

Document Version 1.2

Current Page	Collections	Traversing	Property Checking
@Model	.Children .[DocTypeName]s (Pluralized)	.Parent .First() .Last() .Up() .Up(int) (note 0 = 1) .Down() .Down(int) (note 0 = 1) .Next() .Next(int) (note 0 = 1) .Previous() .Previous(int) (note 0 = 1)	.HasProperty(string propertyAlias) .HasValue(string propertyAlias) .IsNull(string propertyAlias)
Dynamic Node Properties	.Ancestors() .Ancestors(int level) .Ancestors(string nodeTypeAlias) .Ancestors(Func< DynamicNode, bool > func) .AncestorsOrSelf() .AncestorsOrSelf(string nodeTypeAlias) .AncestorsOrSelf(int level) .AncestorsOrSelf(Func< DynamicNode, bool > func) .Descendants() .Descendants(string nodeTypeAlias) .Descendants(int level) .Descendants(Func< INode, bool > func) .DescendantsOrSelf() .DescendantsOrSelf(int level) .DescendantsOrSelf(string nodeTypeAlias) .DescendantsOrSelf(Func< INode, bool > func) .XPath(string xPath) .GetChildrenAsList	Macro Parameters	IsHelpers .IsFirst([valueIfTrue][,valueIfFalse]) .IsNotFirst([valueIfTrue][,valueIfFalse]) .IsLast([valueIfTrue][,valueIfFalse]) .IsNotLast([valueIfTrue][,valueIfFalse]) .IsPosition(int,[valueIfTrue][,valueIfFalse]) .IsNotPosition(int,[valueIfTrue][,valueIfFalse]) .IsModZero([valueIfTrue][,valueIfFalse]) .IsNotModZero([valueIfTrue][,valueIfFalse]) .IsEven([valueIfTrue][,valueIfFalse]) .IsOdd([valueIfTrue][,valueIfFalse]) .IsEqual(DynamicNode[,valueIfTrue][,valueIfFalse]) .IsDecendant(DynamicNode[,valueIfTrue][,valueIfFalse]) .IsDecendantOrSelf(DynamicNode[,valueIfTrue][,valueIfFalse]) .IsAncestor(DynamicNode[,valueIfTrue][,valueIfFalse]) .IsAncestorOrSelf(DynamicNode[,valueIfTrue][,valueIfFalse])
.Parent .Id .Template .SortOrder .Name .Visible (requires umbracoNaviHide property) .Url .UrlName .NodeTypeAlias .WriterName .CreatorName .WriterID .CreatorID .Path .CreateDate .UpdateDate .Version .NiceUrl .Level .PropertiesAsList .ChildrenAsList .Position()	Filtering & Ordering & Extensions	Media (use for media picker property)	Permissions .HasAccess() .IsProtected()
DynamicMedia Properties	.Where("Condition"[,valueIfTrue,valueIfFalse]) .OrderBy("propertyAlias [desc][,propertyAlias]") .GroupBy("propertyAlias") .Pluck("propertyName") .Take(int) .Skip(int) .Count()	@Parameter.ParameterName	
.UmbracoFile .UmbracoSize .UmbracoWidth .UmbrachHeight	Functions	Types for Casting and Newing	
Custom Property Access (Content & Media)	@functions{ public bool isFooBar(string foo, string bar){ return foo == bar; } } @{ var foo = "black"; if(isFooBar(foo,"black")){ <p>Yup</p> } }	DynamicNode(int nodeId) DynamicMedia(int mediaId) DynamicNodeList dynamic (allows .PropertyAlias notation)	Razor Syntax Code Block @{ ... } Comments @* Code comment *@
Note: casing on property aliases are important. All hyphens must be removed for .Notation When accessing using .Notation you need to ensure you capitalize the first letter unless _recursive		Media (use for media picker property)	
Dictionary		.Media("propertyAlias", "mediaPropertyAlias") .Media("propertyAlias")	Conditionals @if(item.HasValue("bodyText"){ @item.BodyText }else if(item.IsNull("bodyText")){ <p>this item is null</p> }else{ <p>Some other text</p> } Looping @for(var i = 0; i < 10; i++){ This is record @i } @foreach(var item in Model.Children){ <p>@item.Name</p> }
@Dictionary .DictionaryItemAlias		Types for Casting and Newing	Conditionals @switch(condition){ case 1 <p>@item.BodyText</p> break; case 2 <p>@item.Children.First().BodyText</p> break; default: break; }
@Library Helper		Macro Parameters	
Loaders	Manipulation		
.NodeById(int string) .MediaById(int string)	.Coalesce(value,value[,value ...]) .Concatenate(value,value[,value ...]) .Join(seperator,value,value[,value ...]) .Truncate(htmlString,int[[,bool addElipsis][,bool treatTagsAsContent]]) .StripHTML(htmlString[, tagsToString])		Visual Studio DynamicNode Intellisense Put the following at the head of the CSHtml file to get some Intellisense support for DynamicNode and DynamicNodeList @using umbraco.MacroEngines @inherits umbraco.MacroEngines.DynamicNodeContext
Conditionals			
.If(booleanProp,valueIfTrue[,valueIfFalse])			