bx:math

Das Tag `math` führt Berechnungen durch. Ab Version 2.5.9.

```
<bx:math pattern="..." locale="de" gs="." ds="," 
rounding="[ceiling|down|floor|half_down|half_even|half_up|up]">{Aufgabe}</bx:math>
```

pattern

Formatierung festlegen (Standard bei Zahl: "0.#####", bei Preis: "0.00"). Tausender-Trennzeichen (gs) und Dezimal-Trennzeichen (ds) können ebenfalls festgelegt werden.

locale

Sprachformatierung (z.B. "de" oder "en_US") `locale` sollte immer angegeben werden, da sonst die Standardeinstellung vom Server genutzt wird, die sich aber nach Updates oder Umzügen ändern kann.

Aufgabe

z.Zt. unterstützte Operatoren:

- `+` (Addition)
- `-` (Subtraktion)
- `*` (Multiplikation)
- `/` (Division)
- `^` (Hochrechnung)
- `%` (Modulo)

rounding

Rundungsmodus:

- **ceiling:** Rundet zur nächsten Ganzzahl auf (in Richtung +unendlich)
- **floor:** Rundet zur nächsten Ganzzahl ab (in Richtung -unendlich)
- **up:** Rundet zur nächsten Ganzzahl auf (aus Richtung null)
- **down:** Rundet zur nächsten Ganzzahl ab (in Richtung null)

- **half_down:** Rundet bei ,5 nach oben auf
- **half_even:** Rundet bei ,5 bei geraden Zahlen ab, bei ungeraden Zahlen auf
- **half_up:** Rundet bei ,5 nach unten ab

Resultat der jeweiligen Methode:

	UP	DOWN	CEILING	FLOOR	HALF_UP	HALF_DOWN	HALF_EVEN
5,5	6	5	6	5	6	5	6
2,5	3	2	3	2	3	2	2
1,6	2	1	2	1	2	2	2
1,1	2	1	2	1	1	1	1
1,0	1	1	1	1	1	1	1
-1,0	-1	-1	-1	-1	-1	-1	-1
-1,1	-2	-1	-1	-2	-1	-1	-1
-1,6	-2	-1	-1	-2	-2	-2	-2
-2,5	-3	-2	-2	-3	-3	-2	-2
-5,5	-6	-5	-5	-6	-6	-5	-6

Beispiele

```
<bx:math pattern="0.00">8*(-3+7)</bx:math>
```

gibt "32,00" aus

```
<bx:math pattern="#,##0.00"><bx:recordfield.Preis/>+<bx:recordfield.Versandkosten/></bx:math>
```

gibt einen Preis incl. Versandkosten aus

```
<bx:math pattern="0.00" locale="de">

<bx:recordfield.Anzahl/>*<bx:recordfield.Nettoeinzelpreis/>*(1+<bx:recordfield.Steuersatz/>/100)
</bx:math>
```

gibt einen Bruttopreis aus.

Debug

```
<bx:math.debug pattern="">{Aufgabe}</bx:math.debug>
```

gibt nicht nur das Ergebnis aus, sondern die gesamte Formel

Beispiele

```
<bx:math.debug pattern="0.00">8*(-3+7)</bx:math.debug>
```

gibt "calc{8*(-3+7)}=32.0" aus

Rechnen mittels Datenbank

```
<bx:math.mysql pattern="...">{Aufgabe}</bx:math.mysql>
```

Rechenoperationen werden mittels einer SQL-Formel gemacht.

Beispiele

```
<bx:math.mysql pattern="0">POW(3,4)</bx:math.mysql>
```

gibt "81" aus (34)

Vergleich von Zahlen

```
<bx:math.compare value1="{Wert1}" operator="{Operator}" value2="{Wert2}"> angezeigter Inhalt
</bx:math.compare>
```

Zeigt Inhalt abhängig vom Vergleich zweier Zahlenwerte aus.(ab V 2.5.9)

value1

erster Vergleichswert es können auch Präfixe verwendet werden

operator

Operator:

- `e` (gleich)
- `gt` (größer als)
- `lt` (kleiner als)
- `lte` (kleiner gleich)
- `gte` (größer gleich)

value2

zweiter Vergleichswert es können auch Präfixe verwendet werden

Beispiele

```
<bx:pagedata.setattribute mode="set" name="kathede1">3</bx:pagedata.setattribute>
<bx:pagedata.setattribute mode="set" name="kathede2">4</bx:pagedata.setattribute>
<bx:pagedata.setattribute mode="set" name="hypotenuse">5</bx:pagedata.setattribute>
<bx:clipboard.cut name="v1">
  <bx:pagedata.attribute name="kathede1"/>*<bx:pagedata.attribute name="kathede1"/>
  +<bx:pagedata.attribute name="kathede2"/>*<bx:pagedata.attribute name="kathede2"/>
</bx:clipboard.cut>
<bx:clipboard.cut name="v2">
  <bx:pagedata.attribute name="hypotenuse"/>*<bx:pagedata.attribute name="hypotenuse/>
</bx:clipboard.cut>
<bx:math.compare value1="clipboard:v1" operator="e"
value2="clipboard:v2">rechtwinklig</bx:math.compare>
```

Vergleiche mit min und max

```
<bx:math.min [attribute|clipboard]="{name}"> {Vergleichszahl} </bx:math.min><!-- Wenn der Ta
Body einer kleineren Zahl entspricht, wird das gespeicherte Attribut aktualisiert. -->
<bx:math.max [attribute|clipboard]="{name}"> {Vergleichszahl} </bx:math.max>           <!--
```

```
Wenn der Tag-Body einer größeren Zahl entspricht, wird das gespeicherte Attribut aktualisiert.
-->
<bx:math.[max|min] [attribute|clipboard]="{name}" extra-source="{text}" extra-
target="[attribute|clipboard]:{name2}"> {Vergleichszahl} </bx:math.max>
```

Vergleicht den in Request-Attribut oder Clipboard-Objekt gespeicherten Zahlenwert mit dem Zahlenwert aus dem Tag-Body und ändert gegebenenfalls den Attributwert. Sinnvoll in Schleifen und den Min- bzw. Max-Wert zu ermitteln.

attribute/clipboard

muß nicht zwangsläufig vorher vorhanden sein, kann aber, um einen Bestimmten Wert als Ausgangswert festzulegen

extra-source

wenn angegeben, wird zusätzlich bei jeder Änderung der Variablen der source-Text in das angegebenen target-Objekt gespeichert.

extra-target

Wohin der Text aus extra-source gespeichert werden soll (siehe Beispiel)

Beispiele

```
<bx:containerfilter.Fragen pool="Umfrage Elemente">
  <bx:math.max clipboard="maxSort"><bx:recordfield.Sortierung/></bx:math.max>
</bx:containerfilter.Fragen>
nächste Sortier-Position: <bx:math pattern="0"><bx:clipboard.paste
name="maxSort"/>+1</bx:math>
```

Innerhalb einer Schleife wird die größte Sortierungsposition ermittelt

```
<bx:containerfilter.Produkt pool="Produkte" idield="pid">
  ...
  <bx:recordfield.Varianzen>
    <bx:clipboard.cut name="größe"><bx:recordfield.groesse/></bx:clipboard.cut>
    <bx:math.min clipboard="preis" extra-source="clipboard:größe" extra-
```

```
target="clipboard:billigsteGröße"><bx:recordfield.NettoPreis/>*1.19</bx:math.min>
  <bx:recordfield.Varianzen>
  ...
  ab <bx:clipboard.paste name="preis"/> Euro (für Größe <bx:clipboard.paste
name="billigsteGröße"/>)
  ...
</bx:containerfilter.Produkt>
```

Die Preise der Varianzen (hier: Größen) eines Produktes werden durchgegangen. Bei jedem Durchlauf wird im Clipboard "größe" die jeweilige Größe gespeichert. Ist der Wert im Recordfield NettoPreis nun kleiner als der im Clipboard "preis" schon gespeicherte, wird der Wert dieses Clipboards überschrieben und der Wert vom Clipboard "größe" ins Clipboard "billigsteGröße" gespeichert (muß nicht vorher angelegt werden) und kann dann am Ende ausgegeben werden.

```
[<bx:clipboard.cut name="gesamtPreis"></bx:clipboard.cut>] <!-- wird angelegt, wenn noch nicht
vorhanden -->
<bx:containerfilter.Warenkorb pool="Warenkorb">
  <bx:math.add clipboard="gesamtPreis"><bx:recordfield.Preis/></bx:math.add>
</bx:containerfilter.Warenkorb >
Gesamtpreis: <bx:clipboard.paste name="gesamtPreis"/>
```

Addiert alle Preise im Warenkorb zusammen.

```
<bx:containerfilter.Warenkorb pool="Warenkorb">
  <bx:recordfield.MWST equals="7"><bx:math.add clipboard="7prozent"
value="1"></bx:math.add><bx:recordfield.MWST>
  <bx:recordfield.MWST equals="19"><bx:math.add clipboard="19prozent"
value="1"></bx:math.add><bx:recordfield.MWST>
</bx:containerfilter.Warenkorb >
Anzahl Artikel mit 7% Mwst: <bx:clipboard.paste name="7prozent"/>
Anzahl Artikel mit 19% Mwst: <bx:clipboard.paste name="19prozent"/>
```

Zählen in verschiedenen Kategorien

Hochzählen / Zusammenzählen

```
<bx:math.add [attribute|clipboard]="{name}"> {Wert} </bx:math.add>   <!-- addiert Werte
innerhalb einer Schleife auf -->
<bx:math.add (attribute|clipboard)="{name}" [value="1"] />□□   <!-- als geschlossenes Tag zum
```

```
Hochzählen mit dem Wert in value -->
```

Beispiele

```
<bx:clipboard.cut name="gesamtPreis"></bx:clipboard.cut>] <!-- wird angelegt, wenn noch nicht
vorhanden -->
<bx:containerfilter.Warenkorb pool="Warenkorb">
  <bx:math.add clipboard="gesamtPreis"><bx:recordfield.Preis/></bx:math.add>
</bx:containerfilter.Warenkorb >
Gesamtpreis: <bx:clipboard.paste name="gesamtPreis"/>
```

Addiert alle Preise im Warenkorb zusammen.

```
<bx:containerfilter.Warenkorb pool="Warenkorb">
  <bx:recordfield.MWST equals="7"><bx:math.add clipboard="7prozent"
value="1"></bx:math.add><bx:recordfield.MWST>
  <bx:recordfield.MWST equals="19"><bx:math.add clipboard="19prozent"
value="1"></bx:math.add><bx:recordfield.MWST>
</bx:containerfilter.Warenkorb >
Anzahl Artikel mit 7% Mwst: <bx:clipboard.paste name="7prozent"/>
Anzahl Artikel mit 19% Mwst: <bx:clipboard.paste name="19prozent"/>
```

Zählen in verschiedenen Kategorien

Formatieren von Zahlenwerten

```
<bx:math.format (attribute|clipboard)="{name}" pattern="#,##0.00" [ ds="," gs="." |
locale="de" ] [rounding="half_up"] />
```

Formatiert den Zahlenwert im Request-Attribut/Clipboard und gibt das Ergebnis aus.

Vergleich von Datumsen

```
<bx:math.datecompare prior="{Wert1}" later="{Wert}" [pattern="{pattern}"] > angezeigter Inhalt
</bx:math.datecompare>
```

Ein Vergleich der beiden übergebenen Datumswerte wird durchgeführt und der Inhalt nur ausgegeben, wenn der Wert prior zeitlich vor dem Wert later liegt.

prior

zu prüfender Datumswert

later

zu prüfender Datumswert

pattern

Eingangsformat von prior und later, Standard ist "dd.MM.yyyy".

Beispiele

```
<bx:clipboard.cut name="Halloween">31.10.2011</bx:clipboard.cut>
<bx:clipboard.cut name="Allerheiligen">1.11.2011</bx:clipboard.cut>
<bx:math.datecompare prior="clipboard:Halloween" later="clipboard:Allerheiligen">immer
angezeigt</bx:math.datecompare>
<bx:math.datecompare prior="clipboard:Allerheiligen" later="clipboard:Halloween">nie
angezeigt</bx:math.datecompare>
```

Datum modifizieren

```
<bx:math.modifydate pattern="{pattern}" locale="{de,en,fr,...}" (in-pattern="{Muster}" out-
pattern="{Muster}") (in-locale="{de,en,fr,...}" out-locale="{de,en,fr,...}")
(year|month|day|hour|minute|second)="[+|-]{Zahlenwert}" > {Datum} </bx:math.modifydate>
```

Eine Funktion, um ein im Body übergebenes Datum zu modifizieren und wieder auszugeben.

pattern

beide Muster sind gleich

in-pattern/out-pattern

Muster unterscheiden sich

in-locale/out-locale

Sprachen unterscheiden sich

Beispiele

```
<bx:math.modifydate in-pattern="yyyyMMdd" month="+1" day="15" out-pattern="dd. MMMM
yy">20110922</bx:math.modifydate>
```

gibt „15. Oktober 11“ aus

```
<bx:clipboard.copy><bx:tools.datum pattern="EEE, MMM d yyyy" locale="en"/></bx:clipboard.copy>
```

gibt aus und speichert im Clipboard “**Wed, Jan 3 2018**”

```
<bx:math.modifydate in-pattern="EEE, MMM d yyyy" out-pattern="EEEEEE, d. MMMMM yy" in-
locale="en" out-locale="fr"><bx:clipboard.paste/></bx:math.modifydate>
```

gibt aus “**mercredi, 3. janvier 18**”

```
<bx:math.modifydate in-pattern="EEE, MMM d yyyy" out-pattern="EEEEEE, d. MMMMM yy" locale="ru"
in-locale="en"><bx:clipboard/></bx:math.modifydate>
```

gibt aus “**среда**, 3. **января** 18**”

```
<bx:math.modifydate pattern="EEE, MMM d yyyy" in-locale="en" out-
locale="es"><bx:clipboard/></bx:math.modifydate>
```

gibt aus “**mié, ene 3 2018**”

```
<bx:math.modifydate in-pattern="EEE, MMM d yyyy" out-pattern="EEEEEE, d. MMMMM yy"
locale="en"><bx:clipboard/></bx:math.modifydate>
```

gibt aus “**Wednesday, 3. January 18**“

Feiertagsausgabe (ab V 2.3.)

```
<bx:math.holiday [inpattern="{pattern}"] [outpattern=("pattern" | "")]
[year|month|day|hour|minute|second)="[+|-]{Zahlenwert}"]> Datumsangabe im konfigurierten
Format </bx:math.holiday>
```

Ausgabe des Namens eines (Thüringer) Feiertags (z.B. Silvester). Das zu prüfende Datum wird im Tagbody angegeben. Leerzeichen an Anfang und Ende werden ignoriert. Ist das Datum kein Feiertag, wird der Wochentagsname ausgegeben (Standard) oder ein in `outpattern` definierter Inhalt.

inpattern

Eingangsformat (Standard ist "dd.MM.yyyy")

outpattern

Art des ausgegebenen Inhalts (siehe Beispiel) oder Wochentagsname (Standard)

Beispiele

```
<bx:math.holiday>30.12.2015</bx:math.holiday> □□<!-- gibt "Silvester" aus -->

<bx:math.holiday inpattern="yyyy:dd MMM">2015:31 Dez</bx:math.holiday> □□<!-- gibt "Silvester"
aus -->
<bx:math.holiday inpattern="yyyy:dd MMM">2015:30 Dez</bx:math.holiday> □□<!-- gibt "Mittwoch"
aus -->

<bx:math.holiday outpattern="">30.12.2015</bx:math.holiday> □□□□<!-- gibt "" (nichts) aus <bx:i
verwendbar, ABER: -->
<bx:math.holiday outpattern="">31.12.2015</bx:math.holiday> □□□□<!-- gibt "Silvester" aus <bx:i
verwendbar -->
```

auch Datums-Modifizierer möglich: `<bx:math.holiday day="+1">31.12.2015</bx:math.holiday>` gibt "Neujahr" aus

Feiertagsabfrage (ab V 2.3.)

```
<bx:math.isholiday testdate="{datum]" [pattern="pattern"] [not]>Inhalt</bx:math.isholiday>
```

Testet, ob das übergebenen Datum ein Feiertag ist und gibt den Taginhalt aus.

testdate

zu prüfendes Datum, über ein **Präfix** angegeben

pattern

Eingangsformat (Standard ist "dd.MM.yyyy")

not

kehrt die Bedingung um

Beispiele

```
<bx:math.isholiday testdate="request:pruefdatum" not pattern="dd.MM.yyyy">kein
Feiertag</bx:math.isholiday>
```

beim Aufruf von ...detail.htm?pruefdatum=30.12.2015 würde "kein Feiertag" ausgegeben

```
<bx:math.isholiday testdate="request:pruefdatum" pattern="dd.MM.yyyy">Feiertag:
<bx:math.holiday inpattern="dd.MM.yyyy"><bx:pagedata.request
name="pruefdatum"/></bx:math.holiday></bx:math.isholiday>
  wär das gleiche wie
<bx:if>Feiertag: <bx:math.holiday inpattern="dd.MM.yyyy" outpattern=""></bx:math.holiday>
</bx:if>
```

Zwei Arten, einen Feiertag abzufragen.

auch Datums-Modifizierer möglich: `<bx:clipboard><bx:tools.datum></bx:clipboard>` `<bx:math.isholiday testdate="clipboard:default" day="+1"> morgen ist ein Feiertag</bx:math.isholiday>`

Tagesdifferenz zwischen zwei Daten

```
<bx:math.datetolong data="{wert}">        // wandelt Datum in Millisekunden (seit 1970) und
<bx:math.datetolong> Wert </bx:math.datetolong>

<bx:math.longtodate>       //wandelt Millisekunden zu Datum
<bx:math.longtodate> Wert <bx:math.longtodate>
```

Die Angabe von `pattern` ist jeweils optional möglich (Standard ist "dd.MM.yyyy").

Beispiele

```
<bx:clipboard name="datum1">25.3.2017</bx:clipboard>
<bx:clipboard name="datum2">2.4.2017</bx:clipboard>

<bx:math>(<bx:math.datetolong data="clipboard:datum2"/>-<bx:math.datetolong><bx:clipboard
name="datum1"/></bx:math.datetolong>)/ 1000 / 3600 /24</bx:math>
```

Da wird „`7.958333333333333`“ Tage ausgegeben. Normalerweise glatte Zahl, aber hier ist Sommerzeitumstellung dazwischen. Wenn man bei `<bx:math pattern=”0”>` schreibt, rundet es auf 8 Tage.