

# Klausurzusammenfassung Webengineering

Florian Kling

2. August 2017

# 1 HTML

## 1.1 Standard HTML Aufbau

```
1      <!DOCTYPE html>
2      <html>
3          <head>
4              <title>Titel der Webseite</title>
5          </head>
6          <body>
7              Hello World!
8          </body>
9      </html>
```

## 1.2 Tabellen

A	B	C
1		2
3	4	

```
1 <table border="1">
2   <tr>
3     <th>A</th>
4     <th>B</th>
5     <th>C</th>
6   </tr>
7   <tr>
8     <td colspan="2">1</td>
9     <td rowspan="2">2</td>
10  </tr>
11  <tr>
12    <td>3</td>
13    <td>4</td>
14  </tr>
15 </table>
```

### 1.3 Listings

Es gibt folgende Listen-Style:

- `<ol type="1">` → numerische Aufzählung
- `<ol type="A">` → Alphanumerische Aufzählung, große Buchstaben
- `<ol type="a">` → Alphanumerische Aufzählung, kleine Buchstaben
- `<ol type="I">` → große Romanische Ziffern
- `<ol type="i">` → kleine Romanische Ziffern

Mit der `start`-Property kann festgelegt werden, mit welchem Zeichen begonnen wird.

Mit der `reverse`-Property kann die Aufzählung rückwärts gestaltet werden.

## 2 JSP

### 2.1 Java-Bean

Eine Bean ist eine Java-Klasse die nur Variablen, getter und setter sowie einen leeren Konstruktor besitzt.

### 2.2 Variablen im JSP-Umfeld

- `request` → Gültigkeitsbereich
- `response`
- `session` → Gültigkeitsbereich
- `pageContext` → Gültigkeitsbereich
- `application` → Gültigkeitsbereich
- `config`
- `out`

## 2.3 Gültigkeitsbereiche

Name	Bedeutung
application	Mit dem Starten der Applikation (bspw. durch Hochfahren des Tomcat) steht dieser Gültigkeitsbereich bis zum Beenden der Applikation zur Verfügung. Dieser ist der umfassendste Gültigkeitsbereich und sollte nur für Attribute genutzt werden, die wirklich für die gesamte Applikation von Bedeutung sind.
session	Eine Session umfasst eine Nutzersitzung und umfasst mehrere Anfragen. So kann der Status einer Benutzers während der Nutzung gespeichert werden (bspw. ein Warenkorb). Ein ausführliches Kapitel zum Umgang mit Session und dabei zu Beachtendes folgt im weiteren Verlauf des Tutorials.
request	Dieser Gültigkeitsbereich umfasst genau eine Anfrage eines Nutzers. Aufgrund eines möglichen Forwardings der Anfrage an weitere Servlets bzw. JSPs kann ein Request sich über mehrere JSPs oder Servlets erstrecken.
page	Dieser Gültigkeitsbereich existiert nur in JSPs und ist nur innerhalb genau einer JSP gültig. Bei einem Forwarding gehen Attribute dieses Gültigkeitsbereichs verloren.

## 2.4 JSP-Snippets

Oberste Zeile einer JSP-Seite:

```
1 | <%@ page language="java" contentType="text/
   | html; charset=UTF-8" pageEncoding="UTF-8"%
   | >
```

Einfache Ausgaben

```
1 | <%= 4+5 %>
2 | <%= request.getHeader("User-Agent") %>
```

Java-Umfeld erschaffen

```
1 | <% JAVA-CODE, bsp. for, if, usw. %>
```

Instanz-Variablen, Methoden deklarieren

```
1 | <%!
2 | private int counter = 0;
3 | public synchronized int next () {
4 |     counter++;
5 |     return counter;
```

```
6 |      }  
7 |      %>
```

Kommentare

```
1 | <%-- Kommentar --%>
```

Bean laden

```
1 | <jsp:useBean id="Objektvariable" class="Klassenname"  
  |      />
```

Setter einer Bean aufrufen

```
1 | <jsp:setParameter name="Objektvariable" property=""  
  |      Variable" param="Wert" />
```















Getter einer Bean aufrufen

```
1 | <jsp:getParameter name="Objektvariable" property=""  
  |      Variable" />
```

## 2.5 JSTL

Import-Tag

```
1 | <%@ taglib prefix = "c" uri = "http://java.sun.com/  
  |      jsp/jstl/core" %>
```

1	<b>&lt;c:out&gt;</b>  Like <%= ... >, but for expressions.
2	<b>&lt;c:set &gt;</b>  Sets the result of an expression evaluation in a ' <b>scope</b> '
3	<b>&lt;c:remove &gt;</b>  Removes a <b>scoped variable</b> (from a particular scope, if specified).
4	<b>&lt;c:catch&gt;</b>  Catches any <b>Throwable</b> that occurs in its body and optionally exposes it.
5	<b>&lt;c:if&gt;</b>  Simple conditional tag which evaluates its body if the supplied condition is true.
6	<b>&lt;c:choose&gt;</b>  Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by <b>&lt;when&gt;</b> and <b>&lt;otherwise&gt;</b> .
7	<b>&lt;c:when&gt;</b>  Subtag of <b>&lt;choose&gt;</b> that includes its body if its condition evaluates to ' <b>true</b> '.
8	<b>&lt;c:otherwise &gt;</b>  Subtag of <b>&lt;choose&gt;</b> that follows the <b>&lt;when&gt;</b> tags and runs only if all of the prior conditions evaluated to ' <b>false</b> '.
9	<b>&lt;c:import&gt;</b>  Retrieves an absolute or relative URL and exposes its contents to either the page, a String in ' <b>var</b> ', or a Reader in ' <b>varReader</b> '.
10	<b>&lt;c:forEach &gt;</b>  The basic iteration tag, accepting many different collection types and supporting subsetting and other functionality .
11	<b>&lt;c:forTokens&gt;</b>  Iterates over tokens, separated by the supplied delimiters.
12	<b>&lt;c:param&gt;</b>  Adds a parameter to a containing ' <b>import</b> ' tag's URL.
13	<b>&lt;c:redirect &gt;</b>  Redirects to a new URL.
14	<b>&lt;c:url&gt;</b>  Creates a URL with optional query parameters

S.No.	Function & Description
1	<b>fn:contains()</b> <a href="#">↗</a> Tests if an input string contains the specified substring.
2	<b>fn:containsIgnoreCase()</b> <a href="#">↗</a> Tests if an input string contains the specified substring in a case insensitive way.
3	<b>fn:endsWith()</b> <a href="#">↗</a> Tests if an input string ends with the specified suffix.
4	<b>fn:escapeXml()</b> <a href="#">↗</a> Escapes characters that can be interpreted as XML markup.
5	<b>fn:indexOf()</b> <a href="#">↗</a> Returns the index within a string of the first occurrence of a specified substring.
6	<b>fn:join()</b> <a href="#">↗</a> Joins all elements of an array into a string.
7	<b>fn:length()</b> <a href="#">↗</a> Returns the number of items in a collection, or the number of characters in a string.
8	<b>fn:replace()</b> <a href="#">↗</a> Returns a string resulting from replacing in an input string all occurrences with a given string.
9	<b>fn:split()</b> <a href="#">↗</a> Splits a string into an array of substrings.
10	<b>fn:startsWith()</b> <a href="#">↗</a> Tests if an input string starts with the specified prefix.
11	<b>fn:substring()</b> <a href="#">↗</a> Returns a subset of a string.
12	<b>fn:substringAfter()</b> <a href="#">↗</a> Returns a subset of a string following a specific substring.
13	<b>fn:substringBefore()</b> <a href="#">↗</a> Returns a subset of a string before a specific substring.
14	<b>fn:toLowerCase()</b> <a href="#">↗</a> Converts all of the characters of a string to lower case.
15	<b>fn:toUpperCase()</b> <a href="#">↗</a> Converts all of the characters of a string to upper case.
16	<b>fn:trim()</b> <a href="#">↗</a> Removes white spaces from both ends of a string.