공부한 내용들 정리

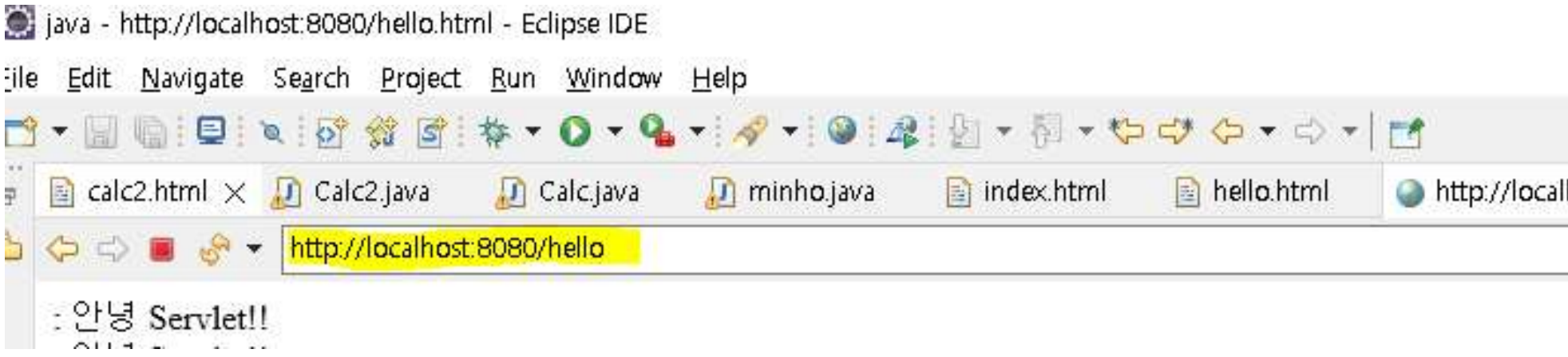
Annotation

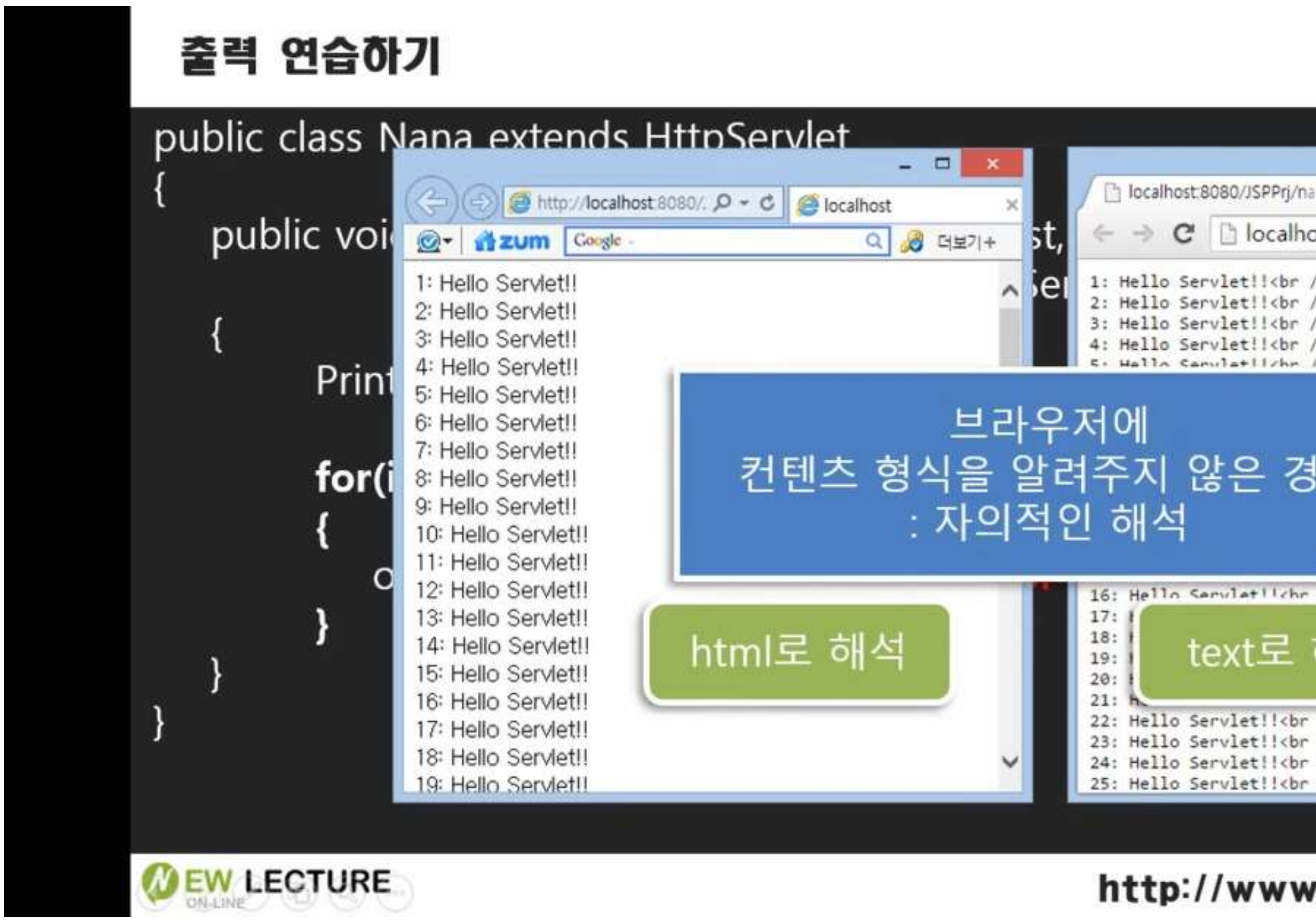
- 어노테이션에 매핑 정보를 남겨 URL과 매핑 시킬 수 있습니다.

minho.java

```
@WebServlet("/hello")
public class minho extends HttpServlet {
    @Override
    public void service(HttpServletRequest request,
    {

        response.setCharacterEncoding("UTF-8");
        response.setContentType("text/html; charset=
        // content를 통해 text/html 지정 , 브라우저에게 c
        PrintWriter out = response.getWriter();
```

- 어노테이션을 이용하여 매핑할 URL 주소 입력





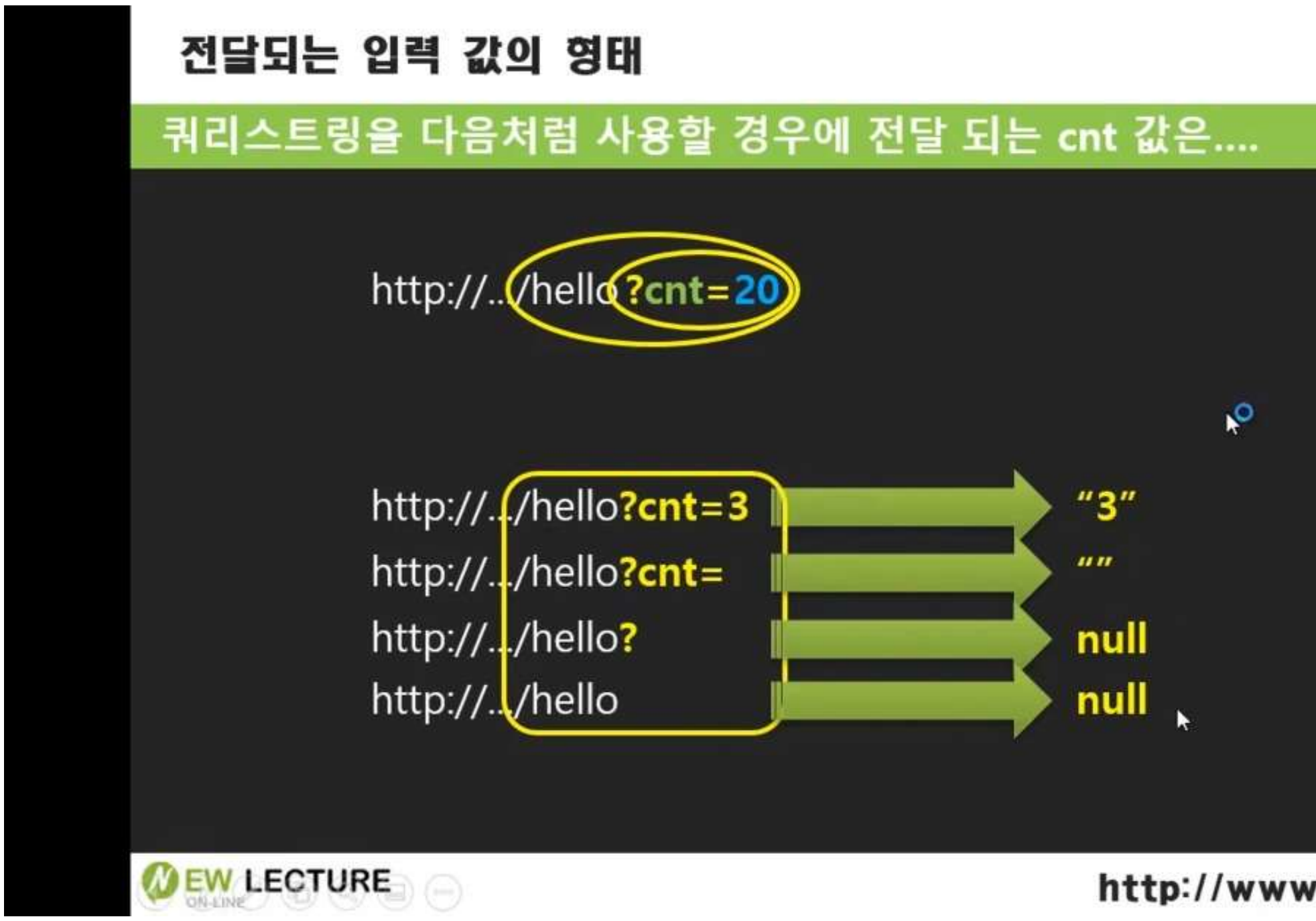
minho.java

```java
@WebServlet("/hello")
public class minho extends HttpServlet {
    @Override
    public void service(HttpServletRequest request
    {

        response.setCharacterEncoding("UTF-8");
        response.setContentType("text/html; charse
        // content를 통해 text/html 지정 , 브라우저에게
        PrintWriter out = response.getWriter();
```

res.setCharacterEncoding("UTF-8")

- **코딩방식을 UTF-8 로 보내줍니다.**

- **쿼리스트링 입력 오류**



- **Null Exception을 피하기 위해 수정**

```
{
        response.setCharacterEncoding("UTF-8");
        response.setContentType("text/html; charset=UTF-8");


        // request.setCharacterEncoding("UTF-8"); 필터 22번째
        // content를 통해 text/html 지정 , 브라우저에게 charset=U


        // request.setCharacterEncoding("UTF-8");
        // Encoding 방식을 UTF-8로 읽어달라고 설정 가능하지만 필터를

        PrintWriter out = response.getWriter();

        String title = request.getParameter("title");
        String content = request.getParameter("content");

        out.print(title);
        out.print(content);
```

사용자가 Page에서 서버에게로 전달하기 위해선, 입력(Post)을 해주어야 하기 때문에

Submit이 필요합니다. (주소창에 입력할 필요없이)

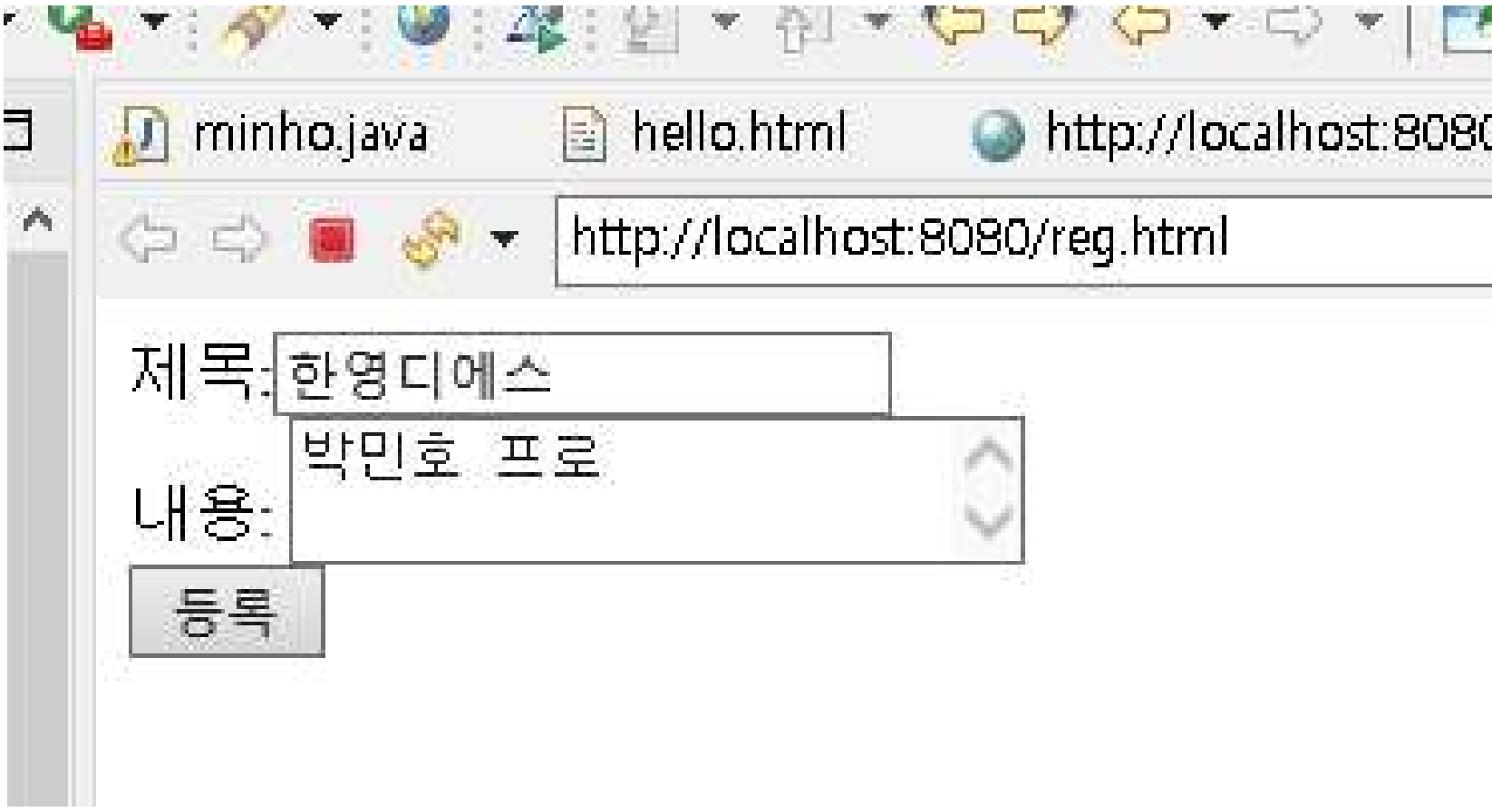
reg.html

```
<form action="hello">
<div>
    <label>"안녕하세요?" 를 몇번 보고 싶으신
    </div>
<div>
<input type="text" name="cnt" >
<input type="submit" value="전송하기" />
```

```
<div>
    <form action="notice-reg" method="post">
    <div>
        <label>제목:</label><input name="title"
        </div>
        <div>
        <label>내용:</label>
        <textarea name="content"></textarea>
        </div>
    <div>
    <input type="submit" value="등록" />
```

NoticeReg.java

```
@WebServlet("/notice-reg")
public class NoticeReg extends HttpServlet {
    @Override
    public void service(HttpServletRequest request, HttpServletRes
    {

        response.setCharacterEncoding("UTF-8");
        response.setContentType("text/html; charset=UTF-8");


        // request.setCharacterEncoding("UTF-8"); 필터 22번째줄
        // content를 통해 text/html 지정 , 브라우저에게 charset=UTF-8 이


        // request.setCharacterEncoding("UTF-8");
        // Encoding 방식을 UTF-8로 읽어달라고 설정 가능하지만 필터를 이용하면


        PrintWriter out = response.getWriter();

        String title = request.getParameter("title");
        String content = request.getParameter("content");
```

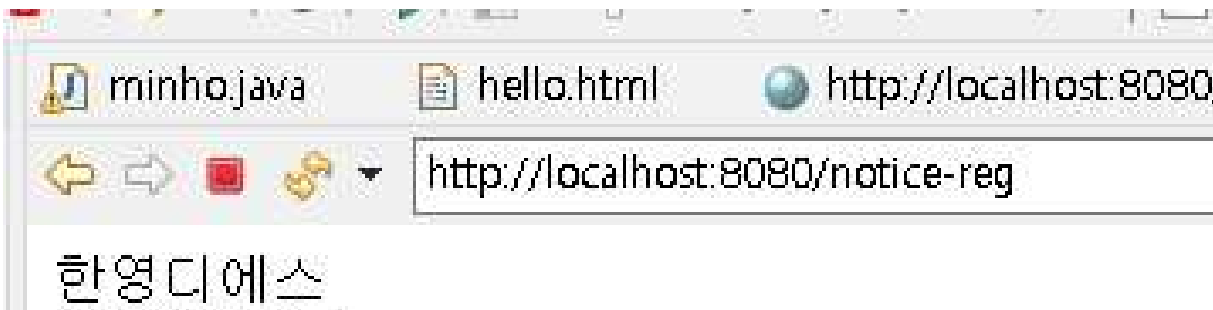


가로로 나와서 title 출력해주는 코드 살짝 수정

- **+
 추가**

NoticeReg.java

```
out.print(title + "<br>");
out.print(content);
```



▼ Servlet Filter를 이용한 예시

- 필터도 Servlet Class와 동일하게 web.xml로 매핑할 수 있으며 또한 어노테이션으로 매핑

1. 각 Servlet Class UTF-8 인코딩 코드 추가
2. 우선 Filter 인터페이스를 implements(구현)한 후 Filter 클래스를 만듭니다.

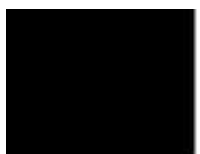
CharacterEncodingFilter.java

```
@WebFilter("/*") // 필터를 어노테이션을 통하여 간편하게 사용
public class CharacterEncodingFilter implements Filter {

    @Override
    public void doFilter(ServletRequest request
            , ServletResponse response
            , FilterChain chain)
    // chain이 넘겨줄까 말까 결정함.
            throws IOException, ServletException {

        //System.out.println("before filter");   -> 요청시

        request.setCharacterEncoding("UTF-8");

        chain.doFilter(request, response); // 필터를 거치는

        //System.out.println("after filter");    -> 응답시
```

상태유지 필요성과 구현의 어려움

Servlet 의 값을 공유 하는데 Java처럼 클래스 변수가 없어 어려움이 있습니다.

Servlet 의 값을 공유하는데 아래 5가지의 도구를 사용합니다.



상태 유지를 위한 5가지 방법

Application 객체와 그것을 이용한 상태값 저장

각 Servlet Class의 값은 Servlet Context 라는 곳에 저장시킬수 있어서 Servlet Class가 죽어도 불리웁니다.

- **Servlet Context (application저장소) 를 사용하여 계산기 만들기**

calc2.html

calc2.java

```java
@WebServlet("/calc2")
public class Calc2 extends HttpServlet {

    @Override
    protected void service(HttpServletRequest request
            , HttpServletResponse response) throws ServletException
    , IOException {

        response.setCharacterEncoding("UTF-8");
        response.setContentType("text/html; charset=UTF-8");
        // (서블릿 컨텍스트) 어플리케이션 저장소이다
        ServletContext application = request.getServletContext();

        // 이때 빈 문자열이 오거나 or 진짜 담긴 값이 오는 경우 발생
        String v_ = request.getParameter("v");
        String op = request.getParameter("operator");

        int v = 0;
        if (!v_.equals("")) v = Integer.parseInt(v_);

        // 계산 버튼
        if(op.equals("=")) {

        // 계산하는곳

            int x = (Integer)application.getAttribute("value");
            int y = v;
            String operator = (String)application.getAttribute("op");

            int result = 0;

            if(operator.equals("덧셈"))
                result = x+y;

            else
                result = x-y;

            // 데이터 출력
            response.getWriter().printf("result is %d\n", result); //
        }
        // 어플리케이션 저장소에 2가지를 저장한다. (객체 속성 Attribute 이용)
        else {
            application.setAttribute("value", v); // = 버튼이 아닐시 값을 저
                application.setAttribute("op", op); // = 버튼이 아닐시 서블
        }
```

HTML