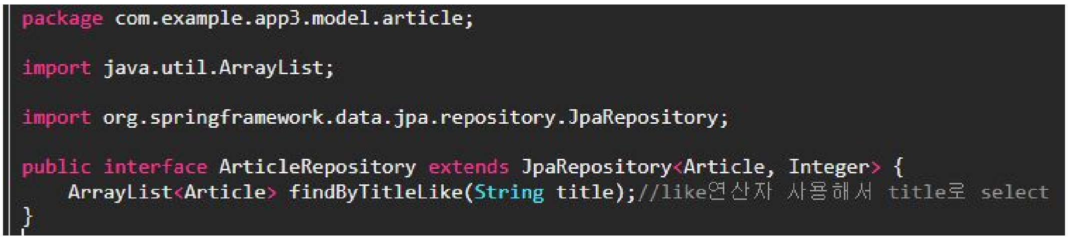
- 백단과 뷰단의 작업 분리
  : 백단에서는 데이터 작업, 뷰단에서는 화면작업
  서로 json/xml 파일로 데이터를 교환함.
  따라서 json/xml 파일 자주 쓰일 것

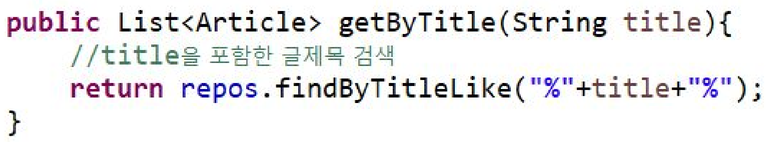
1. [프로젝트 app3]

- ArticleService.java
  Article Repository에서 제공되는 함수만 사용해서 함수 정의.
  기존에 존재하는 함수 이외의 함수는 따로 만들어줘야 함.
  즉, 해당 인터페이스 상속 받는 클래스에서 추상메서드로 만들어 놓은 기능 상세 구현.
  **save(u)에서, 시퀀스 num이 같으면 update, 다르면 insert
- ArticleRepository.java
  함수 이름 중요. 시스템이 인식할 수 있는 이름으로 설정.



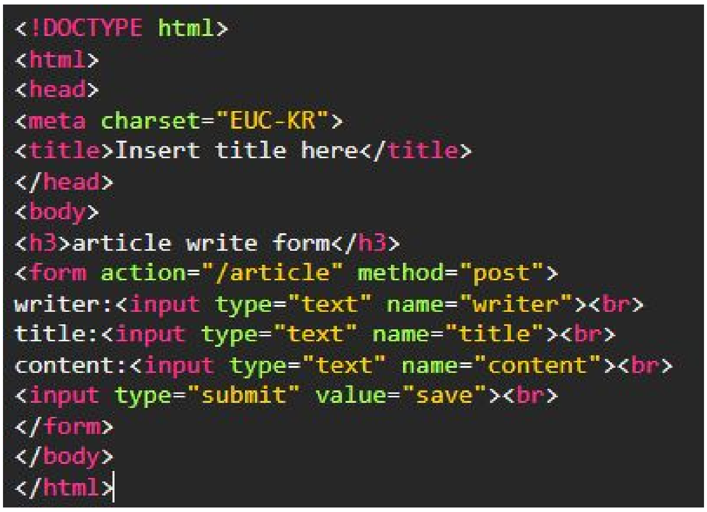
```
package com.example.app3.model.article;

import java.util.ArrayList;

import org.springframework.data.jpa.repository.JpaRepository;

public interface ArticleRepository extends JpaRepository<Article, Integer> {
    ArrayList<Article> findByTitleLike(String title);//like연산자 사용해서 title로 select
}
```

  - service에서 해당함수 사용



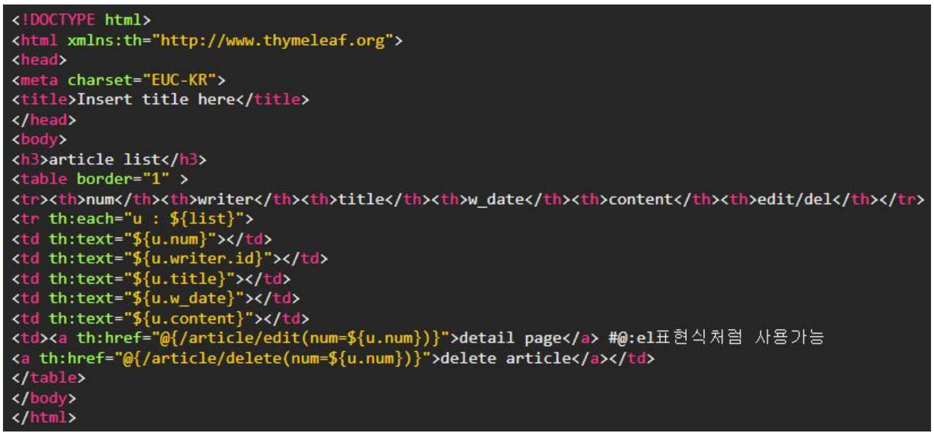
```
public List<Article> getByTitle(String title){
    //title을 포함한 글제목 검색
    return repos.findByTitleLike("%"+title+"%");
}
```

- form.html::글 작성 폼



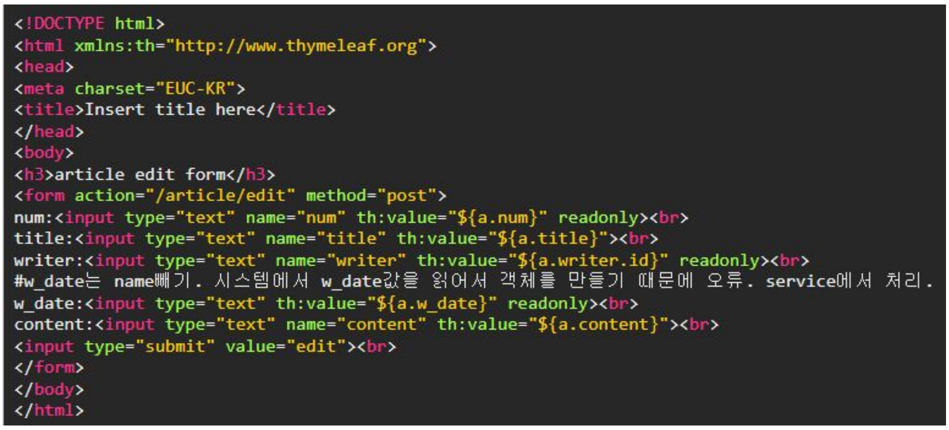
```
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
</head>
<body>
<h3>article write form</h3>
<form action="/article" method="post">
writer:<input type="text" name="writer"><br>
title:<input type="text" name="title"><br>
content:<input type="text" name="content"><br>
<input type="submit" value="save"><br>
</form>
</body>
</html>
```

- list.html::글 목록



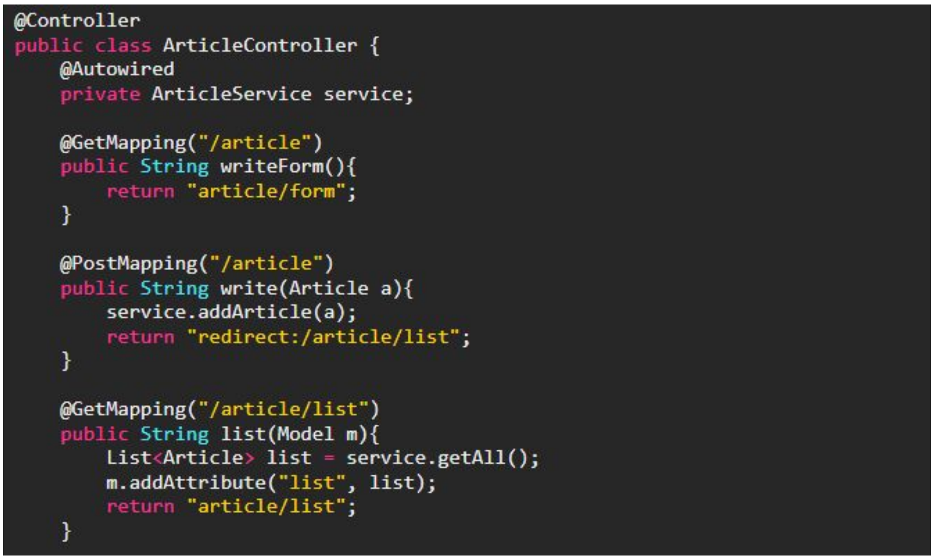
```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
</head>
<body>
<h3>article list</h3>
<table border="1" >
<tr><th>num</th><th>writer</th><th>title</th><th>w_date</th><th>content</th><th>edit/del</th></tr>
<tr th:each="u : ${list}">
<td th:text="${u.num}"></td>
<td th:text="${u.writer.id}"></td>
<td th:text="${u.title}"></td>
<td th:text="${u.w_date}"></td>
<td th:text="${u.content}"></td>
<td><a th:href="@{/article/edit(num=${u.num})}">detail page</a> #@:el표현식처럼 사용가능
<a th:href="@{/article/delete(num=${u.num})}">delete article</a></td>
</table>
</body>
</html>
```

- edit.html:: 글 수정 폼

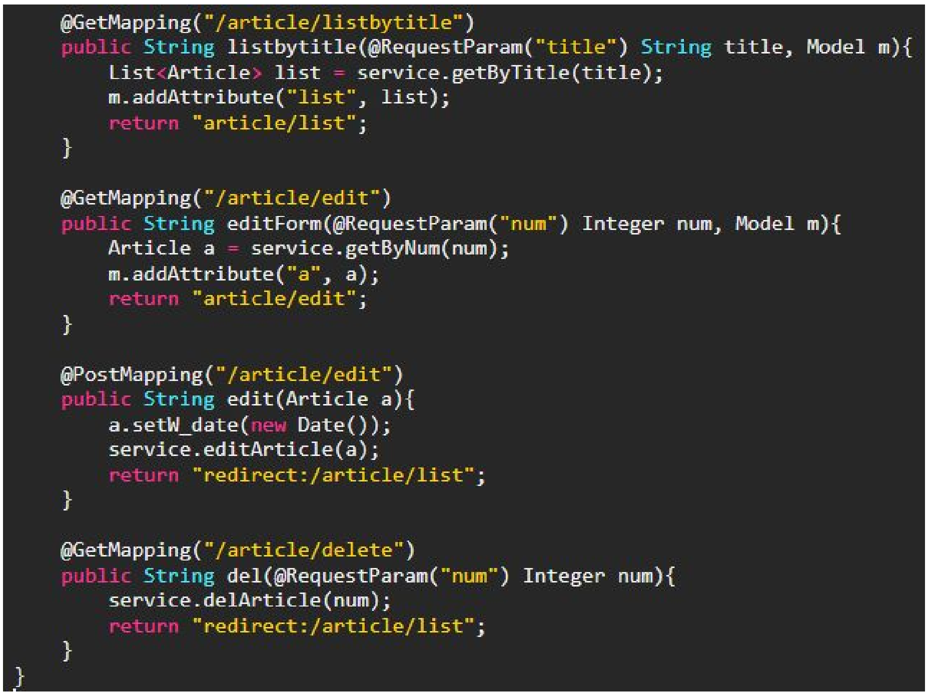


```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
</head>
<body>
<h3>article edit form</h3>
<form action="/article/edit" method="post">
num:<input type="text" name="num" th:value="${a.num}" readonly><br>
title:<input type="text" name="title" th:value="${a.title}"><br>
writer:<input type="text" name="writer" th:value="${a.writer.id}" readonly><br>
#w_date는 name빼기. 시스템에서 w_date값을 읽어서 객체를 만들기 때문에 오류. service에서 처리.
w_date:<input type="text" th:value="${a.w_date}" readonly><br>
content:<input type="text" name="content" th:value="${a.content}"><br>
<input type="submit" value="edit"><br>
</form>
</body>
</html>
```

- ArticleController.java



```
@Controller
public class ArticleController {
    @Autowired
    private ArticleService service;

    @GetMapping("/article")
    public String writeForm(){
        return "article/form";
    }

    @PostMapping("/article")
    public String write(Article a){
        service.addArticle(a);
        return "redirect:/article/list";
    }

    @GetMapping("/article/list")
    public String list(Model m){
        List<Article> list = service.getAll();
        m.addAttribute("list", list);
        return "article/list";
    }
```



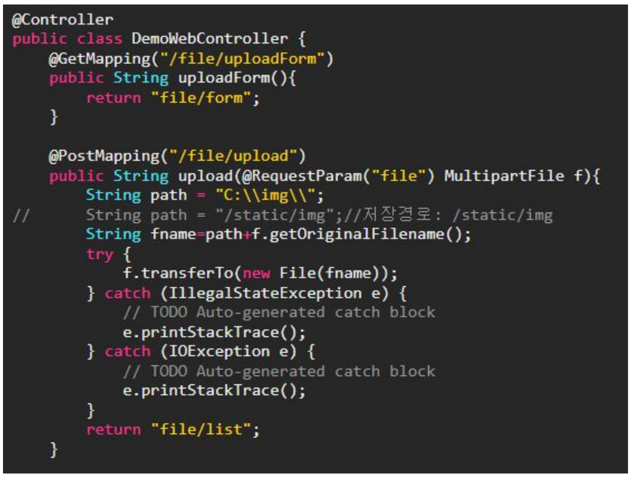
```java
    @GetMapping("/article/listbytitle")
    public String listbytitle(@RequestParam("title") String title, Model m){
        List<Article> list = service.getByTitle(title);
        m.addAttribute("list", list);
        return "article/list";
    }

    @GetMapping("/article/edit")
    public String editForm(@RequestParam("num") Integer num, Model m){
        Article a = service.getByNum(num);
        m.addAttribute("a", a);
        return "article/edit";
    }

    @PostMapping("/article/edit")
    public String edit(Article a){
        a.setW_date(new Date());
        service.editArticle(a);
        return "redirect:/article/list";
    }

    @GetMapping("/article/delete")
    public String del(@RequestParam("num") Integer num){
        service.delArticle(num);
        return "redirect:/article/list";
    }
}
```

2. 파일업로드 기능

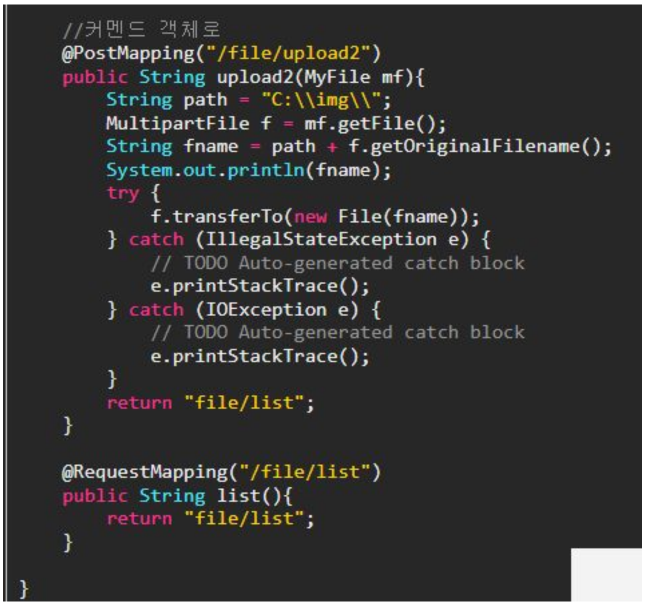
- 테스트용 컨트롤러

  *타임리프와 jsp: https://jongminlee0.github.io/2020/03/12/thymeleaf/

  *pom.xml(의존성 주입): 현재 jsp 안들어가있기 때문에 사용 불가. html로 생성

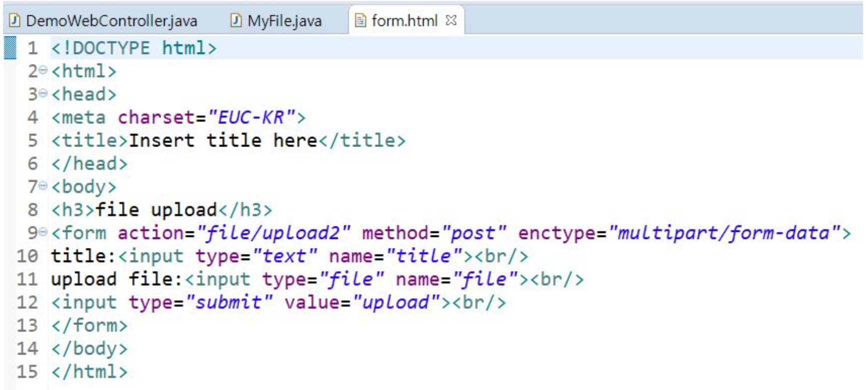


```java
@Controller
public class DemoWebController {
    @GetMapping("/file/uploadForm")
    public String uploadForm(){
        return "file/form";
    }

    @PostMapping("/file/upload")
    public String upload(@RequestParam("file") MultipartFile f){
        String path = "C:\\img\\";
//      String path = "/static/img";//저장경로: /static/img
        String fname=path+f.getOriginalFilename();
        try {
            f.transferTo(new File(fname));
        } catch (IllegalStateException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return "file/list";
    }
```



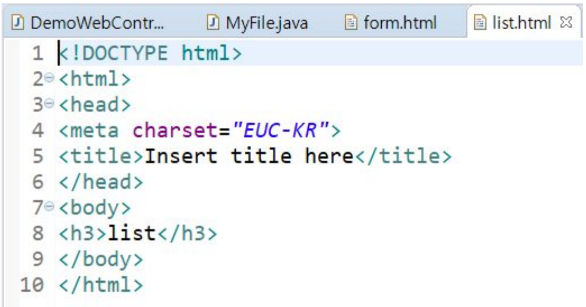
```java
    //커멘드 객체로
    @PostMapping("/file/upload2")
    public String upload2(MyFile mf){
        String path = "C:\\img\\";
        MultipartFile f = mf.getFile();
        String fname = path + f.getOriginalFilename();
        System.out.println(fname);
        try {
            f.transferTo(new File(fname));
        } catch (IllegalStateException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return "file/list";
    }

    @RequestMapping("/file/list")
    public String list(){
        return "file/list";
    }

}
```

- form.html:: 파일 업로드 폼
  *저장경로: c:/img/



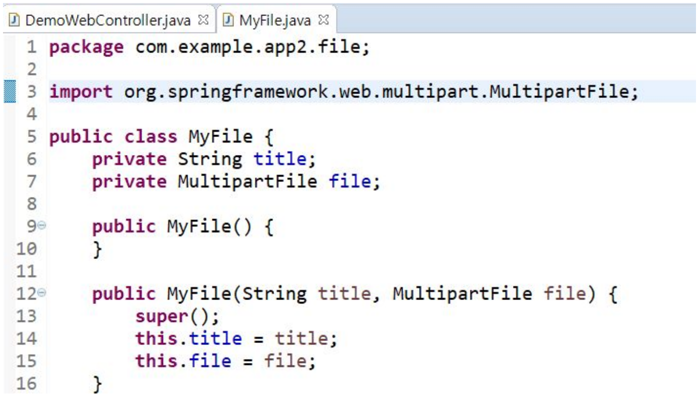
```html
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
</head>
<body>
<h3>file upload</h3>
<form action="file/upload2" method="post" enctype="multipart/form-data">
title:<input type="text" name="title"><br/>
upload file:<input type="file" name="file"><br/>
<input type="submit" value="upload"><br/>
</form>
</body>
</html>
```

- list.html



```html
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
</head>
<body>
<h3>list</h3>
</body>
</html>
```

- dd



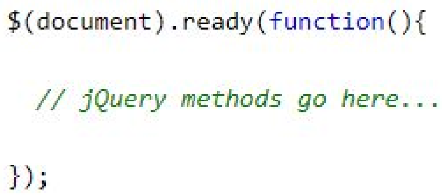
```java
package com.example.app2.file;

import org.springframework.web.multipart.MultipartFile;

public class MyFile {
    private String title;
    private MultipartFile file;

    public MyFile() {
    }

    public MyFile(String title, MultipartFile file) {
        super();
        this.title = title;
        this.file = file;
    }
```

3. j query

- 제이쿼리란?

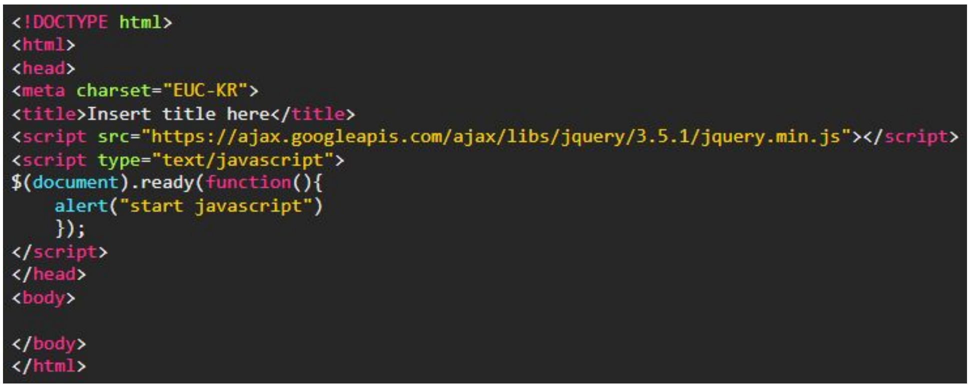
  jQuery는 존 레식이 2006년에 발표한 크로스 플랫폼을 지원하는 경량의 자바스크립트 라이브러리이다. HTML 문서의 탐색이나 조작, 이벤트 핸들링, 애니메이션, Ajax등을 멀티 브라우저를 지원하는 API를 통해 더욱 간편하게 사용할 수 있다.
- :: javascript가 수행하던 부분을 j query로 대체하기.
  cf. https://www.w3schools.com/jquery/default.asp

1) jQuery Tutorial 제이쿼리 튜토리얼
   a. 제이쿼리 사용법
      - 다운로드
      - CDN include로 사용 *인터넷없이 사용불가
   b. 제이쿼리 문법 https://www.w3schools.com/jquery/jquery_syntax.asp
      - 기본문법: $(사용할 셀렉터 객체).함수명()
        예시
        $(this).hide(): 현재 요소를 숨김
        $(“p”).hide(): 모든 p 태그를 숨김
        $(“.test.”).hide(): 모든 text 클래스 그룹의 태그를 숨김
        $(“#test”).hide(): id가 test인 태그를 숨김
      - 함수 세팅

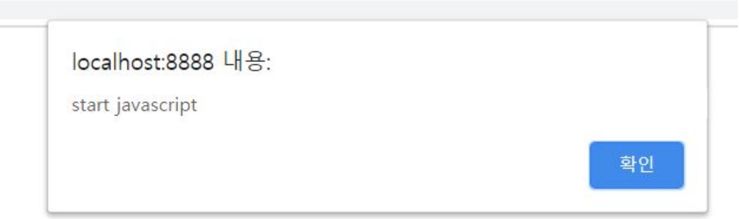


```javascript
$(document).ready(function(){

  // jQuery methods go here...

});
```

- 예시: jq1.html
  화면에 텍스트 출력



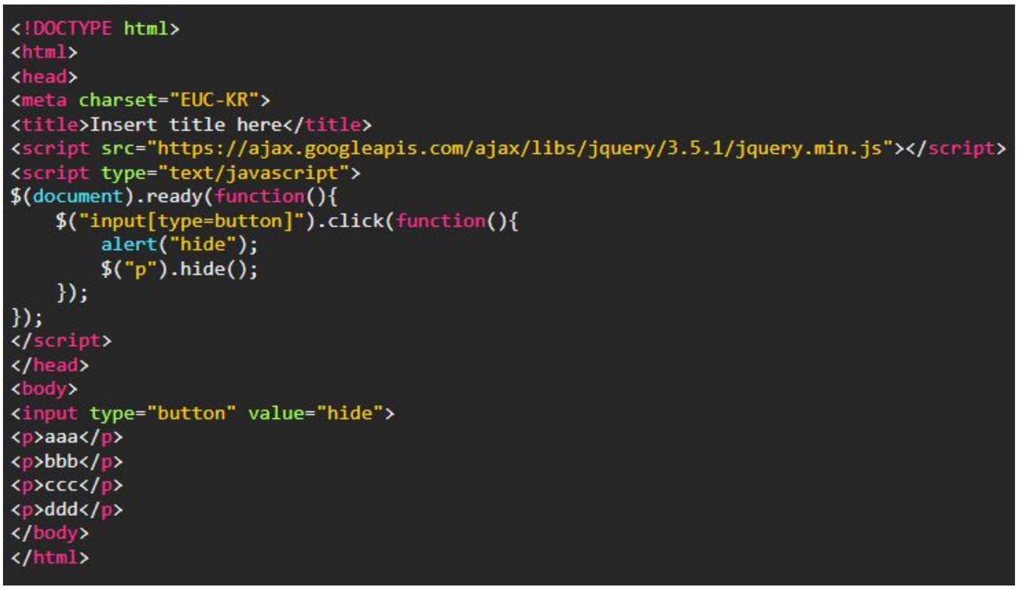
```
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    alert("start javascript")
    });
</script>
</head>
<body>

</body>
</html>
```



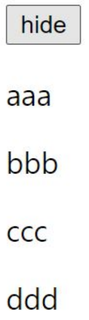
localhost:8888 내용:

start javascript

- 예시: 버튼 누르면 hide 처리



```
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    $("input[type=button]").click(function(){
        alert("hide");
        $("p").hide();
    });
});
</script>
</head>
<body>
<input type="button" value="hide">
<p>aaa</p>
<p>bbb</p>
<p>ccc</p>
<p>ddd</p>
</body>
</html>
```



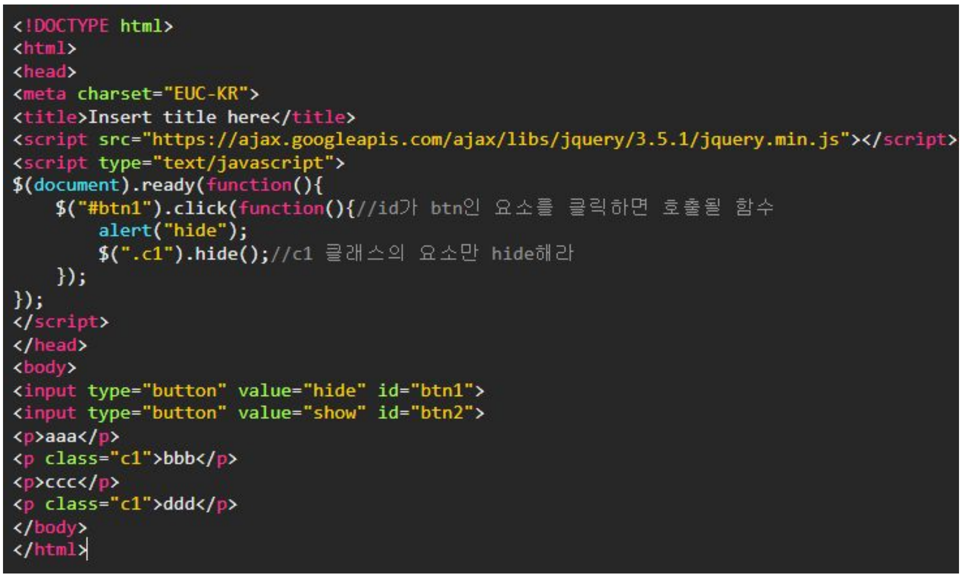
aaa

bbb

ccc

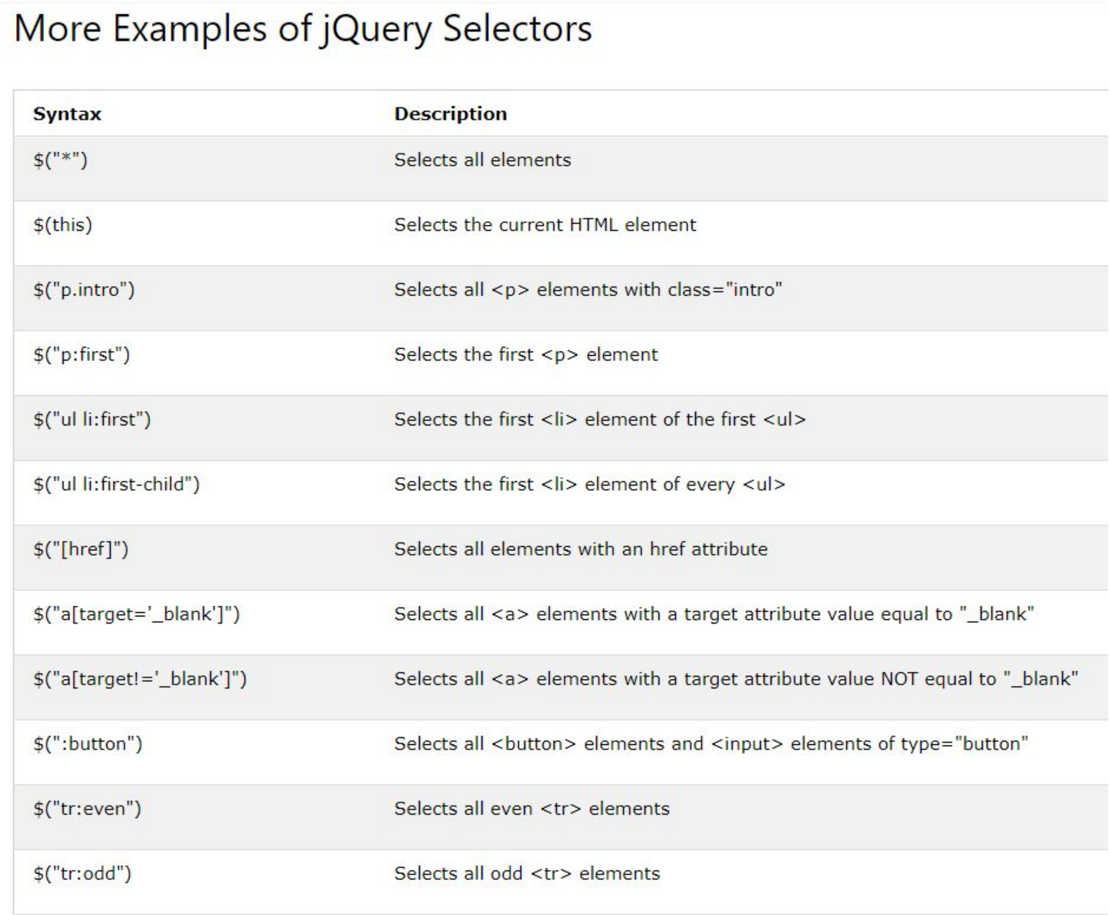
ddd

- 예시: 버튼별 동작 설정



```
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    $("#btn1").click(function(){//id가 btn인 요소를 클릭하면 호출될 함수
        alert("hide");
        $(".c1").hide();//c1 클래스의 요소만 hide해라
    });
});
</script>
</head>
<body>
<input type="button" value="hide" id="btn1">
<input type="button" value="show" id="btn2">
<p>aaa</p>
<p class="c1">bbb</p>
<p>ccc</p>
<p class="c1">ddd</p>
</body>
</html>
```

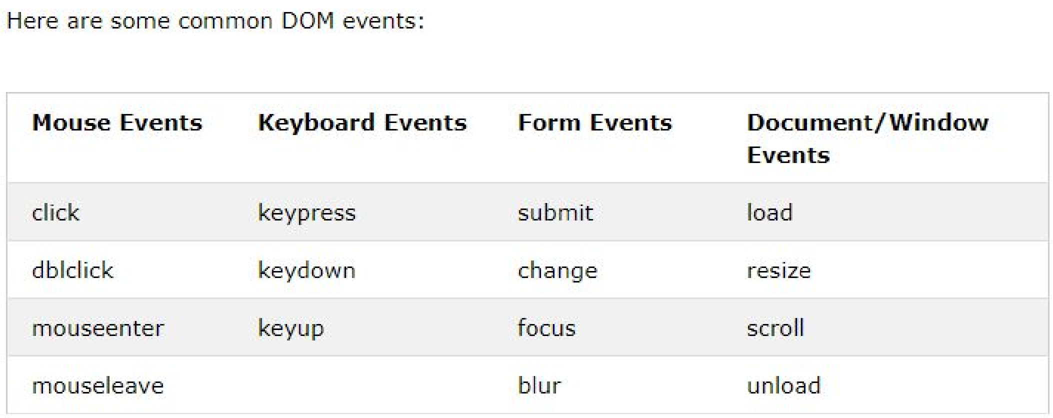
- selector 객체



More Examples of jQuery Selectors

Syntax	Description
$("*")	Selects all elements
$(this)	Selects the current HTML element
$("p.intro")	Selects all <p> elements with class="intro"
$("p:first")	Selects the first <p> element
$("ul li:first")	Selects the first <li> element of the first <ul>
$("ul li:first-child")	Selects the first <li> element of every <ul>
$("[href]")	Selects all elements with an href attribute
$("a[target='_blank']")	Selects all <a> elements with a target attribute value equal to "_blank"
$("a[target!='_blank']")	Selects all <a> elements with a target attribute value NOT equal to "_blank"
$(":button")	Selects all <button> elements and <input> elements of type="button"
$("tr:even")	Selects all even <tr> elements
$("tr:odd")	Selects all odd <tr> elements

- 이벤트



Here are some common DOM events:

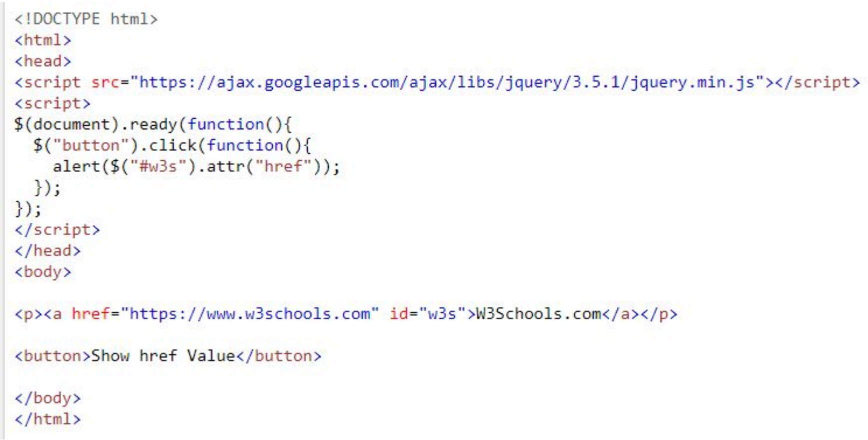
Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

2) jQuery HTML

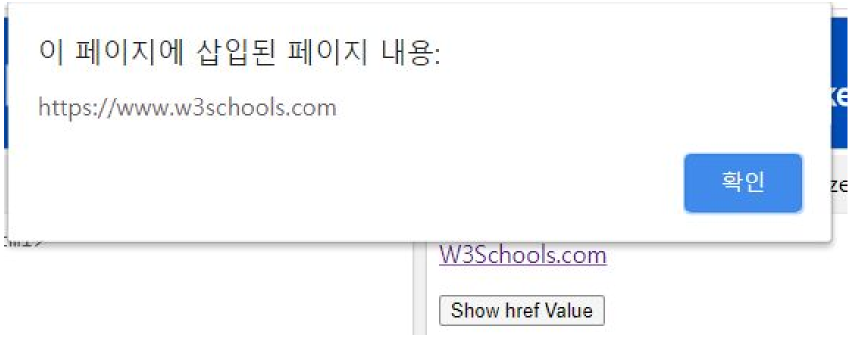
: 제이쿼리의 DOM(다큐먼트 객체 모델) 조작을 쉽게 만든다. DOM:각 다큐먼트를 하나의 객체로 여기고 각각을 조작하는 방법.

a. Get Content
   - text(): 해당 범위 내의 텍스트 값만 가져옴
   - html(): html 해당 범위 내의 html 전체를 가져옴
   - val(): form 영역의 value 값을 가져옴

b. Get Attribute
   - attr(name)
   - 예시



```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    alert($("#w3s").attr("href"));
  });
});
</script>
</head>
<body>

<p><a href="https://www.w3schools.com" id="w3s">W3Schools.com</a></p>

<button>Show href Value</button>

</body>
</html>
```

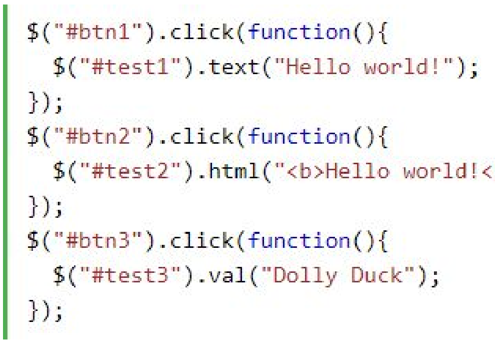


이 페이지에 삽입된 페이지 내용:

https://www.w3schools.com

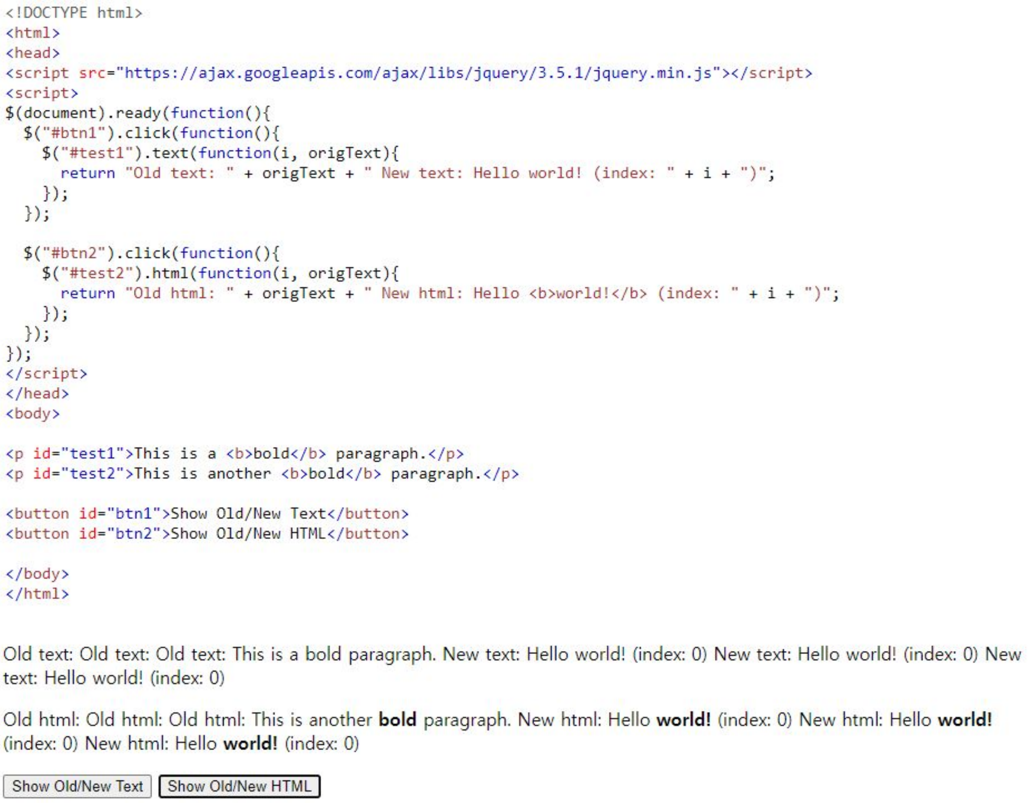
c. Set Content
   - text("입력할 텍스트 값')
   - html("입력할 html")
   - val("입력할 value")

- 예시



```
$("#btn1").click(function(){
  $("#test1").text("Hello world!");
});
$("#btn2").click(function(){
  $("#test2").html("<b>Hello world!</b>");
});
$("#btn3").click(function(){
  $("#test3").val("Dolly Duck");
});
```

- 예시2



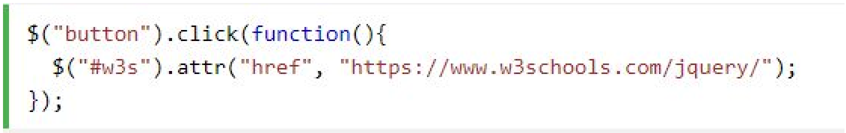
```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("#test1").text(function(i, origText){
      return "Old text: " + origText + " New text: Hello world! (index: " + i + ")";
    });
  });

  $("#btn2").click(function(){
    $("#test2").html(function(i, origText){
      return "Old html: " + origText + " New html: Hello <b>world!</b> (index: " + i + ")";
    });
  });
});
</script>
</head>
<body>

<p id="test1">This is a <b>bold</b> paragraph.</p>
<p id="test2">This is another <b>bold</b> paragraph.</p>

<button id="btn1">Show Old/New Text</button>
<button id="btn2">Show Old/New HTML</button>

</body>
</html>
```

Old text: Old text: Old text: This is a bold paragraph. New text: Hello world! (index: 0) New text: Hello world! (index: 0) New text: Hello world! (index: 0)

Old html: Old html: Old html: This is another **bold** paragraph. New html: Hello **world!** (index: 0) New html: Hello **world!** (index: 0) New html: Hello **world!** (index: 0)

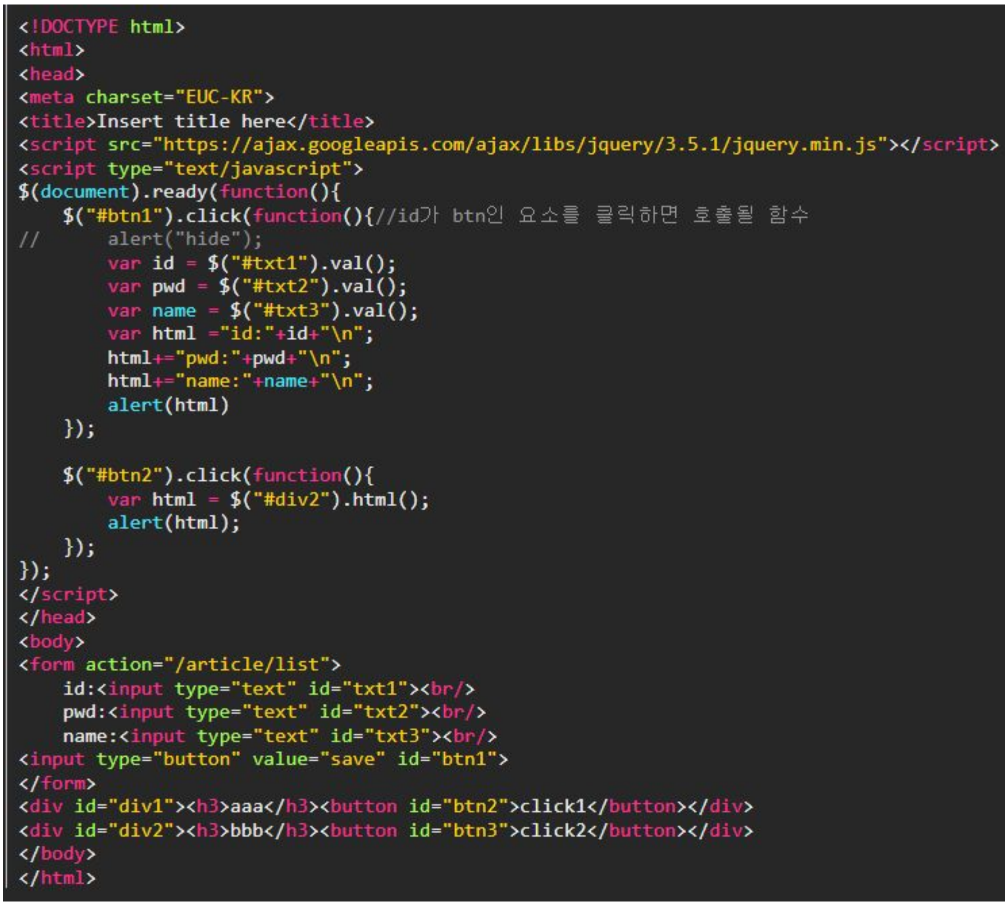
d. Set Attribute

- attr('name', value)
- 예시

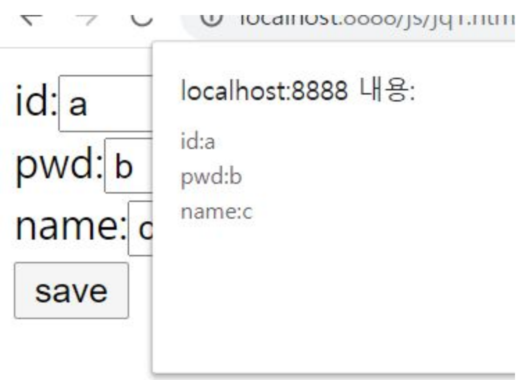


```
$("button").click(function(){
  $("#w3s").attr("href", "https://www.w3schools.com/jquery/");
});
```

- 예시:text()[단순 테스트만], html()[html 효과까지] 함수로 값 읽어오기



```
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    $("#btn1").click(function(){//id가 btn인 요소를 클릭하면 호출될 함수
//        alert("hide");
        var id = $("#txt1").val();
        var pwd = $("#txt2").val();
        var name = $("#txt3").val();
        var html ="id:"+id+"\n";
        html+="pwd:"+pwd+"\n";
        html+="name:"+name+"\n";
        alert(html)
    });

    $("#btn2").click(function(){
        var html = $("#div2").html();
        alert(html);
    });
});
</script>
</head>
<body>
<form action="/article/list">
    id:<input type="text" id="txt1"><br/>
    pwd:<input type="text" id="txt2"><br/>
    name:<input type="text" id="txt3"><br/>
<input type="button" value="save" id="btn1">
</form>
<div id="div1"><h3>aaa</h3><button id="btn2">click1</button></div>
<div id="div2"><h3>bbb</h3><button id="btn3">click2</button></div>
</body>
</html>
```

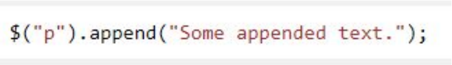


localhost:8888 내용:

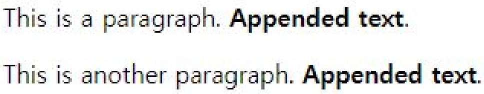
id:a
pwd:b
name:c

e. ADD

- append(): 해당 요소 내의 맨앞 위치에 추가
- prepend(): 해당 요소 내의 맨뒤 위치에 추가
- after(): 해당 요소 뒤에 추가
- before(): 해당 요소 앞에 추가
- 예시1: p 태그 내에 text 추가(뒤)



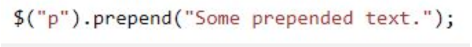
```
$("p").append("Some appended text.");
```



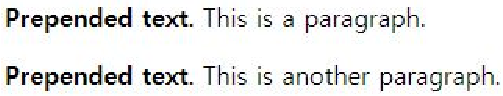
This is a paragraph. **Appended text.**

This is another paragraph. **Appended text.**

- 예시2: p 태그 내에 text 추가(앞)



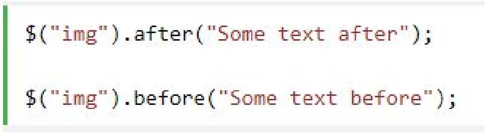
```
$("p").prepend("Some prepended text.");
```



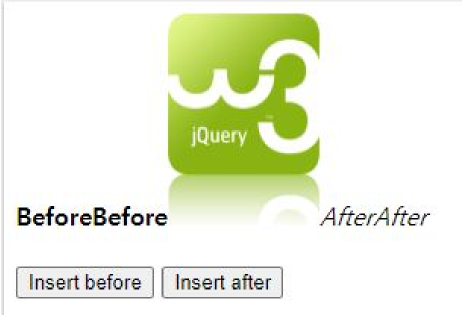
Prepended text. This is a paragraph.

Prepended text. This is another paragraph.

- 예시3: 이미지 태그 앞뒤에 text 추가



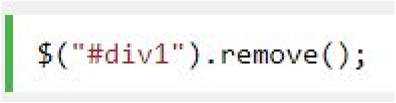
```
$("img").after("Some text after");

$("img").before("Some text before");
```



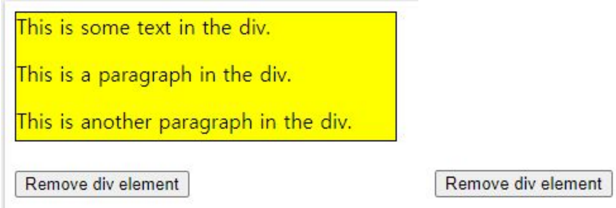
BeforeBefore *AfterAfter*

f. remove

- remove(): 선택된 객체 자체를 지움
- empty(): 선택된 객체 내의 요소 지움
- 예시1



```
$("#div1").remove();
```

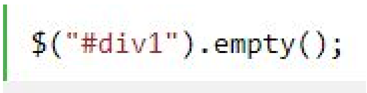


This is some text in the div.

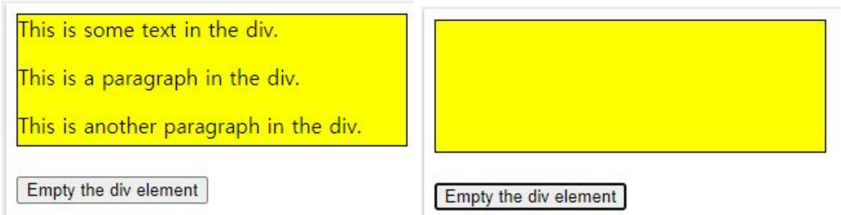
This is a paragraph in the div.

This is another paragraph in the div.

- 예시2



```
$("#div1").empty();
```



This is some text in the div.

This is a paragraph in the div.

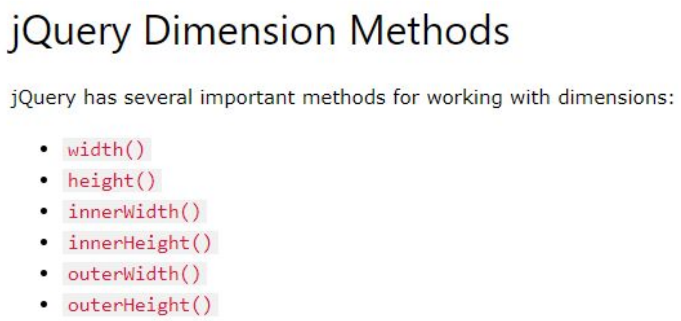
This is another paragraph in the div.

g. css

https://www.w3schools.com/jquery/jquery_css_classes.asp

- addClass()
- removeClass()
- toggleClass()
- css()

h. dimensions
https://www.w3schools.com/jquery/jquery_dimensions.asp



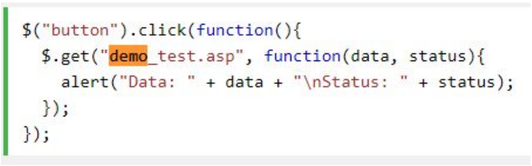
jQuery Dimension Methods

jQuery has several important methods for working with dimensions:

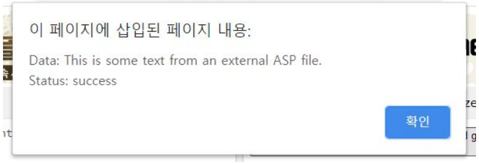
- `width()`
- `height()`
- `innerWidth()`
- `innerHeight()`
- `outerWidth()`
- `outerHeight()`

3) jQuery AJAX: 비동기 Asynchronous JavaScript and XML.

- HTTP Request: GET vs POST
- $.get(URL, 콜백함수)
- 예시



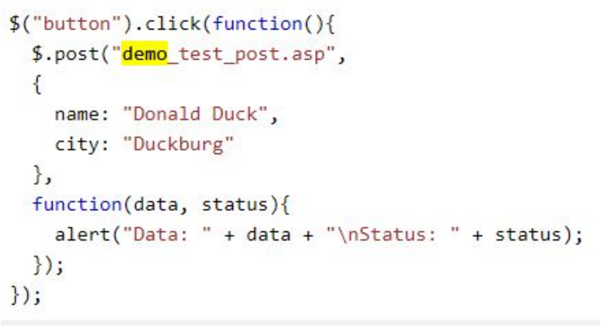
```
$("button").click(function(){
  $.get("demo_test.asp", function(data, status){
    alert("Data: " + data + "\nStatus: " + status);
  });
});
```



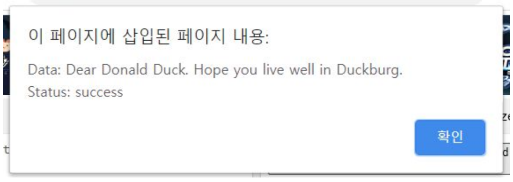
이 페이지에 삽입된 페이지 내용:

Data: This is some text from an external ASP file.
Status: success

- $.post(URL, 데이터, 콜백함수)
- 예시



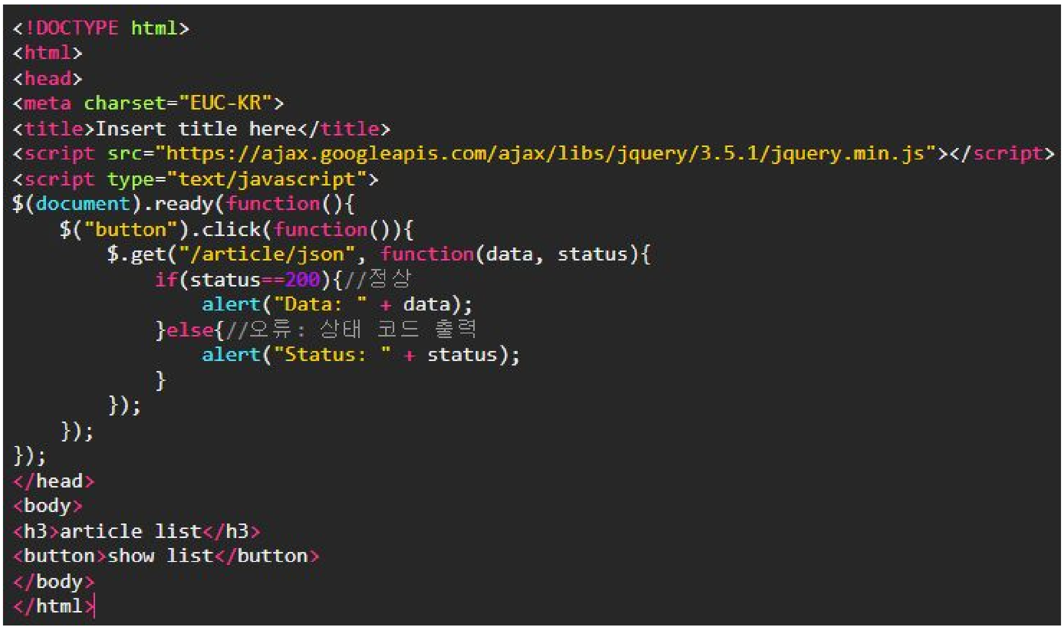
```
$("button").click(function(){
  $.post("demo_test_post.asp",
  {
    name: "Donald Duck",
    city: "Duckburg"
  },
  function(data, status){
    alert("Data: " + data + "\nStatus: " + status);
  });
});
```



이 페이지에 삽입된 페이지 내용:

Data: Dear Donald Duck. Hope you live well in Duckburg.
Status: success

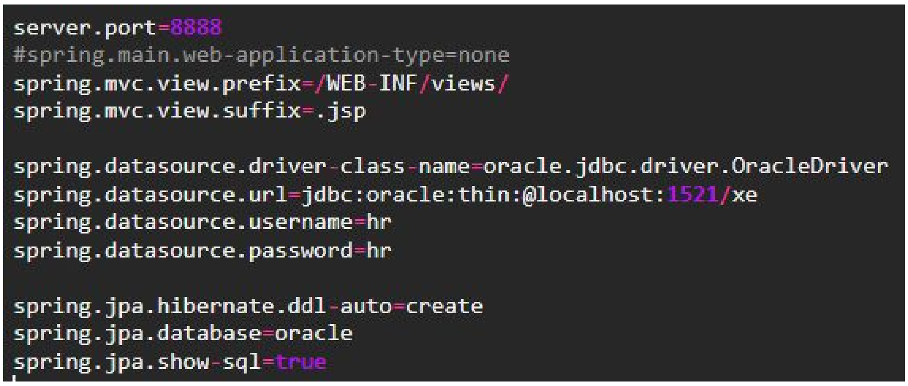
- js2.html



```html
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    $("button").click(function()){
        $.get("/article/json", function(data, status){
            if(status==200){//정상
                alert("Data: " + data);
            }else{//오류: 상태 코드 출력
                alert("Status: " + status);
            }
        });
    });
});
</head>
<body>
<h3>article list</h3>
<button>show list</button>
</body>
</html>
```

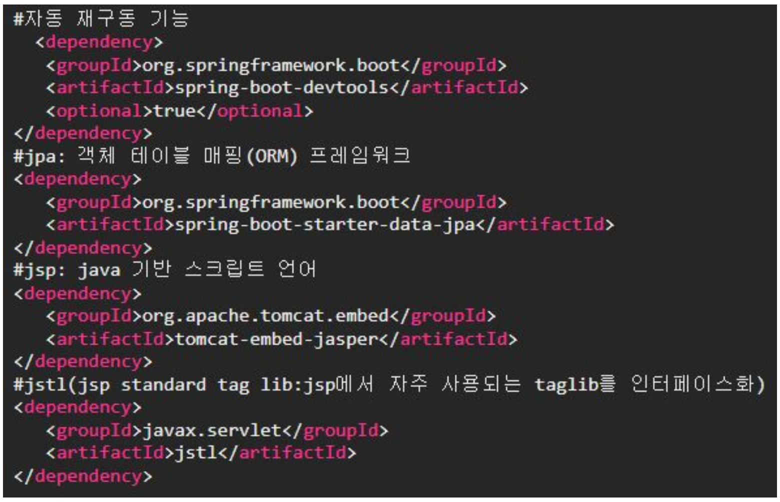
4. [프로젝트 app4]

1) application.properties



```
server.port=8888
#spring.main.web-application-type=none
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp

spring.datasource.driver-class-name=oracle.jdbc.driver.OracleDriver
spring.datasource.url=jdbc:oracle:thin:@localhost:1521/xe
spring.datasource.username=hr
spring.datasource.password=hr

spring.jpa.hibernate.ddl-auto=create
spring.jpa.database=oracle
spring.jpa.show-sql=true
```

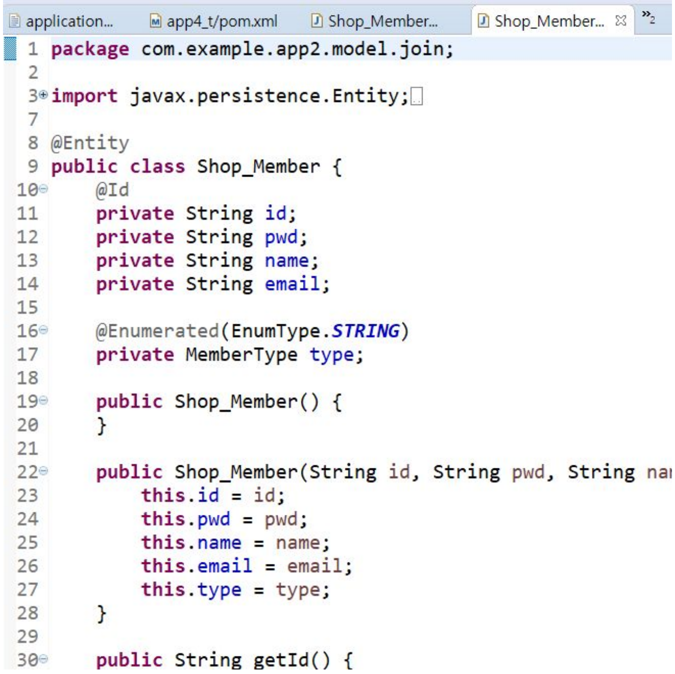
2) pom.xml 세팅(jstl, jpa,jsp,자동 재구동)



```xml
#자동 재구동 기능
  <dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-devtools</artifactId>
   <optional>true</optional>
</dependency>
#jpa: 객체 테이블 매핑(ORM) 프레임워크
<dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
#jsp: java 기반 스크립트 언어
<dependency>
   <groupId>org.apache.tomcat.embed</groupId>
   <artifactId>tomcat-embed-jasper</artifactId>
</dependency>
#jstl(jsp standard tag lib:jsp에서 자주 사용되는 taglib를 인터페이스화)
<dependency>
   <groupId>javax.servlet</groupId>
   <artifactId>jstl</artifactId>
</dependency>
```

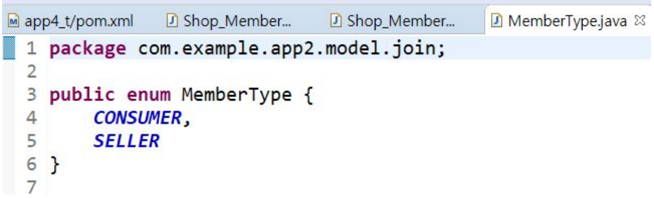
3) model 구축(model.join 패키지)

- Shop_Member.java



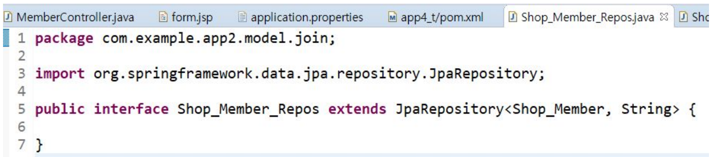
```java
package com.example.app2.model.join;

import javax.persistence.Entity;

@Entity
public class Shop_Member {
    @Id
    private String id;
    private String pwd;
    private String name;
    private String email;

    @Enumerated(EnumType.STRING)
    private MemberType type;

    public Shop_Member() {
    }

    public Shop_Member(String id, String pwd, String na
        this.id = id;
        this.pwd = pwd;
        this.name = name;
        this.email = email;
        this.type = type;
    }

    public String getId() {
```

- MemberType.java(ENUM: 상수 나열) => 멤버 타입 구분용



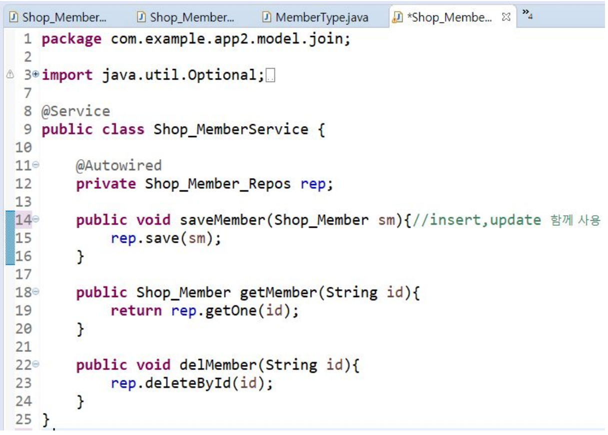
```java
package com.example.app2.model.join;

public enum MemberType {
    CONSUMER,
    SELLER
}
```

- Shop_Member_Repos(인터페이스) <Shop_Member,String>



```java
package com.example.app2.model.join;

import org.springframework.data.jpa.repository.JpaRepository;

public interface Shop_Member_Repos extends JpaRepository<Shop_Member, String> {

}
```

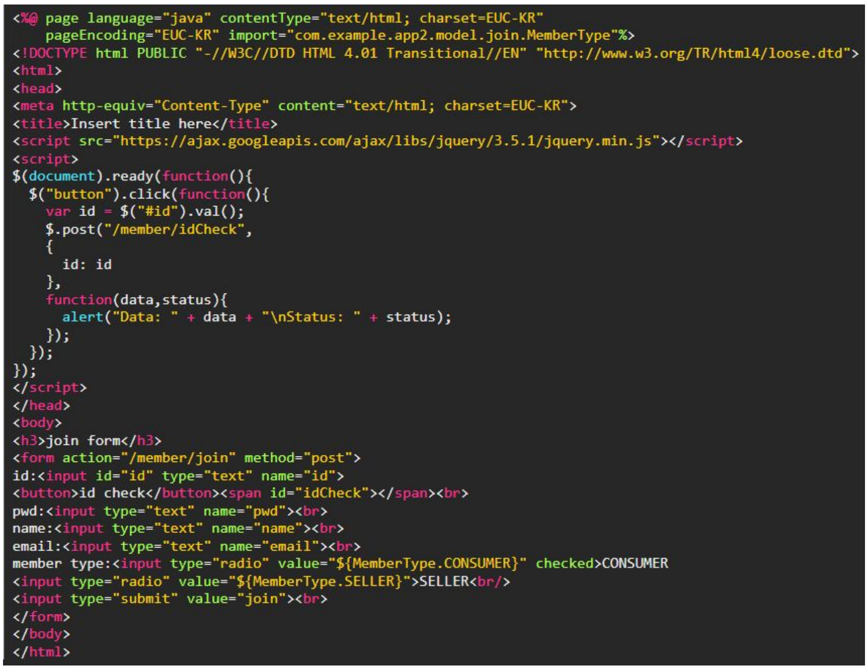
- Shop_Member_Service



```java
package com.example.app2.model.join;

import java.util.Optional;

@Service
public class Shop_MemberService {

    @Autowired
    private Shop_Member_Repos rep;

    public void saveMember(Shop_Member sm){//insert,update 함께 사용
        rep.save(sm);
    }

    public Shop_Member getMember(String id){
        return rep.getOne(id);
    }

    public void delMember(String id){
        rep.deleteById(id);
    }
}
```

4) view 구축(src/webapp/WEB-INF/views/member)

- form.jsp *ajax 활용한 아이디 중복 체크

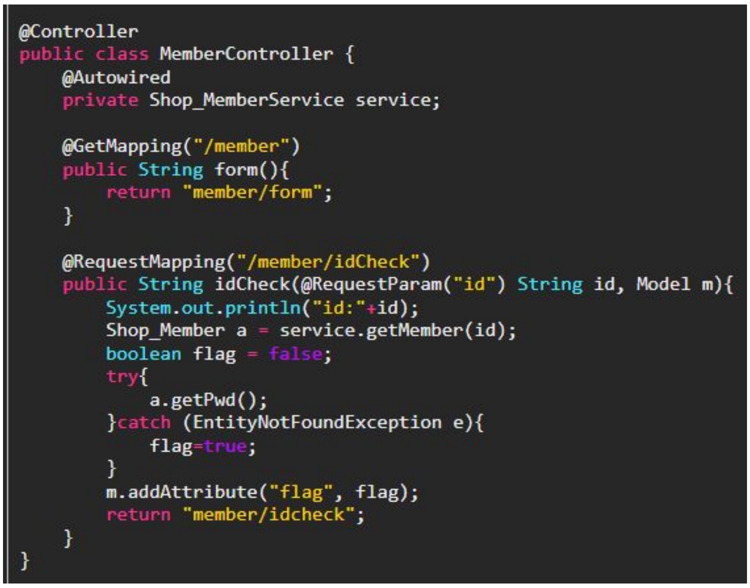


```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR" import="com.example.app2.model.join.MemberType"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    var id = $("#id").val();
    $.post("/member/idCheck",
    {
      id: id
    },
    function(data,status){
      alert("Data: " + data + "\nStatus: " + status);
    });
  });
});
</script>
</head>
<body>
<h3>join form</h3>
<form action="/member/join" method="post">
id:<input id="id" type="text" name="id">
<button>id check</button><span id="idCheck"></span><br>
pwd:<input type="text" name="pwd"><br>
name:<input type="text" name="name"><br>
email:<input type="text" name="email"><br>
member type:<input type="radio" value="${MemberType.CONSUMER}" checked>CONSUMER
<input type="radio" value="${MemberType.SELLER}">SELLER<br/>
<input type="submit" value="join"><br>
</form>
</body>
</html>
```

5) Controller

- MemberController.java

  *json/xml 형태로 데이터를 주고받지 않을 경우. 전/후처리가 복잡해진다. ex.스페이스처리...



```
@Controller
public class MemberController {
    @Autowired
    private Shop_MemberService service;

    @GetMapping("/member")
    public String form(){
        return "member/form";
    }

    @RequestMapping("/member/idCheck")
    public String idCheck(@RequestParam("id") String id, Model m){
        System.out.println("id:"+id);
        Shop_Member a = service.getMember(id);
        boolean flag = false;
        try{
            a.getPwd();
        }catch (EntityNotFoundException e){
            flag=true;
        }
        m.addAttribute("flag", flag);
        return "member/idcheck";
    }
}
```