

- 프레임워크(framework)란?

: 빠른 개발을 위해서 자주 사용되는 기능을 api로 만들어 놓은 것.

살을 붙여서 구체적인 개발 방향을 잡음

ex. SPRING: mvc 패턴으로 개발하도록 한 웹 framework

request.getParameter 없이 데이터 송수신이 간편

ex2. android: 모바일 앱 개발을 위한 앱 framework (linux os 위에 올라가있는 형태)

## 1. 웹 어플리케이션의 구조

: http 프로토콜을 사용한 통신

- a. 클라이언트에서 웹 app으로 요청(get 방식 or post 방식)

두 방식 모두 서버에 요청을 하는 메서드

- get방식: 클라이언트의 데이터를 URL 뒤에 붙여서 보냄

**www.example.com?id=mommoo&pass=1234**

- post방식: URL에 직접 붙이지 않고 BODY에 데이터를 넣어서 보냄

- b. 웹 app에서 클라이언트로 응답 (클라이언트에서 볼 수 있는\* 형태로 요청한 페이지 제공)

- c. 클라이언트단 언어와 서버단 언어?

- 클라이언트단 언어(view단 개발용)

: 웹 컨테이너 없이 클라이언트가 확인할 수 있게 하는 언어

- html(view page의 정적인 text, image, link 단순 조합)
- css(html에서 구현해놓은 단순 조합을 좀 더 세련되게 조합)
- javascript(콜백 함수 처리, if/for문등을 이용해 로직 구축, 클래스 이용, view page내의 동적인 이동을 가능케 함)  
하지만, back 단과 연동 불가능(db 연동, 클라이언트와 커뮤니케이션..)

- 서버단 언어

: 클라이언트에서 바로 볼 수 없는 서버단 언어

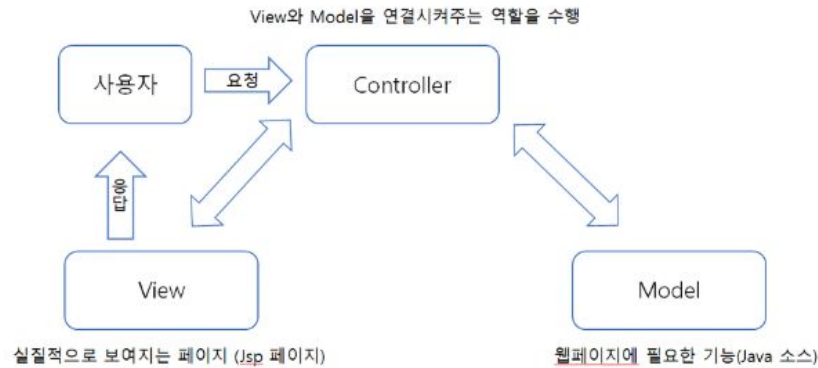
- servlet(controller 역할): 일반 자바지만, http servlet 클래스를 상속받아서 http 프로토콜로 네트워킹을 가능케 함. 웹 기능에 대한 동작을 알고 있는 자바 언어인 셈.  
\*http servlet: (request reponse 기능이 가능한 클래스)연결지향의 tcp 프로토콜과 달리 요청에 따른 응답을 보내주고 연결을 끊음
- jsp(view단 개발 역할-중간다리 역할): 클라이언트 단 언어는 자바 객체 인식 불가. jsp는 자바에서 만든 멤버 객체를 이해하여 service에서 던져준 코드를 이해하여 html로 view단에 뿌려줄 수 있는 언어.  
\*클라이언트단에서 볼 수 있게 하려면? 웹 컨테이너를 사용해 번역.  
: 해당 언어들을 컴파일하여 클라이언트단 언어로 변환해주는 웹 컨테이너 필요. ex. 톰캣

- d. MVC 패턴?

소프트웨어 디자인 패턴 중 하나

사용자 인터페이스와 비즈니스 로직을 분리하여 서로 영향 없이 쉽게 고칠 수 있는 어플리케이션을 만들 수 있다.

- Model(백그라운드 로직을 처리)
- View(사용자가 보게 될 결과 화면을 출력)
- Controller(사용자의 입력처리와 흐름 제어 담당)



- 웹 브라우저 사용자가 요청을 보냄
- 서버릿이 해당 요청을 받아 즉시 view로 보여줄 것인지 model로 보내줄 것인지 정한 뒤 전송
- view: html로 구성된 웹 페이지를 출력
- model: java로 구성된 실질적인 기능 담당

#### e. 웹 컨테이너의 역할

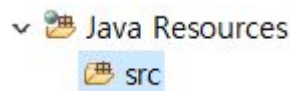
- java 언어: 인터프리터 방식과 컴파일 방식을 모두 가짐
  - jsp(java server pages): html 내에 자바 코드를 삽입하여 웹 서버에서 동적으로 웹 브라우저를 관리하는 언어
- ex. 다음: 몇 만명이 동시접속  
jsp나 서버릿 페이지를 자바로 컴파일하여 메모리 풀에 얹어 놓는다.  
서버에서

## 2. 웹 프로젝트 내 구조

### a. java resources => src 패키지

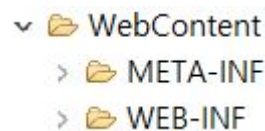
: 확장자가 자바인 파일들이 존재하는 패키지.

ex. model(dto,dao,service), servlet 파일들..

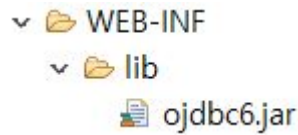


### b. webContent: view단 파일들이 존재하는 디렉토리

ex. html, css, jsp 파일들...



- web-inf
  - : 웹 어플리케이션을 실행하는데 필요한 클래스 파일 및 설정 파일 등이 저장되는 디렉터리
  - \*web.xml: 웹 컨테이너에게 현재 웹 어플리케이션에 대한 설명을 해주는 파일
  - \*메이븐 연동 안한 경우에는 web-inf 디렉토리에 넣으면 not found 404
  - \*메이븐: 버전 관리 tool(버전 맞지 않으면 충돌 多)
  - \*web-inf => lib => 드라이버 저장
  - \*ojdbc6.jar: 데이터 베이스와 연동을 위한 드라이버



#### WEB-INF/lib

: 웹 어플리케이션에서 사용하는 jar 파일이 있는 폴더

#### WEB-INF/classes

: 웹 어플리케이션에서 사용하는 클래스 파일이 위치.

상업적 목적이라면 **com/회사명/분류/클래스** 에 위치시키고,  
그냥 일반 클래스 파일이라면 **org/패키지네임/클래스** 에 위치시키자.

#### WEB-INF/tld

: 태그 라이브러리 관련 설정 파일은 보통 tld 폴더에 저장.

#### WEB-INF/src

: 일반적인 소스파일이 위치한다.

#### WEB-INF/web.xml

: 웹 어플리케이션에 대한 전반적인 설정 정보를 저장하고 있는 파일.

### 3. 웹 프로젝트 내 경로지정 (중요): URL로 접근

- jsp 파일(루트/상위디렉토리/해당파일명)

localhost:8989/web0722/join/a.jsp

ip 주소: port 번호/root 프로젝트명/폴더명/파일명

- servlet 파일(루트/해당파일명) \*변경가능  
annotation: web.xml을 생략하고 대신 정보를 annotation 형태로 넣어줌

```
@import java.io.IOException;
```

```
/**  
 * Servlet implementation class Test  
 */
```

```
@WebServlet("/Test")//이 웹페이지에 접근할 수 있는 URL =>web0722/Test
```

- ../=>한 단계 위  
../../=>두 단계 위

### 4. html 파일 구성

- <!DOCTYPE html>: 문서 종류(html 문서다.)
- 태그(명령어): < 여는 태그 >, </ 닫는 태그 >
  - \*반드시 쌍으로 된 것은 아님
  - \*각 태그 안에 각각 상이한 attribute(속성)이 존재. 갈색
  - \*따옴표 유무 상관 없음. 따옴표 종류 상관 없음
  - \*\*하지만 ajax로 dom(document object model)사용하는 경우 서로 다른 따옴표 사용하여 묶어줌.

ex. '../img'/img1.jpg'

- 주석: <!-- text -->
- <head>와 <body>로 구성  
head: 제목, body: 실제 웹 페이지의 내용
- <meta charset= 인코딩 정보> \*utf-8, post 방식이 가장 안깨짐
- \n : <br> or <br/>
- 글자 크기 조정 + \n : <h1> text </h1>  
h2,h3,h4,h5...
- 이미지 띄우기: <img src=경로 width=n1 height=n2 (단위는 픽셀)>

```
<a href="2.html">2페이지로 이동</a><br/>
<a href="/web0722/a.jsp">a페이지로 이동(절대경로)</a><br/>
<a href="../a.jsp">a페이지로 이동(상대경로:현재파일을 기준으로)</a><br/>

```

- 링크 걸기: <a href=link> 링크명 or 이미지 </a>

```
<a href="https://www.daum.net">다음으로 이동</a><br/>
<a href="https://www.naver.com">네이버로 이동</a><br/>
```

```
<a href="https://www.daum.net"> </a>
<a href="https://www.naver.com"> </a>
```

- 테이블 생성

테이블: <table> \*border = 테두리 두께

행: <tr>

열: <td>, <th>\*글자 진하게

```
<table border="1">
<tr>
<td>a</td><td>b</td><td>c</td>
</tr>
</table>
```

a	b	c
---	---	---

가로칸 합치기: <td colspan="num">

세로칸 합치기: <td rowspan="num">

```
<table border="1">
<tr>
<td colspan="2">a</td><td>c</td>
</tr>
<tr>
<td rowspan="2">a2</td><td>b2</td><td>c2</td>
</tr>
<tr>
<td>b3</td><td>c3</td>
</tr>
</table>
```

a		c
a2	b2	c2
	b3	c3

\*연습: naver 기사 format 만들기

```

<h1>네이버 format</h1>
<table>
<tr>
<td colspan="2" rowspan="2"></td>
<td></td><td>텍스트1</td>
</tr>
<tr>
<td></td><td>텍스트2</td>
</tr>
<tr>
<td colspan="2">텍스트0</td><td></td><td>텍스트3</td>
</tr>
</table>

```

## 네이버 format



 텍스트1

 텍스트2

텍스트0

 텍스트3

- 다른 프레임 추가: iframe  
ex. 약관내용 등... 너무 길 때  
<iframe src=파일경로 width=n1 height=n2></iframe>
- 단락 구분: 이후에 출력으로 쓸 영역을 미리 잡아두기.. ex. 아이디 유무..  
p 아래 위 한줄씩 벌려줌  
div\*\*: 사이에 엔터를 넣어줌  
span: 영역을 만들어줌

```

<body>
<p>aaa</p>aaa-1<br/>
<div>bbb</div>bbb-1<br/>
<span>ccc</span>ccc-1<br/>
</body>

```

```

aaa
aaa-1
bbb
bbb-1
cccccc-1

```

### 5. html form 양식(중요): 기능 구현하기 위해 클라이언트로 부터 데이터를 받는 방식



```

<form action="../a.jsp" method = "post">
id:<input type="text" name="id"><br/>
pwd:<input type="text" name="pwd"><br/>
취미:<input type="checkbox" name="hobby" value="1"> 영화
<input type="checkbox" name="hobby" value="2"> 여행
<input type="checkbox" name="hobby" value="3"> 축구
<input type="checkbox" name="hobby" value="4"> 농구 <br/>
성별:<input type="radio" name="gender" value="f"> 여
<input type="radio" name="gender" value="m"> 남 <br/>
학년:<select name="grade">
<option value="g1">1학년</option>
<option value="g2">2학년</option>
<option value="g3">3학년</option>
<option value="g4">4학년</option>
</select><br/>
가입인사:<textarea rows=5 cols=20 name = "msg"></textarea><br/>
<input type="button" value="일반버튼">
<input type="submit" value="전송버튼">
<input type="reset" value="초기화버튼">
<input type="hidden" name="type" value= "join">
</form>

```

- <form action="서버페이지" method ="전송방식"></form>
  - 서버페이지가 받아서 처리하겠다.
  - 전송방식(get, post)
    - get:URL뒤에 붙어서 가기 때문에 데이터 길이에 제한이 있고 보안상 안 좋음
    - post:패킷 헤더에 담아서 보내기 때문에 데이터 길이에 제한이 없고 보안상 좋음. 한글 덜 깨짐
- 태그 형태
  - ex. form(전송방식), input(), select(단일 선택), textarea(장문 입력)
- type: 해당 태그 내에 입력 형식
  - ex. text(단문 입력), checkbox(다중 선택), radio(단일 선택), button(버튼), submit(전송-전체 폼 데이터가 action에 지정한 서버 페이지로 전송됨), reset(초기화), hidden(클라이언트 화면에는 안보이지만 서버(서블릿)에 값을 보내줘야 하는 경우)
- name: 서버페이지에서 받아갈 때 어떤 값인지 인식하기 위해서
- value: 실제 서버에 전달되는 값

<출력 결과>

id:

pwd:

취미: ☐ 영화 ☐ 여행 ☐ 축구 ☐ 농구

성별: ☐ 여 ☐ 남

학년:

가입인사:

일반버튼    전송버튼    초기화버튼

## 6. jsp

: html로부터 받아온 폼데이터들을 수신(원래 해당 역할은 서블릿이 수행함.)

- 현재 페이지 기본 설명:

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
```

- 자바 코드 삽입: <% %>(스크립트릿)

- 출력: 1)%=, 2)out.print()

\*out이라는 이름의 PrintWriter 객체를 사용하여 응답으로 보낼 웹페이지에 일정 내용을 추가

(servlet 참고)

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    PrintWriter pw = response.getWriter();
    //응답으로 보내줄 html 페이지 정보
    pw.append("<html><body><h1>hello servlet!!</h1></body></html>");
}
```

- request와 response

- request + 폼데이터: 폼데이터와 함께 일정 기능을 요청
- response + html: 출력할 웹페이지 전송

## 7. CSS

- 특징

- style: 속성이름
- 콜론: 값 할당
- 세미콜론: 복수의 속성 나열
- pt: 포인트
- px: 픽셀
- color: 글자 색
- background: 배경

- 예시: p 태그에 css 문법 적용

```
<body>
<p>aaa</p>
<p style="font-size:30pt; color:blue;background:yellow">bbb</p>
<p>ccc</p>
```

aaa

bbb

ccc

- 예시2: 페이지 전체의 p태그에 css 문법 적용 \*재정의한 p 태그는 새로 정의한 속성으로 출력

```

<meta charset="EUC-KR">
<title>Insert title here</title>
<style type="text/css">
p{
    font-size:30pt;
    color:white;
    font-family:arial;
    background-color:orange
}
</style>
</head>
<body>
<p>aaa</p>
<p style="font-size:30pt; color:black;background:yellow">bbb</p>
<p>ccc</p>
</body>

```

aaa

bbb

ccc

- 예시3: class - 일정 그룹에 적용할 때, id - 단일 태그에 적용할 때

```

/* a 클래스 그룹 */
.a{
    font-size:30pt;
    color:white;
    font-family:arial;
    background-color:deeppink
}
/* div 중 클래스 a인 그룹 */
div.a{
    font-size:30pt;
    color:coral;
    font-family:arial;
    background-color:white
}
/* #id 이름 */
#myp{
    color:black;
    font-family:arial;
    background-color: lawngreen;
    width:100px;
    height:50px;
    border:2px dotted lightyellow; /*테두리 두께, 테두리 모양, 테두리 색깔*/
}

```

가나다

bbb

ccc

ddd

fff

gggg

div a

div b

div c

```

</style>
</head>
<body>
<p>가나다</p>
<p style="font-size:30pt; color:black;background:magenta">bbb</p>
<p class="a">ccc</p>
<p id="myp">ddd</p>
<p>fff</p>
<p class="a">gggg</p>
<div class="a">div a</div>
<div class="b">div b</div>
<div class="a">div c</div>
<div class="b">div d</div>
</body>

```

- 예시4: 배경이미지 지정



default: 바둑판 모양으로 반복, repeat-x: x축으로만 반복, repeat-y: y축으로만 반복,  
no-repeat: 반복 없음

```
<body>
<style>
  body{
    background-image: url('../img/universe.png');
    background-repeat:no-repeat
  }
</style>

</body>
```

- 예시5: 위치 제어 position

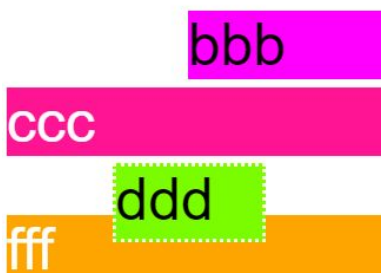
absolute(위치 고정)

relative(바로 앞 요소의 왼쪽 상단 꼭지점을 기준으로 이동)

```
#myp{
  color:black;
  font-family:arial;
  background-color: lawngreen;
  width:100px;
  height:50px;
  border:2px dotted lightyellow; /*테두리 두께, 테두리
  position:absolute; /*고정위치: 왼쪽상단 꼭지점에 고정*/
  top:230px; /*아래로 230px*/
  left:80px; /*왼쪽으로 80px*/
}

<p style="font-size:30pt; color:black;background:magenta;
position:relative;top:34px;left:123px">bbb</p>
```

가나다



- css 파일로 따로 저장 가능.

## 8. javascript

: 동적 텍스트/이미지 사용, 이벤트 핸들링

- 특징

- 세미콜론 생략 가능
- <script> text~~ </script> 태그 안에 작성
- document: 자바스크립트 내장 객체. 웹페이지에 출력 담당
- var: 변수 선언. 하지만 변수 선언없이도 사용 가능하다.

- prompt: 입력박스 창
- alert: 알림 창

```
<script type="text/javascript">
//변수 선언 후 사용
var a=10;
document.write("a="+a+"<br/>");
//변수 선언 없이 사용
b="hellow msg";
document.write("b="+b+"<br/>");

msg = prompt("계산하고 싶은 수식을 입력하십시오");//입력박스 창
res = eval(msg);//문자열 수식 숫자로 바뀌어서 자동 계산
document.write(msg+"="+res+"<br/>");
alert("계산 완료")
```

- for문 if문

```
document.write("<table border='1'><tr>")
for(i=0;i<10;i++){
    document.write("<td>msg"+i+"</td>")
}
document.write("</tr></table>")
```

5 + / = 12

msg0	msg1	msg2	msg3	msg4	msg5	msg6	msg7	msg8	msg9
------	------	------	------	------	------	------	------	------	------

- java script 연습문제: 구구단 => 한 칸 다 채우고 다음 칸

```
<body>
<h3>구구단</h3>
<script>
document.write("<table border='1'><tr>")
for(i=2;i<10;i++){
    document.write("<td>")
    for(j=1;j<10;j++){
        document.write(i+"*"+j+"="+i*j+"<br/>")
    }
    document.write("</td>")
}
document.write("</tr></table>")
</script>

</body>
```

### 구구단

2*1=2	3*1=3	4*1=4	5*1=5	6*1=6	7*1=7	8*1=8	9*1=9
2*2=4	3*2=6	4*2=8	5*2=10	6*2=12	7*2=14	8*2=16	9*2=18
2*3=6	3*3=9	4*3=12	5*3=15	6*3=18	7*3=21	8*3=24	9*3=27
2*4=8	3*4=12	4*4=16	5*4=20	6*4=24	7*4=28	8*4=32	9*4=36
2*5=10	3*5=15	4*5=20	5*5=25	6*5=30	7*5=35	8*5=40	9*5=45
2*6=12	3*6=18	4*6=24	5*6=30	6*6=36	7*6=42	8*6=48	9*6=54
2*7=14	3*7=21	4*7=28	5*7=35	6*7=42	7*7=49	8*7=56	9*7=63
2*8=16	3*8=24	4*8=32	5*8=40	6*8=48	7*8=56	8*8=64	9*8=72
2*9=18	3*9=27	4*9=36	5*9=45	6*9=54	7*9=63	8*9=72	9*9=81

출력창			
AC	%	/	
7	8	9	*
4	5	6	-
1	2	3	+
0	.	=	

출력창			
AC		%	/
7	8	9	*
4	5	6	-
1	2	3	+
0		.	=