



# **VirthosTalk Referenzhandbuch**

Version 2.0  
Stand: 25.08.2006

© 2006

Virthos Systems GmbH

Contrescarpe 8c

28203 Bremen

Germany

E-Mail: [info@virthos.net](mailto:info@virthos.net)

[www.virthos.net](http://www.virthos.net)

Virthos® ist ein eingetragenes Warenzeichen  
der Virthos Systems GmbH, Deutschland.

# Inhaltsverzeichnis

Über dieses Handbuch.....	7
Allgemeine Syntax .....	9
Platzhalter und Anweisungen.....	9
Einbindung in HTML.....	9
Funktionsname und Parameter .....	10
Platzhalter als Parameter.....	10
Seitenangaben.....	11
Seitennummern.....	11
Pfadangaben .....	11
Typangaben .....	13
Spezielle Seiten.....	14
Welche Methode für welchen Zweck? .....	14
Zusammenfassung .....	15
Funktionsreferenz.....	17
(Freier Platzhalter).....	19
vtButton .....	21
vtCalc .....	22
vtComment .....	23
vtCountPages .....	24
vtCountSelection.....	25
vtCreationDate.....	26
vtCreationUser.....	27
vtCurrentAccessRights.....	28
vtCurrentDate .....	29
vtCurrentItem .....	30
vtCurrentPosition .....	31
vtCurrentTime .....	32

vtElse .....33

vtElseif .....34

vtEval .....35

vtExit.....36

vtFile.....37

vtFilename .....38

vtGet.....39

vtGetValue.....40

vtGlobal.....41

vtHeader .....42

vtID .....43

vtIf .....44

vtInclude.....46

vtLink.....47

vtLoop .....48

vtMedia .....51

vtModificationDate .....52

vtModificationUser .....53

vtName.....54

vtNext.....55

vtPackage .....56

vtPageType .....57

vtPath .....58

vtPathIDs .....59

vtPortions.....60

vtPostValue.....61

vtRedirect .....62

vtRepeat.....63

vtResult.....64

vtScript .....65

vtSelect .....66

---

vtSelf.....	70
vtSession.....	71
vtSet.....	72
vtUse.....	73
vtUser.....	74
vtUserAgent.....	75
vtUsername.....	76



# Über dieses Handbuch

Dieses Handbuch wendet sich an Personen, die Templates für das Content-Management-System Virthos gestalten wollen. Es beschreibt den Sprachumfang von VirthosTalk, der virthos-eigenen Template-Sprache, in allen Einzelheiten und ist in erster Linie als Nachschlagewerk gedacht. Um die Beschreibungen nachvollziehen zu können, werden allgemeine Kenntnisse in HTML und in der Gestaltung von Webseiten vorausgesetzt.

Ergänzt wird dieses Handbuch durch weitere Publikationen:

- die „Einführung in Virthos“, die neben einem Abriss der Installationsanleitung eine workshop-ähnliche Einführung in die wichtigsten Virthos-Funktionen enthält
- das „Virthos-Benutzerhandbuch“, das alle Aspekte von Virthos behandelt und eine Einführung in den Umgang mit Virthos-Templates enthält
- eine Reihe zusätzlicher Informationen, die ebenso wie alle anderen Publikationen unter [www.virthos.net](http://www.virthos.net) zum Herunterladen bereitstehen

Bevor Sie dieses Handbuch verwenden, sollten Sie erste Erfahrungen mit Virthos gesammelt und zumindest Teil 3 des Benutzerhandbuchs („Templates“) gelesen haben. Insbesondere die Kapitel über das Erstellen und Installieren von Templates sowie über den Umgang mit Templatevarianten werden hier als bekannt vorausgesetzt.

## Virthos Basic / Virthos Pro

Virthos wird in zwei Versionen angeboten: Virthos Basic und Virthos Pro. Dieses Handbuch gilt grundsätzlich für beide Versionen, allerdings stehen manche VirthosTalk-Sprachelemente nur in der „Pro“-Version zur Verfügung. Sie erkennen dies an den Hinweisen, die in der Einleitung zu jedem Sprachelement unter dem Stichwort „verfügbar in“ zu finden sind.

## Textkonventionen

Um die Texte leichter verständlich zu machen, werden in diesem Handbuch bestimmte Schriftarten verwendet, die jeweils für bestimmte Bedeutungen stehen:

- **Schreibmaschinenschrift** wird für die Wiedergabe von VirthosTalk-Elementen und HTML-Quelltext verwendet
- in *Kursivschrift* werden die Namen von VirthosTalk-Parametern dargestellt sowie die Namen von Dateien und Verzeichnissen





# Allgemeine Syntax

Damit die VirthosTalk-Sprachelemente richtig identifiziert und interpretiert werden können, müssen Sie sich beim Erstellen von Templates an eine strikt vorgegebene Schreibweise (Syntax) halten. Die folgenden Abschnitte beschreiben im Detail, welche Regeln einzuhalten sind.

## Platzhalter und Anweisungen

Die VirthosTalk-Funktionen lassen sich in zwei Gruppen einteilen: Platzhalter und Anweisungen. Zu den Platzhaltern werden alle Funktionen gezählt, die beim Verarbeiten des Templates (in der Regel) durch einen bestimmten Wert ersetzt werden. Alle anderen Funktionen gelten als Anweisungen, da ihr Zweck allein darin besteht, bestimmte Aktionen anzustoßen.

Die Unterscheidung zwischen Platzhaltern und Anweisungen dient in erster Linie dazu, das Verständnis für die Beschreibungen in diesem Handbuch zu erleichtern; Sie können aber auch ohne diese Unterscheidung erfolgreich mit VirthosTalk-Funktionen arbeiten.

## Einbindung in HTML

VirthosTalk-Elemente lassen sich an beliebige Stellen in den HTML-Code eines Templates einbinden, insbesondere auch in den Kopfbereich, in Stildefinitionen und in JavaScript-Funktionen. Damit ein VirthosTalk-Element als solches erkannt und richtig verarbeitet wird, müssen Sie es in doppelt geschweifte Klammern einschließen:

*beliebiger HTML-Code* `{{VirthosTalk-Element}}` *beliebiger HTML-Code*

Beim Verarbeiten wird das VirthosTalk-Element mitsamt den geschweiften Klammern aus dem HTML-Code entfernt und, falls es sich um einen Platzhalter handelt, durch einen anderen Wert ersetzt. Der umgebende HTML-Code wird dabei nicht verändert.

Wenn ein VirthosTalk-Element unmittelbar von HTML-Kommentarzeichen eingeschlossen ist, werden diese Kommentarzeichen beim Verarbeiten ebenfalls aus dem Quellcode entfernt:

*beliebiger HTML-Code* `<!--{{VirthosTalk-Element}}-->` *beliebiger HTML-Code*

Für die Funktionsweise eines Templates spielt es keine Rolle, ob ein VirthosTalk-Element in Kommentarzeichen eingeschlossen ist oder nicht; das Ergebnis der Verarbeitung ist in beiden Fällen gleich. Der HTML-Code bleibt aber übersichtlicher, wenn Sie Platzhalter grundsätzlich ohne und Anweisungen grundsätzlich mit Kommentarzeichen schreiben, so wie es auch in den Codebeispielen in diesem Handbuch durchgehend der Fall ist.

## Funktionsname und Parameter

Ein VirthosTalk-Element besteht mindestens aus einem Funktionsnamen, der manchmal durch zusätzliche Angaben (Parameter) ergänzt wird. Um deutlich zu machen, wo der Funktionsname endet und die Parameter beginnen, wird ein Doppelpunkt gesetzt:

```
{{Funktionsname: Parameter }}
```

Handelt es sich um mehr als einen Parameter, werden diese durch Komma voneinander getrennt:

```
{{Funktionsname: Parameter1, Parameter2, Parameter3 }}
```

Wenn eine Funktion viele mögliche Parameter besitzt, lassen sich diese mit eindeutigen Namen versehen:

```
{{Funktionsname: Name1="Wert1", Name2="Wert2", Name3="Wert3" }}
```

Die Werte können wahlweise in einfache oder doppelte Anführungszeichen eingeschlossen werden. Innerhalb einer Funktion dürfen Sie aber nicht beide Schreibweisen mischen.

## Platzhalter als Parameter

Den Wert eines Parameters können Sie in den meisten Fällen auch durch einen Platzhalter angeben. Bei einer solchen Verschachtelung wird der innere Platzhalter nur mit einfach geschweiften Klammern geschrieben:

```
{{Funktionsname: {Platzhaltername} }} oder
```

```
{{Funktionsname: Name1="{Platzhaltername}" }}
```

Wird ein unbenannter Parameter durch einen Platzhalter angegeben, muss hinter ihm wie im obigen Beispiel ein Leerzeichen eingefügt werden, damit nicht drei schließende Klammern aufeinanderfolgen.

# Seitenangaben

Bei manchen VirthosTalk-Funktionen, zum Beispiel `vtUse` und `vtLink`, ist es erforderlich, eine Seite als Parameter anzugeben. Dies lässt sich in VirthosTalk auf verschiedene Weise bewerkstelligen, wobei jede Methode mit unterschiedlichen Vor- und Nachteilen behaftet ist. Welche Möglichkeiten es gibt und wann Sie welche verwenden sollten, beschreibt dieses Kapitel.

## Seitennummern

Innerhalb von Virthos hat jede Seite eine eindeutige Nummer, die beim Erstellen der Seite automatisch vergeben wird und die stets gleich bleibt, auch wenn die Seite verschoben oder auf andere Weise verändert wird. Mit Hilfe der Nummer lässt sich eine Seite kurz und eindeutig bestimmen. Zum Beispiel wird mit

```
{{vtLink: 381}}
```

eine URL erzeugt, die auf Seite 381 verweist, ganz gleich, wo sich die Seite innerhalb der Virthos-Hierarchie befindet.

Statt die Seitennummer direkt anzugeben, können Sie auch einen Platzhalter verwenden, zum Beispiel

```
{{vtLink: {Kundennummer} }}
```

In diesem Fall würde eine URL erzeugt, die auf die Seite verweist, deren Nummer im Platzhalter „Kundennummer“ eingetragen ist.

## Pfadangaben

Seiten lassen sich auch durch einen Pfad angeben, das heißt durch ihre Position innerhalb der Virthos-Seitenhierarchie. Im Gegensatz zu Seitennummern muss eine Pfadangabe nicht notwendigerweise stets auf dieselbe Seite verweisen. Insbesondere das Verschieben und Umbenennen von Seiten kann dazu führen, dass Pfadangaben ihre Bedeutung ändern.

In Virthos wird zwischen absoluten und relativen Pfadangaben unterschieden. Ein absoluter Pfad geht immer von der Startseite oder der obersten Hierarchieebene aus, ein relativer Pfad von der aktuellen Seite.

## Absolute Pfade

Absolute Pfadangaben beginnen wahlweise auf der als Startseite festgelegten Seite oder auf der obersten Ebene der Seitenhierarchie. Beispielsweise erzeugt der Platzhalter

```
{{vtLink: /News/Aktuell }}
```

einen Verweis auf die Seite „Aktuell“, die der Seite „News“ untergeordnet ist, die wiederum der aktuellen Startseite untergeordnet ist. Der Platzhalter

```
{{vtLink: //Interna }}
```

verweist auf die Seite „Interna“, die sich auf der obersten Ebene der Virthos-Seitenhierarchie befindet.

## Relative Pfade

Relative Pfadangaben gehen stets von der aktuellen Seite aus und können in der Hierarchie sowohl aufwärts als auch abwärts weisen. Mit

```
{{vtLink: Kommentare }}
```

wird auf die Seite „Kommentare“ verwiesen, die der aktuellen Seite untergeordnet ist, und

```
{{vtLink: Kommentare/wichtig }}
```

erzeugt einen Verweis auf die Seite „wichtig“, die der Subseite „Hinweise“ untergeordnet ist. Auf untergeordnete Seiten können Sie auch mit Hilfe von Positionsnummern verweisen. So erzeugt

```
{{vtLink: #2}}
```

einen Verweis auf die zweite Subseite der aktuellen Seite, wobei die Zählung gemäß der in den Seitenvorgaben festgelegten Sortierfolge vorgenommen wird.

Auf die übergeordnete Seite (Containerseite) kann folgendermaßen verwiesen werden:

```
{{vtLink: .. }}
```

Und der Platzhalter

```
{{vtLink: ../.. }}
```

verweist auf die Containerseite der Containerseite.

Schließlich gibt es noch die Möglichkeit, auf eine übergeordnete Seite anhand ihrer Hierarchieebene Bezug zu nehmen. So verweist der Platzhalter

```
{{vtLink: \1 }}
```

auf diejenige Seite der obersten Hierarchieebene, die der aktuellen Seite übergeordnet ist (also die erste Seite innerhalb des Pfades der aktuellen Seite). Entsprechend verweist

```
{{vtLink: \2 }}
```

auf die zweite Seite innerhalb des Pfades der aktuellen Seite usw.

## Typangaben

Die dritte Möglichkeit, auf eine andere Seite Bezug zu nehmen, benutzt den Seitentyp als Hilfsmittel. Dies ist immer dann vorteilhaft, wenn es zwischen zwei Seitentypen eine eindeutige Beziehung gibt. Beispielsweise könnte es sein, dass eine Seite vom Typ „mitarbeiter“ stets einer Seite vom Typ „firma“ untergeordnet ist – in dem Fall können Sie innerhalb der Mitarbeiterseite mit

```
{{vtLink: \firma }}
```

auf die Firmenseite verweisen, und zwar unabhängig davon, ob die Firmenseite der Mitarbeiterseite unmittelbar übergeordnet ist oder ob sich noch andere Seiten (zum Beispiel vom Typ „niederlassung“ dazwischen befinden. Bei einer Typangabe durchsucht Virthos immer den gesamten Pfad der aktuellen Seite rückwärts, bis eine Seite des angegebenen Typs erreicht ist.

Zur Unterscheidung gegenüber Pfadangaben wird bei Typangaben kein Schrägstrich, sondern ein umgekehrter Schrägstrich (Backslash) als Startzeichen verwendet. Dieser soll noch einmal verdeutlichen, dass Typangaben innerhalb des Pfades stets rückwärts gewandt sind. Anders als Pfadangaben lassen sich Typangaben nicht verbinden, das heißt eine Angabe wie „\stadt\firma“ ist unzulässig.

## Spezielle Seiten

Neben den bisher genannten Seitenangaben gibt es in der „Pro“-Version von Virthos noch drei Sonderzeichen, mit denen sich Seiten angeben lassen, die zur aktuellen Seite in einer besonderen Beziehung stehen:

Sonderzeichen	steht für...
>	die nächste Seite innerhalb der aktuellen Liste
<	die vorige Seite innerhalb der aktuellen Liste
^	die Ursprungsseite der Liste

Mit dem Platzhalter `{{vtLink: > }}` können Sie beispielsweise einen Verweis auf die nächste Seite innerhalb der aktuellen Liste erzeugen. Damit dies funktioniert, muss die aktuelle Seite über einen Verweis aufgerufen werden, der sich innerhalb eines `vtLoop`-Blocks mit angegebenem `-store`-Parameter befindet. Ist dies nicht der Fall, wird ein `vtLink`-Platzhalter, der diese Sonderzeichen benutzt, durch ein Rautensymbol (#) ersetzt.

## Welche Methode für welchen Zweck?

Jede der Methoden, eine Seite anzugeben, hat ihre Vor- und Nachteile. Die folgenden Richtlinien sollen dabei helfen, die für den jeweiligen Einsatz geeignetste Methode herauszufinden.

- Im allgemeinen gilt die Empfehlung, Pfadangaben zu verwenden, wann immer dies möglich ist. Pfadangaben verhalten sich am ehesten so, wie Sie es als Webdesigner erwarten.
- Wenn ein klarer inhaltlicher Zusammenhang zwischen zwei Seitentypen besteht, das heißt eine klare Über-/Unterordnung, sollten Sie für Verweise von der unter- auf die übergeordnete Seite eine Typangabe verwenden.
- Für den Aufbau dynamischer Navigationsmenüs, die sich automatisch der Virthos-Seitenhierarchie anpassen, eignen sich relative Pfade mit Angabe von Hierarchieebenen am besten.
- Seitennummern sollten im allgemeinen nur in Verbindung mit anderen Platzhaltern verwendet werden, das heißt, wenn innerhalb einer Seite die Nummer einer anderen Seite als Eintrag gespeichert wurde, um eine feste Beziehung herzustellen.

## Zusammenfassung

Hier noch einmal alle Methoden für Seitenangaben im Überblick:

Methode	Beispiel
Seitennummer direkt	<code>{{vtLink: 381 }}</code>
Seitennummer indirekt	<code>{{vtLink: {Kundennummer} }}</code>
absoluter Pfad, von Startseite ausgehend	<code>{{vtLink: /News/Aktuell }}</code>
absoluter Pfad, von 1. Hierarchieebene ausgehend	<code>{{vtLink: //Interna/Memos }}</code>
relativer Pfad, untergeordnete Seite, nach Name	<code>{{vtLink: Hinweise }}</code>
relativer Pfad, untergeordnete Seite, nach Position	<code>{{vtLink: #2 }}</code>
relativer Pfad, übergeordnete Seite	<code>{{vtLink: .. }}</code>
relativer Pfad, Hierarchieebene	<code>{{vtLink: \2 }}</code>
Typangabe	<code>{{vtLink: \firma }}</code>
Spezialangabe	<code>{{vtLink: &gt; }}</code>





# Funktionsreferenz

Die folgenden Abschnitte beschreiben die verfügbaren VirthosTalk-Funktionen in allen Einzelheiten. Jeder Abschnitt ist nach folgendem Schema aufgebaut:

<b>Funktionsname</b>	
<i>Kurze Erläuterung der Funktion</i>	
Verwendung	<i>Schreibweise, in der die Funktion verwendet werden kann</i>
verfügbar in	<i>Virthos-Versionen, in denen diese Funktion verfügbar ist</i>
Parameter	<i>Erläuterung der verfügbaren Parameter</i>
<i>Ausführliche Erläuterungen und Beispiele</i>	
<a href="#">Siehe auch</a> <i>Funktionen, deren Beschreibungen weitere wichtige Informationen enthalten</i>	

Um ein schnelles Nachschlagen zu ermöglichen, sind die Funktionen in den folgenden Abschnitten alphabetisch sortiert. Wenn Sie nicht genau wissen, wie eine bestimmte Funktion heißt, können Sie in der Übersicht auf der folgenden Seite nachschauen; dort sind die wichtigsten Virthos-Talk-Platzhalter und Anweisungen nach inhaltlichen Kriterien gruppiert.

## Übersicht über die wichtigsten Platzhalter und -Funktionen

### Editierbare Bereiche

- (Freie Platzhalter)
- vtFile
- vtFilename
- vtMedia

### Seiteneigenschaften

- vtName, vtID
- vtCreationDate
- vtModificationDate
- vtCreationUser
- vtModificationUser

### Benutzer & Zugriffsrechte

- vtUser
- vtUsername
- vtCurrentAccessRights

### Kontrollstrukturen

- vtIf, vtElse, vtElseif
- vtLoop, vtNext, vtExit
- vtRepeat
- vtUse
- vtScript

### Listen & Menüs

- vtLoop, vtNext, vtExit
- vtPortions
- vtPath
- vtLink
- vtID, vtPathIDs
- vtName

### Selektionen

- vtSelect
- vtEval

### Variablen & Werte

- vtCurrentDate
- vtCurrentTime
- vtSet, vtGet
- vtCalc, vtResult
- vtGetValue, vtPostValue
- vtSession, vtGlobal

### Verschiedenes

- vtHeader, vtRedirect
- vtInclude
- vtUserAgent

## (Freier Platzhalter)

Wird durch den Text ersetzt, der beim Erstellen einer Seite für diesen Platzhalter eingegeben wurde.

Verwendung	<code>{{Platzhaltername: Codierung}}</code>
Version	Basic, Pro (ab 1.0)
Parameter	<i>Codierung</i> Codierung, die auf den Text angewendet werden soll

Ein freier Platzhalter ist ein Platzhalter, dessen Name frei vergeben wird und der dazu dient, einen editierbaren Textblock zu erzeugen. Beispiele sind `{{Headline}}` oder `{{Inhalt}}`.

Für die Namensgebung von freien Platzhaltern gelten folgende Regeln:

- Der Name darf nicht mit `vt` beginnen.
- Der Name darf keine Leerzeichen enthalten.
- Der Name darf keine Satz- oder Sonderzeichen enthalten. Einzige Ausnahmen sind Bindestrich (-) und Unterstrich (\_).
- Der Name darf keine Umlaute oder fremdsprachige Buchstaben enthalten.
- Der Name darf nicht mit einer Zahl beginnen. Innerhalb des Namens sind Zahlen dagegen erlaubt.
- Die Groß-/Kleinschreibung ist bedeutungslos. Die Namen `headline`, `Headline` und `HEADLINE` bezeichnen aus Virthos-Sicht alle denselben Platzhalter.

Beim Erstellen oder Ändern einer Seite wird der Textinhalt eines Platzhalters so gespeichert, wie er vom Benutzer eingegeben wurde. Beim Abrufen des Inhalts lässt sich eine Codierung auf den Text anwenden:

Codierung	Bedeutung
<code>html</code>	Zeilenumbrüche werden durch <code>&lt;br&gt;</code> -Tags ersetzt, Umlaute und Sonderzeichen durch entsprechende HTML-Entitäten (z. B. „ä“ durch <code>&amp;auml;</code> );
<code>styled</code>	Wie "html", aber zusätzlich werden Internet- und E-Mail-Adressen in <code>&lt;a&gt;</code> -Tags eingeschlossen sowie diverse Auszeichnungen in HTML-Code umgesetzt (s. u.)
<code>raw</code>	Der Text wird unverändert gelassen

Wenn nichts anderes angegeben ist, wird automatisch die `html`-Codierung verwendet. Die `raw`-Codierung ist hilfreich, wenn in einem Platzhalter bereits fertiger HTML-Code gespeichert ist, sowie in mehrzeiligen Textfeldern von HTML-Formularen:

```
<textarea name="Beschreibung">{{Beschreibung: raw}}</textarea>
```

Bei Verwendung der `raw`-Codierung ist zu beachten, dass ein gewisses Sicherheitsrisiko dadurch entsteht, dass ein Benutzer reinen HTML-Code als Inhalt eingeben kann. In Bereichen, in denen anonyme Benutzer Eingaberechte haben (z. B. in Diskussionsforen oder Gästebüchern), sollte diese Option daher mit Vorsicht gebraucht werden.

Bei Verwendung der `styled`-Codierung kann der Redakteur bestimmte Auszeichnungen in der Text einfügen. Folgende Auszeichnungsmöglichkeiten stehen zur Verfügung:

Auszeichnung	wird umgesetzt in
<code>_<b>beliebiger Text</b>_</code>	<code>&lt;strong&gt;beliebiger Text&lt;/strong&gt;</code>
<code>_<i>beliebiger Text</i>_</code>	<code>&lt;i&gt;beliebiger Text&lt;/i&gt;</code>
<code>_<u>beliebiger Text</u>_</code>	<code>&lt;u&gt;beliebiger Text&lt;/u&gt;</code>
<code>_<code>beliebiger Text</code>_</code>	<code>&lt;var&gt;beliebiger Text&lt;/var&gt;</code>
<code>_<code>beliebiger Text</code>_</code>	<code>&lt;code&gt;beliebiger Text&lt;/code&gt;</code>
<code>ein_Wort ((Internetadresse))</code>	<code>&lt;a href="Internetadresse" target="_blank"&gt;ein_Wort&lt;/a&gt;</code>
<code>ein_Wort ((vt:Seite))</code>	<code>&lt;a href="Virthos-URL"&gt;ein_Wort&lt;/a&gt;</code>
<code>ein_Wort ((js:Javascript))</code>	<code>&lt;a href="javascript:void()" onclick="Javascript"&gt;ein_Wort&lt;/a&gt;</code>
- beliebiger Text	<code>&lt;/p&gt;&lt;ul&gt;</code>
- noch mehr Text	<code>&lt;li&gt;beliebiger Text&lt;/li&gt;</code>
...	<code>&lt;li&gt;noch mehr Text&lt;/li&gt;</code>
	<code>...</code>
	<code>&lt;/ul&gt;&lt;p&gt;</code>

Internet- und E-Mail-Adressen werden (ohne besondere Auszeichnung) automatisch in anklickbare Verweise umgesetzt. Damit Virthos eine Internet-Adresse als solche erkennt, muss diese allerdings entweder mit `www.` beginnen oder vollständig notiert werden, also beginnend mit `http://`, `https://` oder `ftp://`. E-Mail-Adressen werden zusätzlich auf eine Weise verschlüsselt, die es Spam-Robotern erschwert, diese Adressen auszuspähen.

Verweise, die in doppelte Klammern eingeschlossen sind, werden immer „hinter“ das davorstehende Wort gelegt. Sollen mehrere Wörter hinterlegt werden, müssen diese durch geschützte Leerzeichen (Mac: Alt-Leertaste, PC: Alt-0160) getrennt werden.

Ein `styled`-Platzhalter sollte immer in `<p>`-Tags eingeschlossen werden, um bei der Verwendung von Aufzählungen eine korrekte HTML-Syntax sicherzustellen.

## vtButton

Wird durch einen Bearbeitungsbutton ersetzt, wenn der aktuelle Benutzer über die erforderlichen Zugriffsrechte verfügt.

Verwendung	<code>{{vtButton: Methode}}</code>
verfügbar in	Basic, Pro (ab 1.3)
Parameter	<i>Methode</i> Name der Methode, die über den Button aufgerufen werden soll

Mit `vtButton` lassen sich kleine Buttons in eine Seite einbinden, mit denen der Redakteur direkt in den Bearbeiten- oder Strukturmodus wechseln, neue Seiten anlegen oder bestehende Seiten löschen kann. Die folgenden Methoden stehen zur Auswahl:

- `edit`    Seite im Bearbeiten-Modus aufrufen
- `sort`    Seite im Sortieren-Modus aufrufen
- `trash`   Seite in den Papierkorb legen
- `new`    neue untergeordnete Seite erstellen (mit vorgegebenem Template)
- `add`    neue untergeordnete Seite erstellen (mit Template-Auswahl)

Bei Verwendung der `new`-Methode muss als zweiter Parameter das gewünschte Template angegeben werden, zum Beispiel

```
{{vtButton: new, artikel}}
```

Ein Button mit der `add`-Methode verhält sich exakt wie der Button „Neue Seite“ in der Werkzeugleiste des Virthos-Managers, das heißt, im Normalfall wird eine Liste der verfügbaren Templates eingeblendet, und der Redakteur kann das gewünschte Template auswählen. Wenn nur ein Template in Frage kommt (durch eine `vtLoop`-Anweisung mit vorgegebenem Seitentyp), wird die Template-Auswahl übersprungen.

Falls der aktuelle Benutzer nicht die nötigen Zugriffsrechte zum Bearbeiten der aktuellen Seite besitzt, wird kein Button eingeblendet. Dies gilt insbesondere für Gastbenutzer, die sich (noch) nicht bei Virthos angemeldet haben.

Siehe auch  
`vtLoop`

## vtCalc

Erlaubt das Berechnen von Werten mittels PHP-Funktionen.

Verwendung	<code>&lt;!--{{vtCalc: <i>Ausdruck</i>}}--&gt;</code>
Version	Basic, Pro (ab 1.3)
Parameter	<i>Ausdruck</i> der zu berechnende PHP-Ausdruck

Mit `vtCalc` lassen sich mathematische, Zeichenketten-, Datums- und Uhrzeit- sowie alle andere Arten von PHP-Funktionen ausführen, um ein Ergebnis zu berechnen. Innerhalb der Formel dürfen VirthosTalk-Platzhalter benutzt werden – sie werden vor der Berechnung durch ihre jeweiligen Werte ersetzt. Das Ergebnis der Berechnung lässt sich mit `vtResult` ausgeben.

Feste Zahlenwerte müssen innerhalb von `vtCalc`, entsprechend der PHP-Syntax, mit Punkt statt mit Komma notiert werden. Platzhalter, die als Zeichenketten verwendet werden sollten in Anführungszeichen gesetzt werden.

Eine vollständige Dokumentation aller PHP-Funktionen befindet sich unter [www.php.net/manual](http://www.php.net/manual).

### Beispiel 1

```
<!--{{vtCalc: round( {Nettopreis} * 1.16, 2 ) }}-->
<p>Der Bruttopreis beträgt: <b>{{vtResult}} Euro</b></p>
```

### Beispiel 2

```
<!--{{vtCalc: substr( '{Nachname}', 0, 1 ) }}-->
<p>Ihr Nachname beginnt mit "{{vtResult}}"</p>
```

Siehe auch  
`vtResult`

## vtComment

Wird durch den Inhalt des Kommentarfeld der aktuellen Seite ersetzt.

Verwendung	<code>{{vtComment}}</code>
verfügbar in	Basic, Pro (ab 1.0)
Parameter	–

Virthos erlaubt es, in den Seitenvorgaben einen Kommentar zu jeder Seite zu speichern. Gedacht ist dies in erster Linie für interne Memos, beispielsweise für Anmerkungen des Webdesigners oder Hinweise des Administrators. Mit Hilfe dieses Platzhalters lässt sich der Kommentar aber auch in eigene Templates einbinden.

## vtCountPages

Wird durch die Anzahl der Seiten ersetzt, die der aktuellen Seite direkt oder indirekt untergeordnet sind.

Verwendung	<code>{{vtCountPages: Tiefe}}</code>
verfügbar in	Basic, Pro (ab 1.2)
Parameter	<i>Tiefe</i> Anzahl der Hierarchiestufen, die beim Zählen berücksichtigt werden sollen (optional)

Die Tiefe muss als ganze Zahl angegeben werden. Folgende Werte sind möglich:

- 0 Es werden sämtliche Seiten gezählt, die der aktuellen Seite untergeordnet sind, unabhängig von ihrer Hierarchiestufe.
  - 1 Es werden nur die unmittelbar untergeordneten Seiten gezählt (Standard).
  - 2 Es werden die unmittelbar untergeordneten Seiten und deren Subseiten gezählt.
  - 3 Wie 2, aber auch die Subseiten der Subseiten werden mitgezählt.
- etc.

[Siehe auch](#)

`vtCountSelection`



## vtCountSelection

Wird innerhalb eines vtSelect-Blocks durch die Zahl der Seiten ersetzt, die zu der Auswahl gehören.

Verwendung	{{vtCountSelection}}
verfügbar in	Pro (ab 1.3)
Parameter	–

### Beispiel

```
<!--{{vtSelect: -filter="{vtName}.bw.A" }}-->
  <!--{{vtIf: {vtCountSelection} .eq. 0 }}-->
    <p>Es liegen keine Einträge für den Anfangsbuchstaben A vor.</p>
  <!--{{vtElse}}-->
    <p>Es liegen {{vtCountSelection}} Einträge vor.</p>
  <!--{{vtEndIf}}-->
<!--{{vtEndSelect}}-->
```

### Siehe auch

vtSelect, vtCountPages, Kapitel „Selektionen“

## vtCreationDate

Wird durch eine Datum-/Uhrzeit-Angabe ersetzt, die beschreibt, wann die aktuelle Seite erstellt wurde.

Verwendung	<code>{{vtCreationDate}}</code>
verfügbar in	Pro (ab 1.0)
Parameter	–

Anders als `vtCurrentDate` liefert `vtCreationDate` keine reine Datumsangabe zurück, sondern einen Text in der Form „23.05.2006, 12:34 Uhr“. Dieser Platzhalter lässt sich daher nicht in `vtIf`- oder `vtSelect`-Anweisungen für Datumsvergleiche benutzen.

[Siehe auch](#)

`vtModificationDate`

## vtCreationUser

Wird durch den Namen des Benutzers ersetzt, der die aktuelle Seite erstellt hat.

Verwendung	<code>{{vtCreationUser}}</code>
verfügbar in	Pro (ab 1.3)
Parameter	–

Wenn eine Seite erstellt wird, speichert Virthos den Namen des aktuellen Benutzers zusammen mit dem Erstellungsdatum in den Seitenvorgaben ab. Wenn in den Benutzereinstellungen ein vollständiger Name hinterlegt ist, wird dieser verwendet, ansonsten der Anmeldename des Benutzers. Der Name bleibt auch dann erhalten, wenn die Benutzereinstellungen nachträglich verändert oder gelöscht werden.

Siehe auch

`vtModificationUser`

## vtCurrentAccessRights

Wird durch eine Liste der Zugriffsrechte ersetzt, über die der aktuelle Benutzer im Hinblick auf die aktuelle Seite verfügt.

Verwendung	<code>{{vtCurrentAccessRights}}</code>
verfügbar in	Basic, Pro (ab 1.3)
Parameter	–

Dieser Platzhalter dient in erster Linie dazu, Bedingungen innerhalb einer `vtIf`-Anweisung zu formulieren. Auf diese Weise lassen sich bestimmte Seitenelemente ausblenden, falls der aktuelle Benutzer nicht die nötigen Zugriffsrechte besitzt.

Die möglichen Zugriffsrechte sind:

<code>view</code>	der Benutzer darf die aktuelle Seite ansehen
<code>update</code>	der Benutzer darf die Inhalte der aktuellen Seite verändern
<code>add</code>	der Benutzer darf neue Subseiten zur aktuellen Seite hinzufügen oder vorhandene entfernen

### Beispiel

```
<!--{{vtIf: {vtCurrentAccessRights}.cn.add }}-->  
  <a href="{{vtLink:-met='vtAdd'}}">neuen Eintrag erstellen</a>  
<!--{{vtEndIf}}-->
```

### Siehe auch

`vtIf`

## vtCurrentDate

Wird durch das aktuelle Systemdatum des Webservers ersetzt.

Verwendung	<code>{{vtCurrentDate}}</code>
verfügbar in	Basic, Pro (ab 1.0)
Parameter	–

Das Datum wird in „deutscher Schreibweise“ ausgegeben, also in der Form „tt.mm.jjjj“.

### Beispiel

```
<!--{{vtIf: {vtCurrentDate} .bw. 24.12. }}-->
  Fröhliche Weihnachten!
<!--{{vtEndIf}}-->
```

### Siehe auch

`vtCurrentTime`

## vtCurrentItem

Wird innerhalb eines `vtRepeat`-Blocks durch den aktuellen Listeneintrag ersetzt.

Verwendung	<code>{{vtCurrentItem}}</code>
verfügbar in	Basic, Pro (ab 1.2)
Parameter	–

Beispiele für die Verwendung dieses Platzhalters finden sich in den Erläuterungen zur `vtRepeat`-Anweisung.

Siehe auch  
`vtRepeat`

## vtCurrentPosition

Wird innerhalb eines vtLoop-Blocks durch die Nummer des aktuellen Schleifendurchlaufs ersetzt.

Verwendung	{{vtCurrentPosition}}
verfügbar in	Basic, Pro (ab 1.2)
Parameter	–

Mit vtCurrentPosition lassen sich Einträge in dynamischen Listen durchnummerieren. In Verbindung mit einer vtIf-Anweisung ist es auch möglich, beispielsweise jeden zweiten Eintrag einer solchen Liste andersfarbig hervorzuheben.

### Beispiel

```
<table>
  <!--{{vtLoop}}-->
  <tr>
    <td>{{vtCurrentPosition}}</td>
    <td>{{Name}}</td>
    <td>{{Telefon}}</td>
  </tr>
  <!--{{vtEndLoop}}-->
</table>
```

### Siehe auch

vtLoop

## vtCurrentTime

Wird durch die aktuelle Systemzeit des Webservers ersetzt.

Verwendung	<code>{{vtCurrentTime}}</code>
verfügbar in	Basic, Pro (ab 1.0)
Parameter	–

Die Systemzeit wird ohne Sekundenangabe in der Schreibweise „hh:mm“ ausgegeben.

### Beispiel

```
<!--{{vtIf: {vtCurrentTime} .gt. 18:00 }}-->
  <h1>Guten Abend!</h1>
<!--{{vtElse}}-->
  <h1>Guten Tag!</h1>
<!--{{vtEndIf}}-->
```

### Siehe auch

`vtCurrentDate`



## vtElse

Leitet innerhalb eines `vtIf`-Blocks den HTML-Code ein, der ausgegeben wird, wenn die angegebene Bedingung *nicht* erfüllt ist.

Verwendung	<code>&lt;!--{{vtElse}}--&gt;</code>
Version	Basic, Pro (ab 1.0)
Parameter	–

Erläuterungen und Beispiele finden Sie bei der Beschreibung von `vtIf`.

Siehe auch

`vtIf`, `vtElseIf`

## vtElseIf

Ermöglicht innerhalb eines `vtIf`-Blocks eine weitere Fallunterscheidung.

Verwendung	<code>&lt;!--{{vtElseIf: <i>Bedingung</i>}}--&gt;</code>
Version	Basic, Pro (ab 1.2)
Parameter	<i>Bedingung</i> Bedingung, unter der der nachfolgende HTML-Code ausgegeben wird

Für das Formulieren der Bedingung gelten dieselben Regeln wie bei `vtIf` (siehe dort).

Siehe auch

`vtIf`, `vtElse`

## vtEval

Wird innerhalb eines `vtSelect`-Blocks durch das Ergebnis einer statistischen Auswertung ersetzt.

Verwendung	<code>{{vtEval: Methode, Platzhalter}}</code>
verfügbar in	Pro (ab 1.3)
Parameter	<p><i>Methode</i> die gewünschte Auswertungsmethode</p> <p><i>Platzhalter</i> Name des (freien) Platzhalters, auf den sich die Auswertung bezieht</p>

Für den Platzhalter *Methode* stehen drei mögliche Werte zur Verfügung:

- `sum` addiert die Inhalte des Platzhalters über alle ausgewählten Seiten hinweg
- `min` ermittelt den kleinsten Wert innerhalb der Auswahl
- `max` ermittelt den größten Wert innerhalb der Auswahl

### Beispiel

```
<!--{{vtSelect: -type="zeugnis"}}-->
  Beste Note: {{vtEval: min, Note}}<br>
  Schlechteste Note: {{vtEval: max, Note}}
<!--{{vtEndSelect}}-->
```

### Siehe auch

`vtSelect`, `vtCountSelection`

## vtExit

Bricht eine `vtLoop`- oder `vtRepeat`-Schleife vorzeitig ab.

Verwendung	<code>&lt;!--{{vtExit}}--&gt;</code> oder <code>&lt;!--{{vtExit: <i>Bedingung</i>}}--&gt;</code>
Version	Basic, Pro (ab 1.2)
Parameter	<i>Bedingung</i> Bedingung, unter der die Schleife abgebrochen wird

In der einfachen Form ohne Parameter führt `vtExit` zu einer sofortigen Beendigung der gerade durchlaufenen Schleife. Wird eine *Bedingung* angegeben, erfolgt der Ausstieg aus der Schleife nur, wenn die Bedingung zutrifft. Für die Formulierung der Bedingung gelten dieselben Regeln, wie bei `vtIf`.

### Beispiel 1

```
<p>Nur den ersten Eintrag ausgeben.</p>
<ul>
<!--{{vtLoop}}-->
  <li>{{vtName}}</li>
  <!--{{vtExit}}-->
<!--{{vtEndLoop}}-->
</ul>
```

### Beispiel 2

```
<p>Die ersten fünf Einträge ausgeben.</p>
<ul>
<!--{{vtLoop}}-->
  <li>{{vtName}}</li>
  <!--{{vtExit: {{vtCurrentPosition}}.eq.5}}-->
<!--{{vtEndLoop}}-->
</ul>
```

### Siehe auch

`vtCurrentPosition`, `vtIf`, `vtLoop`, `vtRepeat`

## vtFile

Wird durch eine URL ersetzt, mit der sich eine zur aktuellen Seite gehörende Download-Datei herunterladen lässt.

Verwendung	<code>{{vtFile: Name}}</code>
Version	Basic, Pro (ab 1.0)
Parameter	<i>Name</i> interner Name für den Dateiplatzhalter

Wird in einem Template dieser Platzhalter verwendet, bietet Virthos im Bearbeiten-Modus automatisch die Möglichkeit, eine Datei hochzuladen. Sollen mehrere Dateien hochladbar sein, können Sie den Platzhalter entsprechend oft in dem Template plazieren, mit jeweils unterschiedlichen Namen.

### Beispiel

```
<!--{{vtIf: {vtFile: Dateil} .neq. }}-->
  <a href="{{vtFile: Downloaddateil}}">jetzt herunterladen</a>
<!--{{vtElse}}-->
```

### Siehe auch

`vtFilename`, `vtMedia`

## vtFilename

Wird durch den Namen einer herunterladbaren Datei ersetzt.

Verwendung	<code>{{vtFilename: Name}}</code>
Version	Basic, Pro (ab 1.0)
Parameter	<i>Name</i> interner Name für den Dateiplatzhalter

Im Gegensatz zu `vtFile` liefert `vtFilename` keinen vollständigen Pfad, sondern nur den Namen der betreffenden Download-Datei, beispielsweise zum Einbinden in eine Hyperlink- oder Button-Beschriftung.

### Beispiel

```
<a href="{{vtFile: Datei2}}">  
    jetzt "{{vtFilename:Datei2}}" herunterladen  
</a>
```

### Siehe auch

`vtFile`

## vtGet

Wird durch den Wert einer mittels `vtSet` gesetzten Variable ersetzt.

Verwendung	<code>{{vtGet: Variable}}</code>
Version	Basic, Pro (ab 1.3)
Parameter	<i>Variable</i> Name einer Variablen

### Beispiel

```
<!--{{vtUse:...}}-->  
  <!--{{vtSet: Container="{vtName}"}}-->  
<!--{{vtEndUse}}-->  
  
<p>Die übergeordneten Seite heißt "{{vtGet:Container}}".</p>
```

Siehe auch

`vtSet`

## vtGetValue

Wird durch einen Parameterwert aus der aktuellen URL ersetzt.

Verwendung	<code>{{vtGetValue: Element}}</code>
Version	Basic, Pro (ab 1.1)
Parameter	<i>Element</i> Name des Werts, der eingefügt werden soll.

Mit `vtGetValue` lassen sich gezielt einzelne Elemente der aktuellen URL auslesen, sowohl virthos-spezifische Elemente wie `-pg` oder `-met` als auch eigene Ergänzungen, die in einen Hyperlink eingebunden sind oder die durch das Absenden eines Formulars mit der „get“-Methode erzeugt werden.

### Beispiel

```
<!--{{vtIf: {vtGetValue: -act} .eq. create }}-->
  Neuer Eintrag wurde erstellt.
<!--{{vtEndIf}}-->
```

### Siehe auch

`vtPostValue`



## vtGlobal

Wird durch den Wert einer globalen Variablen ersetzt.

Verwendung	<code>{{vtGlobal: Name}}</code>
Version	Basic, Pro (ab 1.2)
Parameter	<i>Name</i> Name der globalen Variablen, deren Wert eingefügt werden soll

Mit `vtGlobal` lassen sich Werte ausgeben, die mit Hilfe der Konfigurationsdatei *config.php* oder mit Hilfe einer Einstellungsseite gesetzt worden sind. Informationen zu den beiden Möglichkeiten, globale Variablen zu setzen, finden Sie im Benutzerhandbuch.

### Beispiel

```
<p>Unsere Hotline-Nummer: {{vtGlobal: Hotlinenumber}}.</p>
```

## vtHeader

Erlaubt es, eine Zeile zum HTTP-Header der aktuellen Seite hinzuzufügen.

Verwendung	<code>&lt;!--{{vtHeader: Zeile}}--&gt;</code>
Version	Basic, Pro (ab 1.0)
Parameter	<i>Zeile</i> Textzeile, die zum HTTP-Header hinzugefügt werden soll

Die `vtHeader`-Anweisung ist nur für fortgeschrittene Anwender gedacht. Mit ihrer Hilfe lässt sich beispielsweise der MIME-Typ einer Seite verändern. Auch das Umlenken auf eine andere Adresse ist möglich, allerdings steht dazu auch die leichter anwendbare `vtRedirect`-Anweisung zur Verfügung.

Siehe auch

`vtRedirect`

## vtID

Wird durch die interne Nummer der aktuellen Seite ersetzt.

Verwendung	{{vtID}}
Version	Basic, Pro (ab 1.0)
Parameter	–

Jede Seite innerhalb von Virthos erhält beim Erstellen eine eindeutige Nummer, die automatisch fortlaufend vergeben wird. Diese Nummer lässt sich nachträglich nicht ändern. Sie dient dazu, eine Seite eindeutig anzusprechen, und kann insbesondere beim Aufbau dynamischer Navigationsmenüs helfen, die aktuelle Seite im Menü typografisch hervorzuheben.

### Beispiel

```
<!--{{vtSet: ThisPage="{{vtID}}"}}-->
<!--{{vtUse:..}}-->
  <!--{{vtLoop}}-->
    <!--{{vtIf: {{vtID}} .eq. {{vtGet:ThisPage}}}}-->
      <strong>{{vtName}}<br>
    <!--{{vtElse}}-->
      <a href="{{vtLink}}">{{vtName}}</a><br>
    <!--{{vtEndIf}}-->
  <!--{{vtEndLoop}}-->
<!--{{vtEndUse}}-->
```

Dieses Beispiel erzeugt eine Liste aller Seiten, die der Containerseite der aktuellen Seite untergeordnet sind. Die aktuelle Seite selbst ist in der Liste fett hervorgehoben, die übrigen Seiten sind als Hyperlinks markiert.

Siehe auch

**vtPathIDs**

## vtf

Gibt den nachfolgenden HTML-Code nur aus, wenn eine bestimmte Bedingung erfüllt ist.

Verwendung	<pre> &lt;!--{{vtIf: Bedingung}}--&gt; ... &lt;!--{{vtEndIf}}--&gt; oder &lt;!--{{vtIf: Bedingung}}--&gt; ... &lt;!--{{vtElse}}--&gt; ... &lt;!--{{vtEndIf}}--&gt; oder &lt;!--{{vtIf: Bedingung}}--&gt; ... &lt;!--{{vtElseIf: Bedingung}}--&gt; ... &lt;!--{{vtElseIf: Bedingung}}--&gt; ... &lt;!--{{vtElse}}--&gt; ... &lt;!--{{vtEndIf}}--&gt; </pre>
Version	Basic, Pro (ab 1.0)
Parameter	<i>Bedingung</i> Bedingung, unter der der nachfolgende HTML-Code ausgegeben wird.

Die `vtIf`-Anweisung, in Verbindung mit `vtElse` und `vtElseIf`, erlaubt es, einem Template ein hohes Maß an Flexibilität zu verleihen. Der HTML-Code passt sich dann automatisch an bestimmte Gegebenheiten an, die durch formulierte Bedingungen genau beschrieben sind.

Eine Bedingung wird nach dem Schema

*linker\_Vergleichswert* . *Vergleichsoperator* . *rechter\_Vergleichswert*

aufgebaut. Die Vergleichswerte können durch feste Werte oder Platzhalter (in einfachen geschweiften Klammern) angegeben werden. Wenn der rechte Vergleichswert durch einen Platzhalter angegeben wird, ist darauf zu achten, dass die schließende Klammer dieses Platzhalters durch ein Leerzeichen von der schließenden Klammer der `vtIf`-Anweisung getrennt wird. Die Leerzeichen zwischen den Vergleichswerten und den Punkten des Vergleichsoperators dienen nur der besseren Lesbarkeit, sind aber nicht zwingend.

Soll ein Datum als Vergleichswert dienen, muss es in der Form „tt.mm.jjjj“ angegeben werden, eine Uhrzeit in der Form „hh:mm“. Wenn der Stunden- oder Minutenwert dabei kleiner als 10 ist, muss jeweils eine Null vorangestellt werden; bei Tages- und Monatswerten ist eine vorangestellte Null nicht erforderlich.

Bei Vergleichen von Zeichenketten ignoriert Virthos die Groß-/Kleinschreibung. Die Bedingung `{Name}.eq.meier` wäre also auch erfüllt, wenn im Platzhalter `{Name}` das Wort „Meier“ oder „MEIER“ stünde.

Die folgende Tabelle zeigt, welche Vergleichsoperatoren zur Verfügung stehen:

Operator	Bedeutung	Erläuterung
<code>.eq.</code>	equals	ist gleich
<code>.neq.</code>	not equals	ist ungleich
<code>.gt.</code>	greater than	(Zahl, Datum oder Uhrzeit) ist größer als
<code>.gte.</code>	greater than or equal	(Zahl, Datum oder Uhrzeit) ist größer als oder gleich
<code>.lt.</code>	less than	(Zahl, Datum oder Uhrzeit) ist kleiner als
<code>.lte.</code>	less than or equal	(Zahl, Datum oder Uhrzeit) ist kleiner als oder gleich
<code>.bw.</code>	begins with	(Zeichenkette) beginnt mit
<code>.ew.</code>	ends with	(Zeichenkette) endet mit
<code>.cn.</code>	contains	(Zeichenkette) enthält
<code>.ncn.</code>	not contains	(Zeichenkette) enthält nicht

### Beispiel 1

```
<!--{{vtIf: {vtCountPages} .eq. 0 }}-->
  <p>keine Einträge vorhanden</p>
<!--{{vtEndIf}}-->
```

### Beispiel 2

```
<!--{{vtIf: {Geburtsdatum}.eq.{vtCurrentDate} }}-->
  <p>Herzlichen Glückwunsch zum Geburtstag, {{Name}}!</p>
<!--{{vtElse}}-->
  <p>Guten Tag, {{Name}}</p>
<!--{{vtEndIf}}-->
```

### Beispiel 3

```
<!--
  {{vtIf: {Prioritaet}.eq.1 }}
    {{vtSet: Text="sehr dringend"}}
  {{vtElseIf: {Prioritaet}.eq.2}}
    {{vtSet: Text="dringend"}}
  {{vtElseIf: {Prioritaet}.eq.3}}
    {{vtSet: Text="normal"}}
  {{vtElseIf: {Prioritaet}.eq.4}}
    {{vtSet: Text="kann warten"}}
  {{vtElse}}
    {{vtSet: Text="unwichtig"}}
  {{vtEndIf}}
-->
```

## vtInclude

Wird durch den Inhalt einer externen Textdatei ersetzt.

Verwendung	<code>{{vtInclude: Dateipfad}}</code>
Version	Basic, Pro (ab 1.3)
Parameter	<i>Dateipfad</i> relativer Pfad der einzubindenden Datei (ausgehend vom Speicherort der Template-Datei)

Mit `vtInclude` lassen sich HTML-Codebestandteile, die in mehreren Templates benötigt werden (zum Beispiel Kopfdaten, Navigationselemente o.ä.), in eigenständige Dateien auslagern. Dies erleichtert die Pflege der Templates, weil Änderungen an der Auslagerungsdatei sofort in allen Templates verfügbar sind. Außerdem werden die Templates selbst übersichtlicher.

Eine mit `vtInclude` eingebundene Datei darf wiederum VirthosTalk-Platzhalter oder -Anweisungen, insbesondere auch weitere `vtInclude`-Anweisungen enthalten. Diese werden in dem Kontext interpretiert, in dem der `vtInclude`-Befehl steht.

### Beispiel

```
<!--{{vtSet: PageTitle="Impressum"}}-->
<!--{{vtInclude: includes/page_header.html }}-->
```

In der Auslagerungsdatei *page\_header.html* lässt sich mit `{{vtGet:PageTitle}}` der Wert einbinden, der mit `vtSet` gesetzt wurde, beispielsweise um das `<title>`-Element des HTML-Kopfbereichs mit Inhalt zu füllen.

## vtLink

Wird durch eine URL ersetzt, die auf eine Seite innerhalb von Virthos verweist.

Verwendung	<pre> {{vtLink: Seite}} {{vtLink: Seite, Methode}} {{vtLink: -pg="Seite", -met="Methode", -lg="Sprache", -act="Aktion"}} </pre>	
Version	Basic, Pro (ab 1.2)	
Parameter	<i>Seite</i>	Seitenangabe
	<i>Methode</i>	Name der Methode, mit der die Seite aufgerufen werden soll
	<i>Sprache</i>	Landessprache, in der die Seite dargestellt werden soll (nur Pro)
	<i>Aktion</i>	Aktion, mit der die übergebenen Werte verarbeitet werden sollen

Dieser Platzhalter ist dafür gedacht, innerhalb eines HTML-Ankerelements oder einer Formularaktion plaziert zu werden. Der `-lg`-Parameter steht erst seit Version 2.0 zur Verfügung. Neben den oben genannten Parametern lassen sich beliebige weitere Parameter übergeben. Diese werden nach dem Schema „parameter1=wert1 &parameter2=wert2“ etc. mit in die URL eingebunden.

Die *Seite* kann als Seitennummer oder als Pfad angegeben werden (siehe Kapitel „Seitenangaben“). Wenn die Seite des aktuellen Kontexts gemeint ist, kann der Parameter ganz weggelassen werden. Die *Methode* kann entfallen, wenn die Standardmethode („`vtview`“) gemeint ist.

Die *Sprache* wird in Form eines Kürzels angegeben, das in den Systemeinstellungen hinterlegt sein muss. Die Angabe ist nur erforderlich, wenn die Sprache gewechselt werden soll. Wenn Sie diesen Parameter weglassen, wird die aktuelle Sprache beibehalten. Die Angabe einer *Aktion* ist erforderlich, wenn `vtLink` im `action`-Attribut eines `<form>`-Elements verwendet wird. Nähere Einzelheiten finden Sie im Benutzerhandbuch im Kapitel „Formularverarbeitung“.

### Beispiel 1

```
<p>Klicken Sie <a href="{{vtLink: /Kontakte}}">hier</a>, um Kontakt mit uns aufzunehmen.</p>
```

### Beispiel 2

```
<form action="{{vtLink: -met="feedback", -act="mail"}}" method="post">
```

### Beispiel 3

```
<a href="{{vtLink: -lg="en"}}">English Version</a>
```

### Siehe auch

`vtSelf`, Kapitel „Seitenangaben“

## vtLoop

Wiederholt den nachfolgenden HTML-Code für alle Subseiten der aktuellen Seite oder (innerhalb eines `vtSelect`-Blocks) für alle Seiten, die zur aktuellen Auswahl gehören.

Verwendung	<pre>&lt;!--{{vtLoop}}--&gt; ... &lt;!--{{vtEndLoop}}--&gt; oder &lt;!--{{vtLoop: Seitentyp}}--&gt; ... &lt;!--{{vtEndLoop}}--&gt; oder &lt;!--{{vtLoop: -type="Seitentyp",-show="Anzeige",                -store="Name", -size="Länge"}}--&gt; ... &lt;!--{{vtEndLoop}}--&gt;</pre>	
Version	Basic, Pro (ab 1.0)	
Parameter	<i>Seitentyp</i>	beschränkt die Wiederholung auf Seiten eines bestimmten Typs (nur zulässig, wenn sich der <code>vtLoop</code> -Block <i>nicht</i> innerhalb eines <code>vtSelect</code> -Blocks befindet)
	<i>Anzeige</i>	beschränkt die Wiederholung auf Seiten, die bestimmten Kriterien entsprechen (siehe Erläuterungen)
	<i>Name</i>	Name, unter der die Seitenliste gespeichert werden soll, um ein einfaches Vor- und Zurückblättern zu ermöglichen
	<i>Länge</i>	maximale Zahl von Wiederholungen

Mit `vtLoop` lassen sich dynamische Listen aufbauen, das heißt Listen, die sich beim Hinzufügen und Entfernen von Subseiten automatisch anpassen. Alle innerhalb des `vtLoop`-Blocks verwendeten Platzhalter werden im Kontext der gerade durchlaufenen Seite interpretiert. Auch ein Verschachteln von `vtLoop`-Blöcken ist möglich.

Die Reihenfolge, in der die Subseiten durchlaufen werden, entspricht der Sortierfolge, die in den Seitenvorgaben eingestellt ist. Wenn die `vtLoop`-Anweisung innerhalb eines `vtSelect`-Blocks steht, wird die Sortierfolge verwendet, die in der `vtSelect`-Anweisung durch die Parameter `-sortfield` und `-sortorder` vorgegeben ist.

Grundsätzlich durchläuft `vtLoop` alle Subseiten der aktuellen Seite bzw. alle Seiten der aktuellen Auswahl, für die der aktuelle Benutzer ein Leserecht besitzt. Mit den optionalen Parametern können Sie das Durchlaufen der Subseiten allerdings eingrenzen bzw. erweitern.



Der `-show`-Parameter kann einen der folgenden Werte erhalten:

Anzeige	Bedeutung
<code>all</code>	es werden alle Seiten berücksichtigt
<code>visible</code>	es werden nur „sichtbare“ Seiten berücksichtigt (eine Seite ist sichtbar, wenn in den Seitenvorgaben die Option „Seite nicht in Menüs und Sitemap anzeigen“ deaktiviert ist)
<code>viewable</code>	es werden nur „sichtbare“ und „darstellbare“ Seiten berücksichtigt (eine Seite ist darstellbar, wenn ein Template zu dem betreffenden Seitentyp existiert)

Wenn Sie den `-show`-Parameter weglassen, verhält sich `vtLoop`, als hätten Sie für `-show` den Wert `visible` angegeben.

Die Angabe eines *Seitentyps* hat – abgesehen von der Eingrenzung der zu durchlaufenden Subseiten – noch eine weitere Auswirkung: Beim Erstellen einer neuen Subseite wird die Template-Auswahl übersprungen und sofort das Erstellenformular für den angegebenen Seitentyp eingeblendet. Dies funktioniert selbst dann, wenn zu dem angegebenen Seitentyp überhaupt kein Template existiert. Das Erstellenformular stellt dann für alle freien Platzhalter, die sich innerhalb des `vtLoop`-Blocks befinden, entsprechende Eingabefelder zur Verfügung. Gibt es mehrere nebengeordnete (nicht verschachtelte) `vtLoop`-Anweisungen mit Seitentypangabe, wird beim Erstellen einer neuen Seite eine Auswahl mit den verwendeten Seitentypen eingeblendet.

Wird der `-store`-Parameter angegeben, speichert Virthos die mit `vtLoop` durchlaufene Seitenliste unter dem angegebenen Namen und ermöglicht im weiteren Verlauf das Vor- und Zurückblättern zwischen den einzelnen Seiten mit Hilfe der speziellen Seitenangaben „<“ und „>“ (vgl. Kapitel „Seitenangaben“).

Wenn Sie mit `-size` eine Obergrenze für die Zahl der Wiederholungen festlegen, bricht die Schleife nach Erreichen dieser Zahl ab. Sie können die Schleife an der Abbruchstelle fortsetzen, indem Sie die aktuelle Seite erneut aufrufen und dabei in die URL den Parameter `-prt=2` einbinden. Wird die Schleife dabei noch immer nicht vollständig durchlaufen, kann man mit `-prt=3`, `-prt=4` etc. die jeweils folgenden „Portionen“ aufrufen. Am einfachsten geschieht dies mit Hilfe des Platzhalters `vtPortions`, bei dem dieses Prinzip ausführlicher erläutert ist.

### Beispiel 1

```
<!--{{vtLoop: -show="viewable"}}-->
  <p>{{Headline}}<br>{{Intro}} <a href="{{vtLink}}">mehr</a></p>
<!--{{vtEndLoop}}-->
```

## Beispiel 2

```
<table>
  <tr>
    <th>Name</th>
    <th>Vorname</th>
    <th>Telefon</th>
  </tr>
  <!--{{vtLoop: -type="kontakt", -store="Kontaktliste"}}-->
  <tr>
    <td><a href="{{vtLink}}">{{Name}}</a></td>
    <td>{{Vorname}}</td>
    <td>{{Telefon}}</td>
  </tr>
  <!--{{vtEndLoop}}-->
</table>
```

### Siehe auch

`vtCurrentPosition`, `vtExit`, `vtNext`, `vtSelect`, Kapitel „Seitenangaben“

## vtMedia

Wird durch die URL einer Bilddatei ersetzt, die für die aktuelle Seite hochgeladen wurde.

Verwendung	<code>{{vtMedia: Bildname}}</code>
Version	Basic, Pro (ab 1.3)
Parameter	<i>Bildname</i> Name des Bildplatzhalters (kann frei gewählt werden)

Der `vtMedia`-Platzhalter ist für die Verwendung in `<img>`-Elementen gedacht. Wird er in einem Template platziert, bietet Virthos im Bearbeitenmodus automatisch ein entsprechendes Formular-element, um ein Bild hochzuladen. Sollen mehrere Bilder verwendet werden, können Sie `vtMedia` entsprechend oft im Template platzieren, mit jeweils unterschiedlichen Bildnamen.

Solange kein Bild hochgeladen wurde, wird der Platzhalter durch einen Verweis auf ein unsichtbares Bild („clear.gif“) ersetzt. Auf diese Weise wird verhindert, dass der Webbrowser ein „zerbrochenes Bildsymbol“ einblendet, wenn kein Bild existiert.

### Beispiel 1

```

```

### Beispiel 2

```
<!--{{vtIf: {vtMedia:Bild2} .cn. clear.gif}}-->
  <p>kein Bild vorhanden</p>
<!--{{vtElse}}-->
  
<!--{{vtEndIf}}-->
```

### Siehe auch

`vtFile`

## vtModificationDate

Wird durch eine Datum-/Uhrzeit-Angabe ersetzt, die beschreibt, wann die aktuelle Seite zuletzt bearbeitet wurde.

Verwendung	{{vtModificationDate}}
Version	Basic, Pro (ab 1.0)
Parameter	–

Anders als `vtCurrentDate` liefert `vtModificationDate` keine reine Datumsangabe zurück, sondern einen Text in der Form „23.05.2006, 12:34 Uhr“. Dieser Platzhalter lässt sich daher nicht in `vtIf`- oder `vtSelect`-Anweisungen für Datumsvergleiche benutzen.

[Siehe auch](#)

`vtCreationDate`

## vtModificationUser

Wird durch den Namen des Benutzers ersetzt, der die aktuelle Seite zuletzt bearbeitet hat.

Verwendung	<code>{{vtModificationUser}}</code>
verfügbar in	Pro (ab 1.3)
Parameter	–

Wenn eine Seite bearbeitet wird, speichert Virthos den Namen des aktuellen Benutzers zusammen mit dem Änderungsdatum in den Seitenvorgaben ab. Wenn in den Benutzereinstellungen ein vollständiger Name hinterlegt ist, wird dieser verwendet, ansonsten der Anmeldename des Benutzers. Der Name bleibt auch dann erhalten, wenn die Benutzereinstellungen nachträglich verändert oder gelöscht werden.

[Siehe auch](#)

`vtCreationUser`

## vtName

Wird durch den internen Namen der aktuellen Seite ersetzt.

Verwendung	<code>{{vtName}}</code>
Version	Basic, Pro (ab 1.0)
Parameter	–

Jede Seite innerhalb von Virthos besitzt einen Namen, der aus den ersten Zeichen des Textinhalts gebildet oder frei vergeben wird. Mit `vtName` lässt sich dieser Name beispielsweise in dynamischen Navigationsmenüs ausgeben.

### Beispiel

```
<!--{{vtUse:/}}-->
  <!--{{vtLoop}}-->
    <a href="{{vtLink}}">{{vtName}}</a><br>
  <!--{{vtEndLoop}}-->
<!--{{vtEndUse}}-->
```

## vtNext

Wechselt innerhalb eines `vtLoop`-Blocks schon vor Erreichen von `vtEndLoop` zur nächsten Seite.

Verwendung	<code>&lt;!--{{vtNext}}--&gt;</code>
Version	Basic, Pro (ab 1.2)
Parameter	–

Mit Hilfe von `vtNext` lassen sich dynamische Listen erzeugen, die sich über mehrere Spalten erstrecken, beispielsweise Bildergalerien, bei denen die Liste aus mehreren Reihen besteht, die jeweils mehrere Bilder enthalten.

### Beispiel

```
<table>
  <!--{{vtLoop}}-->
  <tr>
    <td>{{Name}}, {{Vorname}}</td>
    <!--{{vtNext}}-->
    <td>{{Name}}, {{Vorname}}</td>
    <!--{{vtNext}}-->
    <td>{{Name}}, {{Vorname}}</td>
  </tr>
  <!--{{vtEndLoop}}-->
</table>
```

### Siehe auch

`vtLoop`

## vtPackage

Wird durch den Namen des Templatepakets ersetzt, zu dem das Template der aktuellen Seite gehört.

Verwendung	<code>{{vtPackage}}</code>
Version	Basic, Pro (ab 1.0)
Parameter	–

Besonders hilfreich ist `vtPackage`, wenn innerhalb einer `vtLoop`-Schleife bestimmte Arten von Seiten auf spezielle Weise behandelt werden sollen:

### Beispiel

```
<ul>
<!--{{vtLoop}}-->
  <!--{{vtIf: {{vtPackage}}.eq.shop}}-->
    <li><a href="{{vtLink}}">{{vtName}}</a></li>
  <!--{{vtEndIf}}-->
<!--{{vtEndLoop}}-->
</ul>
```

*Hiermit wird eine Liste aufgebaut, die nur Seiten enthält, deren Template zum Templatepaket „shop“ gehört.*

### Siehe auch

`vtPageType`



## vtPageType

Wird durch den Seitentyp bzw. den Namen des Templates ersetzt, auf dem die aktuelle Seite basiert.

Verwendung	<code>{{vtPageType}}</code>
Version	Basic, Pro (ab 1.0)
Parameter	–

Besonders hilfreich ist `vtPageType`, wenn innerhalb einer `vtLoop`-Schleife bestimmte Arten von Seiten auf spezielle Weise behandelt werden sollen. Der Seitentyp entspricht dem Namen des Templates, auf dem die Seite basiert, allerdings ohne die Dateiendung „.html“.

### Beispiel

```
<ul>
<!--{{vtLoop}}-->
  <!--{{vtIf: {vtPageType}.eq.produktgruppe}}-->
    <li><a href="{{vtLink}}">Gruppe "{{vtName}}"</a></li>
  <!--{{vtElseIf: {vtPageType}.eq.artikel}}-->
    <li><a href="{{vtLink}}">{{Artikelnummer}}</a>
      {{Bezeichnung}}</li>
  <!--{{vtEndIf}}-->
<!--{{vtEndLoop}}-->
</ul>
```

*Hiermit wird eine Liste aufgebaut, die nur Seiten vom Typ „produktgruppe“ oder „artikel“ berücksichtigt. Je nach Seitentyp wird der Listeneintrag auf die eine oder die andere Weise zusammengesetzt.*

Siehe auch

`vtPackage`

## vtPath

Wiederholt den nachfolgenden HTML-Code für jede Seite, die zum Pfad der aktuellen Seite gehört.

Verwendung	<code>&lt;!--{{vtPath}}--&gt; ... &lt;!--{{vtEndPath}}--&gt;</code>
Version	Basic, Pro (ab 1.0)
Parameter	–

Mit `vtPath` lassen sich auf einfache Weise Orientierungshilfen erstellen, die dem Besucher der Website zeigen, wo sich die aktuelle Seite innerhalb der Hierarchie befindet. Durch Kombination mit entsprechenden `vtLink`-Platzhaltern wird ihm außerdem ein schnelles Navigieren ermöglicht.

### Beispiel

```
<!--{{vtPath}}-->  
  <a href="{{vtLink}}">{{vtName}}</a> |  
<!--{{vtEndPath}}-->  
{{vtName}}
```

## vtPathIDs

Wird ersetzt durch eine Liste mit den Nummern aller Seiten, die zum Pfad der aktuellen Seite gehören.

Verwendung	<code>{{vtPathIDs}}</code>
Version	Basic, Pro (ab 1.3)
Parameter	–

Der Platzhalter `vtPathIDs` ist hauptsächlich für den Aufbau dynamischer Navigationsmenüs gedacht. Mit seiner Hilfe lässt sich auf einfache Weise prüfen, ob eine Seite, die in einem Navigationsmenü aufgeführt ist, zum Pfad der aktiven Seite gehört und daher typografisch hervorgehoben werden muss. Die Liste, die `vtPathIDs` ausgibt, enthält neben den Nummern der Containerseiten auch die Nummer der aktuellen Seite selbst.

### Beispiel

```
<!--{{vtSet: Pfad="{{vtPathIDs}}"}}-->
<!--{{vtUse:/}}-->
  <!--{{vtLoop}}-->
    <!--{{vtIf: {{vtGet:Pfad}} .cn. {{vtID}} }}-->
      <strong>{{vtName}}<br>
    <!--{{vtElse}}-->
      <a href="{{vtLink}}">{{vtName}}</a><br>
    <!--{{vtEndIf}}-->
  <!--{{vtEndLoop}}-->
<!--{{vtEndUse}}-->
```

Dieses Beispiel erzeugt ein Navigationsmenü mit allen Seiten, die der Startseite untergeordnet sind. In dem Menü ist die Seite, zu deren Zweig die aktuelle Seite gehört, fett hervorgehoben, die übrigen Seiten sind als Hyperlinks markiert.

### Siehe auch

`vtID`

## vtPortions

Wird durch eine Navigationshilfe ersetzt, die das Blättern durch portionierte Listen ermöglicht.

Verwendung	<code>{{vtPortions}}</code> oder <code>{{vtPortions: Klasse}}</code>
Version	Basic, Pro (ab 1.3)
Parameter	<i>Klasse</i> CSS-Klasse, die den erzeugten Ankerelementen zugewiesen werden soll

Der `vtPortions`-Platzhalter ist nur für Templates gedacht, in denen Sie eine `vtLoop`-Anweisung verwenden und die Zahl der Durchläufe mittels `-size`-Parameter begrenzen. In diesem Fall gibt `vtPortions` eine Navigationshilfe aus, mit der ein Leser durch die einzelnen „Portionen“ der Liste blättern kann. Wenn die Gesamtlänge der Liste kleiner ist als die angegebene Portionsgröße gibt `vtPortions` nichts aus.

Beachten Sie, dass `vtPortions` im Template auf jeden Fall *hinter* der `vtLoop`-Anweisung stehen muss. Sollen die Hyperlinks auf besondere Weise formatiert werden, können Sie durch Angabe des zusätzlichen Parameters eine beliebige CSS-Klasse zuordnen. Jedes `<a>`-Element erhält dann ein zusätzliches `class`-Attribut, dessen Wert dem Namen der angegebenen Klasse entspricht.

### Beispiel

```
<p><b>Liste</b></p>
<ul>
<!--{{vtLoop: -size="5"}}-->
  <li><a href="{{vtLink}}">{{vtName}}</a></li>
<!--{{vtEndLoop}}-->
</ul>
<p>{{vtPortions}}</p>
```

Dieser Quellcode wird im Webbrowser auf die folgende (oder ähnliche) Weise dargestellt:

<p><b>Liste</b></p> <ul style="list-style-type: none"> <li>• <a href="#">Artikel 367228</a></li> <li>• <a href="#">Artikel 367229</a></li> <li>• <a href="#">Artikel 367232</a></li> <li>• <a href="#">Artikel 367236</a></li> <li>• <a href="#">Artikel 367240</a></li> </ul> <p>&lt;&lt; 1 2 3 4 &gt;&gt;</p>
---

Siehe auch

`vtLoop`

## vtPostValue

Wird durch einen Wert aus einem mittels „post“-Methode abgesendeten Formular ersetzt.

Verwendung	<code>{{vtPostValue: Feldname}}</code>
Version	Basic, Pro (ab 1.2)
Parameter	<i>Feldname</i> Name des Formularfeldes, dessen Wert eingesetzt werden soll

Normalerweise werden Formularinhalte automatisch durch die jeweils angegebene Aktion (create, update etc.) verarbeitet. Mit `vtPostValue` können Sie aber auch unabhängig davon gezielt auf Formularinhalte zugreifen.

### Beispiel

```
<!--{{vtIf: {vtPostValue: Suchwort} .eq. }}-->
  <p>Bitte geben Sie ein Suchwort ein, bevor Sie auf "go"
  klicken.</p>
<!--{{vtElse}}-->
  <p>Suche nach "{{vtPostValue:Suchwort}}"</p>
  <ul>
    <!--{{vtSelect: -origin="/", -type="artikel",
      -filter="{Inhalt} .cn. {vtPostValue:Suchwort}"}}-->
      <!--{{vtLoop}}-->
        <li><a href="{{vtLink}}">{{Titel}}</a></li>
      <!--{{vtEndLoop}}-->
    <!--{{vtEndSelect}}-->
  </ul>
<!--{{vtEndIf}}-->
```

### Siehe auch

`vtGetValue`

## vtRedirect

Verändert den HTTP-Header der aktuellen Seite so, dass der Browser des Anwenders zu einer anderen Seite umgelenkt wird.

Verwendung	<code>&lt;!--{{vtRedirect: Umlenkziel}}--&gt;</code>
Version	Basic, Pro (ab 1.0)
Parameter	<i>Umlenkziel</i> Angabe der Seite, zu der umgelenkt werden soll.

Falls auf eine externe Webseite umgelenkt werden soll, ist als Umlenkziel eine vollständige URL anzugeben, beginnend mit `http://`, `ftp://` o. ä. Soll auf eine Seite innerhalb von Virthos umgelenkt werden, lässt sich diese auf gleiche Weise angeben wie bei `vtLink`.

### Beispiel 1

```
<!--{{vtRedirect: http://www.virthos.net }}-->
```

### Beispiel 2

```
<!--{{vtRedirect: /News }}>
```

### Siehe auch

`vtLink`, `vtHeader`

## vtRepeat

Wiederholt den nachfolgenden HTML-Code für alle Elemente einer Werteliste.

Verwendung	<code>&lt;!--{{vtRepeat: Werteliste}}--&gt; ... &lt;!--{{vtEndRepeat}}--&gt;</code>
Version	Basic, Pro (ab 1.0)
Parameter	<i>Werteliste</i> mehrere durch Komma oder Zeilenschaltung getrennte Werte

Mit `vtRepeat` lassen sich Listen, die aus mehreren durch Komma oder Zeilenschaltung getrennten Werten bestehen, beliebig formatieren. Die Liste kann direkt angegeben werden oder in Form eines Platzhalters, der für eine entsprechende Liste steht. Bei einem Platzhalter ist darauf zu achten, dass dessen schließende Klammer durch ein Leerzeichen von der schließenden Klammern der `vtRepeat`-Anweisung getrennt ist.

Innerhalb von `vtRepeat` lässt sich mit dem Platzhalter `vtCurrentItem` der aktuelle Listeneintrag ausgeben.

### Beispiel1

```
<ul>
  <!--{{vtRepeat: {Mitarbeiter} }}-->
  <li>{{vtCurrentItem}}</li>
  <!--{{vtEndRepeat}}-->
</ul>
```

### Beispiel2

```
<select name="Wochentag">
  <!--{{vtRepeat: Mo, Di, Mi, Do, Fr, Sa, So}}-->
  <option>{{vtCurrentItem}}</option>
  <!--{{vtEndRepeat}}-->
</select>
```

### Siehe auch

`vtCurrentItem`

## vtResult

Wird durch das Ergebnis der letzten Aktion oder Berechnung ersetzt.

Verwendung	{{vtResult}}
Version	Basic, Pro (ab 1.3)
Parameter	–

Nach dem Ausführen einer Aktion enthält `vtResult` einen Zahlenwert, der angibt, ob die Aktion erfolgreich ausgeführt wurde, oder nicht. Ein negativer Wert bedeutet: Bei der Aktion ist ein Fehler aufgetreten. Eine Null oder ein positiver Wert bedeuten, dass die Aktion erfolgreich ausgeführt wurde.

Neben dem Ergebnis einer Aktion kann `vtResult` auch das Ergebnis einer Berechnung mittels `vtCalc`-Anweisung enthalten. Dabei überschreibt jede `vtCalc`-Anweisung das Ergebnis der vorherigen. Soll das Ergebnis später noch einmal verwendet werden, müssen Sie es mittels `vtSet`-Anweisung zwischenspeichern.

### Beispiel

```
<!--{{vtCalc: {Nettopreis} * 1.16 }}-->
<p>Der Bruttopreis beträgt {{vtResult}} Euro.</p>
```

### Siehe auch

`vtCalc`, `vtSet`



## vtScript

Entfernt die nachfolgenden VirthosTalk- und HTML-Elemente aus dem Quellcode.

Verwendung	<code>&lt;!--{{vtScript}}--&gt; ... &lt;!--{{vtEndScript}}--&gt;</code>
Version	Basic, Pro (ab 1.3)
Parameter	–

Wird eine längere Folge von VirthosTalk-Anweisungen in ein Template eingebunden, entstehen im Quellcode einer abgerufenen Seite hinterher oft größere Lücken mit Leerzeilen und Leerzeichen. Um dies zu unterbinden, lässt sich die Anweisungsfolge in einen `vtScript`-Block einschließen. Nach einer `vtScript`-Anweisung unterdrückt Virthos jede weitere Datenausgabe an den Webbrowser, bis eine `vtEndScript`-Anweisung folgt.

### Beispiel

```
<!--{{vtScript}}-->

    Zuerst berechnen wir die Nettosumme:

    {{vtSelect: -origin="{vtGet:WarenkorbID}" -type="artikel"}}
        {{vtSet: Nettosumme="{vtEval: sum, Preis}"}}
    {{vtEndSelect}}

    ... dann die Bruttosumme:

    {{vtCalc: round( {vtGet:Nettosumme} * 1.16, 2 ) }}
    {{vtSet: Bruttosumme}}

<!--{{vtEndScript}}-->

<p>Die Gesamtsumme beträgt {{Bruttosumme}} EUR brutto.</p>
```

## vtSelect

Definiert eine Auswahl von Seiten (Selektion) für Listendarstellungen, Auswertungen o. ä.

Verwendung	<code>&lt;!--{{vtSelect: -origin="Ausgangsseite", -type="Seitentyp", -filter="Filter", -max="Maximum", -sortfield="Sortierfeld", -sortorder="Sortierfolge"}}--&gt; ... &lt;!--{{vtEndSelect}}--&gt;</code>	
Version	Pro (ab 1.2)	
Parameter	<i>Ausgangsseite</i>	Seite, von der aus die Selektion gebildet wird
	<i>Seitentyp</i>	Seitentyp, der ausgewählt werden soll
	<i>Filter</i>	Bedingungen, die eine Seite erfüllen muss, um in die Auswahl aufgenommen zu werden
	<i>Maximum</i>	Obergrenze für die zu selektierende Anzahl von Seiten
	<i>Sortierfeld</i>	Name des Feldes, nach dem die Auswahl sortiert werden soll
	<i>Sortierfolge</i>	Angabe, ob die Seiten aufsteigend, absteigend oder nach einem Zufallsprinzip sortiert werden sollen

Die `vtSelect`-Anweisung hat keine direkte Auswirkung auf den nachfolgenden HTML-Code; sie definiert lediglich ein „Umfeld“ für andere VirthosTalk-Anweisungen und -Platzhalter, insbesondere für `vtLoop`, `vtEval` und `vtCountSelection`. Es kann immer nur eine Selektion aktiv sein, daher ist das Verschachteln von `vtSelect`-Blöcken nicht zulässig. Um die Ergebnisse mehrerer Selektionen zu mischen, müssen Sie diese zunächst mittels `vtSet` zwischenspeichern und dann an der gewünschten Stelle mit `vtGet` einfügen.

Für die Angabe der *Ausgangsseite* können alle Möglichkeiten genutzt werden, die auch bei `vtLink` zur Verfügung stehen (vgl. Kapitel „Seitenangaben“). Wird dieser Parameter weggelassen, wird die Seite des aktuellen Kontext als Ausgangsseite verwendet. Bei der Selektion werden nur Seiten berücksichtigt, die der Ausgangsseite direkt oder indirekt untergeordnet sind.

Die Angabe eines *Seitentyps* führt dazu, dass – unabhängig von sonstigen Filterbedingungen – nur Seiten berücksichtigt werden, die vom angegebenen Typ sind. Der Seitentyp entspricht dem Namen des Templates, auf dem die Seite basiert, ohne die Endung „.html“. Wenn im `-filter`-Parameter template-spezifische Platzhalter verwendet werden, ist die Angabe des Seitentyps zwingend erforderlich.

Der *Filter* besteht aus einer durch Komma getrennten Liste von Bedingungen, von denen jede dem folgendem Schema entspricht:

*Platzhalter* .*Vergleichsoperator* . *Vergleichswert*

Als *Platzhalter* können Sie alle Platzhalter verwenden, die in dem angegebenen Seitentyp vorkommen, sowie die typunabhängigen Platzhalter `{vtName}` und `{vtPageType}`.

Der *Vergleichswert* kann durch einen festen Wert (Zeichenkette, Zahl, Datum oder Uhrzeit) oder ebenfalls durch einen Platzhalter angegeben werden. Dabei ist zu beachten, dass ein Platzhalter – im Gegensatz zu dem Platzhalter, der sich links vom Vergleichsoperator befindetet – in dem Kontext interpretiert wird, in dem die `vtSelect`-Anweisung steht, und nicht im Kontext der einzelnen, zu prüfenden Seite. Es ist daher z. B. nicht möglich, alle Seiten zu selektieren, bei denen die Platzhalter `{Datum_A}` und `{Datum_B}` denselben Inhalt haben.

Soll ein Datum als Vergleichswert dienen, muss es in der Form „tt.mm.jjjj“ angegeben werden, eine Uhrzeit in der Form „hh:mm“. Wenn der Stunden- oder Minutenwert dabei kleiner als 10 ist, muss jeweils eine Null vorangestellt werden; bei Tages- und Monatswerten ist eine vorangestellte Null nicht erforderlich.

Die möglichen Vergleichsoperatoren sind in der folgenden Tabelle beschrieben:

Operator	Bedeutung	Erläuterung
<code>.eq.</code>	equals	ist gleich
<code>.neq.</code>	not equals	ist ungleich
<code>.is.</code>	is like	(Zeichenkette) ist wie
<code>.gt.</code>	greater than	(Zahl, Datum oder Uhrzeit) ist größer als
<code>.gte.</code>	greater than or equal	(Zahl, Datum oder Uhrzeit) ist größer als oder gleich
<code>.lt.</code>	less than	(Zahl, Datum oder Uhrzeit) ist kleiner als
<code>.lte.</code>	less than or equal	(Zahl, Datum oder Uhrzeit) ist kleiner als oder gleich
<code>.bw.</code>	begins with	(Zeichenkette) beginnt mit
<code>.ew.</code>	ends with	(Zeichenkette) endet mit
<code>.cn.</code>	contains	(Zeichenkette) enthält
<code>.ncn.</code>	not contains	(Zeichenkette) enthält nicht

Der Unterschied zwischen `.eq.` und `.is.` besteht darin, dass `.eq.` bei Zahlen-, Datums- und Uhrzeitwerten einen numerischen Vergleich durchführt, während `.is.` einen Zeichenkettenvergleich erzwingt, auch wenn es sich um numerische Werte handelt. Wenn bei einer Seite beispielsweise der Platzhalter `{PLZ}` den Wert „01023“ enthält, würde die Filterbedingung `{PLZ}.eq.1023` dazu führen, dass die Seite selektiert wird, während sie bei der Bedingung `{PLZ}.is.1023` nicht Teil der Selektion würde.

Wenn Sie mehrere Filterbedingungen angeben, werden alle Seiten selektiert, die *sämtliche* Bedingungen erfüllen; dies entspricht einer logischen „und“-Verknüpfung. Eine „oder“-Verknüpfung von Bedingungen ist nicht möglich.

Das *Sortierfeld* legt fest, in welcher Reihenfolge die selektierten Seiten innerhalb einer `vtLoop`-Anweisung durchlaufen werden. Wenn ein Seitentyp angegeben ist, können Sie als Sortierfeld jeden Platzhalter verwenden, der in dem angegebenen Seitentyp vorkommt. Daneben stehen in

jedem Fall die vordefinierten Platzhalter `vtName`, `vtPageType`, `vtCreationDate` und `vtModificationDate` zur Verfügung. Da beim Sortierfeld nur der Name und nicht der Inhalt des Platzhalters gemeint ist, dürfen keine geschweiften Klammern gesetzt werden.

Bei der *Sortierfolge* haben Sie die Wahl zwischen drei Werten:

<code>ascending</code>	Sortierung in aufsteigender Folge
<code>descending</code>	Sortierung in absteigender Folge
<code>random</code>	Sortierung nach Zufallsprinzip (bei jedem Seitenaufruf anders)

Wenn Sie kein Sortierfeld oder keine Sortierfolge angeben, wird für die Selektion die gleiche Sortierung verwendet wie für die aktuelle Seite, so wie in den Seitenvorgaben festgelegt. Bei einer Sortierung nach dem Zufallsprinzip wird die Angabe zum Sortierfeld ignoriert.

Mit der Angabe eines *Maximums* können Sie die Selektion zahlenmäßig begrenzen. Es werden dann, unter Berücksichtigung von Sortierfeld und -folge, nur die ersten *x* Seiten selektiert, die den Filterbedingungen entsprechen. Im Gegensatz zu dem `-size`-Parameter in einer `vtLoop`-Anweisung wirkt sich die Maximumangabe in `vtSelect` auch auf die Platzhalter `vtCountSelection` und `vtEval` aus. Bei einer Internetpräsenz mit einer großen Zahl von Seiten wird durch eine Maximumangabe in jedem Fall die Serverlast verringert, die mit der Selektion verbunden ist.

### Beispiel 1

```
<!--{{vtSelect: -origin="/", -type="news", -max="3",
               -sortfield="Datum", -sortorder="descending"}}-->
  <ul>
  <!--{{vtLoop}}-->
    <li><a href="{{vtLink}}">{{Headline}}</li>
  <!--{{vtEndLoop}}-->
  </ul>
<!--{{vtEndSelect}}-->
```

*Erstellt eine Liste mit den ersten drei Seiten, die sich im Startseitenzweig befinden und auf dem Template „news.html“ basieren. Die Liste ist nach dem Inhalt des Platzhalters „Datum“ in absteigender Folge sortiert.*

### Beispiel 2

```
<!--{{vtSelect:-origin="//Download", -filter="{{vtName}.ew.pdf"}}-->
  <p>Es stehen {{vtCountSelection}} PDF-Dateien zum Download
  bereit.</p>
<!--{{vtEndSelect}}-->
```

*Gibt die Anzahl der Seiten aus, die sich im Zweig der Seite „Downloads“ (auf der obersten Hierarchieebene) befinden und deren Name mit „pdf“ endet.*

### Beispiel3

```
<!--{{vtSelect: -origin="//Produkte", -type="artikel",  
-filter="{Kategorie}.eq.Sonderangebote, {Verfuegbarkeit}.gt.0"}}-->  
  <p>Sonderangebote schon ab {{vtEval: min, Preis}} Euro!</p>  
<!--{{vtEndSelect}}-->
```

*Selektiert alle Seiten vom Typ „artikel“ im Zweig der Seite „Produkte“ (auf der obersten Hierarchieebene), bei denen der Platzhalter „Kategorie“ das Wort „Sonderangebote“ enthält und der Platzhalter „Verfuegbarkeit“ eine Zahl, die größer ist als 0. Ermittelt dann über alle selektierten Seiten hinweg den Minimalwert für den Platzhalter „Preis“.*

### Siehe auch

`vtCountSelection`, `vtEval`, `vtLoop`

## vtSelf

Wird durch eine URL zum Aufrufen der aktuellen *virtos.php*-Datei ersetzt.

Verwendung	<code>{{vtSelf}}</code>
Version	Basic, Pro (ab 1.3)
Parameter	–

Mit Hilfe von `vtSelf` können Sie Hyperlinks erstellen, die (im Unterschied zu Hyperlinks von `vtLink`) keine Informationen über die aktuelle Sitzung enthalten. Ein Besucher, der einem solchen Hyperlink folgt, wird somit wie ein Besucher behandelt, der von außen (z. B. von einer Suchmaschine aus) auf die Website gelangt. Ein weiteres Einsatzgebiet für `vtSelf` ist das Erzeugen von URLs, die nur zu Informationszwecken dienen (siehe Beispiel).

### Beispiel

```
<p>URL dieser Seite: {{vtSelf}}?{{vtID}}</p>
```

Siehe auch

`vtLink`

## vtSession

Wird durch den Wert einer Sessionvariablen ersetzt.

Verwendung	<code>{{vtSession: Name}}</code>
Version	Pro (ab 1.3)
Parameter	<i>Name</i> Name einer Sessionvariablen

Sessionvariablen lassen sich mit Hilfe einer „updateSession“-Aktion setzen. Sie behalten ihren Wert für die Gesamtdauer einer Arbeitssitzung oder bis sie durch einen anderen Wert überschrieben werden. Mit Hilfe einer Sessionvariablen lässt sich ein einfacher Passwortschutz verwirklichen, wie das Beispiel zeigt.

### Beispiel

```
<!--{{vtIf: {vtSession:Passwort} .neq. ht2y8t}}-->
  <p>Bitte geben Sie zuerst das Passwort ein:</p>
  <form action="{{vtLink:-act='updateSession'}}" method="post">
    <p>Passwort: <input type="password" name="Passwort"></p>
    <p><input type="submit" value="go"></p>
  </form>
<!--{{vtElse}}-->
  ...
<!--{{vtEndIf}}-->
```

### Siehe auch

[vtLink](#)

## vtSet

Speichert einen Wert in einer Variablen.

Verwendung	<pre>&lt;!--{{vtSet: Name="Wert"}}--&gt; oder &lt;!--{{vtSet: Name1="Wert1", Name2="Wert2", ...}}--&gt; oder &lt;!--{{vtSet: Name}}--&gt;</pre>
Version	Basic, Pro (ab 1.3)
Parameter	<p><i>Name</i> Name der Variablen</p> <p><i>Wert</i> der zu speichernde Wert</p>

Mit `vtSet` lassen sich Werte zwischenspeichern, um sie an anderer Stelle innerhalb desselben Templates wiederzuverwenden (mit Hilfe des `vtGet`-Platzhalters). Ein dauerhaftes Speichern über mehrere Seitenabrufe hinweg ist mit `vtSet` nicht möglich; für diesen Zweck ist die Verwendung von Sessionvariablen (vgl. `vtSession`) erforderlich.

Wird nur ein Name, aber kein Wert angegeben, weist Virthos der Variablen den Inhalt des `vtResult`-Platzhalters zu, der das Ergebnis der letzten Formularaktion oder der letzten Berechnung mittels `vtCalc` enthält. Die Anweisung

```
<!--{{vtSet: Ergebnis="{vtResult}"}}-->
```

bewirkt somit dasselbe wie

```
<!--{{vtSet: Ergebnis}}-->
```

*Name* und *Wert* können jeweils in Form eines Platzhalters oder als fester Wert angegeben werden. Auch eine Kombination aus festen Werten und Platzhaltern, und selbst die Verwendung von Anweisungen ist möglich, wie die folgenden Beispiele zeigen:

### Beispiele

```
<!--{{vtSet: Titel="{Headline}", Text="{Inhalt}"}}-->
```

```
<!--{{vtSet: Anrede="Hallo {Anrede} {Nachname}!"}}-->
```

```
<!--{{vtSet: Liste="{vtLoop}{vtName}, {vtEndLoop}"}}-->
```

```
<!--{{vtSet: Var_{vtGet:Nummer}="abcde"}}-->
```

### Siehe auch

`vtCalc`, `vtGet`, `vtResult`, `vtSession`



## vtUse

Ändert den Kontext, in dem nachfolgende Platzhalter und Seitenangaben interpretiert werden.

Verwendung	<code>&lt;!--{{vtUse: Seite}}--&gt; ... &lt;!--{{vtEndUse}}--&gt;</code> <i>oder</i> <code>&lt;!--{{vtUse: Seite, Seitentyp}}--&gt; ... &lt;!--{{vtEndUse}}--&gt;</code>	
Version	Basic, Pro (ab 1.1)	
Parameter	<i>Seite</i>	Angabe der Seite, die den neuen Kontext bilden soll
	<i>Seitentyp</i>	Anweisung, Seite mit angegebenen Typ zu erstellen, falls nicht existent

Mit Hilfe von `vtUse` lassen sich neben den Inhalten der aktuellen Seite auch Inhalte aus jeder anderen Seite in das aktuelle Template einbinden. Um die Seite anzugeben, die den neuen Kontext bilden soll, stehen dieselben Möglichkeiten zur Verfügung wie bei `vtLink` (vgl. Kapitel „Seitenangaben“). Auch ein Verschachteln von `vtUse`-Anweisungen ist möglich, wobei eine relative Seitenangabe in einer „inneren“ `vtUse`-Anweisung in dem Kontext interpretiert werden, der von der „äußeren“ `vtUse`-Anweisung vorgegeben ist (siehe Beispiele).

Die Angabe eines *Seitentyps* ermöglicht es, die angegebene Seite automatisch erstellen zu lassen, falls sie noch nicht existiert. Dies ist allerdings nur möglich, wenn als erster Parameter ein Seitenname (also keine Seitennummer und kein Pfad) angegeben ist. Virthos prüft dann, ob die Seite, die den aktuellen Kontext bildet, eine Subseite mit dem angegebenen Namen besitzt. Falls nicht, wird eine neue Seite mit dem angegebenen Namen und dem angegebenen Seitentyp erstellt.

### Beispiel1

```
<!--{{vtUse: //Firmendaten}}-->
  <p>Zentrale: {{Telefon}}</p>
<!--{{vtEndUse}}-->
```

### Beispiel2

```
<!--{{vtUse: Kommentare, commentlist}}-->
  <p>Anmerkungen:</p>
  <ul>
    <!--{{vtLoop}}-->
    <li><a href="{{vtLink}}">{{Betreff}}</a> {{Datum}}</li>
    <!--{{vtEndLoop}}-->
  </ul>
<!--{{vtEndUse}}-->
```

### Siehe auch

Kapitel „Seitenangaben“

## vtUser

Wird durch Informationen über den aktuellen Benutzer ersetzt.

Verwendung	<code>{{vtUser: Eigenschaft}}</code>
Version	Pro (ab 1.3)
Parameter	<i>Eigenschaft</i> Art der gewünschten Information

Mit `vtUser` können Sie auf Informationen zugreifen, die in den Einstellungen des aktuellen Benutzers gespeichert sind. So lassen sich Templates gestalten, die unterschiedlichen Benutzern unterschiedliche Inhalte präsentieren.

Als *Eigenschaft* können Sie einen der folgenden Werte verwenden:

<i>Eigenschaft</i>	<i>Bedeutung</i>
<code>id</code>	eindeutige Nummer des Benutzers
<code>name</code>	Anmeldename des Benutzers
<code>realname</code>	vollständiger Name des Benutzers
<code>admin</code>	hat der Benutzer Administratorrechte?
<code>startpage</code>	voreingestellte Startseite für den Benutzer
<code>workspace</code>	bekommt der Benutzer nach der Anmeldung den Virthos-Manager eingeblendet?
<code>groups</code>	Liste mit den Namen der Gruppen, denen der Benutzer angehört

Bei den Eigenschaften `admin` und `workspace` wird der Platzhalter durch „1“ ersetzt, falls die jeweilige Eigenschaft zutrifft, ansonsten durch „0“ oder einen leeren Wert.

### Beispiel 1

```
<h1>Willkommen, {{vtUser:name}}!</h1>
```

### Beispiel 2

```
<!--{{vtIf: {{vtUser:groups}.cn.Redaktion }}-->
  <a href="{{vtLink:-met='vtedit'}}">Seite bearbeiten</a>
<!--{{vtElse}}-->
  <i>Bearbeitung nur für Redakteure möglich.</i>
<!--{{vtEndIf}}-->
```

### Siehe auch

`vtCurrentAccessRights`

## vtUserAgent

Wird durch Informationen über den verwendeten Webbrowser ersetzt.

Verwendung	<code>{{vtUserAgent}}</code> oder <code>{{vtUserAgent: Eigenschaft}}</code>	
Version	Pro (ab 1.3)	
Parameter	<i>Eigenschaft</i>	Art der gewünschten Information

Mit Hilfe von `vtUserAgent` können Sie feststellen, welche Art von Webbrowser der aktuelle Benutzer verwendet und unter welchem Betriebssystem er arbeitet. Als *Eigenschaft* kann einer der folgenden Werte übergeben werden:

<i>Eigenschaft</i>	<i>Bedeutung</i>	<i>mögliche Werte</i>
<code>browser</code>	Typ des Webbrowsers	opera, ie, omniweb, konqueror, safari, mozilla, other
<code>version</code>	Version des Webbrowsers	z. B. „5.0“
<code>os</code>	Betriebssystem	Win, Mac, Linux, Unix, OS/2, other

Wenn Sie die *Eigenschaft* weglassen, verhält sich `vtUserAgent`, als hätten Sie den Parameter `browser` übergeben.

### Beispiel

```
<!--{{vtIf: {vtUserAgent} .neq. ie}}-->
  <p>Diese Website ist für die Verwendung mit dem Internet
  Explorer optimiert. Bei anderen Browsern kann es zu
  Darstellungsfehlern kommen.</p>
<!--{{vtEndIf}}-->
```

## vtUsername

Wird durch den Anmeldenamen des aktuellen Benutzers ersetzt.

Verwendung	{{vtUsername}}
Version	Pro (ab 1.1)
Parameter	–

Dieser Platzhalter sollte nicht mehr verwendet werden. Für den gleichen Zweck steht jetzt der vielseitiger einsetzbare Platzhalter `vtUser` zur Verfügung.

Siehe auch

`vtUser`