

# bx:evaluate

Der Body dieses Tags `evaluate` wird ausgewertet und das Ergebnis ein zweites Mal bx-Tag-mäßig ausgewertet.

Ab v2.6.3 ist folgender Anwendungsfall möglich: Ein Konfigurationscontainer enthält ein (Quell-)Textfeld mit HTML + bx-Tags. Eine CM-Seite enthält einen Daten-Datensatz und darin eine Schleife über Datensätze des Konfigurationscontainers. Nun kann mit

`<bx:evaluate><bx:recordfield.*Quelltextfeld* type="plain"/>` eine Ausführung des bx-Codes bewirkt werden. Im Quelltext-Containerfeld können mit z. B. `<bx:recordfield.*Feldname* baseloop="*Datenschleife*"/>` Daten aus dem äußeren Datencontainer ausgegeben werden.

```
<bx:evaluate [charset="( utf-8 | iso-8859-1 )"]> Tag, das ein Tag zurückgibt </bx:evaluate>
```

charset

iso oder utf8 (wenn es Probleme gibt)

## Beispiele

Ein Tag steht im Container, z.B. die Formel zur Berechnung von Bezahlarten oder Versandkosten, die sich dynamisch gestalten. Damit wird die Auswahl über feste IDs innerhalb der Seite gespart.

### Batix-Code im Container

```
<bx:evaluate><bx:recordfield.QuellcodeSnippet type="plain" /></bx:evaluate>
```

Darauf achten, dass Benutzer keine willkürlichen Quelltexte in dem Containerfeld speichern können!

Ein Tag wird im Request übergeben: `test.htm?datum=`

### Batix-Code im Request (nicht empfohlen)

```
<bx:evaluate><bx:pagedata.request name="datum" /></bx:evaluate>
```

Dies ist gefährlich, da der Benutzer im Normalfall den Request-Parameter beliebig manipulieren und somit sämtlichen Batix-Code ausführen kann!