1. Get / Set Method

1)getset.html

```
<!DOCTYPE html>
<html lang='ko'>
<head>
<meta charset="EUC-KR">
<title>GET 과 POST 테스트</title>
</head>
<body>
    <h1>도서 구매 정보 입력</h1>
    <h4>각 사항들에 대해 입력해 주세요</h4>
    <hr>
    <form method="get" action="/DemoWeb/QueryGetServlet">
    이름 : <input type="text" name="buyername"><br>
    주소 : <input type="text" name="buyeraddr" size="30"><br>
    <strong>선택 도서명 : </strong><br>
    <input type="checkbox" name="buyeritem" value="Java Programming">Java Programming<br>
    <input type="checkbox" name="buyeritem" value="Servlet JSP Programming">Servlet JSP Programming<br>
    <input type="checkbox" name="buyeritem" value="JDBC Programming">JDBC Programming<br>
    <p>
    <input type="submit" value="주문"><input type="reset" value="다시 입력">
    </p>
    </form>
</body>
</html>
```

2)QueryGetServlet.java

```
public class QueryGetServlet extends HttpServlet {
    protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        res.setContentType("text/html;charset=utf-8");
        PrintWriter out = res.getWriter();
        //req.setCharacterEncoding("utf-8");
        String name = entoko(req.getParameter("buyername"));
        String addr = entoko(req.getParameter("buyeraddr"));
        String [] item = req.getParameterValues("buyeritem");
        out.println("<html><body>");
        out.println("<font color='red'><h2>");
        out.println("입력 결과</h2></font><hr>");
        out.println("이름 : " + name + "<br>");
        out.println("주소 : " + addr + "<br>");
        if(item == null)
            out.println("선택하신 상품이 없습니다.");
        else{
            out.println("<pre>선택하신 상품은");
            for(int i = 0 ; i < item.length ; i++)
                out.println(entoko(item[i]) + "  ");
            out.println("입니다</pre><br>");
        }
        String getquery = req.getQueryString();
        out.println("getQuery = " + getquery);
        out.println("</body></html>");
        out.close();
    }
    private String entoutf8(String en){
        String utf8 = null;
        try{
            utf8 = new String(en.getBytes("ISO8859_1"), "UTF-8");
        }catch(IOException ex){}
        return utf8;
    }
}
```

2. Cookie

1)CookieSetServlet.java

```
public class CookieSetServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        Cookie c1 = new Cookie("discount", "20");
        Cookie c2 = new Cookie("item", "sports");
        response.setContentType("text/html;charset=utf-8");
        response.addCookie(c1);
        c2.setMaxAge(60 * 60 * 24 * 3);
        response.addCookie(c2);
        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("<h1>Cookie 설정</h1><hr>");
        out.println("설정 완료!!!");
        out.println("</body></html>");
        out.close();
    }
}
```

2)CookieGetServlet.java

```
public class CookieGetServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("<h1>Cookie 추출</h1><hr>");
        Cookie [] cookies = request.getCookies();
        if(cookies != null && cookies.length > 0){
            out.println("다음과 같은 Cookie 정보가 전달 되었습니다.<br>");
            for(int i = 0 ; i < cookies.length; i++){
                out.print("Cookie Name : " + cookies[i].getName() + "<br>");
                out.println("Cookie Value : " + cookies[i].getValue() + "<br>");
            }
        }else
            out.println("전달된 cookie 정보가 없습니다.");
        out.println("</body></html>");
        out.close();
    }
}
```

3. Session

1)SessionTestServlet.java

```
public class SessionTestServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
        HttpSession session = null;
        String msg = null;
        String param = request.getParameter("comm");
        if(param.equals("create")){
            session = request.getSession();
            if(session.isNew())
                msg = "세션 객체가 새로 생성됐습니다";
            else
                msg = "이미 세션 객체가 생성되어 있습니다";
        }else if(param.equals("destroy")){
            session = request.getSession(false);
            if(session != null){
                session.invalidate();
                msg = "세션 객체가 삭제됐습니다.";
            }else
                msg = "전달된 명령을 해석할 수 없습니다";
        }
        out.println("<html><body><h3>처리결과 : </h3>" + msg);
        out.println("</body></html>");
        out.close();
    }
}
```

2)CountTestServlet.java

```
public class CountTestServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=utf-8");
        PrintWriter out = response.getWriter();
        out.println("<html><body bgcolor='yellow'");
        out.println("<h2>누적시킬 숫자를 입력하세요...</h2>");
        out.println("<form method=\"post\" action=\"/DemoWeb/CountTestServlet\">");
        out.println("<p><input type=\"text\" name=\"number\"></p>");
        out.println("<input type=\"submit\" name=\"see\" value=\"전송\">");
        out.println("<input type=\"submit\" name=\"buy\" value=\"재입력\">");
        out.println("</form></body></html>");
        out.close();
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        HttpSession session = request.getSession(true);
        if(session.isNew())
            session.setAttribute("number", new int[]{0});
        int [] sum = (int [])session.getAttribute("number");
        int number = Integer.parseInt(request.getParameter("number"));
        sum[0] += number;
        response.setContentType("text/html;charset=euc-kr");
        PrintWriter out = response.getWriter();
        out.println("<html><body><h3>누적된 값 : " + sum[0] + "</h3>");
        out.println("<h3>세션 ID : " + session.getId() + "</h3>");
        out.println("<h3>세션발생시간 : " + new java.util.Date(session.getCreationTime()).toString() + "</h3>");
        out.println("<hr><a href=" + request.getRequestURI() + ">이전페이지</a>");
        out.println("</body></html>");
        out.close();
    }
}
```

4. Include

1)IncludeTestServlet1.java

```
public class IncludeTestServlet1 extends HttpServlet {
    protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        res.setContentType("text/html;charset=utf-8");
        PrintWriter out = res.getWriter();
        out.println("<html><head><title>Include 예제</title></head>");
        out.println("<body><h3>IncludeTestServlet1의 페이지입니다.</h3></body></html>");
        ServletContext context = getServletContext();
        RequestDispatcher rd = context.getRequestDispatcher("/IncludeTestServlet2");
        out.println("<hr color='red'>");
        rd.include(req, res);
        out.println("<hr color='blue'>");
        rd.include(req, res);
        out.close();
    }
}
```

2)IncludeTestServlet2.java

```
public class IncludeTestServlet2 extends HttpServlet {
    protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        res.setContentType("text/html;charset=utf-8");
        PrintWriter out = res.getWriter();
        out.println("<h3>IncludeTestServlet2의 페이지입니다.</h3>");
    }
}
```

3)InServlet.java

```
public class InServlet extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws IOException, ServletException {
        res.setContentType("text/html;charset=utf-8");
        PrintWriter out = res.getWriter();
        out.println("안녕하세요.");
        out.println("내부 서블릿입니다.");
        //out.close();   ***주의 닫지 말 것
    }
}
```

4)IncludingServlet.java

```java
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class IncludingServlet extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws IOException, ServletException {
        res.setContentType("text/html;charset=utf-8");
        PrintWriter out = res.getWriter();

        out.println("<html><head><title>");
        out.println("Including Servlet");
        out.println("</title></head><body bgcolor='yellow'>");
        out.println("<center><h2>Including Servlet</h2></center>");
        out.println("다음 내용은 다른 서블릿의 내용을");
        out.println("include한 것입니다. <hr>");

        ServletContext sc = getServletContext();
        RequestDispatcher rd = sc.getRequestDispatcher("/servlet/In");
        rd.include(req, res);

        out.println("<hr>이곳은 다시 IncludingServlet 입니다.");
        out.println("</body></html>");
    }
}
```

5)myimage.html

```html
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Insert title here</title>
</head>
<body>
    <h1 style='text-align:center'>여신</h1>
    <div style='text-align:center'><img src="/0613/images/jimin.jpg"></div>
</body>
</html>
```

6)IncludeDemoServlet.java

```java
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


public class IncludeDemoServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException{
        res.setContentType("text/html;charset=utf-8");
        PrintWriter out = res.getWriter();
        out.println("<!DOCTYPE html>");
        out.println("<html lang='ko'>");
        out.println("<body bgcolor='red'>");
        ServletContext context = this.getServletContext();
        RequestDispatcher rd =
                context.getRequestDispatcher("/myimage.html");
        rd.include(req, res);
        out.println("<hr>");
        rd = context.getRequestDispatcher("/servlet/Visit");
        rd.include(req, res);
        out.println("</body></html>");
        out.close();
    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException{
        this.doGet(req, res);
    }
}
```

5. Forward & Redirect

    javax.servlet.Request Dispatcher (요청재전송) --> Forwarding
    1)사용자의 요청을 받은 서블릿 혹은 JSP 에서 다른 컴포넌트(다른 서블릿, 다른 JSP, html)로 요청을 위임할 수 있는 방법
    2)사용이유는 1)보안, 2) 처리작업의 모듈화
    3)직접 요청을 받은 서블릿 혹은 JSP는 모든 작업을 직접 처리하지 않고 모듈화시킨 다른 컴포넌트로 요청을 위임하여 처리할 수 있는 방법
    4)재사용성이 높아지고, 유지보수가 쉬워짐
    5)일반적으로 MVC패턴에서 서블릿이 JSP 페이지로 포워딩할 때 주로 사용함.
    6)사용자로부터 요청을 받은 페이지와 사용자에게 응답을 처리하는 페이지를 구별할 수 있다.
    7)일반적으로 이러한 패턴을 FrontController 패턴이라고 한다.
    8)요청을 받는 페이지는 서블릿으로 구현하고, 응답을 처리하는 페이지는 JSP 로 구현할 수 있다.
    9)Model2(MVC) 의 핵심기능이다.
    10)요청한 자원의 포워딩 방법 2가지
        -RequestDispatcher의 forward() 메소드
        -ServletResponse 의 sendRedirect() 메소드

    11)Lab

    -product.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>상품페이지</title>
</head>
<body>
    <h1>상품 목록</h1>
    <form action="/0613/servlet/CartSave" method="post">
        <input type="radio" name="product" value="Computer">Computer<br />
        <input type="radio" name="product" value="Sonata">Sonata<br />
        <input type="radio" name="product" value="Keyboard">Keyboard<br />
        <input type="radio" name="product" value="Book">Book<br />
        <input type="submit" value="담기">
    </form>
</body>
</html>
```

    -ForwardDemoServlet.java

```
import java.io.IOException;

import javax.servlet.RequestDispatcher;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ForwardDemoServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException{
        ServletContext context = this.getServletContext();
        RequestDispatcher rd =
                context.getRequestDispatcher("/product.html");
        rd.forward(req, res);
    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException{
        this.doGet(req, res);
    }
}
```

    -ForwardTestServlet1.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class ForwardTestServlet1 extends HttpServlet {
    protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        res.setContentType("text/html;charset=utf-8");
        PrintWriter out = res.getWriter();
        out.println("<html><head><title>Forward 예제");
        out.println("</title></head><body bgcolor='red'><h3>ForwardTestServlet1 의 수행결과 </h3>");
        out.println("</body></html>");
        ServletContext context = getServletContext();
        RequestDispatcher rd = context.getRequestDispatcher("/servlet/ForwardTestServlet2");
        rd.forward(req,res);
        out.close();
    }
}
```

-ForwardTestServlet2.java

```
public class ForwardTestServlet2 extends HttpServlet {
    protected void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException {
        res.setContentType("text/html;charset=utf-8");
        PrintWriter out = res.getWriter();
        out.println("<html><head><title>Forward 예제");
        out.println("</title></head><body bgcolor='yellow'><h3>나는 ForwardTestServlet2 입니다.</h3>");
        out.println("</body></html>");
        out.close();
    }
}
```

12)Redirect

-Forward와 마찬가지로 다른 웹 컴포넌트에게 페이지를 위임하는 방법

-차이점은 응답을 먼저하고 클라이언트의 브라우저가 다시 재요청하기 때문에, 브라우저의 URL 과 새로 응답하는 페이지가 같다.

-response.sendRedirect(target)

-target은 동일 컨텍스트의 서브릿, 혹은 JSP, 혹은 Html, 다른 컨텍스트, 다른 도메인, 다른 웹 서버도 가능

-RedirectDemoServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class RedirectDemoServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException{
        res.setContentType("text/html;charset=utf-8");
        PrintWriter out = res.getWriter();
        out.println("<!DOCTYPE html>");
        out.println("<html lang='ko'>");
        out.println("<body bgcolor='red'>");
        //res.sendRedirect("http://www.javaexpert.co.kr"); 1. 첫번째 방법
        out.println("<script>");                                    //2. 두번째 방법
        out.println("location.href = 'http://www.naver.com';");
        out.println("</script>");
        out.println("</body></html>");
        out.close();
    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException{
        this.doGet(req, res);
    }
}
```

5. ServletContext API

1)ServletContext 는 웹 어플리케이션(Context)마다 하나씩 생성되는 객체

2)웹 어플리케이션의 LifeCycle 과 일치

3)application scope

6. Servlet/JSP 의 4가지 Scope

1)Application Scope
- 웹 어플리케이션과 LifeCycle 이 일치한다.
- 웹 어플리케이션이 존재하는 동안에는 모든 서블릿과 JSP 에서 접근 가능.
- ServletContext 객체

2)Session Scope
- 브라우저의 LifeCycle과 동일한 Scope 를 의미
- 브라우저를 종료하지 않으면 서블릿과 JSP 에서 접근 가능
- 일반적으로 time-out 방법을 사용한다.
- HttpSession 객체

3)Request Scope
- 브라우저에게 요청하고 서버에서 응답하는 Scope 까지를 의미
- 사용자 입장에서 요청한 화면이 보여질 때까지의 범위
- HttpServletRequest 해당

4)Page Scope
- 현재 브라우저에 보여지는 웹 페이지 자체
- 자바의 this

7. ServletContext 의 핵심 메소드

```
String getInitParameter(name)
InputStream getResourceAsStream(path)
void setAttribute(name, value)
Object getAttribute(name)
```

8. Context Parameter 설정

1)ContextParamServlet.java

```
@WebServlet("/ContextParam")
public class ContextParamServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
    IOException {
        String driver = getServletContext().getInitParameter("driver");
        String savePath = getServletContext().getInitParameter("savePath");

        response.setContentType("text/html; charset=EUC-KR");
        PrintWriter out = response.getWriter();
        out.print("<html><body>");
        out.print("드라이버명: " + driver +"<br>");
        out.print("저장 경로: " + savePath +"<br>");
        out.print("</body></html>");
    }
}
```

2)web.xml

```
    <context-param>
        <param-name>driver</param-name>
        <param-value>oracle.jdbc.driver.OracleDriver</param-value>
      </context-param>
      <context-param>
        <param-name>savePath</param-name>
        <param-value>c:\\save</param-value>
      </context-param>
```

9. 서블릿에서 파일 접근

- 읽기 모드만 가능

1)testFile.txt

```
안녕하세요.
ServletContext 객체를 이용한 파일읽기입니다.
감사합니다.
```

2)ContextFileServlet.java

```
@WebServlet("/ContextFile")
public class ContextFileServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        String readFile = "/WEB-INF/testFile.txt";
        InputStream is = getServletContext().getResourceAsStream(readFile);
        BufferedReader reader = new  BufferedReader(new InputStreamReader(is));

        response.setContentType("text/html; charset=EUC-KR");
        PrintWriter out = response.getWriter();
        out.print("<html><body>");

        String str = reader.readLine();
        while(str != null){
            out.print(str +"<br>");
            str = reader.readLine();
        }
        reader.close();
        out.print("</body></html>");
    }
}
```

10. 서블릿에서 속성 설정 및 참조

1)ContextSetServlet.java

```
@WebServlet("/ContextSet")
public class ContextSetServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
         // 속성값 설정
         String name = "홍길동";
         int age = 20;

         getServletContext().setAttribute("name", name );
         getServletContext().setAttribute("age", age );
    }
}
```

2)ContextGetServlet.java

```
@WebServlet("/ContextGet")
public class ContextGetServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        //속성값 얻기
        String name = (String)getServletContext().getAttribute("name");
        int age = (Integer)getServletContext().getAttribute("age");

        response.setContentType("text/html; charset=EUC-KR");
        PrintWriter out = response.getWriter();
        out.print("<html><body>");
        out.print("이름 : " + name +"<br>");
        out.print("나이 : " + age +"<br>");
        out.print("</body></html>");
    }
}
```

11. ServletContextListener API

-웹 어플리케이션이 초기화되고 제거되는 이벤트를 감지하는 ServletContextListener API를 사용하면, 언제 초기화되고 제거 되었는지를 쉽게 알 수 있다.

-이 이벤트는 JDBC의 Pooling 기법에 적용 가능

-웹 어플리케이션이 초기화될 때 Pooling 을 활성화하고, 제거될 때 Pooling 을 비활성화 시키면 효율적으로 Connection 을 관리가능

-순서

    1)ServletContextListener interface를 구현한 클래스 작성
    2)web.xml 에 구현한 클래스를 <listener> 태그로 등록
        또는 @WebListener 어노테이션 사용
    3)Tomcat 컨테이너 시작, 종료를 통해 이벤트 감지 확인

-Lab

1)ContextListenerImpl.java

```
public class ContextListenerImpl implements ServletContextListener {
    @Override
    public void contextDestroyed(ServletContextEvent event) {
          System.out.println("웹 어플리케이션 제거");
    }
    @Override
    public void contextInitialized(ServletContextEvent event) {
        System.out.println("웹 어플리케이션 초기화");
    }
}
```

2)web.xml

```
<listener>
   <listener-class>kr.co.javaexpert.libs.ContextListenerImpl</listener-class>
</listener>
```

3)ContextListenerImpl.java

```
@WebListener  <----@WebListener 를 이용한 리스너 처리
public class ContextListenerImpl implements ServletContextListener {
    @Override
    public void contextDestroyed(ServletContextEvent event) {
          System.out.println("웹 어플리케이션 제거");
    }
    @Override
    public void contextInitialized(ServletContextEvent event) {
        System.out.println("웹 어플리케이션 초기화");
    }
}
```