bx:loop

Mithilfe des loop-Tags ist es möglich, sich wiederholende Inhalte mit gleichem Aufbau zusammenzufassen und ggf. zu sortieren.

Dieses Tag ist veraltet. Stattdessen sollte bx:schleife genommen werden - alle Funktionen bleiben gleich, eben nur mit bx:schleife statt bx:loop davor.

Loop erzeugen

** Deprecated - Veraltetes Feature** Die hier aufgeführten Informationen beziehen sich auf eine veraltete Funktion, die nicht mehr weiterentwickelt wird und ggf. in zukünftigen Versionen nicht mehr unterstützt wird. Diese Funktion sollte deshalb nicht mehr benutzt werden.

```
<bx:loop [loop="{name}"] [orderby="{feldname}"] [showmax="{anzahl}"]> Inhalt </bx:loop>
```

Mit `loop` kann der Schleife ein Name gegeben werden (Standard: "defaultLoop") und mit `orderby` kann das Feld festgelegt werden, nach dem sortiert wird. Als Inhalt steht die Formatierung für jedes Element der Schleife. Es können dabei die folgenden Sub-Tags benutzt werden. Die maximale Anzahl an Elementen, die ausgegeben werden, wird über `showmax` festgelegt.

> Die folgenden Tags werden weiterhin in [bx:schleife](/books/cms-handbuch-entwickler/page/bx-schleife) genutzt

loop.count (ab v2.6.6. deprecated) --> neu: loop.index

```
<bx:loop.count [add="{anzahl}"] />
```

Hiermit lässt sich die aktuelle Position der Schleife ermitteln, also wieviele Wiederholungen schon erfolgt sind (z.B. für Nummerierung der Einträge, beginnt mit 0). Dem Ergebnis wird der Wert von "add" hinzuaddiert, sofern vorhanden.

loop.index (neu ab v2.6.6)

```
<bx:loop.index[add="{anzahl}"] />
```

Ist jetzt besser benannt. loop.count geht noch wegen Kompatibilität)

loop.id

```
<bx:loop.id />
```

Diese Funktion gibt die ID des aktuellen Datensatzes zurück.

loop.if

```
<bx:loop.if {first | notfirst | gerade | ungerade}> Inhalt </bx:loop.if>
```

Der Inhalt wird nur ausgewertet, wenn die entsprechende Bedingung zutrifft. Folgende Bedingungen können abgefragt werden:

- first: Datensatz ist der Erste der Liste
- notfirst: Datensatz ist nicht der Erste der Liste
- gerade: Datensatz ist der 2., 4., 6., ...
- ungerade: Datensatz ist der 1., 3., 5., ...

Prüfen auf letztes Element

```
<bx:schleife.Inhalte>
  <bx:clipboard.cut name="LetztesElement"><bx:loop.id/></bx:clipboard.cut>
</bx:schleife.Inhalte>

<bx:schleife.Inhalte>
  <bx:clipboard.cut name="AktuellesElement"><bx:loop.id/></bx:clipboard.cut>
  <bx:clipboard.if name="AktuellesElement" value="clipboard:LetztesElement">{wird ausgeführt
im Falle des letzten Elements}</bx:clipboard.if>
</bx:schleife.Inhalte>
```

Eine direkte Prüfung auf das letzte Element ist derzeit nicht möglich, kann aber mit einen Hilfs-Schleifendurchlauf und dem Clipboard simuliert werden.

loop.cols

```
<bx:loop.cols num="{anzahl der spalten}" equals="{welche spalte}" [not]> Inhalt
</bx:loop.cols>
```

Hier wird der Inhalt nur angezeigt, wenn sich der Datensatz in einer bestimmten Spalte der Ausgabe befindet.

Achtung! Die Anzahl der Spalten wird von 1 bis n angegeben, aber die Angabe der betreffenden Spalte wird von Null aus gezählt. z. B. <bx:loop.cols num="3" equals="1"> Inhalt </bx:loop.cols> spricht die zweite Spalte an.

`num` gibt die Spaltenanzahl und `equals` die gesuchte Spalte an. `not` kehrt die Bedingung um.

loop.previous / loop.next

Diese Tags erzeugen einen Link zum vorherigen bzw. nächsten Element. Der Linktext kann dabei geändert werden (Standard: "zurück" bzw. "weiter").

Nur auf Detailseiten möglich!

```
<bx:loop.previous [loop="{schleifenname}"] />
<bx:loop.previous [loop="{schleifenname}"]>Linktext</bx:loop.previous>
<bx:loop.next [loop="{schleifenname}"] />
<bx:loop.next [loop="{schleifenname}"]>Linktext</bx:loop.next>
```

Über den Parameter `loop` kann eine andere Schleife angesteuert werden.

loop.previousid / loop.nextid

Diese Tags geben die ID des vorherigen bzw. nächsten Datensatzes aus.

Nur auf Detailseiten möglich!

```
<bx:loop.previousid [loop="{schleifenname}"] />
<bx:loop.nextid [loop="{schleifenname}"] />
```

Über den Parameter `loop` kann eine andere Schleife angesteuert werden

loop.showprevious / loop.shownext

Mit `loop.count` lässt sich die aktuelle Position der Schleife ermitteln, also wieviele Wiederholungen schon erfolgt sind (z.B. für Nummerierung der Einträge).

```
<bx:loop.showprevious [schleife="{schleifenname}"] [hidden] [indexparam="{parametername}"] />
zurück </bx:loop.showprevious>
<bx:loop.shownext [schleife="{schleifenname}"] [hidden] [indexparam="{parametername}"] /> vor
</bx:loop.shownext>
```

Über den Parameter `schleife` kann eine andere Schleife angesteuert werden. `indexparam` gibt den Name des Request-Parameters an, welcher den aktuellen Index enthält. Mit `hidden` werden die Links ganz ausgeblendet, wenn sie nicht klickbar sind, ansonsten werden die Links ausgegraut.

loop.total

Dieses Tag gibt die Anzahl der in der Schleife angezeigten Datensätze aus. Es kann außerhalb der Schleife benutzt werden. (ab V2.6.6)

```
<bx:loop.total [object="{schleifenname}"] />
```

Über den Parameter `object` kann die jeweilige Schleife angesteuert werden.

loop.designtitle

Dieses Tag gibt den Titel des Designbausteins aus. (ab V2.6.6)

```
<bx:loop.total [object="{schleifenname}"] />
```

Wenn das umgebende Schleifentag keine Designschleife ist, wird eine Fehlermeldung ausgegeben.