Klausurzusammenfassung Webengineering

Florian Kling

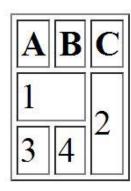
2. August 2017

1 HTML

1.1 Standard HTML Aufbau

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

1.2 Tabellen



```
1
2
 3
 A
 B
4
5
 C
6
 7
 1
8
9
 2
10
 11
 12
 3
13
 4
 14
15
```

1.3 Listings

Es gibt folgende Listen-Style:

- → numerische Aufzählung
- \bullet <
ol type="A"> \to Alphabetische Aufzählung, große Buchstaben
- \bullet \to Alphabetische Aufzählung, kleine Buchstaben
- \bullet \to große Romanische Ziffern
- \bullet \to 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 \rightarrow Gültigkeitsbereich
- response
- session \rightarrow Gültigkeitsbereich
- pageContext \rightarrow 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 wer-
	den, die wirklich für die gesamte Applikation von Bedeutung
	sind.
session	Eine Session umfasst eine Nutzersitzung und umfasst meh-
	rere 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 in-
	nerhalb genau einer JSP gültig. Bei einem Forwarding gehen
	Attribute dieses Gültigkeitsbereichs verloren.

2.4 JSP-Snippets

Oberste Zeile einer JSP-Seite:

Einfache Ausgaben

Java-Umfeld erschaffen

Instanz-Variablen, Methoden deklarieren

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

Kommentare

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

Bean laden

Setter einer Bean aufrufen

Getter einer Bean aufrufen

2.5 JSTL

Import-Tag

1	<c:out> C:out> Like <%= >, but for expressions.</c:out>
2	<c:set> ☑* Sets the result of an expression evaluation in a 'scope'</c:set>
3	<c:remove> ☑* Removes a scoped variable (from a particular scope, if specified).</c:remove>
4	<c:catch> ☑* Catches any Throwable that occurs in its body and optionally exposes it.</c:catch>
5	<c:if></c:if>
6	<pre><c:choose> Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by <when> and <otherwise>.</otherwise></when></c:choose></pre>
7	<pre><c:when> \mathbb{Z}^* Subtag of <choose> that includes its body if its condition evalutes to 'true'.</choose></c:when></pre>
8	<c:otherwise> \(\sigma^* \) Subtag of <choose> that follows the <when> tags and runs only if all of the prior conditions evaluated to 'false'.</when></choose></c:otherwise>
9	<c:import></c:import>
10	$\begin{tabular}{lll} << c: for Each $>$ \mathbb{Z}^* \\ The basic iteration tag, accepting many different collection types \\ and supporting subsetting and other functionality . \\ \end{tabular}$
11	<c:fortokens> ☑ Iterates over tokens, separated by the supplied delimeters.</c:fortokens>
12	<c:param> ☑* Adds a parameter to a containing 'import' tag's URL.</c:param>
13	<c:redirect> □ Redirects to a new URL.</c:redirect>
14	<c:url> Creates a URL with optional query parameters</c:url>

S.No.	Function & Description
1	fn:contains() \mathbb{Z}^* Tests if an input string contains the specified substring.
2	fn:containsIgnoreCase() Tests if an input string contains the specified substring in a case insensitive way.
3	$ \begin{array}{l} \text{fn:endsWith()} \ \ \!$
4	$\begin{tabular}{ll} \textbf{fn:escapeXmI()} $ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $ $$
5	$\begin{array}{l} \textbf{fn:indexOf()} \ \overline{ \ \ } \\ \textbf{Returns the index withing a string of the first occurrence of a} \\ \textbf{specified substring.} \end{array}$
6	fn:join() $\ensuremath{\mathbb{Z}}^{\!$
7	fn:length() ☑ Returns the number of items in a collection, or the number of characters in a string.
8	fn:replace() I Returns a string resulting from replacing in an input string all occurrences with a given string.
9	fn:split() ☐ Splits a string into an array of substrings.
10	fn:startsWith() ☐ Tests if an input string starts with the specified prefix.
11	fn:substring() ☑ Returns a subset of a string.
12	fn:substringAfter() ☐ Returns a subset of a string following a specific substring.
13	fn:substringBefore() ☐ Returns a subset of a string before a specific substring.
14	fn:toLowerCase() ☐ Converts all of the characters of a string to lower case.
15	fn:toUpperCase() ☐ Converts all of the characters of a string to upper case.
16	fn:trim() 🗗 Removes white spaces from both ends of a string.