

xsl-slide.xsl

```
<?xml version="1.0"?>
```

0.1 slide style (plain)

For an example look at the generated slide

stylesheet

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/"<a novref=true
  text=@key href=n-transform.html>Transform</a>" version="1.0">
```

match

```
<xsl:output method="text"/>
<xsl:template match="/slideset">
```

@* generated (XML) slide

(extern) proto(s)

```
#@include proto-slideset.vr
#@include proto-slide.vr
#@include proto-text.vr
#@include proto-line.vr
#@include proto-break.vr
```

```
DEF slides slideset {
slides [
```

```
<xsl:apply-templates/>
] # slides
}
```

timer

```
DEF time TimeSensor { loop TRUE cycleInterval 2 }
DEF script Script {
eventIn SFTime pulse
eventOut SFInt32 next
url "javascript:
function pulse(value) { next = 1; Browser.print('x'); }
"
}
```

```

ROUTE time.cycleTime TO script.pulse
ROUTE script.next TO slides.next
# #<lt;/>
<xsl-value-of select="</gt;" />
</xsl:template>

<xsl:template match="*/slide">
slide { children [
  <xsl:apply-templates/>
] }
</xsl:template>

<xsl:template match="*/text">
text { lines [
  <xsl:apply-templates/>
] }
</xsl:template>

<xsl:template match="*/line">
line { string [ " <xsl:value-of select="@string" />
  <xsl:apply-templates/> " ] }
</xsl:template>

<xsl:template match="*/break">
line { string [ " <xsl:apply-templates/> " ] }
</xsl:template>

<xsl:template match="*/vrmf">
<xsl:apply-templates/>
</xsl:template>

<!-- ignore all not matched -->
<!-- xsl:template match="*" priority="-1" / -->

</xsl:stylesheet>

```