

# String

A string is comprising of one or more letters and numbers forming one or more words. Furthermore, a string might contain some symbols/special characters. In most cases you will note that inside strings there are only . (full stops) and , (commas) - but this is more convention than a fixed rule. Strictly, any number and combination of characters and symbols can be regarded a string - but that's way beyond the scope of this wiki.

Strings are enclosed in quotes, either 'single quotes' or "double quotes". As all the factory macros use double quotes, you should adhere to that rule as well: always enclose strings with double quotes!

Also, this distincts a property's name from a property's value.

See from the example [Timecode - Enable/Disable](#):

```
<step>ActionScript.SetProperty.Boolean("Timecode.Enabled",
!Timecode.Enabled)</step>
```

- [ActionScript.SetProperty.Boolean](#) is the important function here, and expects the name/id of the property to set passed as string, and the value to set passed as [Boolean](#)
- "Timecode.Enabled" is the name of the property - we pass it as string, and hence enclose it in double quotes
- !Timecode.Enabled is the property's value (forget about the ! here - this is a logical NOT, see [Control Structures](#)).
- Hence, in total, this line says 'set the property 'Timecode.Enabled' to the opposite from what the value 'Timecode.Enabled' currently is'. One could write this shorter: toggle it!

## Also used in

- [The Syntax of Functions](#)
- [ActionScript.SetProperty](#)
- [ActionScript.SetProperty.Boolean](#)
- [ActionScript.SetProperty.Double](#)
- [ActionScript.SetProperty.Enum](#)
- [ActionScript.SetProperty.Float](#)
- [ActionScript.SetProperty.Integer](#)
- [ActionScript.SetProperty.SetFlag](#)
- [ActionScript.SetProperty.String](#)
- [Attribute.Mask.IncludeAll](#)
- [Attribute.Mask.SetMask](#)
- [Command.RunCommand](#)
- [Dmx.EnableDmxModule](#)
- [Editor.Shapes.SetCurrentShapesDirection](#)
- [UserMacros.Export](#)
- [PatchFixturesToVacantHandles](#)
- [Group.CheckIfGroupExists](#)
- [Group.QuickCreateGroup](#)
- [Group.SetGroupFaderLevel](#)
- [Handles.CopyDestination](#)

- [Handles.CreateHandleReference](#)
- [Handles.GetHandle](#)
- [Handles.SetGroupPage](#)
- [Handles.SetGroupPageName](#)
- [Handles.SetPendingHaloToSystemColour](#)
- [Handles.SetSourceHandle](#)
- [Handles.SetSourceHandleRange](#)
- [Icons.StoreIcon](#)
- [Include.SelectPlayback](#)
- [Math.EnumAsStringEqualityConverter](#)
- [Math.EqualityConverter](#)
- [Math.Min](#)
- [Math.ToEnum](#)
- [Math.ToString](#)
- [Menu.ErrorReport](#)
- [Menu.InjectInput](#)
- [Menu.PushOrReloadMenu](#)
- [Menu.Stack.PushOrReloadMenu](#)
- [Palette.QuickCreatePalette](#)
- [Panel.Midi.Send](#)
- [PanelManager.SetScreenState](#)
- [PlaybackGroups.CreatePlaybackGroupWithPlaybacks](#)
- [Playbacks.CueList.CreateCueList](#)
- [Playbacks.Editor.Macros.AddLinkFromId](#)
- [Playbacks.Editor.Macros.SetTargetHandle](#)
- [Playbacks.FillCueLegend](#)
- [Playbacks.SetCueLegend](#)
- [Playbacks.StoreCue](#)
- [Programmer.Editor.Fixtures.Fan.SelectCurve](#)
- [Programmer.Editor.Fixtures.GetControllIdFromName](#)
- [Programmer.Editor.Fixtures.Patch.InvertFixtureControls](#)
- [Programmer.Editor.Selection.GetSelectedHandles](#)
- [System.LockConsole](#)
- [Timelines.ImportMarkersFromString](#)
- [Triggers.SelectMappingByName](#)
- [Triggers.SetPendingTargetAction](#)
- [Triggers.SetPendingTargetHardware](#)
- [UserMacros.RecallMacroById](#)
- [Windows.SetWindowProperty.Height](#)
- [Windows.SetWindowProperty.Width](#)
- [Windows.SetWindowProperty.X](#)
- [Windows.SetWindowProperty.Y](#)
- [Workspace.Record](#)
- [Windows](#)
- [Math.Cast](#)
- [Menu.ErrorMessage](#)
- [Playbacks.PendingLegend](#)
- [Windows.ShowLibrary.FilterText](#)
- [Windows.ShowLibrary.SelectedFilter](#)
- [Boolean](#)
- [Enum](#)

- [Flag](#)
- [Object](#)
- [Types](#)

From:

<https://www.avosupport.de/wiki/> - **AVOSUPPORT**

Permanent link:

<https://www.avosupport.de/wiki/macros/type/string>

Last update: **2017/11/19 14:22**

