세요.

■선택자(selector) – [속성], [속성=값]

- 특정 속성을 가진 html태그를 컨트롤하기 위해 사용

■실습 ex05

- css() 함수를 이용하여
  class 속성을 가진 li엘리먼트의 색을 빨간색으로 변경하세요

- css() 함수를 이용하여
  class 의 값이 first 인 li엘리먼트의 색을 파란색으로 변경하세요

■객체조작 - .text(문자열)

•파라미터로 문자열을 넘기면 태그 안의 텍스트를 괄호안의 문자열로 변경한다.

■객체조작 - .text()

•파라미터가 없으면 태그에 포함된 텍스트를 가져온다.

■실습 ex06

•p1태그의 텍스트를 "가나다라마바사" 로 변경해 보세요

•p2태그의 텍스트값을 가져와 console에 찍어보세요

■객체조작 - .html(html문자열)

•파라미터로 html 문자열을 넘기면 태그가 반영된다.

■객체조작 - .html()

•파라미터가 없으면 태그에 포함된 html을 가져온다.

■실습 ex07

•p1태그의 내용을 "가나다라마바사아자차카타라하" 로 변경해 보세요

•p2태그의 내용을 text() 를 이용하여 ""가나다라마바사아자차카타라하" 로 변경해 보세요

•p1태그의 html을 값을 가져와 console에 찍어보세요

•p2태그의 내용을 text() 를 이용하여 가져와 console에 찍어보세요

■객체조작 - .prepend(html문자열)

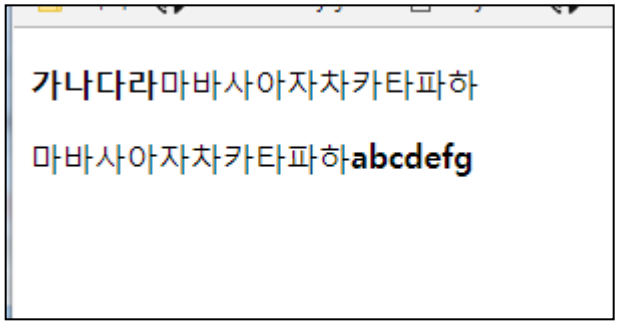
•지정한 태그안의 **내용 앞**에 html문자열을 추가한다.

■객체조작 - .append(html문자열)

•지정한 태그안의 **내용 뒤**에 html문자열을 추가한다.

■실습 ex08

•prpend(), append() 을 이용하여 아래와 같이 출력되도록 하세요



■객체조작 - .remove()

•선택자로 지정된 태그들을 제거한다.

■실습 ex09

•리스트에서 삭제할 아이템을 삭제해 보세요.

- 첫번째 아이템입니다.
- 두번째 아이템입니다.
- 삭제할 아이템입니다.
- 네번째 아이템입니다.
- 삭제할 아이템입니다.
- 다섯 번째 아이템입니다.

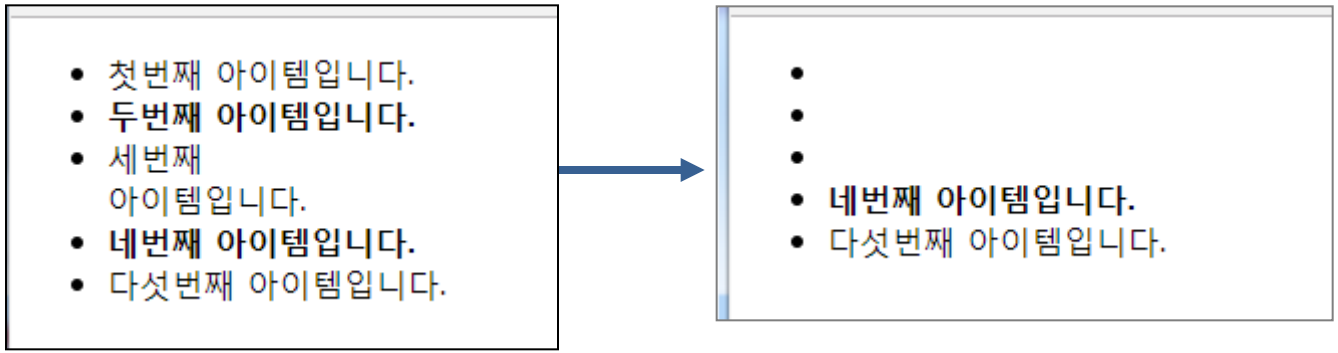
- 첫번째 아이템입니다.
- 두번째 아이템입니다.
- 네번째 아이템입니다.
- 다섯 번째 아이템입니다.

■객체조작 - .empty()

- 선택자로 지정된 태그의 하위의 텍스트와 태그들을 제거한다.
- 선택된 태그는 삭제되지 않음

■실습 ex10

- 아래와 같이 되도록 텍스트값을 제거해 보세요

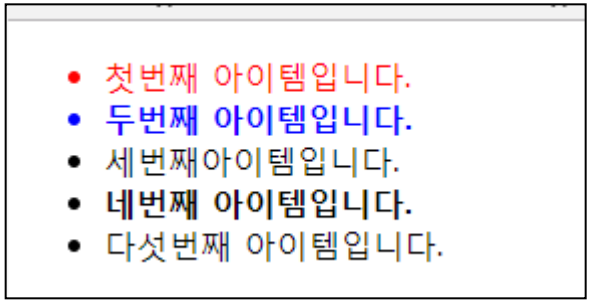


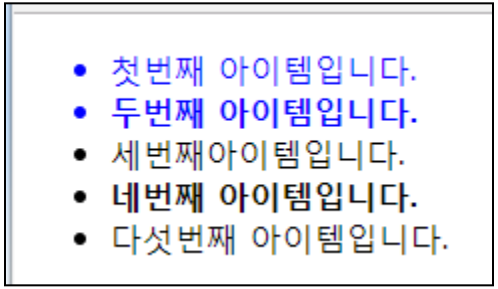
■객체조작 - .addClass("클래스명"), .removeClass("클래스명")

•선택자로 지정된 태그의 에 클래스가 추가/제거 된다.

■실습 ex11

•아래와 같이 되도록 class값을 변경해 보세요



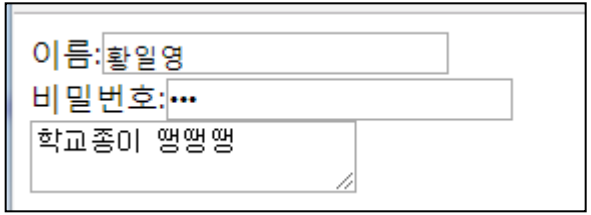


■객체조작 - .val() .val("값")

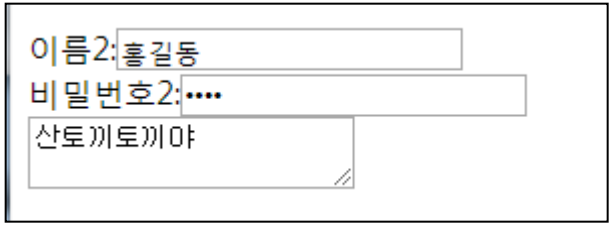
- 주로 form 엘리먼트에서 사용됨
- form 태그의 value값을 읽어오거나, value값을 입력하는데 사용

■실습 ex12

- 아래와 같이 되도록 val() 함수를 이용하여 value값을 추가해 보세요



- 아래의 value값을 읽어 console에 출력해 보세요



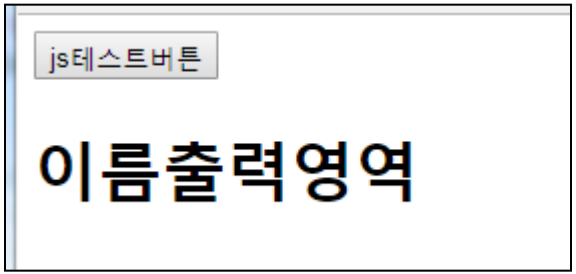
■이벤트 - click

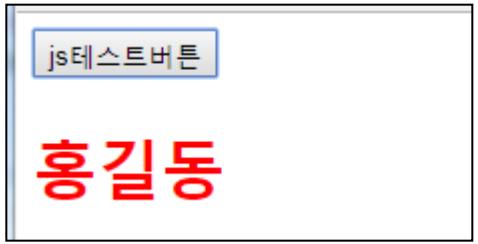
•마우스가 클릭 되었을때

```
$(셀렉터).on("click", function(){
    /* 셀렉터로 지정한 태그가 클릭되었을 때 실행하는 처리 */
});
```

■실습 ex13

•버튼을 클릭하면 아래와 같이 되도록 작성하세요





■객체조작 - $(this)

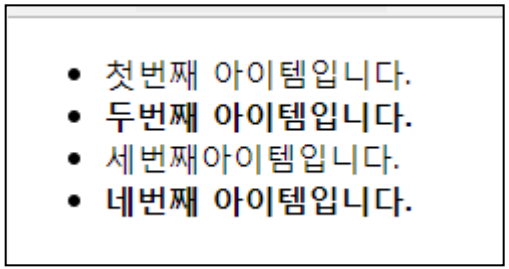
•현재 클릭(선택) 한 엘리먼트

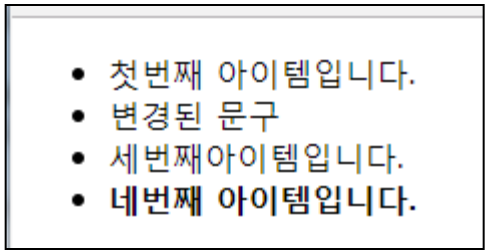
```
$(셀렉터).on("click", function(){
    var $this = $(this);
     console.log($this);
});
```

■실습 ex14

•클릭한 li 엘리먼트의 id 값을 console 에 출력하세요 attr() 이용

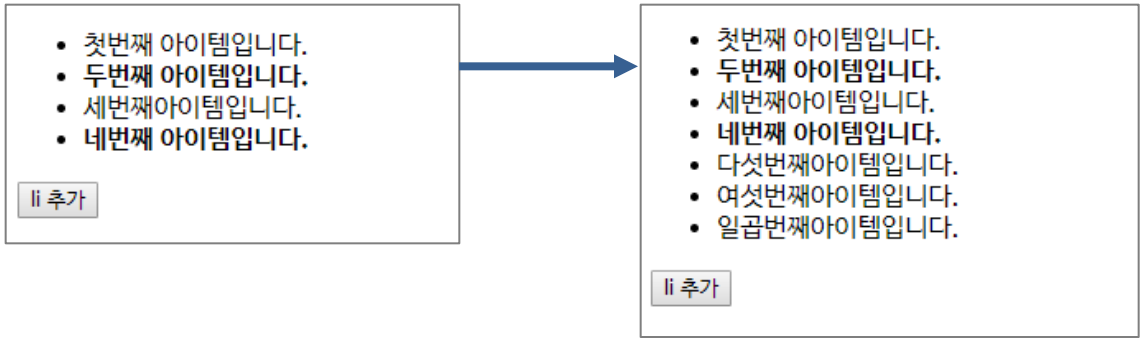
•클릭한 li 엘리면트의 출력값을 변경해보세요





■실습 ex15

•버튼을 클릭하여 li 요소를 추가해 보세요



•세번째 요소 클릭시 텍스트를 콘솔창에 출력해 보세요

•여섯번째 요소 클릭시 텍스트를 콘솔창에 출력해 보세요

chapter05

JavaScript
jQuery
Ajax

1. JavaScript
2. jQuery
3. **Ajax, json**

■Ajax (Asynchronous Javascript XML)

- JavaScript를 이용해서 서버에서 데이터를 가져와 페이지 전체의 갱신없이 특정부분만 변경
- 주로 JSON으로 서버와 데이터를 주로 주고 받는다.

```
$("button").click(function(){

        $.ajax({

                url : "${pageContext.request.contextPath }/api/gb/add",
                type : "post",
                contentType : "application/json",
                data : {name: ″홍길동"},

                dataType : "json",
                success : function(result){
                        /*성공시 처리해야될 코드 작성*/
                },
                error : function(XHR, status, error) {
                        console.error(status + " : " + error);
                }
        });
});
```

■Json (JavaScript Object Notation)

- 경량의 DATA-교환 형식

■메시지 컨버터

- XML 이나 JSON을 이용한 AJAX 기능이나 웹서비스 개발에 이용
- HTTP 요청 메세지 본문(Request Body), HTTP 응답 메세지 본문(Response Body)을 통째로 메세지로 다루는 방식
- 파라미터의 @RequestBody, 메소드에 @ResponseBody를 이용
- 메세지 컨버터는 AnnotationMethodHandlerAdapter를 통해 하나 이상의 컨버터가 등록, 선택 동작하게 된다.
- 응답(Response)의 경우 해당 핸들러 메소드에 @ResponseBody 와 함께 반환되는 객체의 종류에 따라 메세지 컨버터가 선택되고 응답바디 내용이 채워져 브라우저로 전달된다.

■MappingJacksonHttpMessageConverter :

•pom.xml 에 jackson core library 추가

```
<!-- jackson -->
<dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-databind</artifactId>
        <version>2.8.7</version>
</dependency>
```

■MappingJacksonHttpMessageConverter :

•spring-servlet.xml

```
<mvc:annotation-driven>
	<mvc:message-converters>
		<bean class="org.springframework.http.converter.json.MappingJackson2HttpMessageConverter">
			<property name="supportedMediaTypes">
				<list>
					<value>application/json; charset=UTF-8</value>
				</list>
			</property>
		</bean>
	</mvc:message-converters>
</mvc:annotation-driven>
```