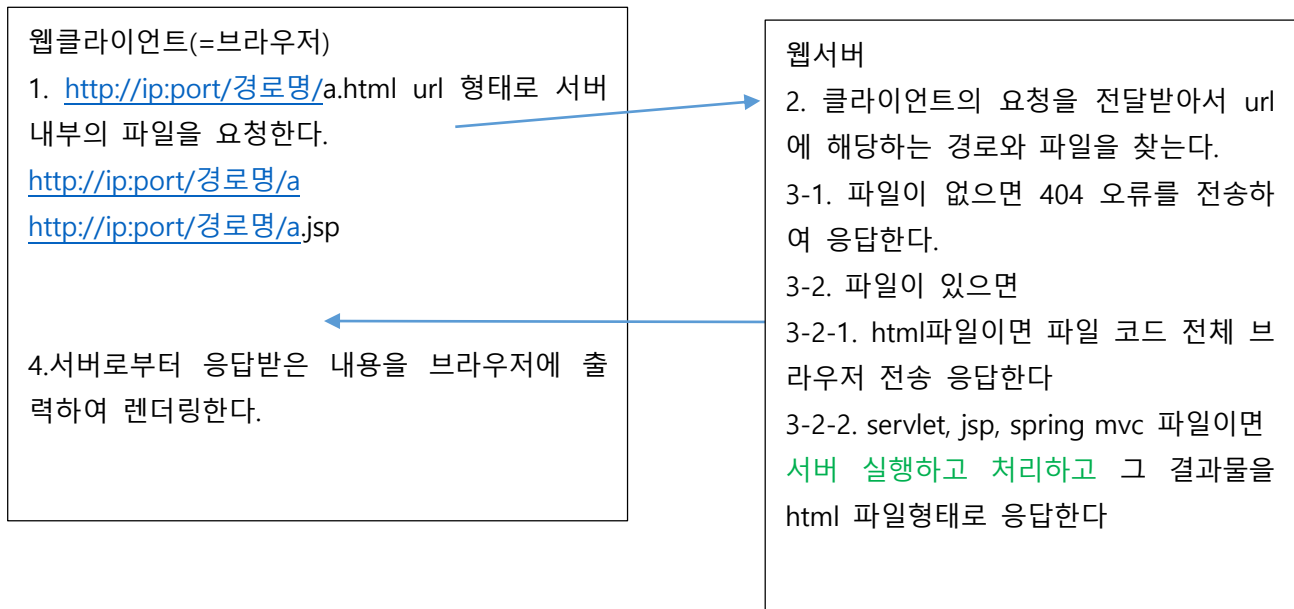


- 웹개발 기술들

1. http 프로토콜을 사용하여 통신하는 웹서버와 웹클라이언트 관련 기술

웹클라이언트	웹서버 + servlet 서버 + jsp 서버
웹브라우저	tomcat=> http 요청 - 처리 - 응답 웹서버 + servlet 서버 + jsp 서버 iis, apache - 웹서버역할
html - 정적 웹페이지 css - 스타일효과 더 추가 정적 웹페이지 java script ==> jquery react vue - "동적" 웹페이지 db 연동 서버자원 기능 못한다	자바언어 - servlet jsp spring (mvc) rdb 이용 - jdbc -> mybatis

- 웹 흐름



- 웹서버에서 실행 가능한 자바 파일들

servlet 4

jsp 3

스프링 + mybatis ??

1.2 웹

http

url

html

css

javascript

3. tomcat + eclipse + jdk

4장 웹 어플리케이션 이해하기

dynamic web project- test 다양한 파일들 포함 1개 프로젝트 생성 위해 필요한 파일들 = web application – test = context - test	java resources 폴더 jdk라이브러리폴더 srcW*.java==.서블릿클래스
	WEBCONTENT 폴더 W a.html http://ip:port/test/a.html WEBCONTENTWWEB-INF폴더Wa.html http://ip:port/test/WEB-INF/a.html ==> FILE NOT FOUND 404오류 WEBCONTENTWWEB-INF폴더 WEBCONTENTW *.HTML *.css *.js *.jpg *.mp3 *.mp4 *.jsp

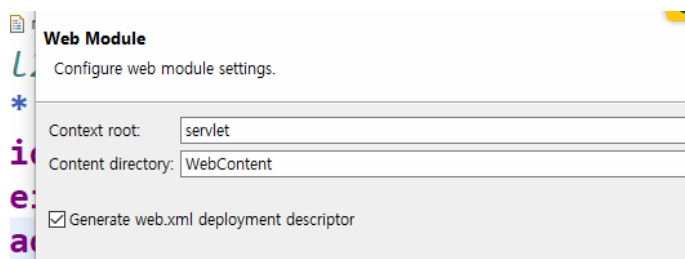
- servlet

1. servlet 프로젝트 생성하기

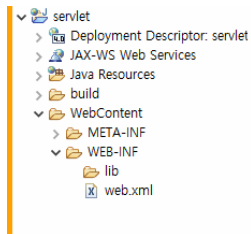
servlet을 실행하기 위해 이클립스에서 dynamic web project를 생성

마지막 화면,

이 때 Web Module 화면에서 아래와 같이 generate.. 부분을 체크한다.



아래와 같은 경로에서 web.xml 파일을 생성했음을 확인한다.



이 때 체크하지 않았다면 , 프로젝트 선택 - 마우스 우클릭 - javaee tools - generate deployment descriptor stub 을 선택할 수 있다.

브라우저 못찾는 오류 발생시 preferences-general-web browser 에서 해당 브라우저 경로 수정할 것

- 실습

자바코드

annotation - 자바소스파일 '@' 시작 단어들. 컴파일러 주석, 실행시 동작 지시

@Override - override annotation - 상속 부모클래스 메소드와 매개변수 동일해야 한다

comment - 자바소스파일 주석-컴파일러 인식, 컴파일 제외

```
/* */
```

```
//
```

서블릿 @WebServlet 사용하기/web.xml 사용하기

1> 서블릿 클래스 정의

file-new-servlet- 패키지명, 클래스명 입력 - url mapping ('/FirstServlet')-종료

```
@WebServlet("/FirstServlet")
class FirstServlet extends HttpServlet { ==> A 서블릿 정의=server side applet(자바클래스)
    doGet(){ 실행 결과 요청 브라우저 전송 }
```

자동 컴파일

2> 실행

run - run on server - 브라우저 자동 오픈 주소 입력

<http://ip:port/servlettest/FirstServlet> 요청시

servlet3 이상 1. @WebServlet 추가 2. web.xml 파일 활용	url mapping시 @WebServlet("/second") class A...{}
servlet 2.5 web.xml 파일 활용	url mapping시 <servlet> <servlet-name>ss</servlet-name> <servlet-class>SecondServlet</servlet-class>

```

</servlet>
<servlet-mapping>
    <servlet-name>ss</servlet-name>
    <url-pattern>/second</url-pattern>
</servlet-mapping>

```

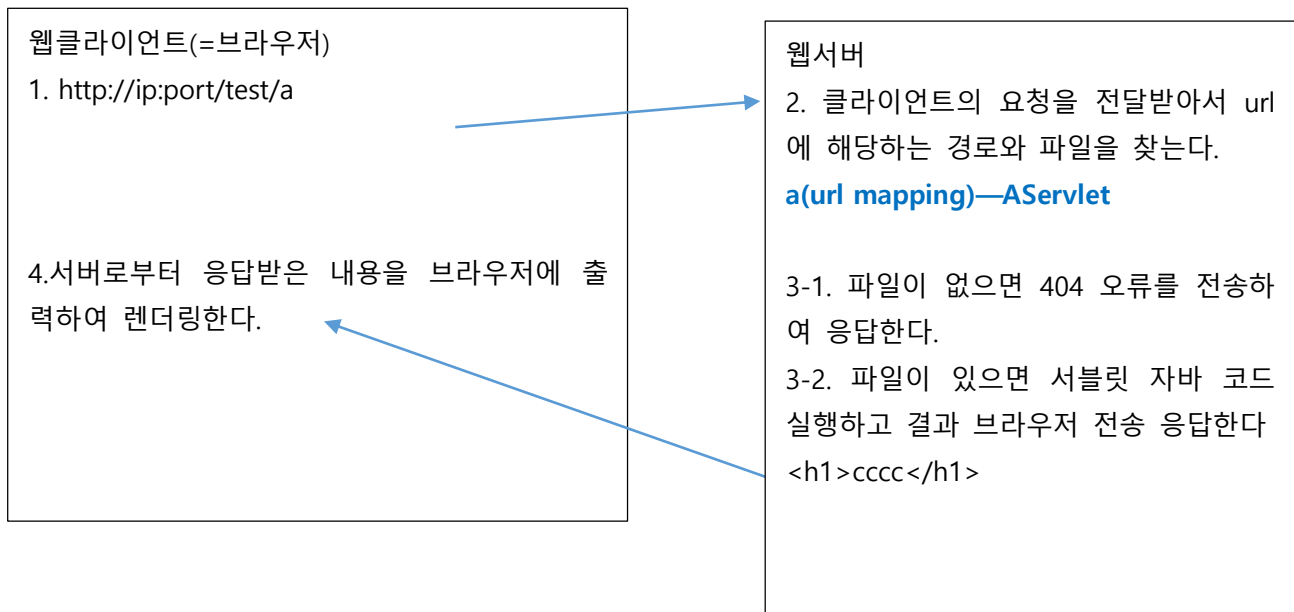
==> 서블릿 3 이상

서블릿 2.5 ==> spring 5(내부 서블릿 2.5 연동)

5장

서블릿 = server side java class + 웹클라이언트 요청 실행 구조

- 웹 흐름



- 구현(http요청방식 – get/post방식 나누어 요청)

@WebServlet("/a")

```
class A extends HttpServlet {
```

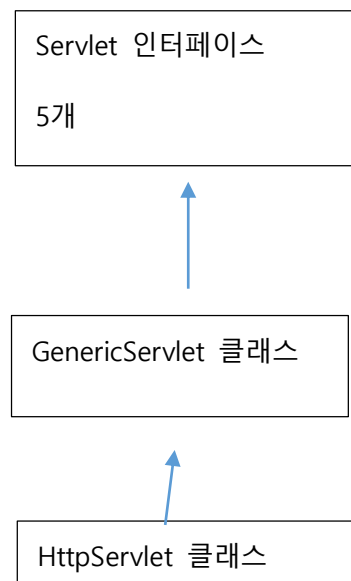
```
doGet(){.....}
```

```
doPost(){}
```

```
}
```

```
class A extends GenericServlet { service(){.....} }
```

```
class A implements Servlet { 모든 메소드 오버라이딩 5개 }
```



- 서블릿 실행 흐름

클라이언트	서버
1. 서블릿 요청(/a)	2. 서블릿컴파일클래스 찾아서 객체 생성. 메모리 로딩
6. 브라우저 출력	3. init 메소드 호출
	4. 8. doGet 메소드 호출- 요청/응답
7. 서블릿 요청(/a)	5. 9. 응답 브라우저 전송
10. 브라우저 출력	11.서버종료 / 서블릿수정
	메모리정리=서블릿삭제
	destroy() 메소드 호출

4,5장

6장 요청과 응답

서블릿==> 요청 - 분석처리 - 응답

= 다른 컴퓨터 자원 (db, 자바 로직 실행....) 이용하는 웹페이지

1>클라이언트 요청받는다

2> 요청 분석(로그인, 회원가입, 게시물1개, 다른페이지이동,...)

db 연결 - 회원정보테이블 id, pw 존재 판단한다

3>"정상로그인", "비정상로그인" 클라이언트에게 응답한다

```
@WebServlet("/a")
class A extends HttpServlet{
doGet( HttpServletRequest request , HttpServletResponse response ) {
    request.xxxx() / 자바모든로직 / response.xxxx()
}
}
```

login.html <form action="http://localhost:9090/servlettest/login" <input...type=text name="i" id="d" > <input .. <input type=submit value="로그인"	login-->LoginServlet.java doGet(..HttpServletRequest request , HttpServletResponse response){ request.getParameter("i"); ...
---	--

<http://localhost:9090/servlettest/login?id=bluejeansh&pw=32453242353>

<http://localhost:9090/servlettest/login?id=spring&pw=spring>

<https://search.daum.net/search>

?

w=tot

&DA=UME

&t_nil_searchbox=suggest

&sug=

&sugo=5

&sq=servlet

&o=5

&q=servlet