



웹프로그래밍 기초



생각해봅시다:





- dynamic web project
 - lib copy
 - C:\a01_prog\tomcat\lib\servlet-api
 - \jspexp\WebContent\WEB-INF\lib
 - css, jquery
 - WebContent
 - com 폴더 생성 후, jquery-1.10.2.js copy
-



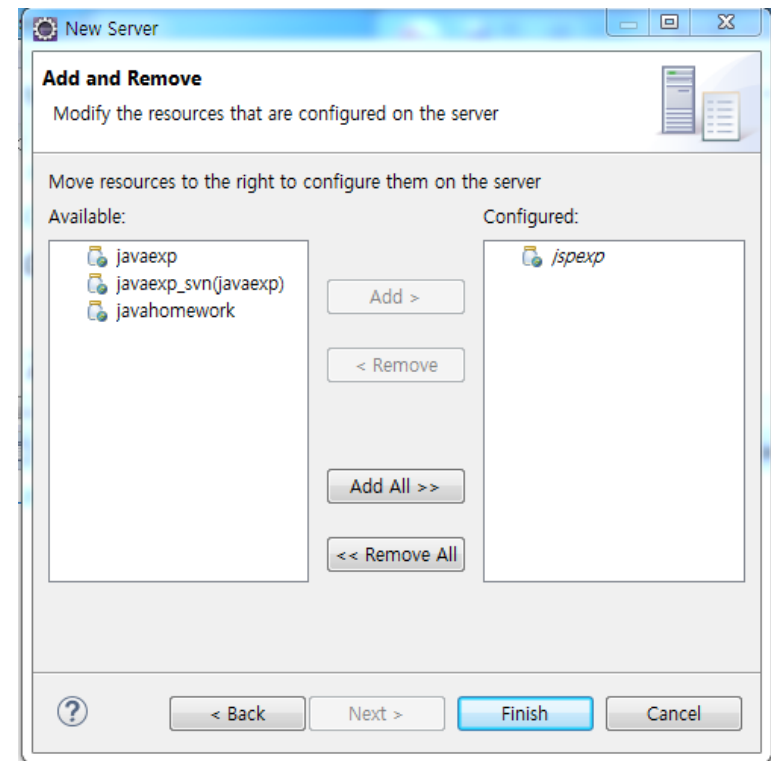
css, jquery:

Menu → Window → Preferences → Web → JSP Files →
Editor → Templates

New JSP File(html) 에 edit 아래 내용 추가

```
<style type="text/css">
/**/
</style>
<script src=" /jspexp/com/jquery-1.10.2.js">
</script>
<script type="text/javascript">
$$(document).ready(function(){
    alert("jquery");
});
</script>
```

- WebCotent에서 new → jsp File
 - index.jsp
- Servers
 - new server





index.jsp

*Tomcat v7.0 Server at localhost

her common settings.

Tomcat v7.0 Server at localhost

localhost

Apache Tomcat v7.0

ervers/Tomcat v7.0 Server at loca

Browse...

atalina.base) and deploy path. Server must be
esent to make changes.

(does not modify Tomcat installation)

Publishing

Timeouts

Ports

Modify the server ports.

Port Name	Port Number
Tomcat admin port	7005
HTTP/1.1	7080
AJP/1.3	7009

Overview

Modules

Markers

Properties

Servers

Data Source Explorer

Snippets

Problems

Console

Tomcat v7.0 Server at localhost [Stopped, Synchronized]



jsp 페이지 시작하면서 :

- jsp의 구성요소
 - 설정부분(Directive) : `<%@ page ~~~ %>`
 - 문서타입 html, excel...
 - custom tag : `<html, <h1, jsp에서 정의한 tag`활용하게..
 - 활용할 자바클래스 import
 - 스크립트 : java 코드를 선언, 호출, 할당
 - 표현식(Expression) : 화면출력
`<% 자바코드(변수)%> System.out.println(...)`
 - 스크립트릿(Scriptlet) : 자바코드 실행(선언, 조건식, 반복자, 객체호출) `<% 자바코드%> if, for, int numer`
 - 선언부(Declaration) : 자바 메서드를 선언, 변수 선언 `<%! %>`

- 데이터가 나오는 곳은 **java** 변수로 출력..

Daily Schedule!!

시간	계획내용	달성여부	기타
9:30	수업시작	달성	기분 좋은시작
13:20	점심시간메뉴		오늘은?
16:30	프로젝트		이틀?



page 디렉티브 :

- jsp페이지 대한 정보
 - 어떤 문서를 생성하는지?
 - 어떤 자바클래스 사용하는지?
- 자바클래스 호출하여 사용하는 방법
 - class 객체 생성.
 - package와 클래스 생성..
 - package : `jspexp.a01_basic`
 - class : `Person.java`
 - 속성 : name, age, loc
 - setXXX, getXXX 생성
 - 메뉴 → Source → Generator Getter/Setter
 - `<%@ page`
 - `import = "jspexp.a01_basic.Person"`
 - `<% Person p = new Person();`
 - `p.setName("홍길동");`
 - `%>`
 - `<tr><td>이름</td><td><%=p.getName()%></td></tr>`



page 지시자 import 활용 연습 :

- 클래스 선언
 - *jspexp.a01_basic.Woman*
 - 멤버(필드, set/get메서드) : name, telnum, height
 - jsp 화면
 - a03_pageDirectExp.jsp
 - 그녀의 이름 : @@@
 - 전화번호 : @@
 - 키 : @@@
-



jsp의 디렉티브 :

- 디렉티브 : jsp 페이지에 대한 설정
 - `<%@ [디렉티브명 page, taglib, include]`
 - 속성1 = "값1" 속성2="값" %>
 - **page** : 페이지 정보로, jsp가 생성한 문서의 타입, 출력버퍼 크기, 에러 페이지 등
 - **taglib** : 태그 라이브러리
 - **include** : 특정 영역에 다른 문서 포함
-



page 디렉티브의 속성들 :

- <%@ page 속성 = 값.....
 - contentType : 문서 타입, 한글처리 ex) html, excel...
 - import : java api 클래스, 사용자 정의 클래스 호출
 - session : 세션 사용 여부 true/false
 - buffer : 출력을 효과적으로 처리하기 위한 메모리 공간. jsp페이지의 출력 버퍼 크기 지정. none/@@kb
 - autoFlush : 출력 버퍼가 찼을 경우 자동으로 출력 스트림에 보내고 비워주는 여부를 boolean
 - info : jsp페이지에 대한 설명 입력
 - errorPage : jsp 페이지에서 에러발생시, 보여줄 페이지 지정. 보여줄 에러페이지 : [error01.jsp](#)
 - isErrorPage : 에러발생시, 지정된 에러페이지 인지?
 - [error01.jsp](#) 에서 속성 `isErrorPage="true"`
 - pageEncoding : 페이지 자체의 인코딩 지정(한글처리)



스크립트 요소 :

- 스크립트릿
 - 화면, 자바 소스 구분 처리 <% 자바코드 %>
 - 조건문, 반복문 화면
 - <% if(조건){ %>
 - 화면처리.... <%=자바변수/메서드()%>
 - <% } %>
- 표현식(expression) <%= %>
 - 스크립트릿과 함께 처리할 출력내용
- 선언부(declaration)
 - 함수(메서드) 선언부. <%! %>



request 기본 객체 :



■ request 기본 객체의 기능

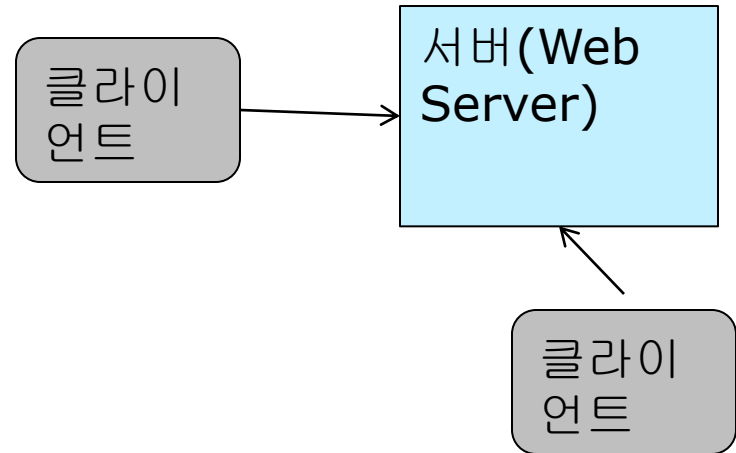
- 클라이언트(웹 브라우저)와 관련된 정보 읽기 기능
- 서버 관련된 정보
- 클라이언트가 전송한 요청 파라미터 정보
- 클라이언트가 전송한 요청 헤더 읽기 정보
- 클라이언트가 전송한 쿠키
- 속성 처리 기능



request 통한 정보 읽기 :

`http://localhost:8080/jspexp/start.jsp`

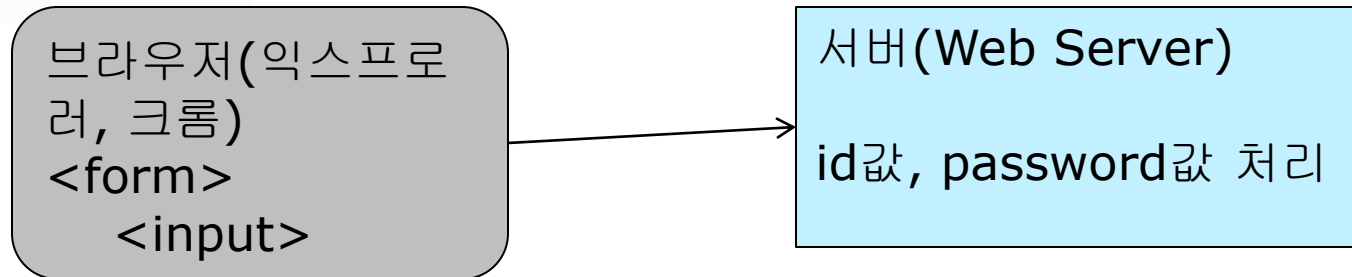
- 형식
 - `request.getXXXX();`
- `getRemoteAddr()`
 - 클라이언트 접속 ip
- `getProtocol()` :
 - 클라이언트 요청 프로토콜
- `getMethod()` : get/post 전송 때 사용한 방식
- **`getContextPath()`** : 컨텍스트 경로
 - `/jspexp/WebContent/시작위치`
- `getServerName(), getServerPort()`





폼과 요청 파라미터 처리 :

- request 처리..



- <화면단 client 요청화면>

- id : [himan]
- id : <input type="text" name="name01"/>

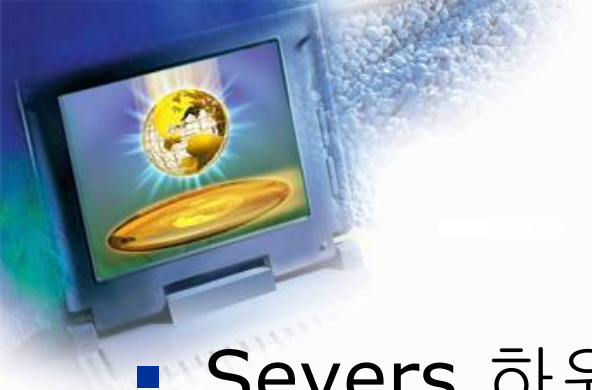
- <서버단 요청된 데이터 처리>

- String
data=request.getParameter("name01");



form의 속성처리 :

- send.jsp
 - [good job] → name="send01"
 - <form action="receive.jsp"
 - <input type="text" name="send01"/>
 - <input type="submit" value="전송" />
 - type이 submit 버튼모양을 클릭하면, form에 하위에 있는 <input <select ... 객체들의 데이터가 지정된 페이지(action)에 전송이 된다.
- receive.jsp
 - String
send01=request.getParameter("send01");

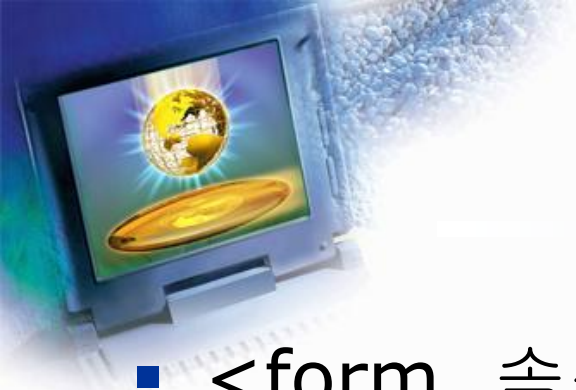


전송된 데이터 한글처리 :

- Servers 하위에
 - server.xml 파일에
 - <Connector connectionTimeout="20000"
 - port="7080" **URIEncoding="utf-8"**



- 오늘의 명언(1단계)
 - a09_sendExp.jsp
 - 명언을 입력하세요:[]
 - a10_revExp.jsp(다음 page)
 - Do be in mind! : @@@@ @@@@
- 인증처리(2단계)
 - ID []
 - PASSWORD []
 - [로그인]
 - ➔ @@@ 님 환영합니다// @@@님은 회원가입을?



form의 속성 :

- `<form 속성 1=값1 속성 2=값2....>`
- `action`
 - form 하위에 데이터를 전송하는 대상 `page`를 지정
 - `<form action="page01.jsp">`
 - `page01.jsp`에 form 하위에 `name="전달key"`로 전달한다.
 - `action`의 속성을 생략하면 현재 `page`로 데이터를 전송
 - `main.jsp`
 - `<form action=""(X) : 데이터 전달이 main.jsp로 다시 재로딩되면서 전달처리.`



form의 속성 :

- `method="get|post"`
 - `<form method="post">`
 - get방식 : url을 기반으로 query String 형식으로 데이터를 전송
 - 속성값을 지정하지 않으면 default가 get방식으로 url에 데이터 전송이 보인다.
 - query string : 데이터를 전송시,
`key1=value1&key2=value2&key3=value3`
 - `< name="key1" value="value1" />`
 - `< name="key2" value="value2" />`
 - `< name="key3" value="value3" />`
-



get방식 데이터 전송 :

- http://localhost:7080/jspexp/a01_basic/a10_reload.jsp?favFruit=himan
 - url에 직접으로 입력해서 데이터를 전송할 수 있다.
 - 데이터 전송용량에 한계
 - 보안면에서 취약 : url통해서 데이터가 전송되는 것을 확인할 수 있다.
-



post방식 데이터 전송 :

- url이 아니라, 데이터 영역을 이용해서 데이터를 전송하기에 웹브라우저나 서버 상관없이 전송할 수 있는 파라미터 길이 제한 없다.
 - query string 으로 데이터는 전달하지만, url에 통해서 볼 수 없다.



한글 처리 **get/post** :

웹 브라우저
파라미터 홍길동 → 인코딩(####)

웹 서버(WAS)
파라미터 ##### → 디코딩(홍길동)

■ GET방식으로 전달

- 파라미터 값을 지정한 캐릭터 셋으로 알맞게 읽어 오는 방법 WAS 따라 차이..
- TOMCAT 7.0 server.xml 으로
URIEncoding속성값으로 처리
 - euc-kr → euc-kr, uft-8 → utf-8

■ POST방식으로 전달

- request.setCharacterEncoding("utf-8");



form에서 데이터 전송 :

- request
 - .getParameterValues("name01 (멀티 name)")
→ String[]
 - < name="name01"/>
 - < name="name01"/>
 - < name="name01"/>



multi check data :

- 동일한 name 값의 type="checkbox"일 때, 처리 내용
 - `<input type="checkbox" name="fav" value="사과"/>` 사과
 - `<input type="checkbox" name="fav" value="바나나"/>` 바나나
 - `<input type="checkbox" name="fav" value="딸기"/>` 딸기
- `request.getParameterValues("fav");`
- **checked** 된 값만 배열로 가져 옴.

- 아래 내용을 form과 jquery 를 활용하여 처리하세요..

■ 1단계

- 5월에 주요 할 일과들

- 1. []
- 2. []
- 3. []
- [리스트등록]

May schedule
@@@
@@
@@@

■ 2단계

- Bain Trainning(랜덤)
- 1. 5 * 2 = []
- 2. 3 + 4 = []
- 3. 4 % 2 = []

입력내용	결과
@@ * @@=@@	O
@@ + @@=@@	X
@@ %@@=@@	O



request 기본객체의 파라미터 읽기 :

- `.getParameter("name01")`
 - `<input name="name01" />`
- `.getParameterValues("cho")`
 - `<input type="checkbox" name="cho"/>`
 - `<input type="checkbox" name="cho"/>`
- `.getParameterNames()`
 - form 하위에 입력값으로 넘겨지는 모든 name의 값을 Enumeration 호출
 - `<input name="data01"/>`
 - `<input type="checkbox" name="data02"/>`
 - `<select name="cho01">`
 - `Enumeration param = request.getParameterNames();`
 - `while(param.hasMoreElements)`
 - `(String)param.nextElement() → data01, data02, cho01`



request 기본객체의 파라미터 읽기 ●

- .getParameterMap()
 - 웹브라우저에 전송한 마라미터를 맵(이름,값)으로 가져온다.
 - <input name="name01" value="값01"/>
 - <input name="name02" value="값02"/>
 - <input name="name03" value="값03"/>
 - 서버단
 - Map map01 = request.getParameterMap();
 - String[]
names=(String[])map01.get("name01");
 - names[0] → 값01



요청 헤더 정보 처리 :

- 웹 브라우저는 **HTTP** 프로토콜에 따라 요청 정보를 웹 서버에 전송한다.
 - 웹 브라우저의 종류에 대한 정보를 헤드에 담아서 넘겨 줌.
 - **request** 기본 객체에 포함된 헤더정보
 - `getHeader("헤드key값")`
 - `Enumeration getHeaders("헤드값")`
 - `Enumeration getHeaderName()` : 모든 헤더의 이름
-





자신 **page**로딩(a10_reload.jsp) :

```
<%
```

```
// request.getParameter("favFruit") 초기 (X)
```

```
// favFruit ==> null
```

```
String favFruit=request.getParameter("favFruit");
```

```
// 초기 page 이거나 name="favFruit" 값을 입력하지
```

```
// 않을 때는, null 이 있기 때문에 이에 처리를 해 주어야  
한다.
```

```
if(favFruit==null){
```

```
favFruit="";
```

```
}
```



자신 **page**로딩(a10_reload.jsp) :

`%>`

`<body>`

`<form><!-- form에 action의 속성을 지정하지 않으면
현재 page를 호출한다. -->`

좋아하는 과일 : `<input type="text" name="favFruit"
value="<%=favFruit%>" />
`

`<input type="submit" />
`

`</form>`

입력한 과일 : `<%= favFruit%>`



정리 및 확인하기 :



감사합니다 !
