

5장. Tomcat 서버 설치 및 eclipse 연결

1. Tomcat 서버 설치

(1) Tomcat

- Servlet 이나 JSP의 컨테이너
- tomcat.apache.org
 - => Download/Tomcat 8.0
 - => 8.0.36/Core/32-bit/64-bit Windows Service Installer 클릭, 저장
 - => "apache-tomcat-8.0.36.exe" 저장됨

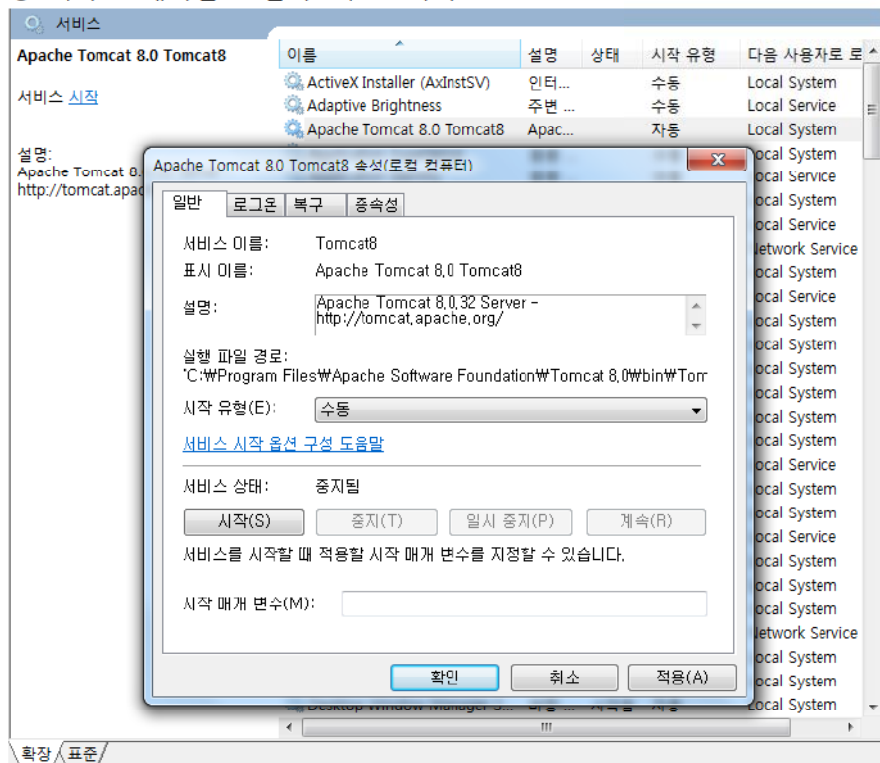
(2) 설치

- "Choose Components"에서만 "Full"로 변경하고, 기본 설정으로 설치한다.
- 다 설치 후, 아래 항목 체크 해제후, "Finish" 눌러서 마무리 한다.
 - => "Run Apache Tomcat" 체크 해제
 - => "Show Readme" 체크 해제

(3) 서비스 수정 (시작 유형 수정)

- Oracle : 자동 실행으로 설정
- Tomcat : 수동 실행으로 설정

① 시작 - 제어판 - 관리도구 - 서비스

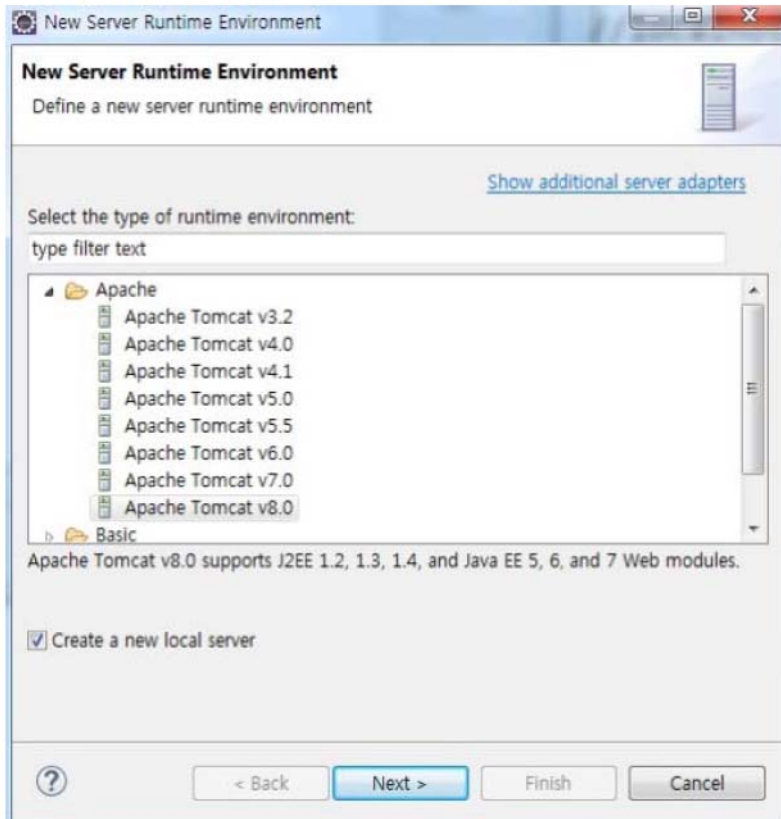


(4) Tomcat 실행하기

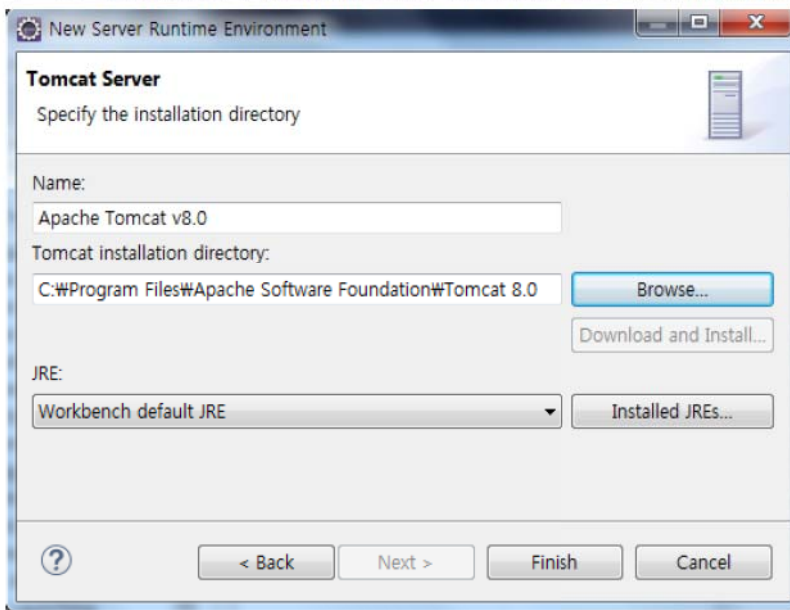
- C:\Program Files\Apache Software Foundation\Tomcat 8.0\bin\Tomcat8.exe
- 브라우저에서 테스트하기
 - => URL
 - http://localhost:8080/index.jsp
 - => Tomcat 8.0/webapps/ROOT/index.jsp
 - http://localhost:8080
- 고양이 그림이 올라오면 성공.

2. Eclipse에 Tomcat 연결하기

- (1) Perspective에서 "Java EE" 선택하기
- (2) 메뉴 -> Window -> Preferences 선택
- (3) Server -> Runtime Environments 선택
 - => "Server Runtime Environment"에서 "Add" 버튼 클릭
 - => "New Server Runtime Environment" 창에서
 - > "Apache/Apache Tomcat v8.0" 선택 -> "Next" 버튼 클릭

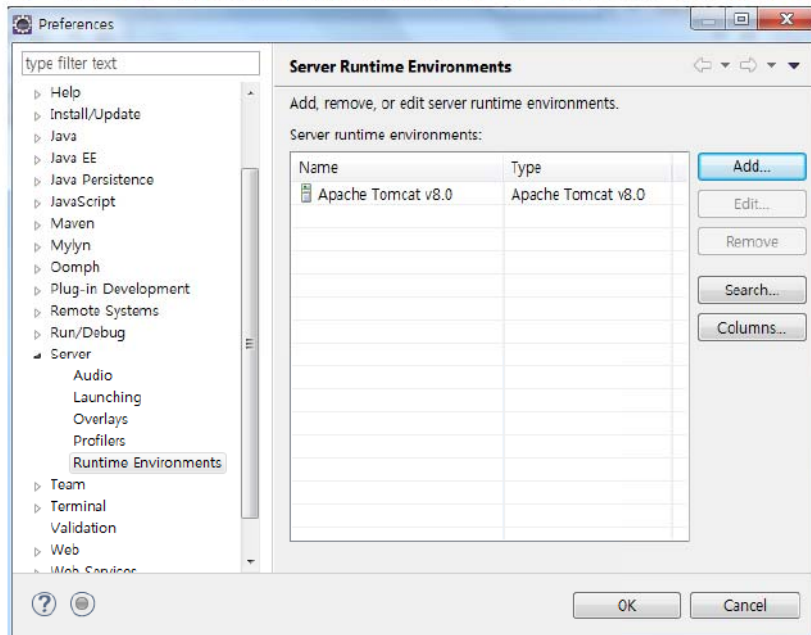


- => "Tomcat Server" 창에서
 - > "Browse" 버튼 클릭
 - > "C:\Program Files\Apache Software Foundation\Tomcat 8.0" 선택



- > "Finish" 버튼 클릭

=> "Server Runtime Environment"에 "Apache Tomcat v8.0"이 추가된다.

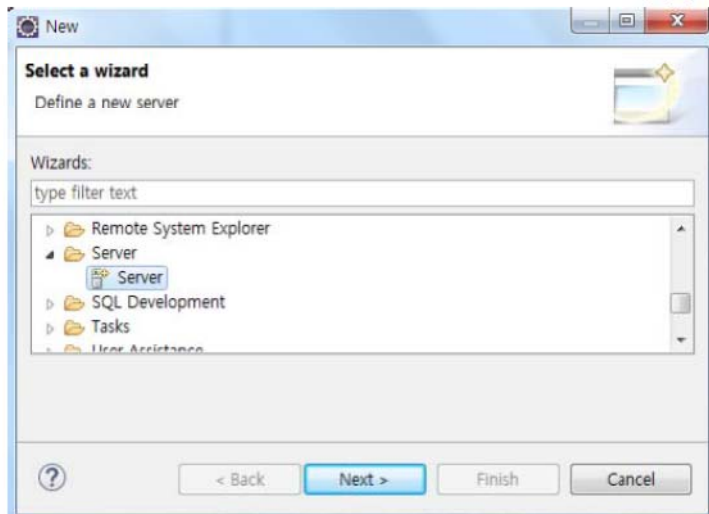


=> "OK" 버튼 클릭

(4) Server 추가하기

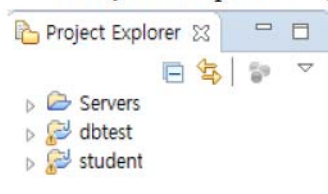
=> "Project Explorer" -> 오-마 -> New -> Other 선택

=> "Select a wizard" 창 -> "Server/Server" 선택 -> "Next" 버튼 클릭



=> "Finish" 버튼 클릭

=> "Project Explorer"에 "Servers" 가 추가된다.



(5) Encoding 설정하기

=> 메뉴 -> Window -> Preferences 선택

=> "Web" 항목 선택

- > CSS Files -> Encoding을 "utf-8"로 변경
- > HTML Files -> Encoding을 "utf-8"로 변경
- > JSP Files -> Encoding을 "utf-8"로 변경

* eclipse에 연결 가능한 기능

- (1) JDK
- (2) DB
- (3) Tomcat
- (4) 브라우저
- (5) spring

* 용어 정리

Project : eclipse -> java

Context : tomcat -> jsp

예제1. html 파일 만들기

Dynamic Web Project : 5-test

HTML File : WebContent/hello.html

<실행 결과>

안녕하세요!! 홍길동

<작업 순서>

1. 프로젝트 만들기

- ① "Project Explorer" -> 오-마 -> New -> Project 선택
- ② "Select a wizard" 창 -> "Web/Dynamic Web Project" 선택 -> "Next" 버튼 클릭
- ③ "Dynamic Web Project" 창
 - > Project name : test
 - > "Next" 버튼 클릭
- ④ "Java" 창
 - > "Next" 버튼 클릭
- ⑤ "Web Module" 창
 - > "Generate web.xml deployment descriptor" 체크
 - > "Finish" 버튼 클릭

2. HTML 파일 추가하기

- ① "test/WebContent" 폴더 -> 오-마 -> New -> "HTML File" 선택
- ② "HTML" 창
 - > File name : hello
 - > "Finish" 버튼 클릭

3. hello.html 실행하기

- ① "Servers" View
 - > "Tomcat v8.0 Server at localhost" 선택
 - > 오-마 -> "Add and Remove..." 클릭
 - > "Add and Remove" 창 -> "test" 선택 -> "Add" 클릭
 - > "Finish" 버튼 클릭
 - > 실행 버튼을 눌러서 서버 실행 시킨다.
- ② 브라우저를 실행한후, URL 주소 추가후 결과 확인한다.
 - http://localhost:8080/5-test/hello.html => 본인 컴퓨터
 - http://59.5.12.23:8080/5-test/hello.html => 다른 컴퓨터

4. hello.html의 위치에 따른 실행 방법

① C:\Program Files\Apache Software Foundation\Tomcat 8.0\webapps\ROOT\hello.html

Context : ROOT ← default Context

webapps

http://localhost:8080/ROOT/hello.html (X) ← ROOT는 쓰면 안된다.

ROOT가 아닌 다른 Context일 경우에는 반드시 URL에 써 주어야 한다.

<http://localhost:8080/hello.html>

② D:\jsp\workspace\5-test\WebContent\hello.html

Context : 5-test

<http://localhost:8080/5-test/hello.html>

5. 파일 작업

① 5-test/WebContent/hello.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Insert title here</title>
```

```
</head>
```

```
<body>
```

```
안녕하세요!! 홍길동<br>
```

```
</body>
```

```
</html>
```


6장. HTML 기본동작

* URL의 구조

http:// www.macho.com :80 / webpubs / home.html

값	구분	설명
http://	프로토콜	통신 규격의 종류를 의미한다. • http:// → 웹 • ftp:// → 파일전송
www.macho.com	도메인	접속 대상 컴퓨터에 부여된 주소값 실제 접속시에는 도메인이 IP주소로 변환된다.
:80	포트번호	컴퓨터에서 특정 프로그램에 대하여 허용하는 통신 채널. 웹의 경우 80포트를 기본으로 사용 하고, 80포트일 경우 생략 가능하다.
webpubs	디렉토리	접속 대상 컴퓨터에 설치된 웹 서버 프로그램 과 연결된 폴더의 하위 폴더
home.html	파일명	폴더 안에 존재하는 파일이름. 웹 서버는 이 파 일의 내용을 웹 브라우저에게 전송한다.

1. HTML을 시작하며

(1) HTML (Hyper Text Markup Language) 이란?

- 인터넷 전용 브라우저에 그림이나 text를 표현할 수 있도록 해 주는 언어
- Hyper Text는 Link를 사용하여 공간의 초월의미가 부여되는 글자들을 말한다.
- Markup이란 컴퓨터가 문서를 생성할 수 있게 하는 신호체계를 설정하는 것이다.

(2) 홈페이지는 HTML로 Page를 구성하여 만들어진다.

① 일반 편집문서 : 일반 text로 만들어진 완성형 아스키(ASCII)방식

한글, MS-Word, 메모장, UltraEdit, Editplus, SimpleText(Mac)

② 전용 에디터 : 나모, Frontpage, DreamWeaver, VisualPage(Mac)

※ WYSIWYG(What You See Is What You Get 위지윅)방식이란?

Drag and Drop 만으로 화면에 보여지는 그대로를 문서로 만들 수있는 방식

※ HTML전용 에디터를 과신하지 말 것

- 에디터 작업도중 실행취소하는 경우 소스는 그대로 남아 있는 경우 발생할 수 있다
- 적용할 수 없는 곳에 강제적용시 에디터가 다운 될 수도 있다.
- 직접 손으로 짜맞추는 습관이 있어야 많은 테크닉을 기를 수 있고,
문제 발생시 대처할 수 있다.

(3) HTML을 꼭 배워야 하나?

웹환경의 직업을 가지려면 HTML은 기본이다.

(4) 기본적인 HTML문서 만들기

① 작업도구 설정 : Internet Explorer같은 Browser와 메모장 같은 Editor

② Editor로 입력하고 저장 : 확장자는 반드시 htm, html 이어야 한다.

요즘은 동적인 웹사이트를 위해 jsp, asp, php 파일로도 만든다.

③ 확인 : 웹브라우저로 확인

2. HTML 사용하기

(1) HTML의 특징

- ① 태그는 대소문자 구분이 없다. (그러나, CSS에서의 속성명은 대소문자 구분있음)
- ② 확장자는 반드시 htm, html로 만들어야 한다.
IBM계열의 PC와 Window운영체제는 3자리코드의 확장자 개념을 쓰고
Unix나 매킨토시 계열은 중요시되진 않지만 4자리코드의 확장자 개념을 사용하므로
htm, html 둘 다 사용하게 되었다. (htm이 컴퓨터상에서 html보다 빨리 읽힌다)
- ③ Enter, Spacebar, Tab은 작용되지 않는다. (특정기호나 태그로 구현)
- ④ 태그의 순서(시작태그, 종료태그)를 반드시 지켜야 웹브라우저가 인식한다.
- ⑤ 파일이나 폴더명은 반드시 여백이 없는 영문자일 것.

(2) HTML의 구성요소

- ① 태그(Tag) : '<'와 '>'로 묶인 명령어
<태그>...</태그>의 형태를 이룬다. --> 요소(Elements)라고 한다.
- ② 속성(Attribute) : 시작태그에 필요에 따라 속성이라는 옵션을 가진다.
- ③ 값(Value) : 속성에 대한 값을 의미 (" "또는 ' '안에 넣어 표시)
<태그 속성1="값1" 속성2="값2">...</태그>

(3) HTML문서의 기본구조(필수표기)

<html> -----> html문서의 시작
 <head></head>-----> 문서의 머리부분 (웹페이지상에는 보이지 않는다)
 <body></body>-----> 문서의 몸통부분 (웹페이지상에 보이는 모든 정보)
</html> -----> html문서의 끝

(4) 주석달기

- <!-- 삽입할 설명 -->
- 차후의 수정이나 보완을 쉽게 할 수 있게 한다.
 - <!-- 이하의 모든태그는 없는 것처럼 인식된다.

3. <head> 태그

(1) <title>...</title> ----> 페이지의 제목달기

- 웹브라우저 상단에 사이트의 제목을 보여주는 태그
 - 즐겨찾기에 적히는 제목이기도 하다 : 사이트를 홍보할 수 있는 훌륭한 마케팅 수단이 됨
 - 되도록 한글로 작성: 브랜드명, 명료한 사이트특성. 한글도메인명도 좋다.
- c.f, 꼭 <head>...</head>사이에 삽입하지 않아도 보는데는 문제없음

(2) <meta>...</meta> ----> 홈페이지의 홍보담당

① 한글폰트 지정

<meta http-equiv="content-type" content="text/html; charset=euc-kr"> 또는
<meta http-equiv="content-type" content="text/html; charset=utf-8">

② keywords 지정 : 검색사이트의 검색엔진이 수시로 수집하는 키워드 ----> 홍보효과

<meta name="keywords" content="웹디자인,웹마스터,webdesign,webmaster,html">
coma(.)를 제외한 최대 256자 까지 지정할 수 있다.

③ description 지정 : 검색엔진에서 listup된 홈페이지에 대한 설명문

<meta name="description" content="웹디자인을 배우는 사람을 위한 사이트입니다.">

④ 문서이동 : 방문시 몇 초 후에 다른 곳으로 이동하게 하는 태그

<meta http-equiv="refresh" content="초간격:url= 이동할 홈페이지주소">
초간격을 0으로 하면 접속됨과 동시에 바로 다른 사이트로 이동한다.

⑤ 제작자명 지정

<meta name="author" content="(주)스타트디자인">

(3) <base href="절대경로url">

절대경로를 지정해 두면 이후에 쓰여진 상대경로들은 절대경로에 붙여져서 인식된다.

<head>에서 <base href="http://www.hd4u.co.kr">라 지정하고 다음과 같이 쓰면

이렇게 링크했을때 주소는? <a>

실제로 인식되는 경로는 http://www.hd4u.co.kr/service/service-1.htm이 된다.

(4) 이 외에도 <head>에는 CSS와 Javascript 가 삽입되기도 한다.

4. <body> 태그

(1) <body>는 웹 브라우저상에 보여지는 모든 내용들이 표현되는 곳이다.

(2) <body bgcolor="#16진수색상값/색상명" -----> 배경색상

text="#16진수색상값/색상명" -----> 문자색상

link="#16진수색상값/색상명" -----> 한번도 방문한 적이 없는 링크의 색상(파랑)

vlink="#16진수색상값/색상명" -----> 한번 방문한 적이 있는 링크의 색상(보라)

alink="#16진수색상값/색상명" -----> 링크를 클릭했을때의 색상(빨강)

요즘은 CSS 가상클래스로 활용되는 추세

leftmargin="숫자" topmargin="숫자" -----> internet Explorer에서 인식하는 태그

marginwidth="숫자" marginheight="숫자" >Netscape에서 인식하는 태그

지정하지 않으면 기본값은 8pixel offset한다.

background="이미지파일명" -----> 배경그림

bgproperties="fixed" -----> 배경그림 고정하기 (Watermark :불법복제방지)

CSS의 background-repeat 지정으로도 구현

</body>

(3) <address>...</address>

홈페이지의 주소, E-mail, 회사명등을 이탤릭체로 보여준다. (링크기능이 있는 건 아니다)

5. 텍스트 처리 태그

(1)
 줄바꿈 태그

- line BReak 의 약자. HTML 문서의 어느곳에서든 강제로 줄바꿈할 수 있다.
- 여러개 줄바꿈은
을 여러번 이어 쓰면 된다.
- 종료태그가 없다.
예)

 은 세줄 줄바꿈

(2) <hn>...</hn> n:1~6 의 숫자

<h1 align = "left/right/center/justify"> ---> 글자의 정렬(왼쪽/오른쪽/중앙/가지런한정렬)
</h1>

- Heading 의 약자. 문장의 타이틀 제목으로 사용된다. 볼드체로 표현
- 글자의 크기는 H1>H2>H3>H4>H5>H6 순이다
- 자동 줄 바꿈기능을 포함한다.
- 숨겨진 기능
메타태그처럼 검색엔진이 <H1>,<H2>등의 큰 제목을 검색수집하여 결과에 보여주기도 한다.
시각장애인용 컴퓨터에서 <H>태그를 사용한 문장은 더 강하게 읽어주게 된다.

(3) <hr> 선그리기

Horizontal Rule 의 약자. 텍스트사이에 선을 긋는다.

<hr width = "선의 길이값, 퍼센트값"
size = "선의 두께값" -----> 퍼센트값은 없다.
align = "left/righ/center"
color = "색상값"
noshade > -----> 입체감 없애기

- 세로선은 그을 수 없다. 자동 줄바꿈기능 포함
- 종료태그가 없다.

(4) ... 글자조절

----> 최근에는 CSS로 표현하게 되어 잘 사용하지 않게됨
 1~7까지 지정되어 있고 7이 가장 크다.
기본값은 3 이고 +, -상대값 으로 지정할 수 있다.
face = "서체명" -----> 운영체제 일반글꼴을 쓸 것(사용자배려)
color = "색상값"> ...

(5) <basefont>글자의 기본값 설정

<basefont size="글자크기" face="서체명" color="색상값">
- 페이지전체에 적용될 글자의 기본값 지정
- <head>...</head>내에 쓰여짐.
- 종료태그가 없다.

(6) <p>와 <p>...</p> 단락태그

- Paragraph의 약자
- <p> :

과 같은 효과.
과 달리 연속으로 사용해도 <p>한번으로 인식한다.
- <p align="left/right/center">...</p> : 전체 텍스트에서 독립된 문단을 만들어준다.

(7) <center>...</center> 글자, 이미지, 표 등을 중앙에 정렬

(8) <div>...</div> 그룹화 태그

<div align = "left/right/center" style = "css적용">...</div>
- 레이아웃을 만들 때도 활용된다.

6. 문자장식 태그

(1) 물리태그 : 글자의 내용과는 상관없이 외향적으로 드러나는 문자 장식에 쓰는 태그

...	: 굵은 글자로 만든다.
<i>...</i>	: 이탤릭체로 만든다.
<tt>...</tt>	: 타자체로 만든다
^{...}	: 윗첨자
_{...}	: 아래첨자
<u>...</u>	: 글자에 밑줄을 그어준다.
<s>...</s>	: 글자의 중앙에 선을 그어준다. 취소 또는 삭제의 의미로 사용
<big>...</big>	: 기존에 사용된 글자보다 크게 표시
<small>...</small>	: 기존에 사용된 글자보다 작게 표시
<blink>...</blink>	: 글자를 깜박이게 한다. 넷스케이프전용

(2) 특수문자 표기법

- HTML 문서상에서 직접 쓸수 없는 문자들(<, >, ", & 등등)을 하나의 문자로 표현하기 위해 특수문자를 사용한다.
- 특수문자 엔티티 코드(Entity code) 형식
=> &로 시작하여 ;로 끝난다.

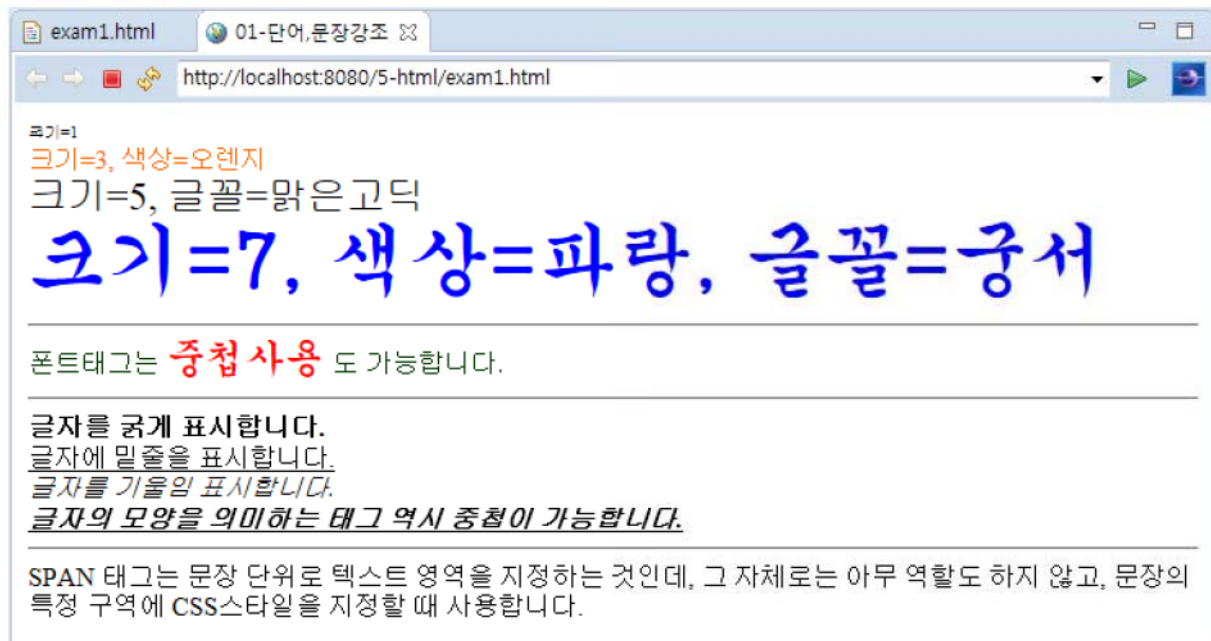
기호	내용	문자실체 참조	수치문자 참조
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
©	Copyright (저작권)	©	©
®	Registered trademark (등록상표)	®	®
™	Trademark (상표)	™	™
"	Quotation mark (이중 인용부호)	"	"
'	Opening single quote mark (인용부호 시작)	‘	‘
'	Close single quote mark (인용부호 끝)	’	’
·	Medium list dot (가운데 점)	·	·
	Space (공백)	 	

예제1. 단어,문장강조

Dynamic Web Project : 6-html

HTML File : /WebContent/exam1.html

<실행 결과>



예제2. 이미지

Dynamic Web Project : 6-html

HTML File : /WebContent/exam2.html

Image File : /WebContent/img/

=> img1.jpg, img2.jpg

<실행 결과>



7. <LIST> 리스트 목록만들기

(1) 비순차적 목록

- unordered list 의 약자. 세부리스트항목은 태그로 사용
- 에는
태그 속성이 포함된다.

```
<ul type="disc/circle/square"> -----> bullet 모양지정 검정색원/흰색원/검정사각형
  <li type="disc/circle/square">내용1 ----> <li>에서 개별적으로도 지정할 수있다.
    :
  <li>내용n
</ul>
```

(2) 순차적 목록

- Ordered List의 약자. 순서를 표현하는 방식을 지정할 수 있다.

```
<ol type="A/a/I/i/1" -----> A,a(영문대,소문자)/I,i(그리스수대소문자)/1(아라비아숫자)
  start="적용할 시작순서">
  <li type="A/a/I/i/1" value="적용할 시작순서"> ----> <ol>에서는 start, <li>에서는 value
</ol>
```

(3) <dl>정의 목록 : 어떤 용어를 정의하여 풀어 나열한다.

Definition List의 약자. DT(Definition Term), DD(Definition Description)과 함께 쓴다.

```
<dl>
  <dt>정의 -----> <dd>는 <dt>가 적용된 곳에서 한탭 들여쓰기 적용된다.
  <dd>정의에 대한 해설          마찬가지로 <br>태그 속성이 포함된다.
</dl>
```

8. <a> 링크결기

(1) 텍스트나 이미지에 링크결기

Anchor(닻)으로 원하는 곳으로 자유롭게 이동하기

```
<a href = "파일이름 or URL">텍스트나 이미지</a>
```

```
<a href = "링크할 파일주소"
```

```
  name = "현재문서의 위치표시명(책갈피)"
```

```
  target = "frame 명" ----->frame에서 사용
```

```
  title="도움말 형식의 툴팁으로 사용할 글"
```

```
  style = "적용할 스타일시트 적용" ></a>
```

(2) 특정위치로 이동하기

① 현재 페이지내에서

```
<a href="#원하는 위치의 책갈피명"></a>
```

```
<a name="원하는 위치의 책갈피명"></a>
```

② 다른 페이지의 특정위치이동

```
<a href="URL#원하는 위치의 책갈피명"></a>
```

```
<a name="원하는 위치의 책갈피명"></a>
```

(3) 링크로 전자메일 보내기

```
<a href="mailto:메일주소"></a>
```

메일의 제목, 본문내용의 미리 설정방법은

```
<a href="mailto:메일주소?subject=편지제목&BODY=본문의 내용을 적습니다">
```

(4) 또 다른 링크대상

이미지 파일(gif, jpg), 사운드파일(wav, au, mid), 동영상파일(mpeg, mov),

압축파일(zip): 사용자의 pc로 다운로드 하게 한다.

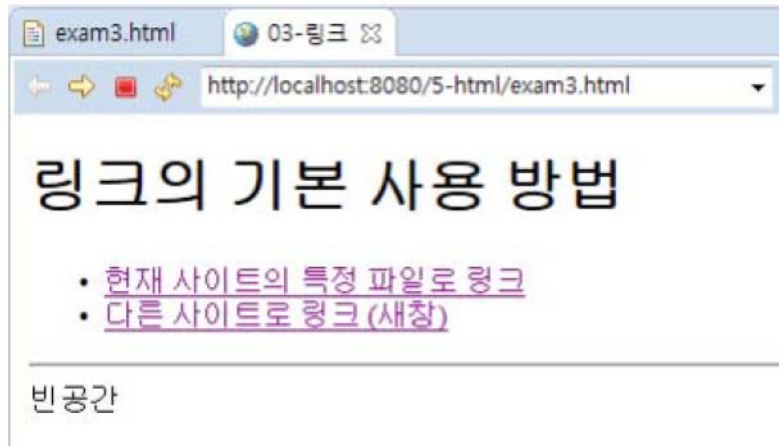
——> 단 MS Office가 있는 pc에서는 바로 볼수 있는 파일을 다운로드하고 싶으면
마우스 오른쪽 버튼의 다른이름으로 대상을 눌러 다운로드한다.

예제3. 링크

Dynamic Web Project : 6-html

HTML File : /WebContent/exam3.html

<실행 결과>



9. <form> 입력양식 만들기

(1) FORM이란?

- 정보를 입력 또는 선택하고 버튼을 클릭하며 정보를 서버에 전달하고 응답받는 양방향의 의사소통을 지원하는 입력양식이다.
- Form 의 내부에는 <input>, <select>, <textarea> 등의 태그들이 포함된다.
- name 속성의 값과 사용자가 입력한 값이 한 쌍이 되어 서버측에 전달된다.

(2) 구조

<form action="URL" -----> 동작시킬 파일이 있는 경로
 enctype="MIME형식" -----> 폼을 서버에 전송하는데 사용하는 형식
 method="get/post" -----> 데이터가 반환되는 유형을 결정. 기본값은 get, 주로 post
 target="프레임명"> -----> 폼입력된 데이터가 서버를 거쳐 처리된 결과 표시할 프레임
<input type="입력받는 형식" --> text, password, checkbox, radio, submit, reset이 있다.
 name="입력변수명" ----> 서버에 전송될 양식의 이름
 value="기본입력값" ----> 입력변수의 초기값
 size="입력양식길이" ----> 입력필드길이
 maxlength="입력문자수" --> 실제 입력할 수 있는 최대문자수
 src="URL" -----> 이미지버튼을 만들때 이미지의 주소
 checked> -----> 체크박스과 라디오버튼일 경우 기본선택값 설정

각 type별 적용 속성이 다르다. 예를들면,

<input type="text" name="var1" value="이름" size="10" maxlength="8">
<input type="password" name="var1" value="이름" size="10" maxlength="8">
<input type="checkbox" name="var1" value="노랑" checked>-->하나의 name 에
<input type="checkbox" name="var1" value="빨강" > value값만 다르게 구성
<input type="checkbox" name="var1" value="파랑" checked> 여러개 선택가능
<input type="radio" name="var1" value="노랑" checked>--> 단 하나의 선택만 가능
<input type="radio" name="var1" value="빨강" >
<input type="radio" name="var1" value="파랑" >
<input type="submit" value="버튼명" >
<input type="reset" value="버튼명" >
<input type="image" src="이미지경로명" border="이미지테두리두께" >
<input type="file" name="이름" size="크기"> -----> 찾아보기 버튼이 생긴다.
<input type="hidden" name="이름" value="데이터값"> -> 웹상에는 보이지 않는 변수처리

<select name="입력변수명" -----> 서버에 전송될 양식의 이름
 size="보여질 항목갯수" > 드롭다운메뉴의 줄수지정
 multiple> -----> 다중선택메뉴일 경우
 <option value="메뉴값" -----> 실제 데이터처리되는 입력값
 selected> -----> 기본선택 지정
 메뉴항목 -----> 화면상에 리스트되는 항목
 사용되는 구조
 <option value="메뉴값1" selected> 메뉴항목1
 <option value="메뉴값n">메뉴항목n
</select>

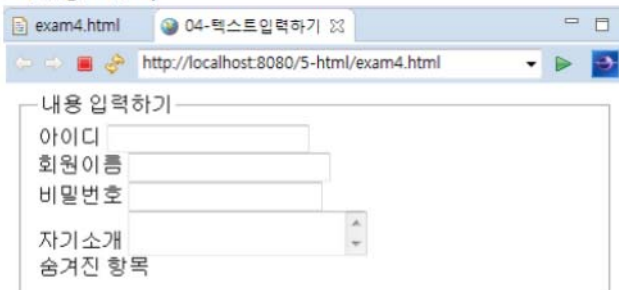
</form>

예제4. 텍스트입력하기

Dynamic Web Project : 6-html

HTML File : /WebContent/exam4.html

<실행 결과>




The screenshot shows a web browser window with the address bar displaying 'http://localhost:8080/5-html/exam4.html'. The page content is a form titled '내용 입력하기' (Content Input). The form contains five input fields: '아이디' (ID), '회원 이름' (Member Name), '비밀번호' (Password), '자기소개' (Self-introduction), and '숨겨진 항목' (Hidden Item). The '자기소개' field has a small upward arrow icon on its right side, indicating it might be a text area or have a scroll function.

예제5. 항목선택하기

Dynamic Web Project : 6-html

HTML File : /WebContent/exam5.html

<실행 결과>



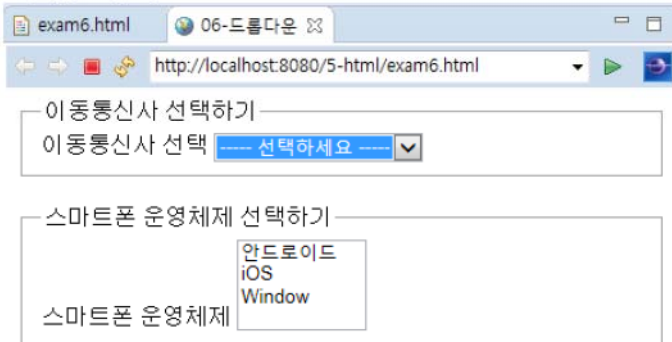
The screenshot shows a web browser window with the address bar displaying 'http://localhost:8080/5-html/exam5.html'. The page content is a form titled '취미 선택하기' (Hobby Selection). The form contains two sections: '취미 선택하기' (Hobby Selection) with three checkboxes labeled '축구' (Soccer), '농구' (Basketball), and '야구' (Baseball); and '성별 선택하기' (Gender Selection) with two radio buttons labeled '남자' (Male) and '여자' (Female).

예제6. 드롭다운

Dynamic Web Project : 6-html

HTML File : /WebContent/exam6.html

<실행 결과>

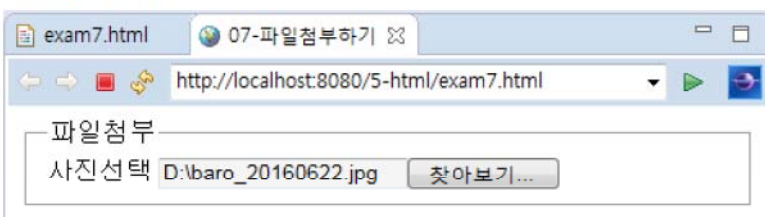


예제7. 파일첨부하기

Dynamic Web Project : 6-html

HTML File : /WebContent/exam7.html

<실행 결과>

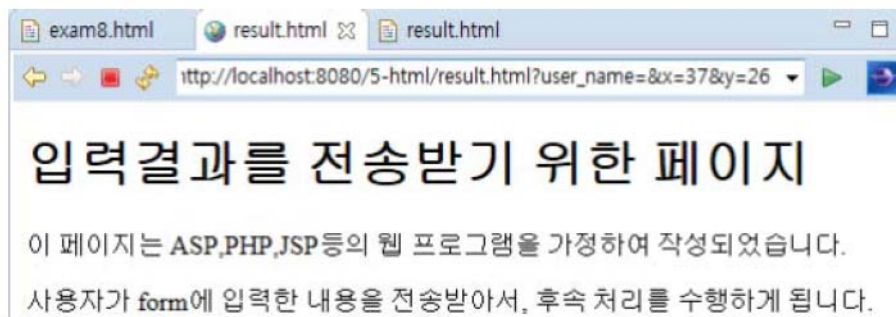
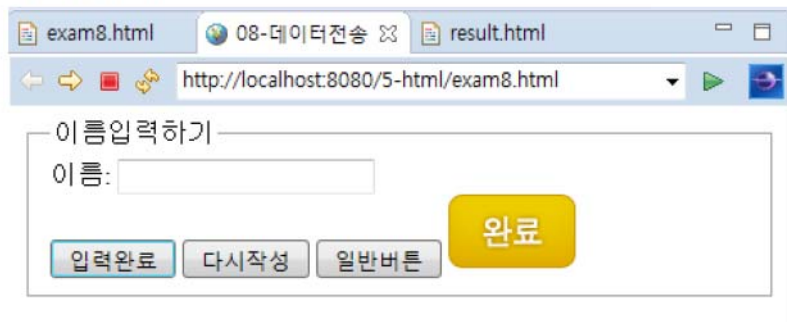


예제8. 데이터전송

Dynamic Web Project : 6-html

HTML File : /WebContent/exam8.html
/WebContent/result.html

<실행 결과>



10. <table>

(1) Table

- ① 표는 분할된 이미지를 모아준다.
무게가 큰 이미지를 분할하여 이음새없는 하나의 이미지로 재구성해주는 역할
- ② 홈페이지의 전체 레이아웃은 표를 이용한다.
- ③ 테이블 셀 하나에도 배경이미지나 색상을 삽입할 수 있다.

(2) 기본형식

```
<table>
  <caption>...</caption>    ---> 표의 상단에 제목을 적을때 사용. 생략가능.
  <tr>                      ---> Table Row. 표의 한줄을 정의
    <th>...</th>            ---> 생략가능. Table Header의 약자. 표의 내부의 제목으로 사용
    <td>...</td>            ---> Table Data. 표안에 있는 하나의 셀. 아주 중요한 요소
  </tr>
</table>
```

- 테이블안에 테이블을 또 지정할 수 있다.

(3) 속성

```
<table border="0"          ---> 외곽테두리 두께를 0로 지정. 기본값은 "1"
  width="너비"              ---> 전체 테이블의 가로크기. 픽셀값과 %값이 있음
  height="높이"             ---> 전체 테이블의 세로크기. 픽셀값과 %값이 있음
  align="left/right/center" ---> 문서에서의 테이블의 위치
  cellpadding="픽셀값"     ---> 테두리와 내용데이터와의 간격. 상하좌우 모두 적용
  cellspacing="픽셀값"     ---> cell과 cell사이의 투명한 간격
  bgcolor="색상값/명"       ---> 배경색
  background="이미지명"     ---> 배경이미지 파일 주소
  bordercolor="색상값/명"   ---> 테이블의 전체 외곽선에 색상을 지정
  bordercolorlight="색상값/명" ---> 외곽선의 밝은 부분의 색상지정(익스플로러만)
  bordercolordark="색상값/명" ---> 외곽선의 어두운 부분의 색상지정

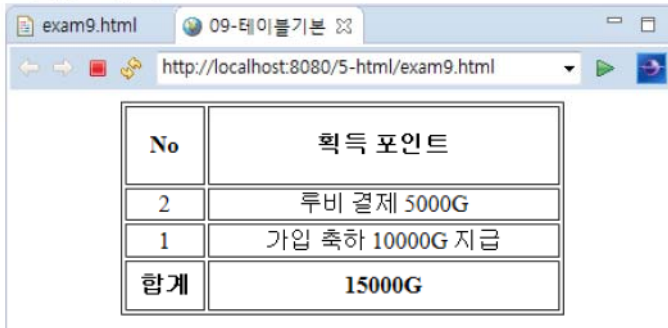
  <caption align="top/bottom"> ---> 상하조절
</caption>
  <tr align="left/right/center" ---> 행에 있는 모든요소에 적용된다.
    bgcolor="색상값/명"
    background="이미지명"> ---> 행 전체요소에 이미지 삽입 (넷스케이프에서만)
    <td align="left/right/center" ---> 셀내에서의 좌/우/중앙 위치지정
      valign="top/middle/bottom" ---> 셀내에서의 상/중앙/하단 위치지정
      width="너비"                ---> 셀의 가로크기. 픽셀값과 %값이 있다.
      height="높이"              ---> 셀의 세로크기. 픽셀값과 %값이 있다.
      bgcolor="색상값/명"
      background="이미지명"
      rowspan="합칠 셀갯수"       ---> 세로 셀들을 합칠때 사용
      colspan="합칠 셀갯수"       ---> 가로 셀들을 합칠때 사용
      nowrap >                  ---> 셀내에서 자동으로 줄바꿈하지 못하게 한다.
                                셀의 width가 데이터의 크기에 따라 넓혀진다.
    </td>
  </tr>
</table>
```

예제9. 테이블 기본

Dynamic Web Project : 6-html

HTML File : /WebContent/exam9.html

<실행 결과>



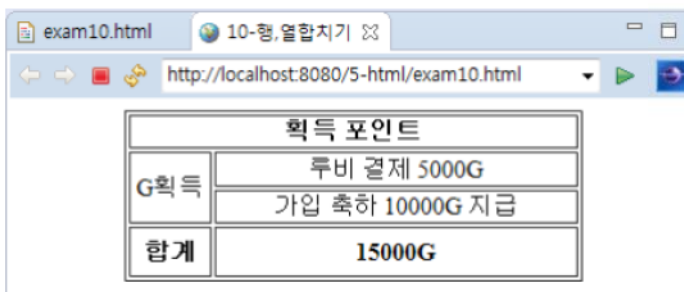
No	획득 포인트
2	루비 결제 5000G
1	가입 축하 10000G 지급
합계	15000G

예제10. 테이블 행, 열 결합

Dynamic Web Project : 6-html

HTML File : /WebContent/exam10.html

<실행 결과>



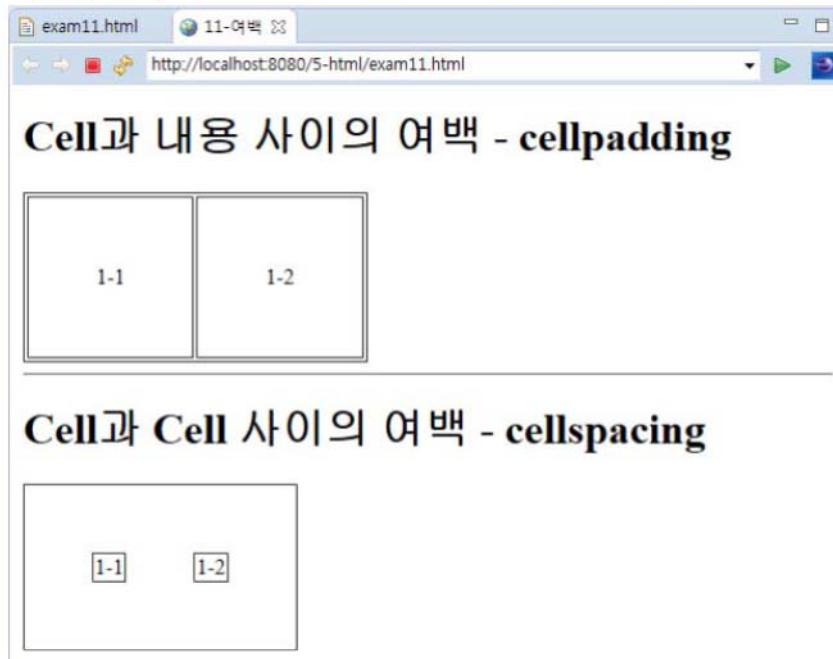
	획득 포인트
G획득	루비 결제 5000G
	가입 축하 10000G 지급
합계	15000G

예제11. 테이블 여백

Dynamic Web Project : 6-html

HTML File : /WebContent/exam11.html

<실행 결과>

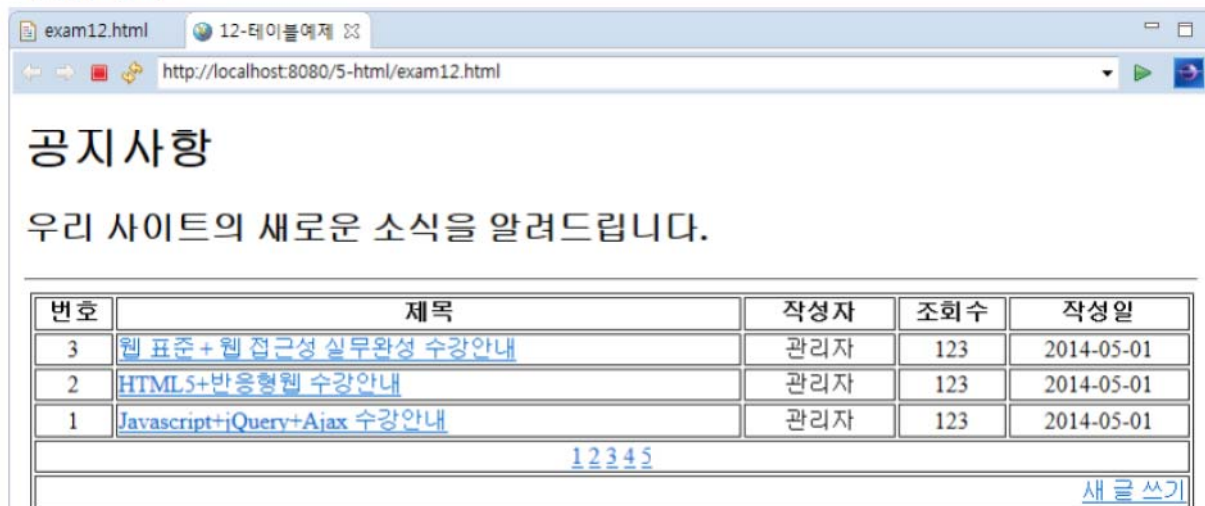


예제12. <table> 활용

Dynamic Web Project : 6-html

HTML File : /WebContent/exam12.html

<실행 결과>



11. 블록 / 인라인 레벨 요소

(1) 블록 / 인라인 레벨 요소

① 블록 레벨 요소

- 블록 레벨 요소는 자신을 감싸는 상위의 요소가 허용한 가로 폭 전체 영역을 차지한다.
- 블록 레벨 요소는 너비나 높이를 지정할 수 있고, 줄 바꿈 속성을 가지고 있다.
- address, article, aside, audio, blockquote, canvas, dd, div, dl, fieldset, figcaption, figure, footer, form, h1~h6, header, hr, noscript, ol, output, p, pre, section, table, tfoot, ul, video 등

② 인라인 레벨 요소

- 인라인 레벨 요소는 흘러가는 요소이다.
- 인라인 레벨 요소는 너비나 높이를 지정할 수 없으며, 줄이 바뀌지 않는 형식(inline)으로 표현된다.
- 줄이 바뀌지 않지만, 내용이 많아서 그 상위 블록요소의 영역을 가득 채우게 되면 줄이 바뀐다.
- a, abbr, acronym, b, bdo, big, br, button, cite, code, dfn, em, i, img, input, kbd, label, map, object, q, samp, script, select, small, span, strong, sub, sup, textarea, tt, var 등

(2) 콘텐츠 모델 별 분류

=> HTML5에서는 HTML의 요소를 비슷한 속성에 따라 묶어서 분류한다.

=> 몇몇 요소들은 여러 분류에 중복해서 포함되기도 한다.

① 메타데이터(Metadata) 콘텐츠의 종류

- base, command, link, meta, noscript, script, style, title

② 플로우(Flow) 콘텐츠의 종류

- a, abbr, address, area(map 요소의 자손인 경우), article, aside, audio, b, bdi, bdo, blockquote, br, button, canvas, cite, code, command, datalist, del, details, dfn, div, dl, em, embed, fieldset, figure, footer, form, h1~h6, header, hgroup, hr, i, iframe, img, input, ins, kbd, keygen, label, map, mark, math, menu, meter, nav, noscript, object, ol, output, p, pre, progress, q, ruby, s, samp, script, section, select, small, span, strong, style(scope 속성이 있으면), sub, sup, svg, table, textarea, time, ul, var, video, wbr, Text

③ 섹션(Section) 콘텐츠의 종류

- article, aside, nav, section

④ 제목(Heading) 콘텐츠의 종류

- h1~h6

⑤ 구문(Phrasing) 콘텐츠의 종류

- a(구문 콘텐츠만을 포함하는 경우), abbr, area(map 요소의 자손인 경우), audio, b, bdi, bdo, br, button, canvas, cite, code, command, datalist, del(구문 콘텐츠만을 포함하는 경우), dfn, em, embed, i, iframe, img, input, ins(구문 콘텐츠만을 포함하는 경우), kbd, keygen, label, map(구문 콘텐츠만을 포함하는 경우), mark, math, meter, nav, noscript, object, output, progress, q, ruby, s, samp, script, select, small, span, strong, sub, sup, svg, textarea, time, var, video, wbr, Text

⑥ 포함된(Embedded) 콘텐츠의 종류

- audio, canvas, embed, iframe, img, math, object, svg, video

⑦ 대화형(Interactive) 콘텐츠의 종류

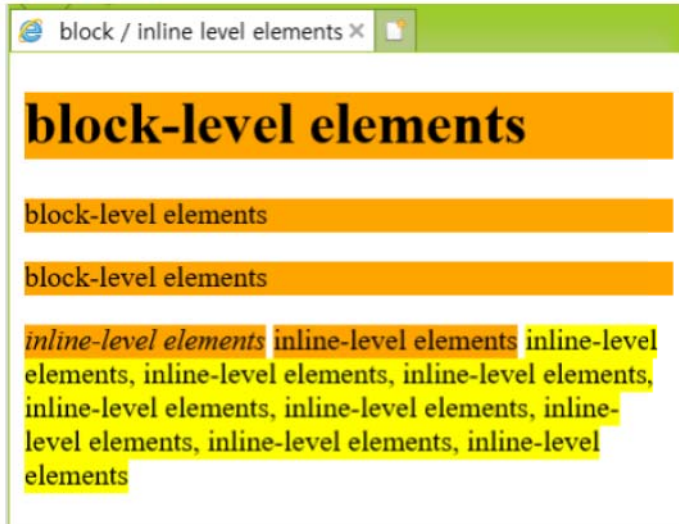
- a, audio(controls 속성이 있으면), button, details, embed, iframe, img(usemap 속성이 있으면), input(type 속성이 hidden 상태가 아니면), keygen, label, menu(type 속성이 toolbar 상태면), object(usemap 속성이 있으면), select, textarea, video(controls 속성이 있으면)

예제13. 블록 / 인라인 레벨 요소

Dynamic Web Project : 6-html

HTML File : /WebContent/exam13.html

<실행 결과>



12. CSS 스타일시트에 대한 이해

(1) CSS(Cascading Style Sheets) :

웹페이지에 자주 적용되는 부분들은 미리 정리하고 조합하여 정의한 후 페이지 내부에서 필요할 때마다 해당부분을 적용하는 스타일시트중 하나. HTML로는 부족한 레이아웃이나 폰트등에 다양성을 부여한다.

- 폰트 크기는 지정하는 수치에 따라 마음대로 조절가능
- 자간/행간 배치가 자유로워 가독성을 높인다.
- 링크상의 밑줄 변형이 자유롭다.
- 페이지의 여백을 원하는 만큼 만들어 줄 수 있다.

<만드는 방법 3가지>

- ① <head></head>사이 기술
- ② <body>안에서 직접기술
- ③ *.css(파일)로 따로 저장

(2) 기본구성

<HEAD>

<STYLE type="text/css"> ----> 스타일의 유형이 텍스트이고 그 파일은 css 이라는 뜻.
선택자 {속성1:값1 ; 속성2:값2} 선언문(속성과 값)간의 구분은 :(세미콜론)이다.

</STYLE>

</HEAD>

(3) CSS의 특성

- ① 기존의 HTML기능을 확장해서 사용할 수 있다.
- ② 한번의 지정으로 웹문서 모든 곳에 적용할 수 있다.
- ③ CSS는 상위의 기능을 상속받을 수 있다.
- ④ 복잡하지 않고 쉽게 파악할 수 있어 관리가 용이하다.

13. JavaScript

(1) 역사

네스케이프사가 사이트의 처리능력을 높이기 위해 개발한 스크립트 언어
처음엔 Live Script라고 발표되었다가 선마이크로시스템의 자바의 기능을 결합
자바스크립드라 부르게 되었다.

* 스크립트란?

컴파일되지 않고 Application이 실행되는 동안 Line 단위로 해석되는 명령어나 문장들의 집합

(2) 특징

- 브라우저 안에서만 쓰이는 언어
- Html태그로는 부족한 동적인 효과를 내준다.
- Html태그에 쓰이는 명령어를 다시 응용해서 사용하게 된다.

(3) 자바와 자바스크립트의 차이

- 자바스크립트 : 넷스케이프에서 제작. 특별한 개발환경이 없다.
Client-side-script(클라이언트인 브라우저에서 소스자체가 해석된다.)
객체기반(Based)언어. 단 클래스를 지원하지 않는다.
변수형을 따로 선언할 필요가 없다.
동적Binding, 실행시 객체 참조가 체크된다.
- 자바 : 선마이크로시스템에서 제작. Visual J++이나 Visual Cafe, Latte의 개발환경
Server-side-script(서버에서 먼저 컴파일되고 그코드를 클라이언트에서 해석한다.)
객체지향(Oriented)언어. 자바애플릿은 클래스의 상속을 지원한다.
변수형을 반드시 선언해야 한다.
정적Binding. 컴파일시 객체참조가 체크된다.

(5) 기본구조

```
<SCRIPT language="JavaScript1.3"> ----> 스크립트의 종류와 버전표기  
소스코드 (새버전을 사용시 반드시 표기)  
</SCRIPT>
```

(6) 주석문 : 프로그램 실행에는 아무런 영향을 미치지 않는다.

```
// -----> 한줄 주석  
/* */ -----> 두줄 이상의 주석 : debugging의 용도로도 쓰임
```

(8) HTML문서내에서의 위치

가급적 <HEAD>...</HEAD>사이에 위치하는 것이 좋다
그러나, 페이지내 다른 객체와 어울려 작동시 또는 HEAD에서는 작동하지 않는경우
BODY의 어떤 곳에도 사용가능

(9) 스크립트문 작성시 주의점

- 대소문자를 구분한다.
- 한 line (실행단위)을 끝내려면 :(세미콜론)으로 달아준다.
----> 요즘은 ;없이도 enter로 line구분이 가능하게 되었다
- 한줄에 2문장기술 :를 생략할수 없다

7장. Servlet

* Servlet

- 웹에서 실행하는 프로그램
- html in JAVA
- main() 메소드가 없다
- 반드시 public 이어야한다.
- html 이 필요
- 주기함수(Life Cycle)
- new X
- 서버안에 저장

* 환경 설정

- 시작 - 컴퓨터 - 속성 - 고급 시스템 설정 - 고급(탭) - 환경변수 - 새로 만들기
- JDK 설정 (JDK의 위치)
JAVA_HOME
C:\Program Files\Java\jdk1.8.0_71
- TOMCAT 설정 (Tomcat의 위치)
TOMCAT_HOME or CATALINA_HOME
C:\Program Files\Apache Software Foundation\Tomcat 8.0

1. 주기함수 (Life Cycle)

init() : 맨처음에 1번만 호출
↓
service() - doGet() : 클라이언트가 요청시마다 호출
 - doPost()
↓
destroy()

2. 서비스

(1) get방식

- default
- 주소표시줄(Query String)를 통해서 이동
- 이동되는 데이터가 보인다
- 이동되는 데이터가 문자열만(String) 처리

(2) post방식

- 클라이언트가 post 로 요청시
- 내부적으로(페이지단위) 이동
- 이동되는 데이터가 안보인다.
- 대량데이터

* 실제 서버에 사용할 때는 ~.war로 만들어서 사용한다.

- ~.war => 압축파일
- 서블릿 등록 - 어노테이션

예제1. 서블릿을 이용해서 응답하기

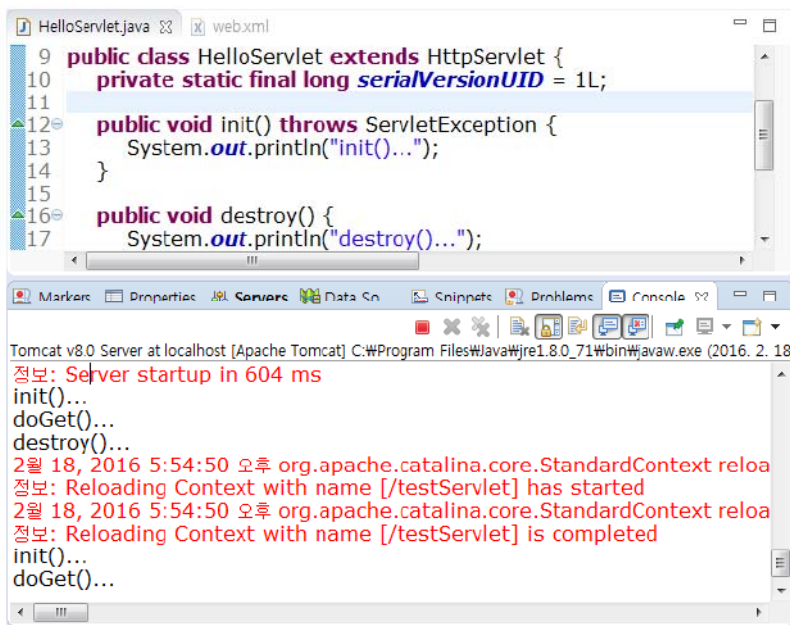
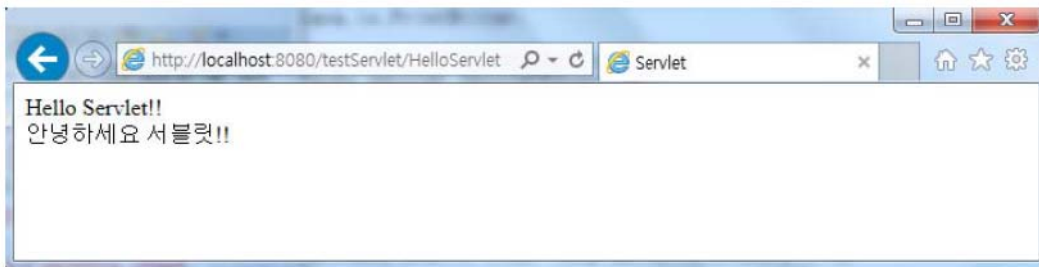
Dynamic Web Project : 7-Servlet

Package : com.hello

Servlet File : src/com.hello/HelloServlet.java

XML File : /WebContent/WEB-INF/web.xml

<실행 결과>



<작업 순서>

1. 프로젝트 만들기

- ① "Project Explorer" -> 오-마 -> New -> Project 선택
- ② "Select a wizard" 창 -> "Web/Dynamic Web Project" 선택 -> "Next" 버튼 클릭
- ③ "Dynamic Web Project" 창
-> Project name : 7-Servlet
-> "Next" 버튼 클릭
- ④ "Java" 창
-> "Next" 버튼 클릭
- ⑤ "Web Module" 창
-> "Generate web.xml deployment descriptor" 체크
-> "Finish" 버튼 클릭

2. 패키지 만들기

- ① "7-Servlet/Java Resources/src" -> 오-마 -> New -> Package 선택
- ② "Java Package" 창
 - > Name : com.hello
 - > "Finish" 버튼 클릭

3. 서블릿 파일 만들기

- ① "7-Servlet/Java Resources/src/com.hello" -> 오-마 -> New -> Servlet 선택
- ② "Create Servlet" 창
 - > Class name : HelloServlet
 - > "Next" 버튼 클릭
 - > "Next" 버튼 클릭
 - > "init, destroy, doPost, doGet" 체크
 - > "Finish" 버튼 클릭
- ③ HelloServlet.java 파일 정리하기
 - > @WebServlet("/HelloServlet") 지우기
 - > 주석 지우기
 - > init() 메서드의 매개변수 지우기
- ④ web.xml 파일에 서블릿을 등록한다.
 - 7-Servlet/WebContent/WEB-INF/web.xml
 - 아래의 파일 내용을 참조하여 등록한다.
- ⑤ HelloServlet.java에 동작시킬 내용을 코딩한다.
 - 아래의 "파일 작업"을 참조한다.

4. 실행하기

- ① "Servers" View
 - > "Tomcat v8.0 Server at localhost" 선택
 - > 오-마 -> "Add and Remove..." 클릭
 - > "Add and Remove" 창 -> "6-Servlet" 선택 -> "Add" 클릭
 - > "Finish" 버튼 클릭
 - > 실행 버튼을 눌러서 서버 실행 시킨다.
- ② 브라우저를 실행한후, URL 주소 추가후 결과 확인한다.
 - URL
 - webapps
 - http://localhost:8080/7-Servlet/HelloServlet
 - => /HelloServlet = com.hello.HelloServlet (web.xml에 등록되어 있음)
 - http://localhost:8080/7-Servlet/com.hello.HelloServlet
- ③ HelloServlet.java 파일을 수정하면, destroy()가 호출되고, 다시 처음부터 동작된다.

예제2. 서블릿으로 성인 인증 처리하기

- 서블릿 등록 - web.xml

Dynamic Web Project : 7-Servlet

XML File : /WebContent/WEB-INF/web.xml

[클라이언트]

Folder : param

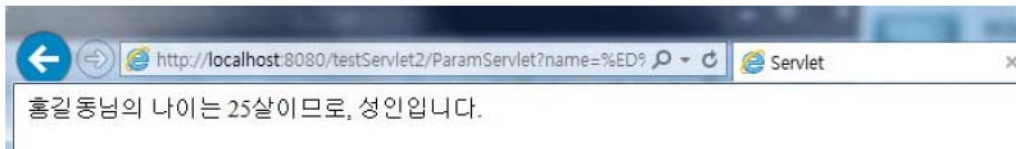
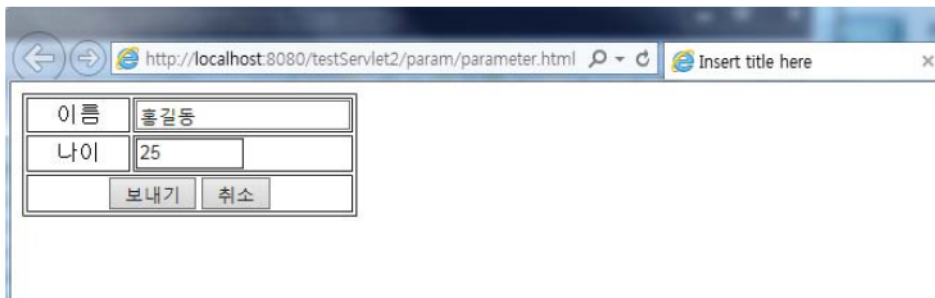
HTML File : WebContent/param/parameter.html

[서버]

Package : com.param

Servlet File : src/com.param/ParamServlet.java

<실행 결과>



<작업 순서>

1. 클라이언트 화면을 만든다.
2. 서버를 서블릿으로 처리하고, 응답을 한다.
 - => 기본 설정으로 파일을 만든다.
 - => web.xml에 서블릿을 등록한다.

URL

http://localhost:8080/7-Servlet/param/parameter.html 요청

↓ 페이지 이동

http://localhost:8080/7-Servlet/ParamServlet?name=hong&age=25

예제3. 서블릿으로 회원등록 처리하기

- 서블릿 등록을 web.xml을 사용하지 않고 어노테이션으로 하기.

Dynamic Web Project : 7-Servlet

[클라이언트]

Folder : person

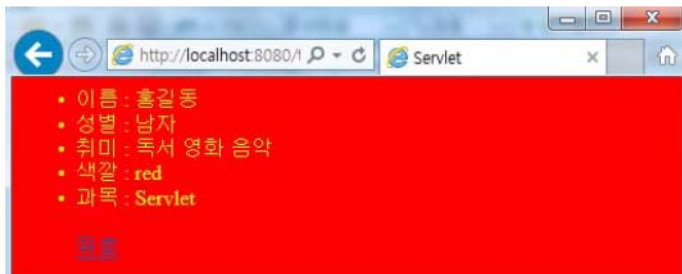
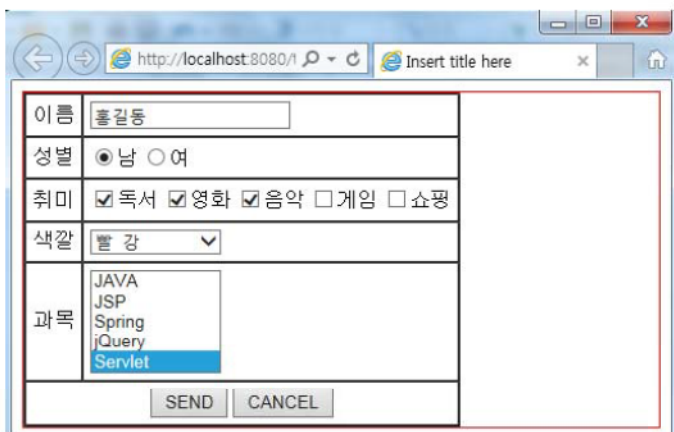
HTML File : WebContent/person/person.html

[서버]

Package : com.person

Servlet File : src/com.person/PersonServlet.java

<실행 결과>



<작업 순서>

1. 클라이언트 화면을 만든다.
2. 서버를 서블릿으로 처리하고, 응답을 한다.

URL

http://localhost:8080/7-Servlet/person/person.html

요청

http://localhost:8080/7-Servlet/PersonServlet
/6-Servlet/PersonServlet

http://localhost:8080/7-Servlet/PersonServlet?name=hong&gender=0&color=red&subject=Servlet

http://localhost:8080/7-Servlet/PersonServlet?name=%ED%99%8D%EA%B8%B8%EB%8F%99&gender=0&color=red&subject=Servlet