

Funktion	Beschreibung
TYPO3.TypoScript:Array	Rendert mehrere TypoScript-Objekte
beliebige Unterelemente mit Namen	Vergleichbar mit dem "COA" aus TYPO3 CMS, rendert mehrere TypoScript-Objekte in einem. Die Unterelemente werden jeweils mit einem selbst gewählten Namen angelegt. Um die Reihenfolge zu bestimmen kann @position benutzt werden.
TYPO3.TypoScript:Template	Rendert ein Fluidtemplate
templatePath	Pfad + Dateiname der Template-Datei oder Flow "resource".
sectionName	Wenn angegeben, wird nur die Section mit diesem Namen gerendert und der Rest des Templates ignoriert.
partialRootPath	Setzt den Pfad, in dem Partials zu finden sind, die in diesem Template genutzt werden.
layoutRootPath	Setzt den Pfad, in dem das Layout zu finden ist, das in diesem Template genutzt wird.
variables	Variablen mit beliebigem Namen der nicht in der Liste hier vorkommt (so darf z. B. keine Variable "templatePath" heißen). Diese Variablen sind im Template verfügbar, es können auch beliebige, verschachtelte Strukturen als Variable genutzt werden.
TYPO3.TypoScript:Collection	Rendert eine Sammlung von Elementen (in der Regel "Nodes"), diese können z. B. mit FlowQuery selektiert werden.
collection	Übergibt die zu rendernden Elemente als Collection, in der Regel steht hier eine FlowQuery. Beispiel: collection = \${q(node).children('[instanceof TYPO3.Neos.ContentTypes:Page])} Selektiert alle Unterseiten der aktuellen Seite.
itemName	Mit dieser Bezeichnung kann innerhalb des itemRenderers auf das aktuell durchlaufende Element der Collection zugegriffen werden; d. h. setzt man itemName auf "currentItem", dann wäre das aktuelle Element innerhalb einer FlowQuery über \${q(currentItem)} verfügbar.
iterationName	Wird diese Bezeichnung gesetzt, stehen Informationen über die Iteration zur Verfügung, die zum Rendern benutzt werden können. Ist der iterationName auf "iteration" gesetzt, kann über \${iteration.isFirst} abgefragt werden, ob es sich um das erste Element handelt. Folgende Informationen sind verfügbar: isFirst isLast index (startet bei 0 zu zählen und gibt den aktuellen Durchlauf an) cycle (startet bei 1 zu zählen und gibt auch den aktuellen Durchlauf an)
itemRenderer	TypoScript-Objekt, das benutzt werden soll, um jedes einzelne Element zu rendern. Normalerweise ein Case-Objekt, um zwischen verschiedenen Node-Typen zu unterscheiden.
TYPO3.TypoScript:Case	Bedingungelement, das sogenannte "Matcher" überprüft und den ersten Matcher nutzt, dessen Bedingung erfüllt ist.
beliebige Unterelemente mit Namen	Unterelemente sind vom Typ TYPO3.TypoScript:Matcher und sollten über @position sortiert werden, um zu bestimmen, welche Matcher am wichtigsten sind. Generell gilt: Spezielle Matcher vor allgemeinen.
TYPO3.TypoScript:Matcher	In der Regel nur in Verbindung mit Case. Kombiniert eine Bedingung mit einem TypoScript-Objekt
condition	Eine FlowQuery, die TRUE oder FALSE zurückliefert und so entscheidet, ob dieser Matcher auf das Element anwendbar ist.
type	Das TypoScript-Objekt, das zum Rendern benutzt werden soll.
TYPO3.Neos.ContentTypes:Page	Ein Template-Objekt mit einigen vordefinierten Variablen und Templates.
body	Diese Variable sollte gesetzt werden und muss den Content beinhalten, der zwischen die body-Tags soll. Vorbelegt ist hier ein Template-Objekt, so dass nur noch ein templatePath angegeben werden muss und die entsprechenden Variablen gesetzt werden müssen.
headerData	Bereits als TYPO3.TypoScript:Array definiert. Alle enthaltenen TypoScript-Objekte werden in den <head> der Webseite gerendert.
TYPO3.Neos:Menu	Rendert ein Menu aus dem Seitenbaum.
templatePath	Hierüber kann das Template des Menüs überschrieben werden, ein Standardtemplate ist aber gesetzt.
entryLevel	Seitenbauebene, auf der das Menü beginnt.
lastLevel	Seitenbauebene, auf der das Menü endet.
maximumLevels	Maximale Anzahl von Unterebenen, die das Menü darstellen soll. Der Standard ist hier 1, muss überschrieben werden, wenn mehr Ebenen gewünscht werden.
TYPO3.Neos.ContentTypes:Section	Spezialisiertes Collection-Objekt, um Inhalte auf der Seite zu rendern.
nodePath	Der nodeName der Section, die gerendert werden soll. Kann selbst bestimmt werden.
TYPO3.TypoScript:Value	Ein einfacher Wert, vergleichbar mit dem Text-Objekt von TYPO3 CMS.
value	Der Wert der ausgegeben werden soll.
TYPO3.Neos.ContentTypes:Plugin	Standard-Renderer für Plugins, alle eigenen Plugins sollten davon im TypoScript erben.
Prototypdefinition	Prototypen sind TypoScript-Objekte. Ein Prototyp wird auch überall oben erwartet, wenn von TypoScript-Objekt die Rede ist. Alle oben angegebenen TypoScript-Objekte sind Prototypen. prototype(TYPO3.Neos:Menu).@class = ,TYPO3\Neos\TypoScript\MenuImplementation'
Prototype copy	prototype(TYPO3.Neos:Template) < prototype(TYPO3.TypoScript:Template)
Meta Eigenschaften	Es gibt einige "Meta" Eigenschaften, die mit mehreren TypoScript-Objekten benutzt werden können. Alle Meta-Eigenschaften fangen mit @ an. Die wahrscheinlich wichtigste nach @class ist @position. Im einfachsten Fall funktioniert @position über die Angabe von Zahlen, d. h. @position = ,10' Will man vor allem anderen positionieren, kann man entweder einfach @position = ,start' benutzen oder auch dort mit einer Zahl kombinieren, um zu priorisieren, also @position = ,start 10' Gleiches gilt für "end". Außerdem lassen sich mit "before" und "after" auch relative Positionen setzen.