

Document Version 1.3

Current Page @Model	Collections .Children .[DocTypeName]s (Pluralized) .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	Traversing .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) .AncestorOrSelf() .AncestorOrSelf(string nodeTypeAlias) .AncestorOrSelf(int level) .AncestorOrSelf(Func< DynamicNode, bool > func) .DescendantOrSelf() .DescendantOrSelf(string nodeTypeAlias) .DescendantOrSelf(int level) .DescendantOrSelf(Func< DynamicNode, bool > func)	Property Checking .HasProperty(string propertyAlias) .HasValue(string propertyAlias) .IsNull(string propertyAlias)
Dynamic Node Properties .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 .Where("Condition",[valueIfTrue,valueIfFalse]) .OrderBy("propertyAlias [desc],[propertyAlias]") .GroupBy("propertyAlias") .Pluck("propertyName") .Take(int) .Skip(int) .Count()	Macro Parameters @Parameter.ParameterName Media (use for media picker property) .Media("propertyAlias", "mediaPropertyAlias") .Media("propertyAlias")	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]) .IsDescendant(DynamicNode[,valueIfTrue],[valueIfFalse]) .IsDescendantOrSelf(DynamicNode[,valueIfTrue],[valueIfFalse]) .IsAncestor(DynamicNode[,valueIfTrue],[valueIfFalse]) .IsAncestorOrSelf(DynamicNode[,valueIfTrue],[valueIfFalse])
DynamicMedia Properties .UmbracoFile .UmbracoSize .UmbracoWidth .UmbrachHeight		Types for Casting and Newing DynamicNode(int nodeId) DynamicMedia(int mediaId) DynamicNodeList dynamic (allows .PropertyAlias notation)	Permissions .HasAccess() .IsProtected()
Custom Property Access (Content & Media) .PropertyAlias .propertyAlias (recursive) .GetProperty("propertyAlias").Value Notes: 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	Functions @functions{ public bool isFooBar(string foo, string bar){ return foo == bar; } } @{ var foo = "black"; if(isFooBar(foo,"black")){ <p>Yup</p> } }	Razor Syntax Code Block @{ ... } Conditionals @if(item.HasValue("bodyText"){ @item.BodyText }else if(item.IsNullOrEmpty("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){	Comments @* Code comment *@ @switch(condition){ case 1: <p>@item.BodyText</p> break; case 2: <p>@item.Children.First().BodyText</p> break; default: break;
Dictionary @Dictionary .DictionaryItemAlias @Dictionary["dictionaryItemAlias"]			
@Library Helper Loaders .NodeById(int string) .MediaById(int string) Conditionals .If(booleanProp,valueIfTrue[,valueIfFalse])	Manipulation .Coalesce(value,value[,value ...]) .Concatenate(value,value[,value ...]) .Join(seperator,value,value[,value ...]) .Truncate(htmlString,int[[,bool addEllipsis][,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