

bx:containerloop

Das Tag `containerloop` listet Datensätze einer Datenliste.

In den unten aufgeführten Formen muss die ID eines vorhandenen Containers angegeben werden (`<id>`). Alternativ kann statt `<bx:containerloop pool="<id>"` auch `<bx:containerloop.<bez> definition="<id>"` verwendet werden. Hier muss dann ein Bezeichner (`<bez>`) angegeben werden, welcher in der Administration angezeigt wird. Ebenfalls wird hier die ID einer Container-Definition angegeben. In der Verwaltung wird dann eine Auswahlliste mit allen verfügbaren Containern angezeigt, die auf dieser Definition beruhen.

eine Anzahl an Elementen

```
<bx:containerloop pool="{id}" [force={single | list | random | choice}] [indexparam="{param}" | index="{param}" | forceindex="{n}"] [max="{n}"] [show="all"] [orderbyparam="[+|-]{param}" [directionparam="{param}"] | orderby="{feld}" [desc]] >
  Inhalt
</bx:containerloop>
```

Es werden die Listenelemente ab einer bestimmten Position ausgegeben, diese wird über ein Request-Parameter (`indexparam` oder `index`) festgelegt oder direkt dem Tag übergeben (`forceindex`). Standardmäßig werden alle Elemente ausgegeben, falls `max` angegeben ist, dann maximal nur `<n>`.

pool	enthält die Containerid bzw. den Containernamen
indexparam	enthält den Request-Parameter-Namen, der die Startposition für die Ausgabe enthält
index	enthält die Startposition für die Ausgabe
forceindex	feste Angabe für die Startposition der Ausgabe
max	maximal Anzahl ausgegebener DS im Schleifendurchlauf
show="all"	auch Anzeige von inaktiven Elementen
orderbyparam	Sortierungsfeldangabe über einen Request-Parameter (+ für aufsteigend und - für absteigend vor den Feldnamen) NEU: reagiert jetzt auch auf mehrfach vorkommende Request-Parameter (siehe Beispiel) (ab V2.7, siehe auch bei <code>containerfilter</code>)
directionparam	Sortierrichtung über einen Request-Parameter (wenn nicht bei <code>orderbyparam</code> vorangestellt)

pool	enthält die Containerid bzw. den Containernamen
orderby	Sortierfeld für die Anzeigereihenfolge (standardmäßig aufsteigend) NEU: Der Parameter orderby nimmt jetzt mehrere Feldnamen, getrennt durch Komma, Leerzeichen, Semikolon oder Pipe an und es kann über eine Einzelverknüpfung sortiert werden (siehe Beispiel) (ab V2.7)
desc	kehrt die Sortierreihenfolge um
force	list: Listenanzeige single: einzelner Datensatz random: Erzwingt Einzelanzeige eines Zufallsdatensatzes choice: Erzwingt Listenanzeige einer im Request übergebenen Auswahl von Datensätzen, braucht idfield=... (oder der Requestparameter heißt recordid)

Beispiele

Containeranzeige: Listen- und Detailseite

Listenseite

```
<form>
  Sortierung nach:<br>
  Titel Preis
</form>

<bx:containerloop pool="Artikel" indexparam="anzeigeAb" max="10" orderbyparam="sortierfeld"
directionparam="sortierrichtung" orderby="Titel">
  <a href="detail.htm?art=<bx:recorddata.id/>"><bx:recordfield.Titel/></a><br>
</bx:containerloop>
```

In der Liste werden pro Seite 10 Artikel angezeigt (max).

Detailseite

```
<bx:containerloop pool="{id}" [indexparam="{param}" | index="{param}" | forceindex="{n}"]
[max="{n}"] [show="all"] [orderbyparam="{+|-}{param}" [directionparam="{param}"] |
orderby="{feld}" [desc]] >
  Inhalt
</bx:containerloop>
```

mehrere Sortierparameter: ** (ab V2.7)**

Der Parameter `orderby` nimmt jetzt mehrere Feldnamen, getrennt durch Komma, Leerzeichen, Semikolon oder Pipe an. Dem Feldnamen kann man ein Pluszeichen für aufsteigende Sortierung (A..Z) oder ein Minuszeichen für absteigende Sortierung (Z..A) voranstellen.

zugehöriger Filter

```
<bx:containerloop orderby="-Datum,+Titel"> <!-- wenn kein + oder - vorangestellt ist, wird nach desc geschaut -->

<bx:containerloop orderby="-Datum,Titel" [asc]>
<bx:containerloop orderby="Datum,+Titel" desc>□□□□<!-- ergeben die gleiche Sortierung wie das Beispiel oben -->

<bx:containerloop orderby="-Jahr -Monat -Tag">
<bx:containerloop orderby="Jahr Monat Tag" desc>           <!-- ergeben auch beide die selbe Sortierung.
```

Sortierung über Einzelverknüpfung: (ab V2.7) Sortierung über Einzelverknüpfung ist auch für jedes Feld möglich:

zugehöriger Filter

```
<bx:containerloop orderby="Kategorie/Titel;-Preisgruppe/Nummer">
```

ein zufälliges Element

```
<bx:containerloop pool="<id>" force="random" [show="all"] >
  Inhalt </bx:containerloop>
```

id	Objekt-ID
Es wird ein zufälliges Element der Liste ausgewählt. Falls <code>show="all"</code> gesetzt ist, werden auch inaktive Elemente in Betracht gezogen.	

Beispiele

```
<bx:tag.feature param="value" />
```

ein bestimmtes Element

```
<bx:containerloop pool="<id>" force="single" [idfield="<param>"] >  
  Inhalt </bx:containerloop>
```

id	Objekt-ID
param	Name eines Request-Parameters
Die ID des gewünschten Datensatzes wird durch ein Request-Parameter (festgelegt durch <code>idfield</code> , Standard ist "recordid") übergeben.	

mehrere bestimmte Elemente

```
<bx:containerloop pool="<id>" force="choice" [idfield="<param>"] [id="<idlist>"]  
  [boundary="<text>"] [show="all"]  
  [oderbyparam="["+|-]<param>" [directionparam="<param>"] | orderby="<feld>" [desc]] >  
  Inhalt </bx:containerloop>
```

id	Objekt-ID
param	Name eines Request-Parameters
text	beliebiger Text
idlist	Liste von IDs
<code>idfield</code> gibt den Request-Parameter an, in welchem die Auswahl an Datensätzen steht. Es kann entweder eine Liste an IDs (getrennt durch Komma oder anderen Zeichen laut <code>boundary</code>) angegeben werden oder es wird der gleiche Request-Parameter (gleicher Name) mehrfach spezifiziert (für jeden gewünschten Datensatz einmal mit der entsprechenden Datensatz-ID). Alternativ kann die Liste der IDs im Attribut <code>id</code> übergeben werden.	

Um auch hier inaktive Elemente anzuzeigen, muss `show="all"` gesetzt sein.

Die Möglichkeiten zur Sortierung sind die selben, wie unter [eine Anzahl an Elementen](#)

Verweise auf diese Seite:

[bx:recorddata](#) [bx:tableloop](#) [bx:tableloop](#) [bx:recorddata](#)

(4 Verweise)