

# 내부객체, 제어문

# 내부객체의 개요

- ❖ 내부 객체는 JSP 페이지 내에서 특정한 변수로 호출하고자 하는 변수와 메서드로 접근을 한다.
- ❖ JSP 페이지에서 사용하게 되는 특정한 변수가 아무런 선언과 객체 생성 없이 사용 할 수 있는 이유는
  - JSP 페이지가 서블릿으로 변환이 될 때 JSP 컨테이너가 자동적으로 제공을 하기 때문이다.
- ❖ 9개의 내부 객체의 사용되는 범주에 따라 4가지 형태로 분류.
  - JSP 페이지 입출력 관련 기본 객체
  - JSP 페이지 외부 환경 정보 제공 기본 객체
  - JSP 페이지 서블릿 관련 기본 객체
  - JSP 페이지 예외 관련 기본 객체
- ❖ 내부 객체의 종류
  - request, response, out, pageContext, session, application, config, page, exception

## ❖ 내부 객체가 제공하는 메소드

- request, session, application, pageContext 내부 객체는 임의 속성(attribute) 값을 저장하고 읽을 수 있는 메서드를 제공

메서드	리턴 타입	설명
setAttribute (String key, Object value)	void	주어진 key속성의 값을 value로 지정한다.
getAttributeNames()	java.util.Enumeration	모든 속성의 이름을 구한다.
getAttribute (String key)	Object	주어진 key속성의 값을 얻어낸다.
removeAttribute (String key)	void	주어진 key속성의 값을 제거한다.

<%@ page contentType="text/html; charset=euc-kr"%>

<%

String info = application.getServerInfo();

int major = application.getMajorVersion();

int minor = application.getMinorVersion();

String path = application.getRealPath("/");

application.log("로그 기록");

%>

<h2>Application내장객체 예제</h2>

웹 컨테이너의 이름과 버전 : <%=info%><p>

서블릿의 버전 : <%=major%>.<%=minor%><p>

웹 어플리케이션 폴더의 로컬 시스템 경로 : <%=path%>

## ❖ request 내부객체

- request 객체는 웹 브라우저에서 JSP 페이지로 전달되는 정보의 모임으로 HTTP 헤더와 HTTP 바디로 구성
- 웹 컨테이너는 요청된 HTTP 메시지를 통해 HttpServletRequest 객체 타입으로 사용되고 request 객체 명으로 사용

메서드	설명
String getParameter(name)	이름이 name인 파라미터에 할당된 값을 리턴 하며 지정된 파라미터 값이 없으면 null값을 리턴 한다.
String[] getParameterValues(name)	이름이 name인 파라미터의 모든 값을 String 배열로 리턴 한다. checkbox에서 주로 사용된다.
Enumeration getParameterNames()	요청에 사용된 모든 파라미터 이름을 java.util.Enumeration 타입으로 리턴 한다.

# Request.jsp

```
<%@ page contentType="text/html; charset=euc-kr"%>
<html>
<body>
<h2>Request내장객체 예제1</h2>
<FORM METHOD=POST ACTION="requestTest1.jsp"><!--active:한글, inactive:영문 --%>
이름 : <INPUT TYPE="text" NAME="name" style="ime-mode:active;"><br>
나이 : <INPUT TYPE="text" NAME="age" style="ime-mode:inactive;"><br>
성별 : 남자 <INPUT TYPE="radio" NAME="gender" VALUE="m" checked>
      여자 <INPUT TYPE="radio" NAME="gender" VALUE="f"><br>
취미 : <SELECT NAME="hobby" >
      <OPTION SELECTED VALUE="잠자기">잠자기</OPTION>
      <OPTION VALUE="TV시청">TV시청</OPTION>
      <OPTION VALUE="만화보기">만화보기</OPTION>
      <OPTION VALUE="게임">게임</OPTION>
      <OPTION VALUE="등산">등산</OPTION>
    </SELECT><br>
<INPUT TYPE="submit" value="보내기">
</FORM>
</body>
</html>
```

# RequestTest1.jsp

```
<%@ page contentType="text/html; charset=euc-kr"%>
<%//파라미터에대한 한글처리
    request.setCharacterEncoding("euc-kr");
%>
<h2>Request내장객체 예제1</h2>
<% //request객체에서 파라미터값을 얻어냄
    String name = request.getParameter("name");
    String age = request.getParameter("age");
    String gender = request.getParameter("gender");
    String hobby = request.getParameter("hobby");
    //성별값 처리
    if(gender.equals("m")){
        gender = "남자";
    }else{
        gender = "여자";
    }
%>
<%-- DB연동을 위한 작업--%>
<%-- 화면 출력--%>
<%=name%> 님의 정보는 다음과 같습니다.<p>
나이 : <%=age%><p>
성별 : <%=gender%><p>
취미 : <%=hobby%>
```

- **request 객체는 또한 웹 브라우저와 웹 서버의 정보도 가져 올 수  
가 있다**
- **request 내부 객체의 클라이언트 정보 메서드**

메서드	설명
String getMethod()	요청에 사용된 요청 방식(GET, POST, PUT)을 리턴 한다.
String getRequestURI()	요청에 사용된 URL로부터 URI을 리턴 한다.
String getQueryString()	요청에 사용된 Query 문장을 리턴 한다.
String getRemoteHost()	클라이언트의 호스트 이름을 리턴 한다.
String getRemoteAddr()	클라이언트의 주소를 리턴 한다.
String getProtocol()	사용 중인 프로토콜을 리턴 한다.
String getServerName()	서버의 도메인 이름을 리턴 한다.
int getServerPort()	서버의 port번호를 리턴 한다.
String getHeader(name)	HTTP 요청 헤더에 지정된 name의 값을 리턴 한다.
String getContextPath()	해당 JSP페이지가 속한 웹 어플리케이션의 컨텍스트 경로를 리턴 한다.



# RequestTest2.jsp

```
<%@ page contentType="text/html; charset=euc-kr"%>
<%
    String protocol = request.getProtocol();
    String server = request.getServerName();
    int port = request.getServerPort();
    String clientIp = request.getRemoteAddr();
    String clientHost = request.getRemoteHost();
    String methodType = request.getMethod();
    String url = new String(request.getRequestURL());
    String uri = request.getRequestURI();
    String contextPath = request.getContextPath();
    String browser = request.getHeader("User-Agent");
    String mediaType = request.getHeader("Accept");
%>
```

<h2>Request내장 객체 예제2</h2>

프로토콜명: <%=protocol%><p>

접속한 서버명 : <%=server%><p>

접속한 서버의 포트 번호 : <%=port%><p>

클라이언트의 IP : <%=clientIp%><p>

클라이언트의 호스트명 : <%=clientHost%><p>

현재 페이지의 method방식 : <%=methodType%><p>

요청한 현재 페이지의 경로(URL) : <%=url%><p>

요청한 현재 페이지의 경로(URI) : <%=uri%><p>

웹어플리케이션에서의 컨텍스트 경로 : <%=contextPath%><p>

사용한 웹 브라우저 : <%=browser%><p>

웹 브라우저가 지원하는 매체(media)의 타입 : <%=mediaType%><p>

## ❖ response 내부객체

- response 객체는 웹 브라우저로 응답할 응답 정보를 가지고 있음.
- 웹 컨테이너는 요청된 HTTP 메시지를 통해 HttpServletResponse 객체 타입으로 사용되고 response 객체명으로 사용.
- response 객체는 응답정보와 관련하여 주로 헤더 정보 입력, 리다이렉트하기등의 기능을 제공.

메서드	설명
void setHeader(name, value)	응답에 포함될 Header를 설정한다.
void setContentType(type)	출력되는 페이지의 contentType 설정한다.
void sendRedirect(url)	지정된 URL로 요청을 재전송한다

## ResponseTest1.jsp

```
<%@ page contentType="text/html; charset=euc-kr"%>
<h2>Response내장객체 예제</h2>
  현재 페이지는 responseTest1.jsp 페이지입니다.
```

```
<%
  response.sendRedirect("responseTest1_1.jsp");
%>
```

## ResponseTest\_1.jsp

```
<%@ page contentType="text/html; charset=euc-kr"%>
<html>
  <body>
    <h2>리다이렉트된 페이지</h2>
    지금 보시는 페이지는 responseTest1_1.jsp입니다.
  </body>
</html>
```

## ❖ pageContext 내부객체

- pageContext 객체는 현재 JSP 페이지의 컨텍스트를 나타내며 pageContext 내부 객체를 통해서 다른 내부 객체에 접근을 할 수 있음.
- pageContext 객체는 javax.servlet.jsp.PageContext 클래스 타입으로 제공 되는 JSP 기본 객체.

메서드	설명
ServletRequest getRequest()	페이지 요청 정보를 가지고 있는 request 기본 객체를 리턴 한다.
ServletResponse getResopnse()	페이지 요청에 대한 응답 정보를 가지고 있는 response 기본 객체를 리턴 한다.
JspWriter getOut()	페이지 요청에 대한 응답 출력 스트림인 out 기본 객체를 리턴 한다.
HttpSession getSession()	요청한 클라이언트의 세션 정보를 담고 있는 session 기본 객체를 리턴 한다.
ServletContext getServletContext()	페이지에 대한 서블릿 실행 환경 정보를 담고 있는 application 기본 객체를 리턴 한다.
Object getPage()	page 기본 객체를 리턴 한다.
ServletConfig getServletConfig()	페이지의 서블릿 초기 정보 설정 정보를 담고 있는 config 기본 객체를 리턴 한다.
Exception getException()	페이지 실행 중에 발생하는 에러 페이지에 대한 예외정보를 갖고 있는 exception 기본 객체를 리턴 한다.

pageTest1.jsp

```
<%@ page contentType="text/html; charset=euc-kr"%>
<%@ page info = "Page 내장 객체 예제"%>
<%
    String info = this.getServletInfo();
%>
<h2>Page내장객체 예제</h2>
page디렉티브의 info속성값 " <%=info%> " 을 얻어낸다.
```

# 내부객체의 영역(scope)

- ❖ 웹 어플리케이션은 page, request, session, application 이라는 4개의 영역을 가지고 있음.
- ❖ 기본객체의 영역은 객체의 유효기간이라고도 불리며, 객체를 누구와 공유할 것인가를 나타냄.
  - page영역
    - 한번의 웹 브라우저(클라이언트)의 요청에 대해 하나의 JSP페이지가 호출.
    - 웹 브라우저의 요청이 들어오면 이때 단 한 개의 페이지만 대응.
  - request영역
    - 한번의 웹 브라우저(클라이언트)의 요청에 대해 같은 요청을 공유하는 페이지가 대응.
    - 웹 브라우저의 한번의 요청에 단지 한 개의 페이지만 요청될 수 있고, 같은 request 영역이면 두개의 페이지가 같은 요청을 공유 . include 액션 태그, forward 액션 태그를 사용시
  - session영역
    - 하나의 웹 브라우저당 1개의 session객체가 생성.
    - 같은 웹 브라우저 내에서는 요청되는 페이지 들은 같은 객체를 공유.
  - application영역
    - 하나의 웹 어플리케이션당 1개의 application 객체가 생성. 같은 웹 어플리케이션에 요청되는 페이지들은 같은 객체를 공유.

# JSP페이지의 연산자

❖ 자바의 기본적인 문법을 그대로 사용

❖ 식별자(identifier)규칙

- JSP는 자바 식별자 규칙을 따르는데, 식별자(identifier)란 클래스명, 메소드명, 멤버변수명, 자동변수명등을 일컫는다.
- 자바 식별자는 길이에겐 제한이 없고 첫 글자는 반드시 영문자,\_,\$,로 시작해야 한다. 자바의 클래스명, 메소드명, 인스턴스변수, 자동변수등에 자바 식별자 규칙이 적용
- 자바는 대소문자를 구별하므로 주의해야한다.

❖ 기본데이터타입(primitive data type)

- 자바 및 JSP는 byte, short, int, long, float, double, char, boolean등의 기본 데이터 타입(primitive data type)을 제공한다.

타입	크기(byte)	자료범위	기본값
byte	1byte	-128 ~ +127	0
short	2byte	-32,768 ~ +32,767	0
int	4byte	-2,147,243,648 ~ +2,147,243,647	0
long	8byte	-9,223,372,036,854,775,808 ~ +9,223,372,036,854,775,807	0
float	4byte	-3.40292347E+38 ~ +3.40292347E+38	0
double	8byte	-1.79769313486231570E+308 ~ +1.79769313486231570E+308	0
char	2byte	'\u0000' ~ '\uFFFF'	0
boolean	1bit	true or false	false

## ❖ 연산자

- JSP에서는 자바와 마찬가지로 산술연산자, 관계연산자, 논리연산자, 비트연산자, shift연산자, 증감연산자, 조건연산자, 대입연산자들을 사용할 수 있다.

### ■ 연산자의 종류

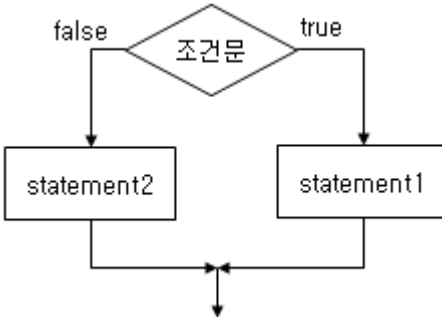
- 산술연산자: \* , / , % , + , -
- 관계연산자: < , > , <= , >=
- 논리연산자: && , || , !
- 비트연산자: & , | , ^
- shift연산자: << , >> , >>>
- 증감연산자: ++ , --
- 조건연산자: ? :
- 대입연산자: = , += , -= , \*= , /= , % =

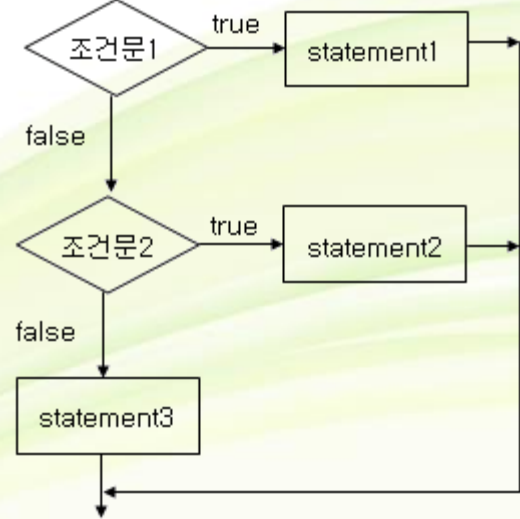


# JSP페이지의 제어문

## ❖ if문

- if문은 조건비교 분기문의 하나로 주어진 조건을 비교해서 그 결과에 따라 여러 대안들 중에서 하나를 선택할 때 사용된다.
- if문의 조건에 들어갈 수 있는 타입은 리턴타입 또는 결과 값이 boolean 값을 경우만 가능하다.

문법	순서도(Flowchart)
<pre>if(조건){     statement1; }else{     statement2; }</pre>	

문법	순서도(Flowchart)
<pre>if(조건1){     statement1; }else if(조건2){     statement2; } else{     statement3; }</pre>	

# ifTestForm.jsp

```
<html>
<head>
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<title>숫자를 입력하는 폼</title>
</head>
<body>
    <h2>숫자를 입력하세요.</h2>

    <form method="post" action="/ch04_web/ifTestPro.jsp">
        <input type="text" name="number">
        <input type="submit" value="입력완료">
    </form>
</body>
</html>
```

# ifTestPro.jsp

```
<html>
<head>
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<title>입력받은 숫자 비교</title>
</head>
<body>
    <h2>입력받은 숫자가 10보다 작거나 같은지 비교</h2>
    <%
        String strNumber = request.getParameter("number");
        int number = Integer.parseInt(strNumber);

        if (number <= 10) {
    %>
        입력받은 숫자는 <%=number %> 입니다.
    <%} %>
</body>
</html>
```

## IfDxample.html

```
<h1>If-else Example</h1>
<FORM METHOD=POST ACTION="/ch04_web/IfExample.jsp">
이름 : <INPUT TYPE="text" NAME="name"><p>
좋아하는 색깔 : <SELECT NAME="color">
    <option value="blue" selected>파란색</option>
    <option value="red">붉은색</option>
    <option value="orange">오렌지색</option>
    <option value="etc">기타</option>
</SELECT><p>
<INPUT TYPE="submit" VALUE="보내기">
</FORM>
```

## IfExample.jsp

```
<h1>If-else Example</h1>
<%!    String msg; %>
<%    String name = request.getParameter("name");
    String color = request.getParameter("color");
    if (color.equals("blue")) {    msg = "파란색";
    } else if (color.equals("red")) {    msg = "붉은색";
    }else if (color.equals("orange")){    msg = "오렌지색";
    }else{    color = "white";    msg = "기타색";
    }
%>
<body bgcolor=<%=color%>>
<%=name%>님이 좋아하는 색깔은 <%=msg%>입니다.
</body>
```

## ifTestMultiForm.jsp

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<html>
<head>
<title>이름과 전화번호를 입력하는 폼</title>
</head>
<body>
    <h2>이름과 전화번호를 입력하세요.</h2>
    <form method="post" action="/ch04_web/ifMultiTestPro.jsp">
        이름 : <input type="text" name="name"><br>
        전화번호 :
        <select name="local">
            <option value="서울">서울</option>
            <option value="경기">경기</option>
            <option value="강원">강원</option>
        </select>
        - <input type="text" name="tel"><br>
        <input type="submit" value="입력완료">
    </form>
</body>
</html>
```

# ifMultiTestPro.jsp

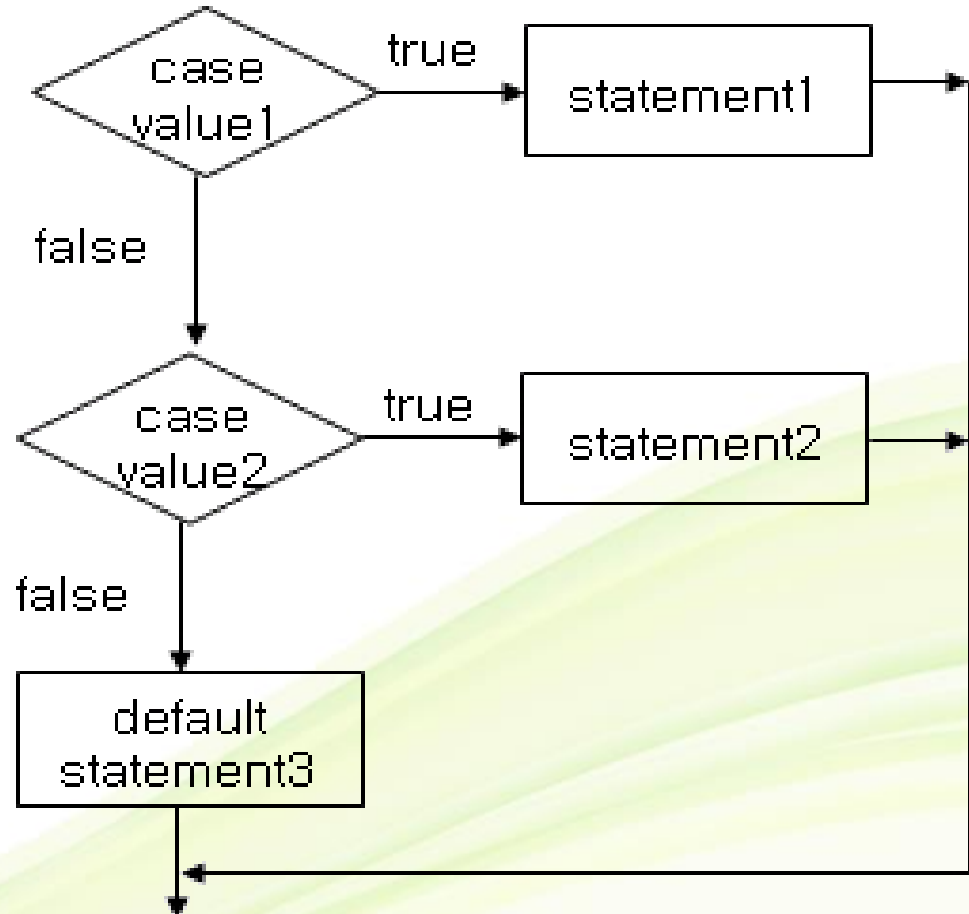
```
<%@ page language="java" contentType="text/html; charset=EUC-KR"  
    pageEncoding="EUC-KR"%>  
<% request.setCharacterEncoding("euc-kr");%>  
<%    String name = request.getParameter("name");  
    String local = request.getParameter("local");  
    String tel = request.getParameter("tel");  
    String localNum = "";  
    if (local.equals("서울")){ //local변수의 값이 서울일경우  
        localNum = "02";  
        out.println("<b>" + name + " </b>님의 전화번호는 " + localNum + "-" +  
            tel + " 입니다");  
    }else if (local.equals("경기")){ //local변수의 값이 경기일경우  
        localNum = "031";  
        out.println("<b>" + name + " </b>님의 전화번호는 " + localNum + "-" +  
            tel + " 입니다");  
    }else if (local.equals("강원")){ //local변수의 값이 강원일경우  
        localNum = "033";  
        out.println("<b>" + name + " </b>님의 전화번호는 " + localNum + "-" +  
            tel + " 입니다");  
    }  
%>
```

# ❖ Switch문

## 문법

```
switch(expression)
{
    case value1:
        statement1;
        break;
    case value2:
        statement2;
        break;
    default:
        statement3;
        break;
}
```

## 순서도(Flowchart)



# switchTestForm.jsp

```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<html>
<head>
<title>권역을 선택하는 폼</title>
</head>
<body>
    <h2>권역을 선택하세요.</h2>
    <form method="post" action="switchTestPro.jsp">
        <input type="radio" name="localNum" value="0" checked>0권역<br>
        <input type="radio" name="localNum" value="1">1권역<br>
        <input type="radio" name="localNum" value="2">2권역<br>
        <input type="radio" name="localNum" value="3">4권역<br>
        <input type="radio" name="localNum" value="4">5권역<br>
        <input type="radio" name="localNum" value="5">6권역<br>
        <input type="radio" name="localNum" value="6">7권역<br>
        <input type="submit" value="입력완료">
    </form>
</body>
</html>
```



# switchTestPro.jsp

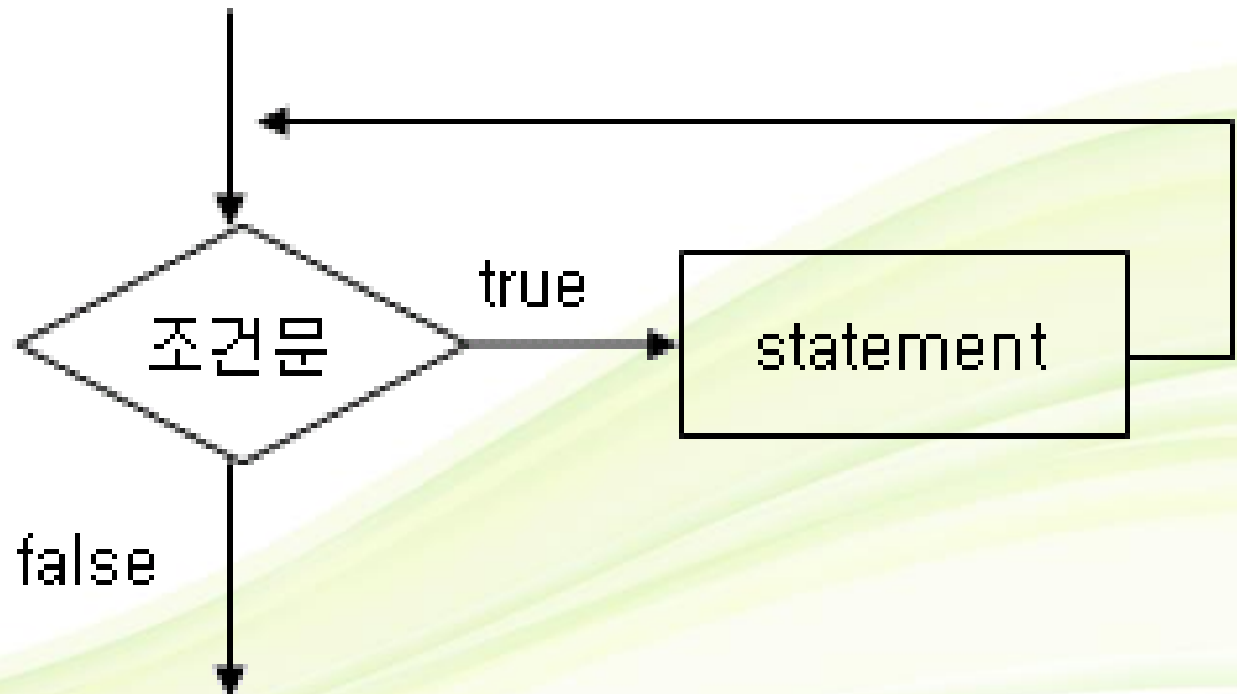
```
<%@ page language="java" contentType="text/html; charset=EUC-KR"
    pageEncoding="EUC-KR"%>
<% request.setCharacterEncoding("euc-kr");%>
<%
    int localNum = Integer.parseInt(request.getParameter("localNum"));
    String localName = "";
    switch (localNum){ //localNum변수의 값은 0~7사이의 값
        case 0:    localName = "종로, 중구, 용산";        break;
        case 1:    localName = "도봉, 강북, 노원, 성북";    break;
        case 2:    localName = "동대문, 성동, 중랑, 광진";    break;
        case 3:    localName = "강동, 송파";                break;
        case 4:    localName = "서초, 강남";                break;
        case 5:    localName = "동작, 관악, 금천";            break;
        case 6:    localName = "강서, 양천, 영등포, 구로";    break;
        case 7:    localName = "은평, 마포, 서대문";        break;
        default:    localName = "없는 권역";                break;
    }
    out.println("선택하신 지역은 <b>" + localName + " </b> 입니다");
}%>
```

# 반복문 – for

## ❖ 문법

```
for(초기값; 조건문; 증감값){  
    statement;  
}
```

## 순서도 (Flowchart)



# forTest.jsp

```
<%@ page contentType="text/html; charset=euc-kr"
import="java.util.*, java.text.*"
%>
```

```
<html>
```

```
    <body>
```

```
        <font color="blue" size=5>
```

```
            <u> 오늘 날짜 출력 </u></font><br>
```

```
        <%
```

```
            java.util.Date date=new java.util.Date();
```

```
            SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd");
```

```
        %>
```

```
        <%= sdf.format(date) %>
```

```
        <p>
```

```
        <%--
```

```
        <%
```

```
            for(int i=0;i<10;i++)
```

```
                { %>
```

```
                    안녕하세요<br>
```

```
                <% } %> --%>
```

```
</body>
```

# whileTest.jsp

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

```
<%  
    int i = 0;
```

```
    while( i<10){ //0~9까지 값이 출력된다.  
        out.println( "출력되는 값 : " + i + "<br>");  
        i++;  
    }  
%>
```

# 짝수 입력

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><head><meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<script type="text/javascript">
do {
    var num = prompt("짝수를 입력하시오", 1);
}while(num%2!=0);
document.write("당신은 짝수 " + num + "을 입력했습니다");
</script>
</body>
</html>>
```

# 웹 애플리케이션 배포

- ❖ 모든 파일을 직접 배포 가능
- ❖ war(Web Application Archive) 파일로 압축해서 배포
  - ❖ Eclipse에서는 [File] – [Export] – [WAR file]을 선택해서 생성이 가능

