

Üst Düzey Programlama

JSP

JSP

JSP durağan HTML ile dinamik oluşturulan içeriği birleştirmeyi sağlar.

Bir web sayfası tasarlama programı ile web sayfasını tasarlar daha sonra dinamik içeriğin bulunduğu noktalara **<% %>** etiketleri arasında Java kodu yazılır.

JSP sayfası arka planda otomatik olarak Servlet sınıfına çevrilir ve derlenir. Derlenen servlet sınıfı çalıştırılır ve isteğe cevap vermesi sağlanır.

Servletlerin dezavantajları

- * Sunumda iyi değildir.
- * HTML kodu yazmak ve değiştirmek kolay değildir.
- * Standart HTML araçlarını kullanamazsınız.
- * Java bilmeyen fakat iyi bir HTML tasarımcısı, Servlet içerisindeki kodu kolaylıkla değiştiremez.

JSP nin avantajları

- * HTML kolaylıkla yazılabilir ve değiştirilebilir.
- * İstenilen web sayfası tasarım programı kullanılabilir.
- * Geliştirme takımı ile tasarım takımını ayırabilirsiniz.

Basit JSP Sözdizilimleri

JSP Yorum: **<%-- yorum --%>**

JSP İfadesi(expression): **<%=degisken%>**

JSP Betikleri(Script): **<% ifadeler %>**

JSP Tanımlama: **<%! değişken %>**
<%! metot %>

JSP Yönergeleri(directive):
<%@ yonerge ozellik="deger" %>

JSP de tanımlı olan değişkenler

***request** : HttpServletRequest

***response** : HttpServletResponse

***session** : HttpSession = request.getSession();

***out** : Writer = response.getWriter();

***application** : ServletContext

Java Betiklerinin Yazılması

`<% Java kodu%>`

`<%`

`String sorguVerisi=request.getQueryString();`

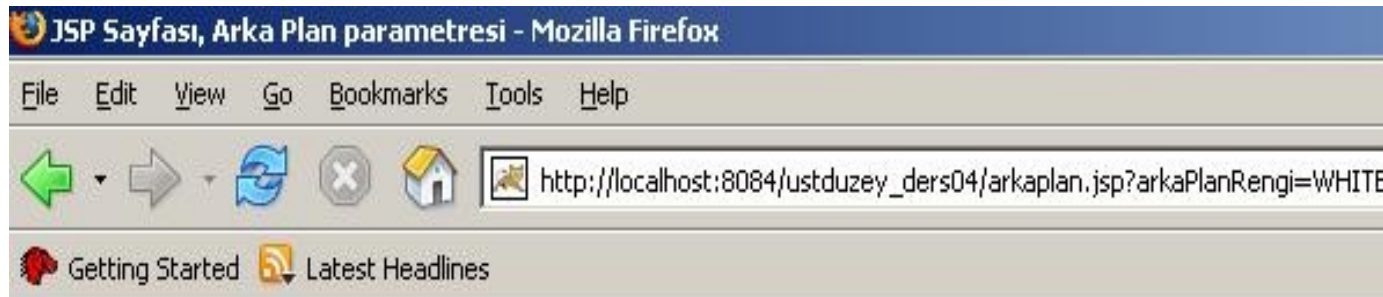
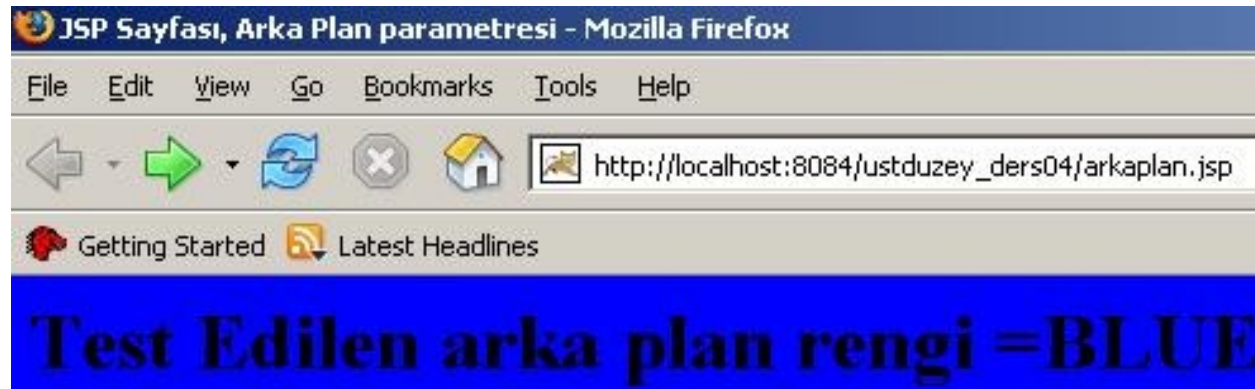
`%>`

`<%=ifade%> ==> out.println(ifade);`

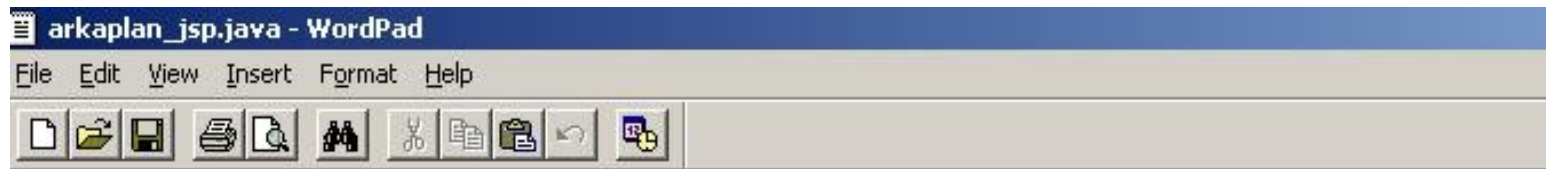
Yazılan betik satırları, oluşacak Servletin `_jspService()` isimli metoduna doğrudan eklenir.

```
arkaplan.jsp x|
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Sayfası, Arka Plan parametresi</title>
</head>
<%
String arkaPlanRengi = request.getParameter("arkaPlanRengi");
if((arkaPlanRengi==null) || (arkaPlanRengi.trim().equals(""))){
    arkaPlanRengi="BLUE";
}
%>
<body bgcolor="<%=arkaPlanRengi%>">
<h1>Test Edilen arka plan rengi =<%=arkaPlanRengi%></h1>
</body>
</html>
```

Test Edilen arka plan rengi =WHITE



```
package org.apache.jsp;

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;

public final class arkaplan_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {

    private static java.util.List _jspx_dependants;

    public Object getDependants() {
        return _jspx_dependants;
    }

    public void _jspService(HttpServletRequest request, HttpServletResponse response)
        throws java.io.IOException, ServletException {

        JspFactory _jspxFactory = null;
        PageContext pageContext = null;
        HttpSession session = null;
        ServletContext application = null;
        ServletConfig config = null;
        JspWriter out = null;
        Object page = this;
        JspWriter _jspx_out = null;
        PageContext _jspx_page_context = null;
```

```

try {
    _jspxFactory = JspFactory.getDefaultFactory();
    response.setContentType("text/html;charset=UTF-8");
    pageContext = _jspxFactory.getPageContext(this, request, response,
        null, true, 8192, true);
    _jspx_page_context = pageContext;
    application = pageContext.getServletContext();
    config = pageContext.getServletConfig();
    session = pageContext.getSession();
    out = pageContext.getOut();
    _jspx_out = out;

    out.write("\n");
    out.write("\n");
    out.write("\n");
    out.write("<html>\n");
    out.write("    <head>\n");
    out.write("        <meta http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\">\n");
    out.write("        <title>JSP Sayfası, Arka Plan parametresi</title>\n");
    out.write("    </head>\n");
    out.write("    ");

    String arkaPlanRengi = request.getParameter("arkaPlanRengi");
    if((arkaPlanRengi==null) || (arkaPlanRengi.trim().equals(""))){
        arkaPlanRengi="BLUE";
    }
}

```

```
out.write("\n");
out.write("    <body bgcolor=\"\"");
out.print(arkaPlanRengi);
out.write(">\n");
out.write("        <h1>Test Edilen arka plan rengi =");
out.print(arkaPlanRengi);
out.write("</h1>    \n");
out.write("    </body>\n");
out.write("</html>\n");
} catch (Throwable t) {
    if (!(t instanceof SkipPageException)){
        out = _jspx_out;
        if (out != null && out.getBufferSize() != 0)
            out.clearBuffer();
        if (_jspx_page_context != null) _jspx_page_context.handlePageException(t);
    }
} finally {
    if (_jspxFactory != null) _jspxFactory.releasePageContext(_jspx_page_context);
}
}
```

Koşullu HTML sayfası üretme

```
<%  
    if (kosul){  
%>  
HTML eger kosul dogru ise  
<%  
}  
else{  
%>  
HTML eğer koşul yanlış ise  
<%  
}  
%>
```

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Koşullu çıktı üretme</title>
</head>
<body>
  <% if (Math.random()<0.5) { %>
    <h1> Bugün hava güneşli</h1>
  <%}else{%>
    <h1> Bugün hava yağmurlu </h1>
  <%}%>
</body>
</html>
```

```

out.write("\n");
out.write("\n");
out.write("\n");
out.write("<html>\n");
out.write("    <head>\n");
out.write("        <meta http-equiv=\"Content-Type\" content=\"text/html; charset=UTF-8\">\n");
out.write("        <title>Koşullu İşletme Örneği</title>\n");
out.write("    </head>\n");
out.write("    <body>\n");
out.write("        ");
if (Math.random()<0.5) {
    out.write("\n");
    out.write("        <h1> Bugün hava güzel</h1>\n");
    out.write("        ");
}else{
    out.write("\n");
    out.write("        <h1> Bugün hava yağmurlu </h1>\n");
    out.write("        ");
}

out.write("\n");
out.write("    </body>\n");
out.write("</html>\n");

```

Değişken ya da metot bildirimi

<%!

değişken ya da metotlar tanımlanır.

%>

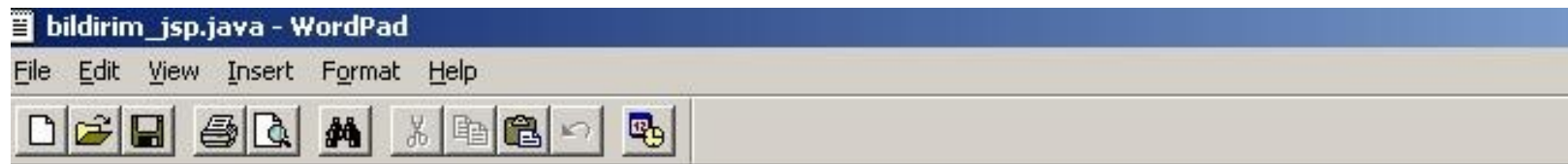
yazılan metot ve değişken isimlerine dikkat edilmelidir. Servlete özel olan metot ve değişken isimleri kullanılmamalıdır.(doGet,doPost,init,...)

Oluşan Servlette, _jspService() metodu dışına eklenirler.


```
bildirim.jsp x
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Bildirim(Declaration)</title>
</head>
<body>
<%!
    private String yazi;
    private int sayi=0;
    private String rastgeleYazi(){
        yazi="<H2>"+Math.random()+"</H2>";
        return yazi ;
    }
%>

    Rastgele Yazi = <%=rastgeleYazi()%><br>
    Rastgele Yazi = <%=yazi%>
    Sayi = <%= ++sayi%>
</body>
</html>
```



```
package org.apache.jsp;

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;

public final class bildirim_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent {

    private String yazi;
    private int sayi=0;
    private String rastgeleYazi(){
        yazi="<H2>" + Math.random() + "</H2>";
        return yazi;
    }

    private static java.util.List _jspx_dependants;

    public Object getDependants() {
        return _jspx_dependants;
    }

    public void _jspService(HttpServletRequest request, HttpServletResponse response)
        throws java.io.IOException, ServletException {
```



Rastgele Yazi =

0.06013470868394355

Rastgele Yazi =

0.06013470868394355

Sayi = 4

Öntanımlı Değişkenler

- ***request** : doGet ve doPost metotlarında bulunan HttpServletRequest nesnesidir.
- ***response**: doGet ve doPost metotlarında bulunan HttpServletResponse nesnesidir.
- ***out**: JspWriter sınıfındandır. PrintWriter sınıfına tamponlama özelliği eklenmiştir. Kullanılan tampon boyutu *page* yönergesinin *buffer* değişkeni kullanılarak değiştirilebilir.

Öntanımlı Değişkenler

- ***session**: HttpSession nesnesidir. Session'lar JSP lerde otomatik olarak oluşturulurlar. JSP sayfasında page yönergesinin session özelliği ile bu durumu iptal edebilirsiniz.
- ***application**: getServletContext() metodu ile elde edilen ServletContext değişkenidir. Servletler ve JSP ler kalıcı verilerini ServletContext de saklarlar. Bu değişkenler tüm uygulamadaki Servlet ve JSP lerce ortak kullanılırlar.

setAttribute(String,Object)
getAttribute(String)

Öntanımlı Değişkenler

***config**: Bu değişken ServletConfig nesnesidir. Sadece sayfaya özel değişkenler saklanabilir.

***pageContext** : bu nesne JSP ile birlikte eklenmiştir. Bu değişken sayfanın birçok özelliğine ortak erişim için kullanılır. Örn; `getRequest()`, `getResponse()`, `getOut()`, `getSession()` gibi metotları vardır. Eğer bir metot sayfa ile ilgili birçok değişkene ihtiyaç duyuyorsa, bu nesne parametre olarak gönderilir.

***page** : sayfadaki `this` nesnesine karşılık gelmektedir.

Oluşturulan Servletin yapısını sayfa yönergeleri ile değiştirme

Yazılan JSP yönergeleri(directive) oluşturulan Servletin yapısını değiştirebilir.

```
<%@ yonerge degisken="deger" %>
```

```
<%@ yonerge degisken1="deger1"  
      degisken2="deger2"
```

```
      ...  
      degiskenN="degerN"
```

```
%>
```

Oluşturulan Servletin yapısını sayfa yönergeleri ile değiştirme

JSP de üç temel yönerge bulunmaktadır.

page, include, taglib

page yönergesi sayfada istenilen yerde yazılabilir. En fazla kullanılan değişkenler(import, contentType, pageEncoding, session, buffer, autoFlush, info, errorPage, isErrorPage, isThreadSafe ve extends) dir.

page import özelliği

Sınıfları import etmek için kullanılır.

```
<%@page import="paket.sinif"%>
```

```
<%@page import="paket.sinif1,paket.sinif2,..."%>
```

Örn:

```
<%@ page import="java.util.*"%>
```

```
yonerge01.jsp x|
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<%@ page import="ce.comu.ServletYardimci" %>

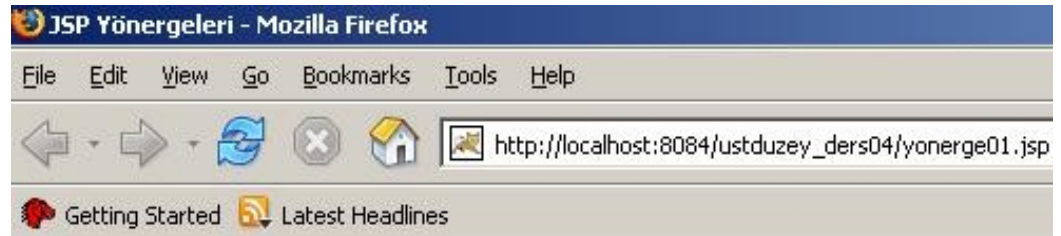
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Yönergeleri</title>
</head>
<body>
RastgeleSayi = <%=ServletYardimci.RastGeleYazi()%>

</body>
</html>
```

```
ServletYardimci.java x|
package ce.comu;

public class ServletYardimci {

    public static String RastGeleYazi(){
        String yazi;
        yazi = ""+Math.random();
        return yazi;
    }
}
```



page contentType ve pageEncoding özellikleri

contentType değişkeni Content-Type response bağıllığını değiştirir. istemciye gönderilen dökümanın MIME tipini değiştirir.

```
<%@ page contentType="MIME-tipi" %>
```

```
<%@ page contentType="MIME-tipi;charset=KarakterKümesi" %>
```

```
<%@ page contentType="application/vnd.ms-excel" %>
```

```
<% response.setContentType("application/vnd.ms-excel"); %>
```

```
<%@ page pageEncoding="UTF-8" %>
```

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Sayfa Yönergeleri</title>
</head>
<body>
<%
String bicim = (String) request.getParameter("bicim");
if((bicim!=null) && (bicim.trim().equals("excel"))){
    response.setContentType("application/vnd-excel");
}
%>
<table border="1">
<tr><th>Ad</th><th>Numara</th></tr>
<tr><td>AAA</td><td>90</td></tr>
<tr><td>BBB</td><td>80</td></tr>
</table>
</body>
</html>
```

page session özelliği

sayfada session değişkeninin olup olmayacağına karar verir. Varsayılan değeri "true" dır.

```
<%@page session="true"%>
```

```
<%@page session="false"%>
```

Eğer sayfada session yok ise bu değeri "false" yapmak sunucunun bellek kullanımını azaltır.

page buffer ve autoFlush özelliği

out nesnesi için kullanılacak tampon bilgisini belirler.

```
<%@page buffer="kb boyut" %>
```

```
<%@page buffer="none" %>
```

```
<%@page autoFlush="true" %>
```

```
<%@page autoFlush="false" %>
```

Üst Düzey Programlama

JSP