

# Entwicklung einer Java Web-Applikation

Das folgende Beispiel verwendet den **Glassfish** JavaEE server  
erfordert Setzen der Umgebungsvariablen `GLASSFISH_HOME`, & `CLASSPATH`

## 1. Bereitstellen der Servlet-Klasse:

das Servlet muss Informationen bereitstellen,  
wie es deployt werden soll

Annotations

Alternative: separater Deployment Descriptor (XML-Datei)

compilieren der Java-Klassen

## 2. Starten des JavaEE Servers, z.B. mit

`asadmin start-domain domain1`

## 3. Deployen der Web-Applikation, z.B. über

■ **Admin Console** Web Interface (Applications → Deploy)

■ oder mit dem `asadmin` Command Line Interface, z.B.

`asadmin deploy`

## 4. Zugriff auf die Web-Applikation, z.B. über Browser



# Attribute in impliziten Objekten

Durch Zuweisung an Attribute können **Java-Objekte** über die aktuelle JSP-Seite hinaus **gespeichert** werden  
Attribute können in den **impliziten Objekten** `pageContext`, `request`, `session` und `application` gespeichert werden  
Der **Gültigkeitsbereich** (Scope) des Attributs ist identisch mit dem des impliziten Objekts, in dem es gespeichert wird  
Die entsprechenden impliziten Objekte stellen **Methoden** zur **Verwaltung** der darin abgelegten **Attribute** bereit

**setzen** eines Attributs:

```
void setAttribute(String name, Objekt object);
```

**auslesen** eines Attributs:

```
(Objekt) getAttribute(String name);
```

auslesen der **Namen** aller im **Objekt** abgelegten Attribute:

```
Java.util.Enumeration getAttributeNames();
```

**löschen** eines Attributs aus dem **Objekt**:

```
void removeAttribute(String name);
```

Auch in einem **Servlet** können Attribute in einem **Scope** abgelegt werden, z.B.

- `getServletContext().setAttribute("xyz", obj);`
- `request.setAttribute("xyz", obj);`
- `request.getSession().setAttribute("xyz", obj);`

