

JSP & Servlets

JSTL

JSTL

JSTL

JSTL = JavaServer Pages Standard Tag Library

JSTL = verzameling libraries met **server-sided tags**.

Server-sided tags (in tegenstelling tot HTML tags) worden verwerkt door de webserver. De browser ziet ze nooit.

JSTL biedt nog **meer functionaliteit** aan JSP :

standaard functionaliteit:
enz....

iteratie, condities, internationalisatie,

eigen functionaliteit:

custom tags

Downloaden

JSTL tags zijn geschreven in Java en bevinden zich in:

- jstl1.2.jar

Hierin zitten de **tag-classes** en de **tag library descriptor** (TLD) vervat.

download link: <http://jstl.java.net>

of <http://tomcat.apache.org/taglibs/standard/>

(Apache Standard Taglib, een implementation van JSTL)

JSP Standard Tag Library

The current JSTL can be downloaded from the maven repositories:

- [JSTL API](#)
- [JSTL Implementation](#)



MAIN

[Home](#)
[License](#)
[Governance](#)

DOWNLOAD

[Download](#)

CONTRIBUTE

[Mailing Lists](#)
[Membership](#)

PROJECT INFO

[Source Control](#)
[Issue Tracking](#)
[Mailing Lists](#)
[JSP Specification](#)



Importeren van de jar bestanden

Kopieer de library naar de map
WebContent/WEB-INF/lib/ van
je project

URI associëren met een prefix

Herinner je nog de directives?

Page directive:

`<%@page ...%>`

Include directive:

`<%@include ... %>`

Taglib directive:

`<%@taglib .. %>`

Om taglibs te kunnen gebruiken → eerst definiëren bovenaan de jsp

Het definiëren van een taglib directive met een prefix:

`<%@taglib prefix="c"`

`uri="http://java.sun.com/jsp/jstl/core"%>`

URI associëren met een prefix

Definitie:

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

Het gebruik van een prefix:

Als je in een pagina een tag gebruikt uit de URI die je met een prefix associeerde, doe je dit zo:

```
<prefix:naamVanDeTag>
```

Een voorbeeld:

Het gebruik van de “set” tag (waarde toekennen)

```
<c:set var "a" value="5">    (een variabele a krijgt de waarde 5)
```


Tag names en tag URI's

Regels:

- Iedere JSTL tag heeft een **naam**,
 - set
 - foreach, if,...
- Iedere JSTL tag heeft een **URI** (of een associatie ernaar met een prefix, dat is in het voorbeeld hierboven die letter 'c'), bijvoorbeeld:
 - de uri <http://java.sun.com/jsp/jstl/core>
- Alle JSTL tags uit **één library** delen **dezelfde URI**
- Bij gebruik van een tag library, kan je de URI associëren met een zelfgekozen **prefix**. (= dus die 'c' in ons voorbeeld) Dit doe je met de taglib directive

JSTL Tag Libraries

| Taglib | Prefix | Omschrijving |
|------------------|--------|--|
| Core | c | De taglib met algemene (core) tags |
| Fmt | fmt | Taglib voor formatering van gegevens volgens I18N normen |
| Sql | sql | Taglib voor het gebruik van SQL |
| XML | x | Taglib voor het werken met XML bestanden |
| Functions | fn | Taglib met allerlei functies |

JSTL Core Tags: demo

JSTL Core

Algemene tags om

- gegevens in een document op te nemen
- variabelen te declareren en te gebruiken
- Itereren
- Conditie
- URL beheer

De absolute URI van deze Core Taglib is:

<http://java.sun.com/jsp/jstl/core>

De taglib directive:

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

<c:url>

De tag <c:url> helpt

- bepaalde problemen van relatieve/absolute URL's binnen je website op te lossen.
- een query string (met parameters) toe te voegen aan een URL

ALTERNATIEF

Er bestaat een alternatieve manier om correcte URL's te maken: de EL expressie **`${pageContext.servletContext.contextPath}`** geeft het context path van je website (/pizzamamamia).

Probeer nu de tag <c:url> uit op de oefening van daarnet

<c:import>

Toevoegen van kleine (gemeenschappelijke) jsp code, bijvoorbeeld een menu.

Opdracht

Maak een afzonderlijke jsp voor het menu aan en importeer deze in het vervolg in alle pagina's

<c:forEach>

Algemeen

Met **<c:forEach>** wordt over de elementen van een verzameling geïtereerd.

Syntax

```
<c:forEach var = "eenVariabele" items="${eenVerzameling}">
```

...

```
</c:forEach>
```

Itereren zonder verzameling

Met de attributen **begin** en **end** in een **forEach** tag maak je een lus zonder verzameling.

Voorbeeld:

4 keer itereren met behulp van een index:

```
<c:forEach var="index" begin="1" end="4">
```

In de volgende opdracht gaan we 5 keer een ster afbeelden

<c:forEach varstatus="status">

Je kunt aan de tag `forEach` een attribuut `varStatus` toevoegen.

Je vermeldt daarbij een vrij te kiezen variabelenaam

```
<c:forEach var="eenVar" items="{${eenVerzameling}" varStatus="status">
```

```
...  
</c:forEach>
```

Deze variabele bevat binnen de `forEach` iteratie de status van de iteratie.

Je leest deze status via enkele eigenschappen van deze variabele.

De meest gebruikte eigenschappen van de `varStatus` variabele

| eigenschap | betekenis |
|------------|---|
| count | aantal iteraties, te beginnen vanaf 1. |
| index | aantal iteraties, te beginnen vanaf 0. Als ook het attribuut <code>begin</code> is meegegeven, wordt de inhoud van <code>begin</code> bijgeteld. |
| first | true bij de eerste iteratie, false bij de volgende iteraties |
| last | true bij de laatste iteratie, false bij de vorige iteraties |

<c:forEach varstatus="status">

Je kunt `varStatus` gebruiken als je over een verzameling itereert, maar ook als je itereert zonder bijbehorende verzameling.

De volgende `forEach` itereert drie keer

```
<c:forEach begin="4" end="6" varStatus="status">
  ...
</c:forEach>
```

Bij de opeenvolgende iteraties hebben `status` eigenschappen volgende waarden

| <code>\${status.count}</code> | <code>\${status.index}</code> | <code>\${status.first}</code> | <code>\${status.last}</code> |
|-------------------------------|-------------------------------|-------------------------------|------------------------------|
| 1 | 4 | true | false |
| 2 | 5 | false | false |
| 3 | 6 | false | true |

Defining op <c:forEach>

oefening op <c:forEach>

- Maak een nieuwe servlet aan die de GET request naar “/personen” verwerkt.
- In deze servlet maak je een lijst van 10 personen aan. Vul telkens hun naam en adres in.
- Stuur de data door naar een view.
- In deze view toon je de gegevens van de vierde en de vijfde persoon

Nog enkele Core Tags

<c:if>

Syntax

<c: if test="voorwaardeUitgedruktAlsEExpressie> ... </c:if>

<c:choose>

Deze tag is vergelijkbaar met het switch statement uit Java

Syntax

```
<c:choose>
  <c:when test="${voorwaarde1}">
    Dit deel wordt enkel uitgevoerd als voorwaarde1 true is
  </c:when>
  <c:when test="${voorwaarde2}">
    Dit deel wordt enkel uitgevoerd als voorwaarde2 true is
  </c:when>
  <c:otherwise>
    Dit deel wordt enkel uitgevoerd als voorwaarde1 én voorwaarde2 false zijn.
    (vergelijkbaar met 'default' onderdeel van java switch)
  </c:otherwise>
</c:choose>
```

Oefening

oefening op

Herhaal de vorige oefening.

Zorg ervoor dat enkele personen in dezelfde gemeente wonen.

Druk in uw view eerst de personen van die gemeente af,
vermeld dan het aantal

Druk daaronder alle andere personen af
vermeld ook het aantal

<c:out>

Je mag sommige tekens niet letterlijk naar HTML output sturen.
Je moet ze vervangen door hun character entity code.

Voorbeelden

| teken dat je naar de HTML output wil sturen | character entity code die je naar de HTML output moet sturen |
|---|--|
| & | & |
| > | > |
| < | < |

<c:out> doet deze vertaalslag waar nodig.

Syntax

```
<c:out value="tekst_of_ELExpressie"/>
```

<c:out> stuurt de tekst of de EL expressie bij `value` naar de HTML output, en vertaalt tussendoor tekens naar bijbehorende character entity codes.

<c:set>

Algemeen

Wordt gebruikt om een variabele in te vullen. Deze kan dan in de rest van je JSP gebruikt worden.

Deze variabele heet een **page scope variabele**. De levensduur (scope) van de variabele is beperkt tot de pagina waarin de variabele werd aangemaakt

Syntax

```
<c: set var="naamVanDeVariabele" value="inhoud" />
```

Opmerking

Page directives (regels die beginnen met `<%@`)

+ oproepen van server-sided tags (`<c:set ..>`, `<c:forEach>`, ...)

veroorzaken lege regels in de HTML instructies die de JSP naar de browser stuurt.

Je kan dit vermijden met volgende aanpassing:

```
<%@page contentType="text/html"  
    pageEncoding="UTF-8"  
    session="false"  
    trimDirectiveWhitespaces = "true"
```

```
%>
```