

10. 파일 업로드

1. 파일 업로드의 개요
2. 파일 업로드하고 폼을 분석하는 라이브러리 `cos.jar`
3. 파일 업로드 예제 작성

1. 파일 업로드의 개요

□ 파일 업로드에 필요한 요소

- ▣ 파일을 입력받는 폼의 method 및 enctype타입 :
 - 예) `method="post" enctype="multipart/form-data"`
- ▣ 업로드 할 파일이 저장될 폴더
 - 예) [fileSave] 폴더

□ 파일 업로드에 필요한 요소

- ▣ 파일을 업로드하고 폼을 분석하는 라이브러리
 - 예) cos.jar
- ▣ Ajax 기반의 업로드를 하려면 추가적으로 [jQuery Form Plugin], [jQuery Multiple File Upload Plugin]이 필요

1. 파일 업로드의 개요

□ 파일 업로드용 폼의 형태

- method가 "post"이고, enctype이 "multipart/form-data"

▣ 파일 입력 방법

- <input>태그의 type속성의 값을 "file"로 지정
- 사용 예
 - `<input type="file" name="selectfile">`

1. 파일 업로드의 개요

□ 파일 업로드용 폼의 형태

▣ 작성 방법

- 방법1 : <form> 태그의 속성을 사용하는 방법

```
<form name="formName" method="post"
      enctype="multipart/form-data">
```

- 방법2 : <input type="submit">에서 별도의 속성을 지정하는 방법
 - <input type="submit">에 HTML5에서 새로 추가된 *formenctype*, *formmethod* 속성을 사용

```
<input type="submit" formmethod="post",
      enctype="multipart/form-data" value="업로드 전송">
```

1. 파일 업로드 개요

□ 파일 업로드용 폼의 형태

▣ 작성 방법

■ 방법3 : Ajax기반의 자바스크립트에서 지정

- jQuery Form Plugin]을 다운로드 받아 사용하거나 \$.ajax() 메소드에 FormData를 생성
- <script >추가

```
<script src="../js/jquery-1.11.0.min.js"> </script>  
<script src="../js/jquery.form.min.js"> </script>
```

1. 파일 업로드 개요

□ 파일 업로드용 폼의 형태

▣ 코딩 방법

■ HTML 태그

```
<form id="formid" method="post" action="upPro.jsp"
      enctype="multipart/form-data">
  <input type="text" name="title">
  <input type="file" name="selectfile">
  <input type="submit" value="전송">
</form>
```

■ 자바스크립트 코드

```
$("#formid").ajaxForm({
  success: function(data, status){
    $("#result").html(data);
  }});
```

1. 파일 업로드 개요

□ 다중 파일 업로드용 폼의 형태

▣ 작성 방법

■ 방법1 : [jQuery Form Plugin]을 사용

- 여러 개의 파일을 입력받을 경우 `<input type="file">` 태그를 입력받을 파일의 수만큼 생성

■ 방법2 : [jQuery Multiple File Upload Plugin]을 사용

- 코딩 방법 :

`<script>` 추가

```
<script src="../js/jquery-1.11.0.min.js"> </script>
<script src="../js/jquery.form.min.js"> </script>
<script src="../js/jquery.Metadata.js"> </script>
<script src="../js/jquery.MultiFile.js"> </script>
<script src="../js/jquery.blockUI.js"> </script>
```

1. 파일 업로드 개요

□ 다중 파일 업로드용 폼의 형태

▣ 코딩 방법

■ HTML 태그

```
<input type="file" id="file1" name="file1" class="multi"
      maxlength="3">
```

■ 자바스크립트 코드

```
$("#formid").ajaxForm({
  success: function(data, status){
  }});
```


1. 파일 업로드 개요

□ 다중 파일 업로드용 폼의 형태

▣ 작성 방법

- 방법3 : HTML5에서 추가된 multiple속성을 사용
 - 코딩 방법

```
<input type="file" name="selectfile" multiple>
```

2. 파일을 업로드 및 폼 분석라이브러리-cos.jar

- ▣ 파일을 업로드하고 폼으로부터 넘어오는 데이터를 얻어내는데 사용

▣ 다운로드 및 파일 배치

- ▣ <http://www.servlets.com/cos> 사이트에서 다운로드
- ▣ cos.jar 파일을 톰캣 공용 라이브러리 폴더(톰캣홈\lib)와 해당 프로젝트의 라이브러리 폴더([프로젝트]-[Webcontent]-[WEB-INF]-[lib])에 넣어서 사용

2. 파일을 업로드 및 폼 분석라이브러리-cos.jar

□ 파일 업로드 및 폼 요소를 처리하는 **multipartRequest** 클래스

▣ 생성자

```
MultipartRequest(javax.servlet.http.HttpServletRequest request, //request 객체  
    java.lang.String saveDirectory, //파일 업로드할 폴더  
    int maxPostSize, //업로드 할 파일의 최대크기  
    java.lang.String encoding, //인코딩 방식  
    FileRenamePolicy policy //같은 파일 덮어쓰기 방지 설정  
);
```

2. 파일을 업로드 및 폼 분석라이브러리-cos.jar

□ 파일 업로드 및 폼 요소를 처리하는 **multipartRequest** 클래스

▣ 객체 생성의 예

```
MultipartRequest upload = new MultiPartRequest(  
    request,  
    fileSave,  
    1024*5,  
    "utf-8",  
    new DefaultFileRenamePolicy( )  
);
```

2. 파일을 업로드 및 폼 분석라이브러리-cos.jar

□ 파일 업로드 및 폼 요소를 처리하는 `multipartRequest` 클래스

▣ 메소드

- `getContentType(java.lang.String name)` : `java.lang.String`
 - 업로드된 파일의 콘텐츠 타입을 반환하는 것으로, 업로드된 파일이 없으면 `null`을 리턴
- `getFile(java.lang.String name)` : `java.io.File`
 - 서버 상에 업로드된 파일을 `File` 객체 타입으로 리턴한다. 업로드된 파일이 없다면 `null`을 리턴
- `getFileNames()` : `java.util.Enumeration`
 - 폼의 요소들 중 `<input type="file">`로 된 파라미터들을 받아서 `Enumeration` 타입의 객체를 리턴
- `getFileName(java.lang.String name)` : `java.lang.String`
 - 사용자가 업로드한 파일명을 리턴한다. 서버상에 업로드된 파일명이 리턴

2. 파일을 업로드 및 폼 분석라이브러리-cos.jar

□ 파일 업로드 및 폼 요소를 처리하는 multipartRequest 클래스

▣ 메소드

- **getOriginalFileName(java.lang.String name) : java.lang.String**
 - 사용자가 업로드한 파일의 원래 파일명을 리턴한다. 파일명이 중복될 경우 이름이 변경되므로, 변경되기 전의 원래 파일명이 리턴
- **getParameter(java.lang.String name) : java.lang.String**
 - *name*에 해당하는 파라미터의 값을 리턴
- **getParameterNames() : java.util.Enumeration**
 - 폼의 요소들 중 `<input type="file">` 아닌 파라미터들을 *Enumeration* 객체 타입으로 리턴
- **getParameterValues(java.lang.String name) : java.lang.String[]**
 - 하나의 파라미터에 대해 여러 개의 값을 가지는 `<input type="checkbox">` 와 같은 경우에 파라미터의 값을 얻어내기 위해 사용

3. 파일 업로드 예제 작성

- 파일을 업로드하면 업로드한 파일의 서버상의 특정 폴더에 저장되고, 업로드한 파일 정보는 데이터베이스 테이블에 저장
- 단일 파일 업로드 시 처리할 작업
 - ▣ 파일 업로드 폴더 생성
 - [upload] 폴더 생성
 - ▣ jQuery Form Plugin 다운로드 및 배치
 - jquery.form.min.js 파일을 [studyjsp/WebContent/js]폴더에 넣음
 - ▣ 파일 업로드 폼과 업로드 처리 페이지 작성
 - 폼 : singleUploadForm.jsp, 파일 업로드 처리 : singleUploadPro.jsp

3. 파일 업로드 예제 작성

□ 다중 파일 업로드 시 처리할 작업

▣ 파일 업로드 폴더 생성

- [upload] 폴더 없으면 생성

▣ jQuery Multiple File Upload Plugin 다운로드 및 배치

- jquery.MetaData.js, jquery.MultiFile.js, jquery.blockUI.js 파일을 [studyjsp/WebContent/js] 폴더에 넣음

▣ 파일업로드 폼과 업로드 처리 페이지 작성

- 폼 : multiUploadForm2.jsp, 파일 업로드 처리 : multiUploadPro.jsp