

# **Einführung in XSLT**

**Eine Programmiersprache zur Transformation von XML**

Tag 1 (heute)

# **Grundlagen und Einführung Elementare Sprachelemente**

Tag 2 (April)

# **Fortgeschrittene Sprachelemente Praxisbeispiele**

# Was ist XSLT?

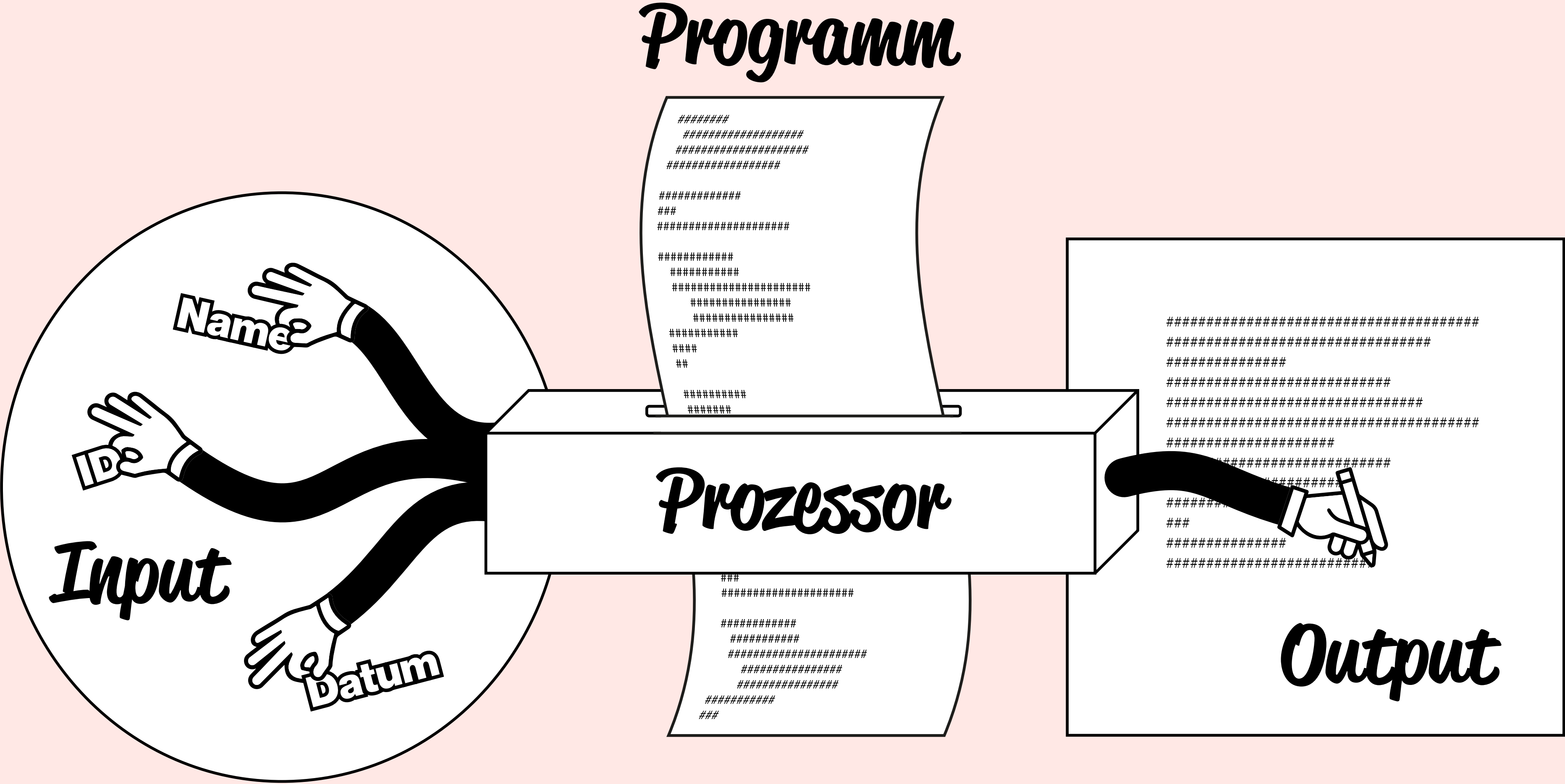
- XSLT = für XSL Transformation
- XSL = Extensible Stylesheet Language
- XSLT ist Bindeglied um aus XML (Content) XSL-FO (Presentation) zu erzeugen
- Während XSL durch CSS Paged Media verdrängt wird, ist XSLT (zumindest in der XML-Community) äußerst vital.
- Eine Turing-vollständige, funktionale, deklarative Programmiersprache um XML-Daten (oder andere Formate) in anders strukturierte XML-Daten (oder andere Formate) umzuwandeln.

# Versionsgeschichte

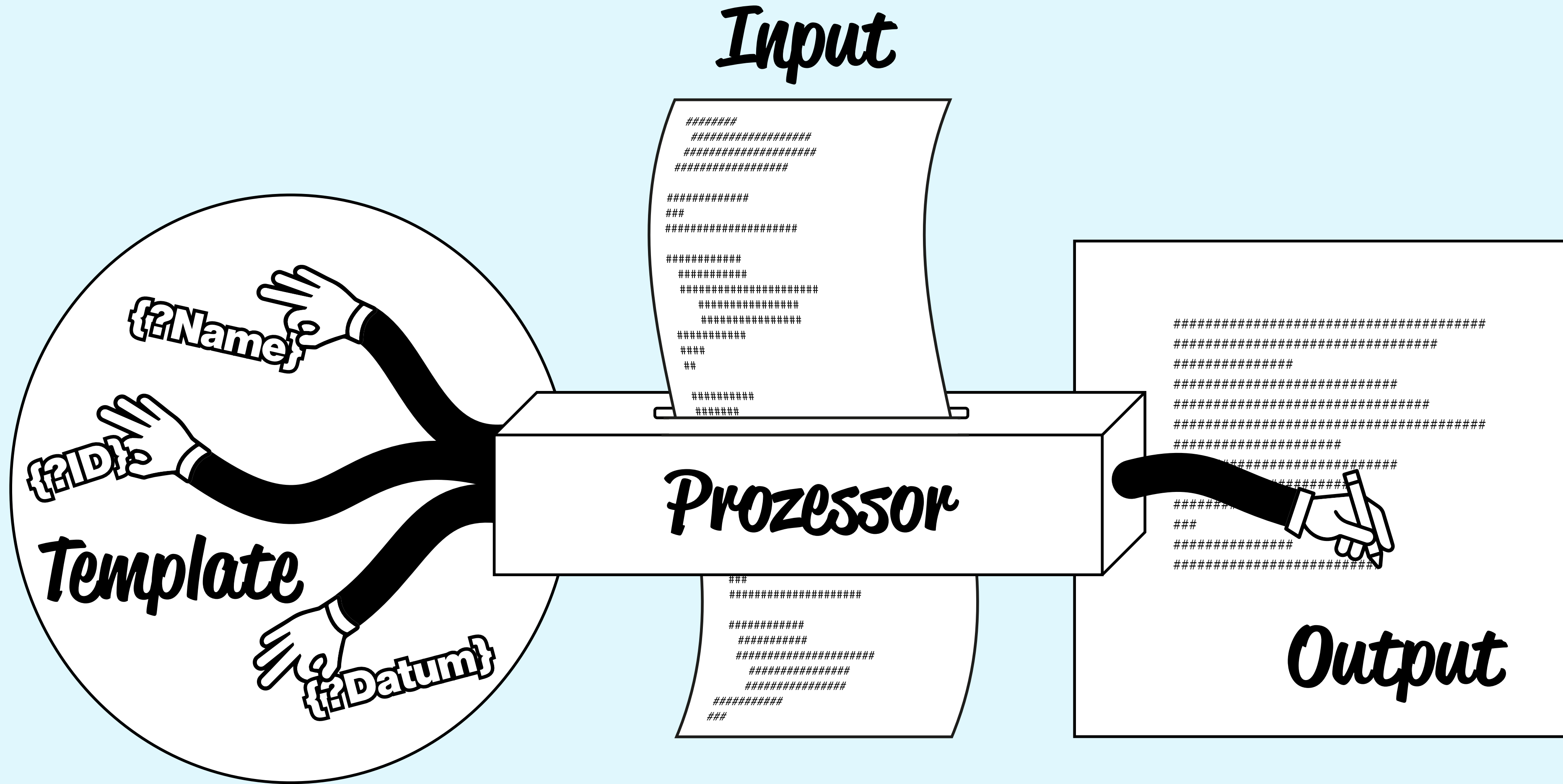
XSLT 1.0	1999	Teil des XSL-Standards
XSLT 2.0	2007	Grouping, Datentypisierung, u.v.m.
XSLT 3.0	2017	Arrays, Maps, JSON, viele neue Sprachelemente
XSLT 4.0	Draft	<a href="https://www.saxonica.com/qt4specs/XT/Overview-diff.html">https://www.saxonica.com/qt4specs/XT/Overview-diff.html</a>

XPath 1.0	1999	Abfragesprache für XML-Dokumente
XPath 2.0	2007	Sequenzen statt Node-Sets, Datentypisierung, for-Anweisung
XPath 3.0	2014	Higher Order Functions, let-Anweisung
XPath 3.1	2017	Maps und Arrays
XPath 4.0	Draft	<a href="https://www.saxonica.com/qt4specs/XP/xpath-40-diff.html">https://www.saxonica.com/qt4specs/XP/xpath-40-diff.html</a>

# Imperativ vs. Deklarativ



# Imperativ vs. Deklarativ



# PULL

# Programm

[illegible]

# Prozessor

```
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
```

[illegible]

# Output

# Input

# Datum

**Names**





# XSLT-Stylesheet

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">  
  
    <!--Hier steht der Code-->  
  
</xsl:stylesheet>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">  
  
    <!--Hier steht der Code-->  
  
</xsl:transform>
```

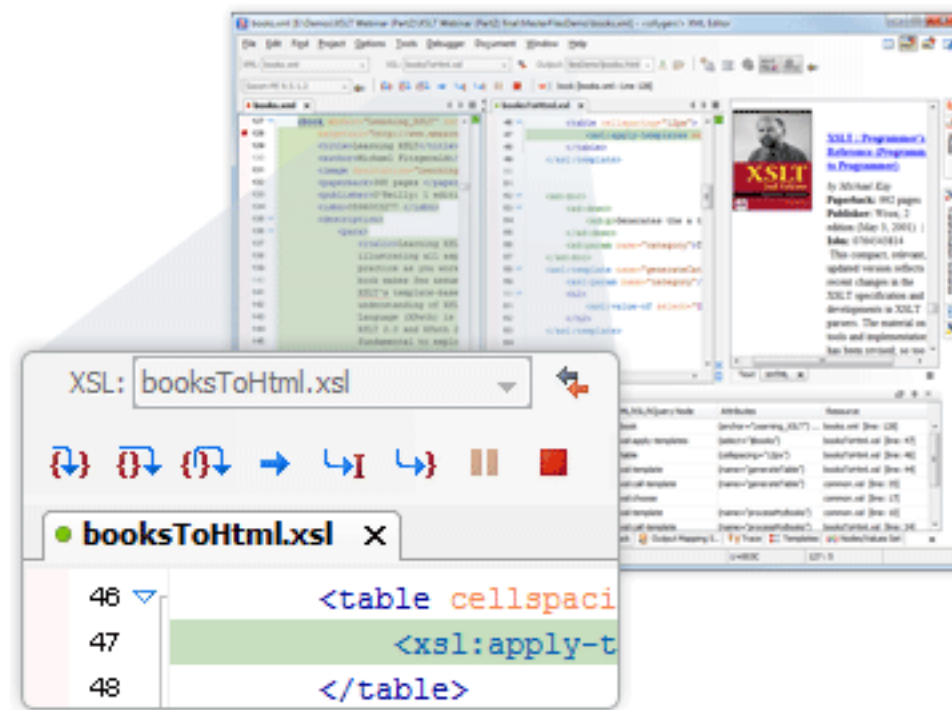
**Dateiendung: \*.xsl**

# XSLT Entwicklungsumgebungen

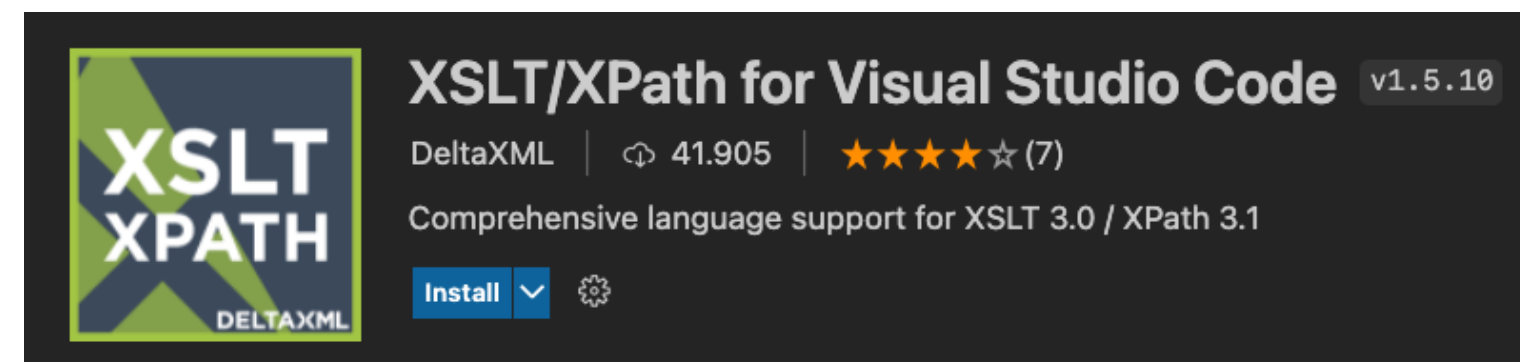
**Saxon** (<https://github.com/Saxonica/Saxon-HE>)

```
java -jar ~/saxon/saxon-he-10.9.jar -s:input.xml -xsl:stylesheet.xsl -o:output.xml
```

**Oxygen XML**



**Visual Studio Code**



**Online** (<http://xsltransform.net/>)

# **Exkurs**

## **Das XML-Dokument als Datenmodell**

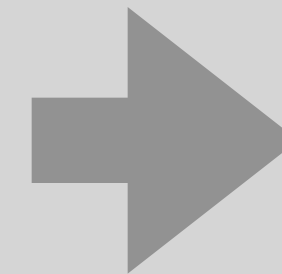
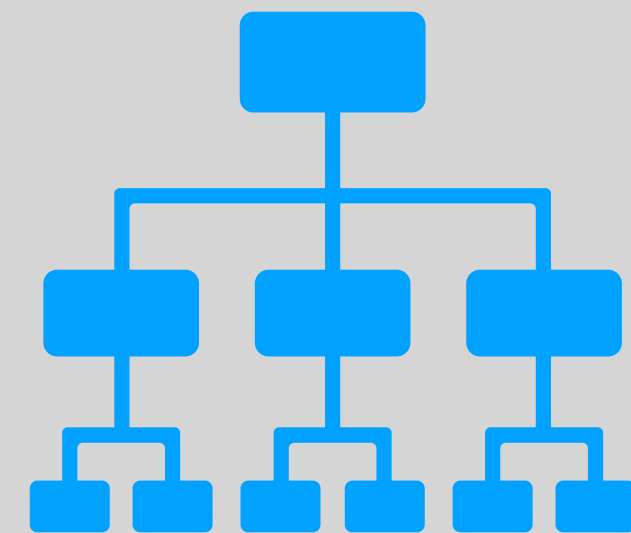
XML

Data-Model

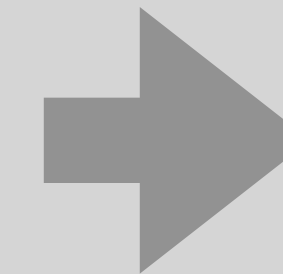
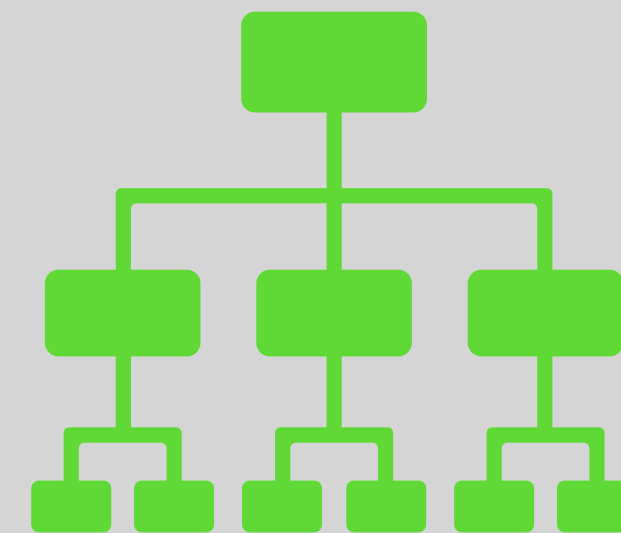
XML



parse



transform



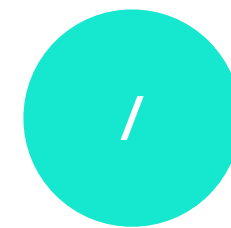
serialize



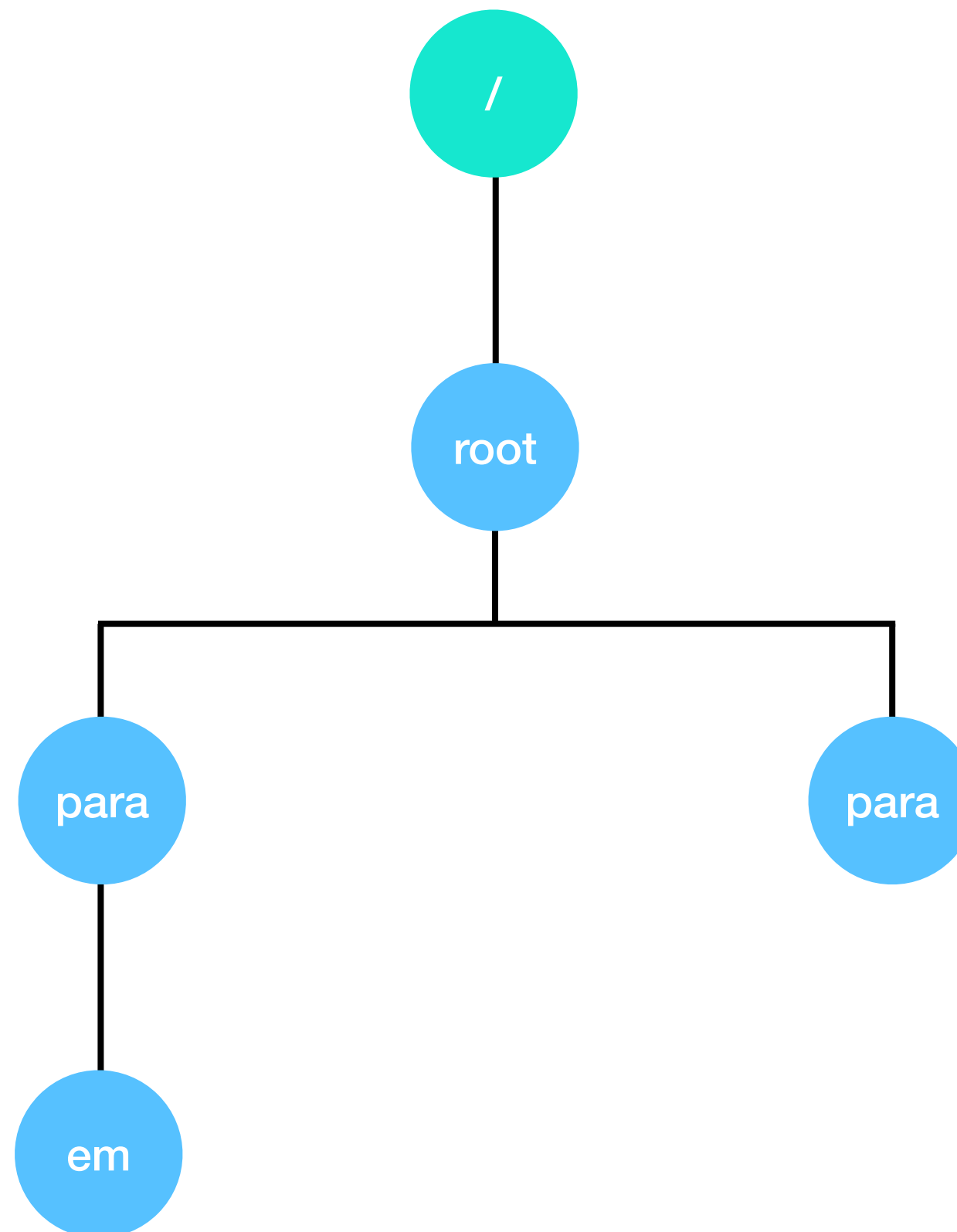
XSLT

```
<!--Beispiel-->
<root>
  <para>XSLT <em style="bold">rocks!</em></para>
  <para><?nr 1?>Punkt.</para>
</root>
```

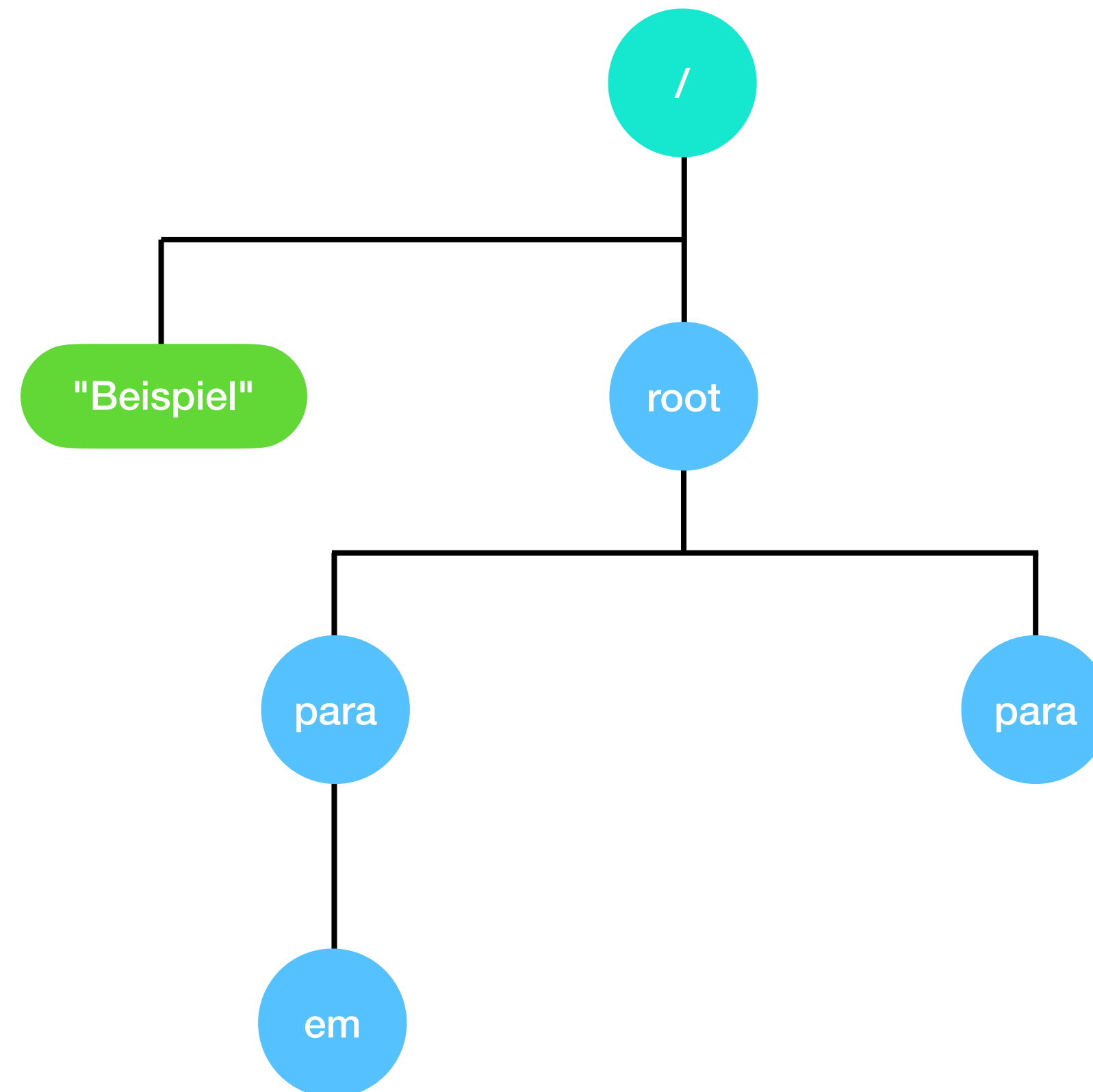
```
<!--Beispiel-->  
<root>  
  <para>XSLT <em style="bold">rocks!</em></para>  
  <para><?nr 1?>Punkt.</para>  
</root>
```



```
<!--Beispiel-->  
<root>  
  <para>XSLT <em style="bold">rocks!</em></para>  
  <para><?nr 1?>Punkt.</para>  
</root>
```

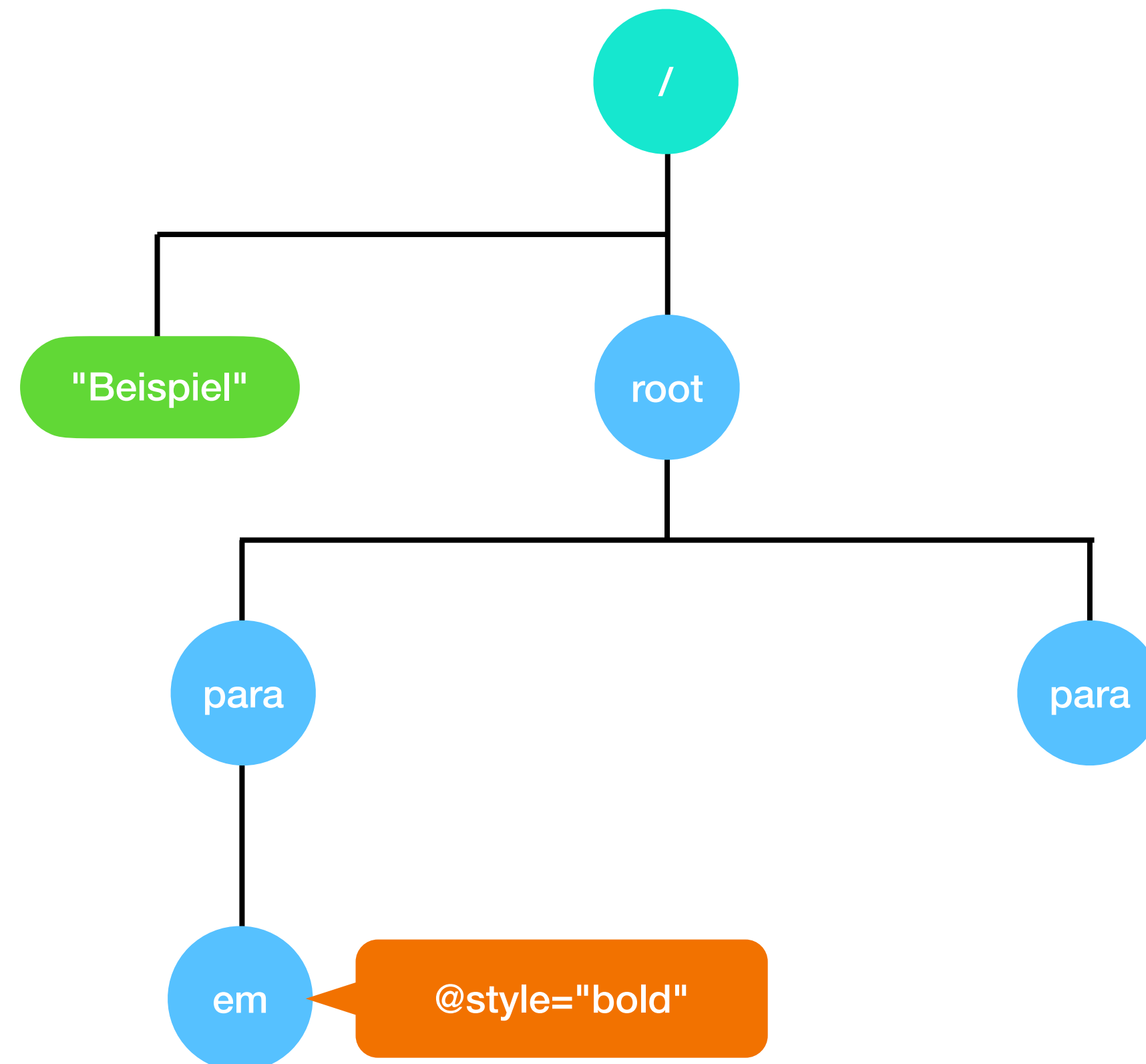


```
<!--Beispiel-->  
<root>  
  <para>XSLT <em style="bold">rocks!</em></para>  
  <para><?nr 1?>Punkt.</para>  
</root>
```

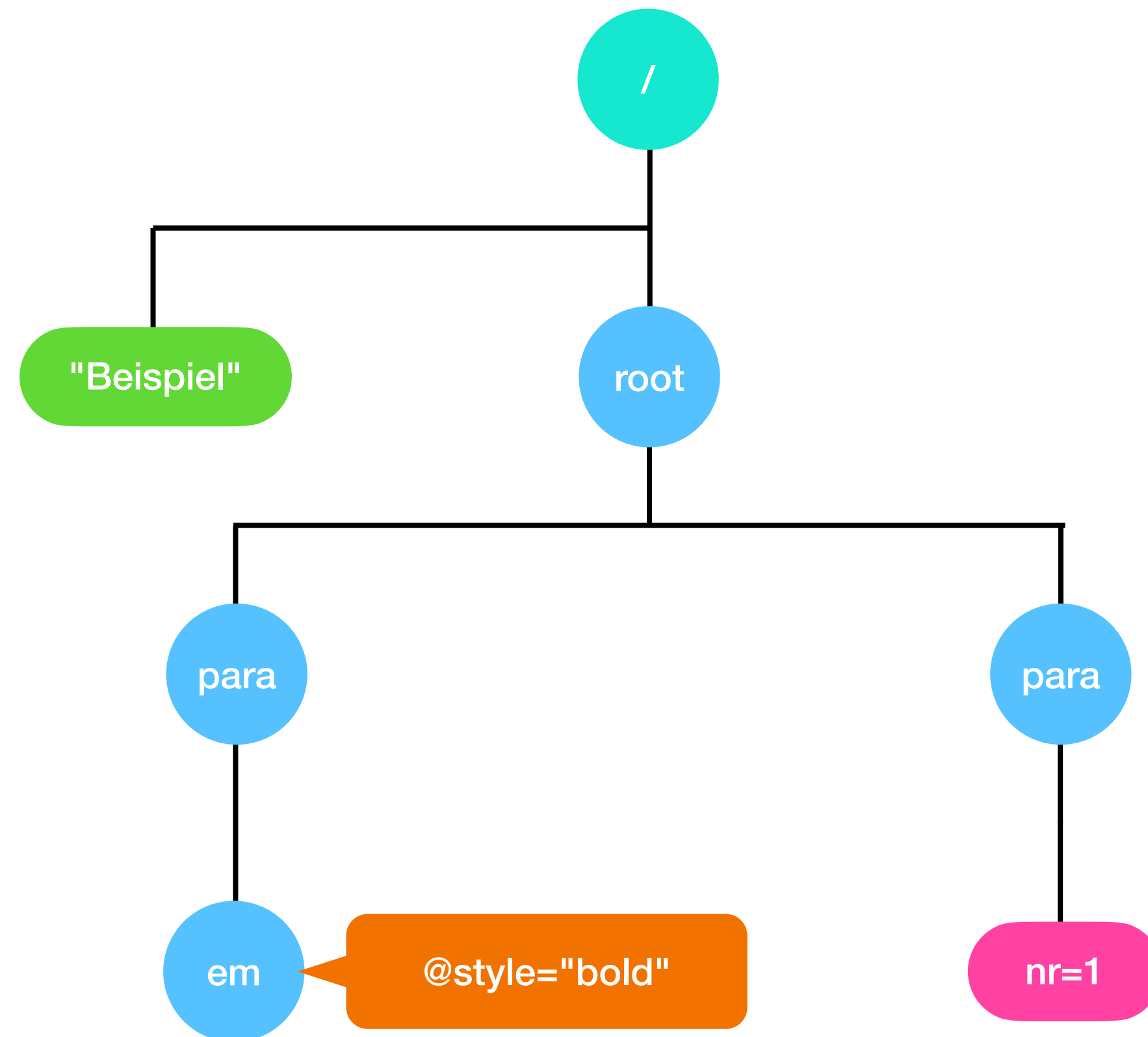




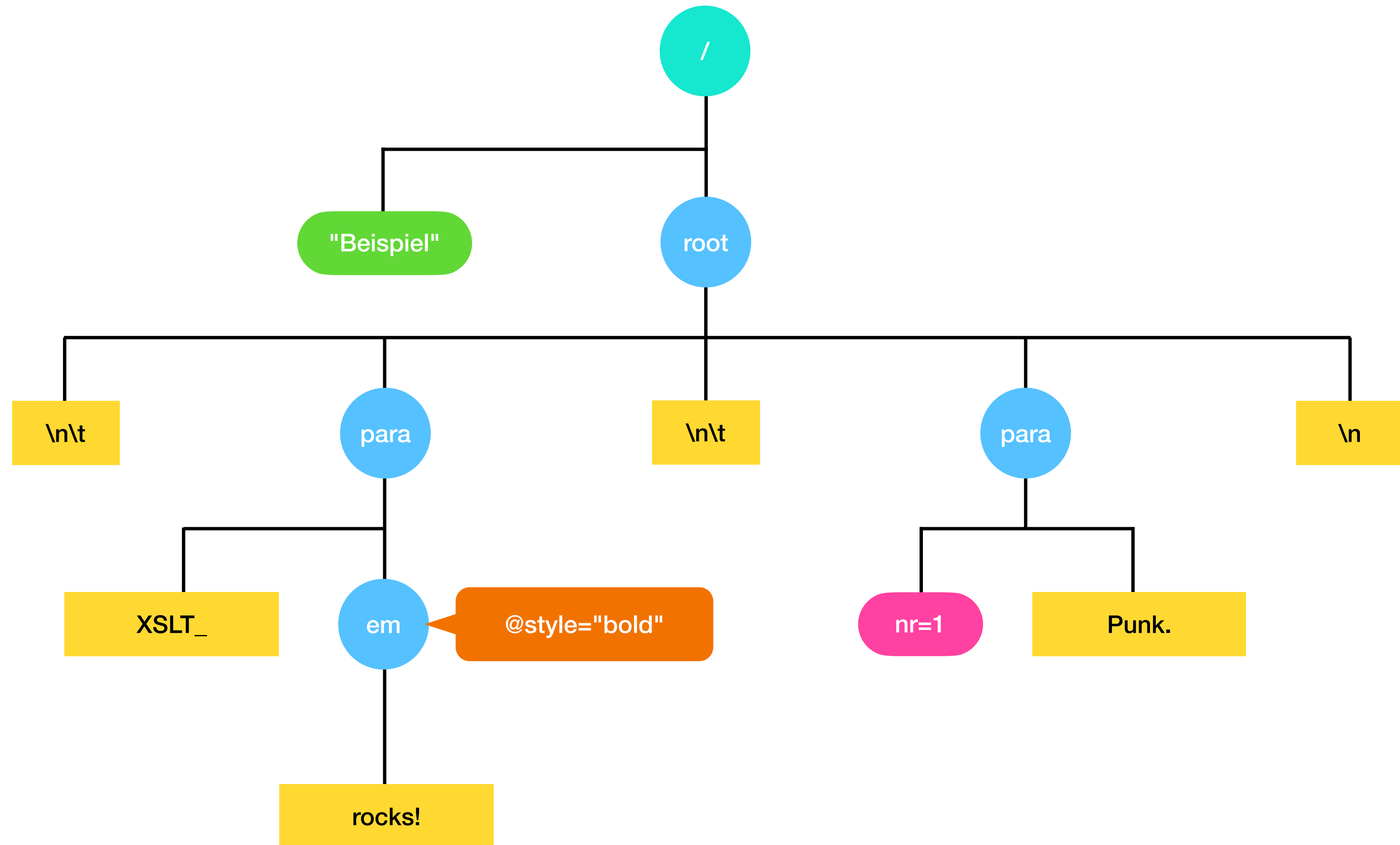
```
<!--Beispiel-->
<root>
  <para>XSLT <em style="bold">rocks!</em></para>
  <para><?nr 1?>Punkt.</para>
</root>
```



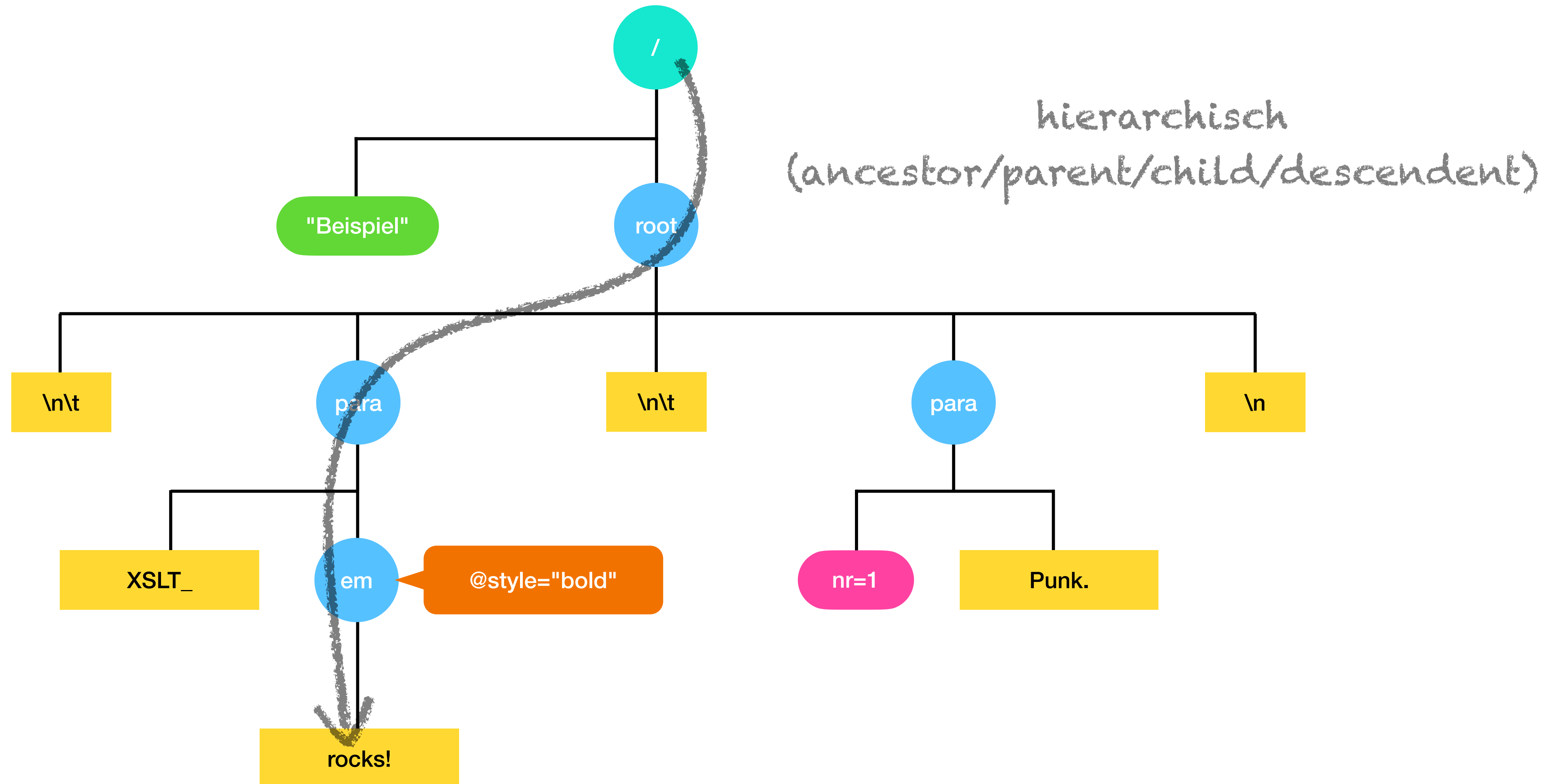
```
<!--Beispiel-->
<root>
  <para>XSLT <em style="bold">rocks!</em></para>
  <para><?nr 1?>Punkt.</para>
</root>
```



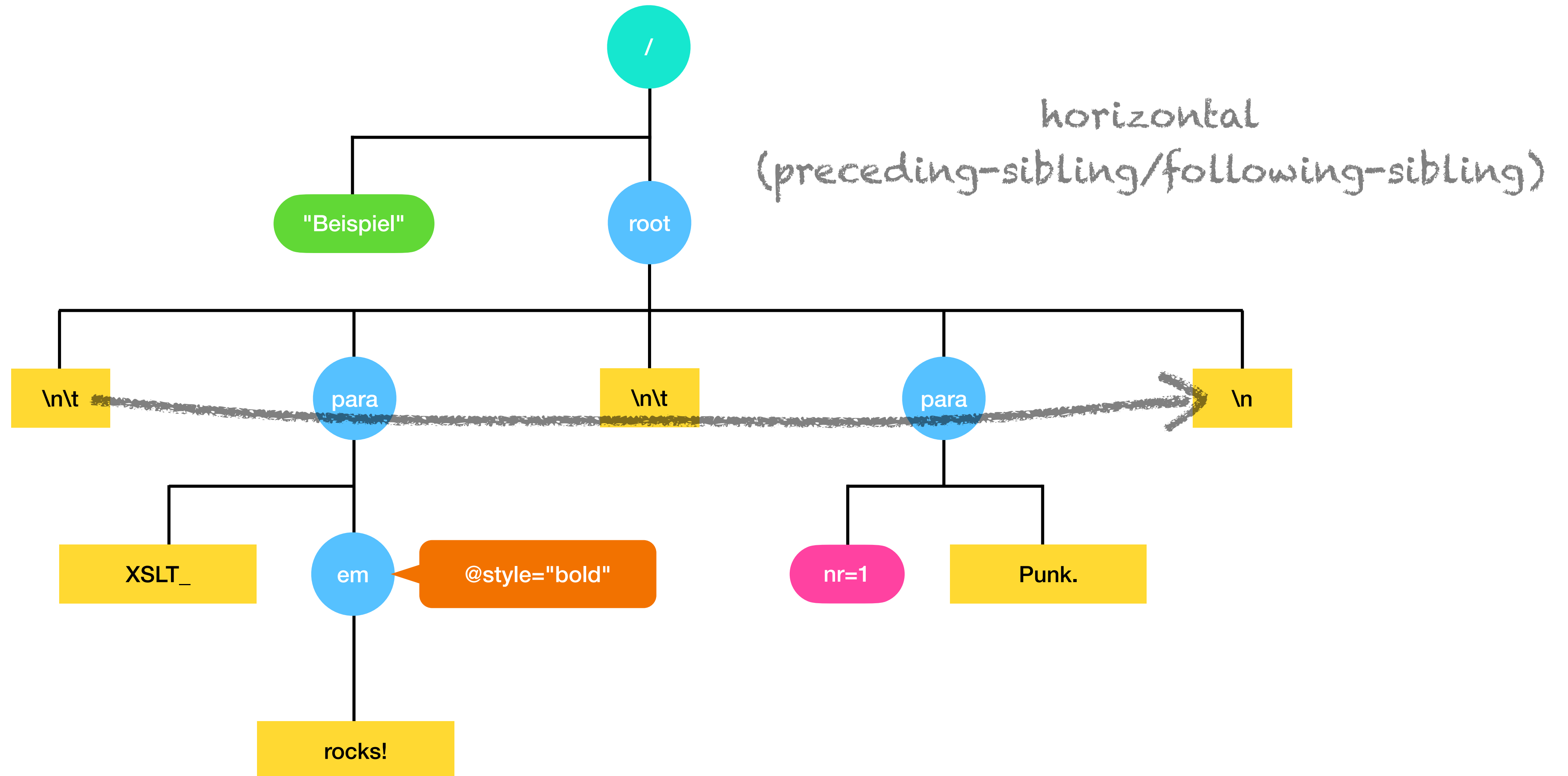
```
<!--Beispiel-->
<root>
  <para>XSLT <em style="bold">rocks!</em></para>
  <para><?nr 1?>Punkt.</para>
</root>
```



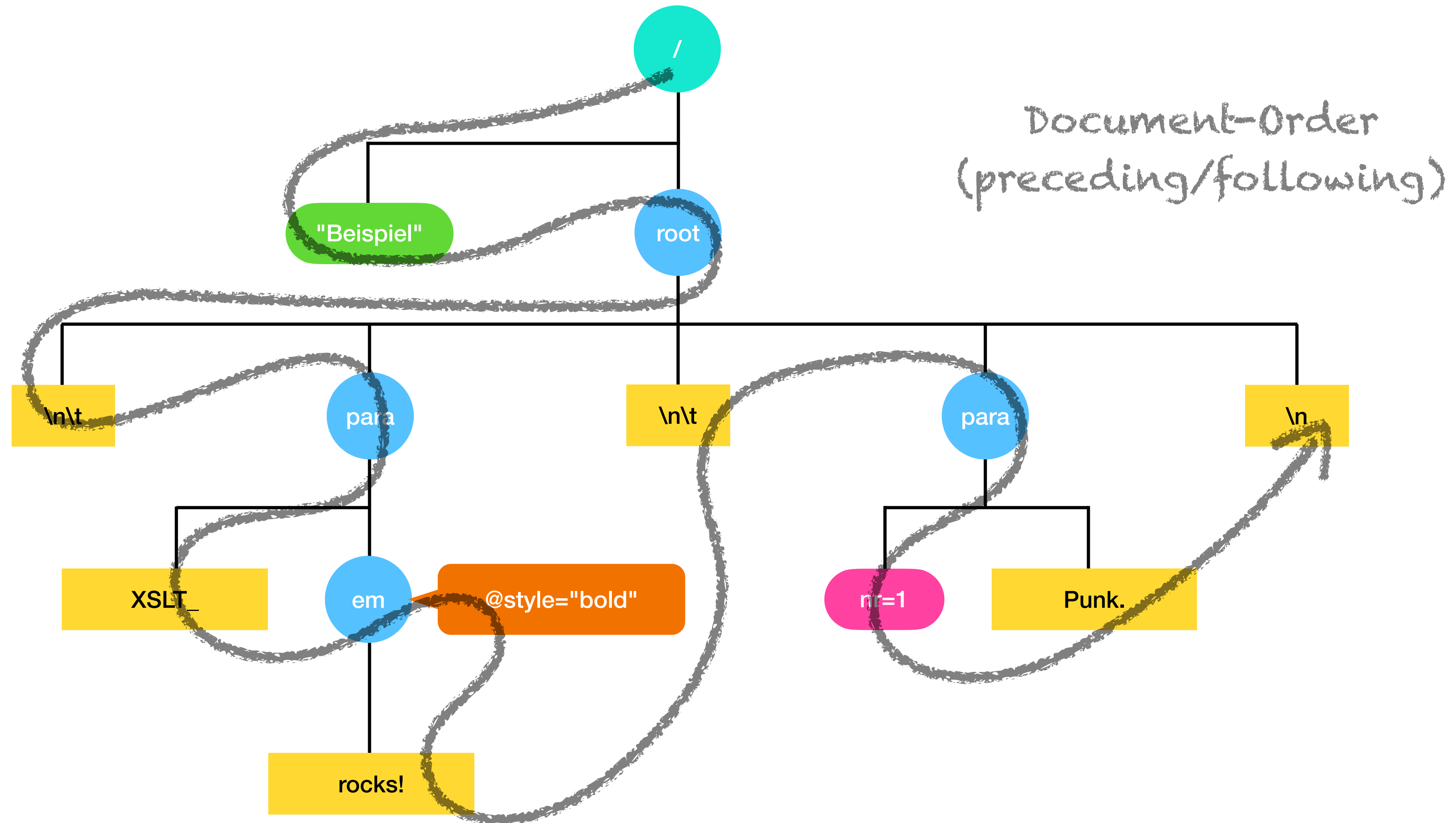
```
<!--Beispiel-->
<root>
  <para>XSLT <em style="bold">rocks!</em></para>
  <para><?nr 1?>Punkt.</para>
</root>
```




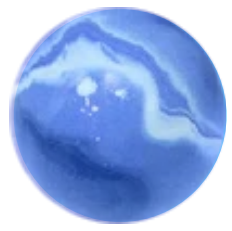




```
<!--Beispiel-->
<root>
  <para>XSLT <em style="bold">rocks!</em></para>
  <para><?nr 1?>Punkt.</para>
</root>
```



```
<!--Beispiel-->
<root>
  <para>XSLT <em style="bold">rocks!</em></para>
  <para><?nr 1?>Punkt.</para>
</root>
```



# XPath

	document-node()	/	node()	node()   @*
	element()	*		
	text()			
	comment()			
	processing-instruction()			
	attribute()	@*	⚠	