# Control Structures

When a macro is called, all its steps are executed sequentially.

However, you can define conditional steps - steps which are only executed if a condition is met. In order to do this, simply give the step a condition property, like in Chase - Double speed:

```
<step condition="Math.IsEqual(Playbacks.Editor.Times.ChaseSpeed, 0.0)">
  ActionScript.SetProperty.Float("Playbacks.Editor.Times.ChaseSpeed", 1.0)
</step>
```

This step is only executed if its condition is met - and the condition is a function written in double quotes: Math.IsEqual takes two values as arguments and returns true if both values are equal. In total, if the property `Playbacks.Editor.Times.ChaseSpeed` equals 0.0 then this step is performed and sets this property to 1.0.

Another mechanism is also related to control structures albeit strictly it is just juggling with booleans: simple toggle logic, see Timecode - Enable/Disable:

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

This simply negates a variable, effectively turning it into its reciprocal value: if the variable 'Timecode.Enabled' is true it will be set to false, and vice versa.

From:
https://avosupport.de/wiki/ - **AVOSUPPORT**

Permanent link:
**https://avosupport.de/wiki/macros/control_structures?rev=1511102858**

Last update: **2017/11/19 14:47**