

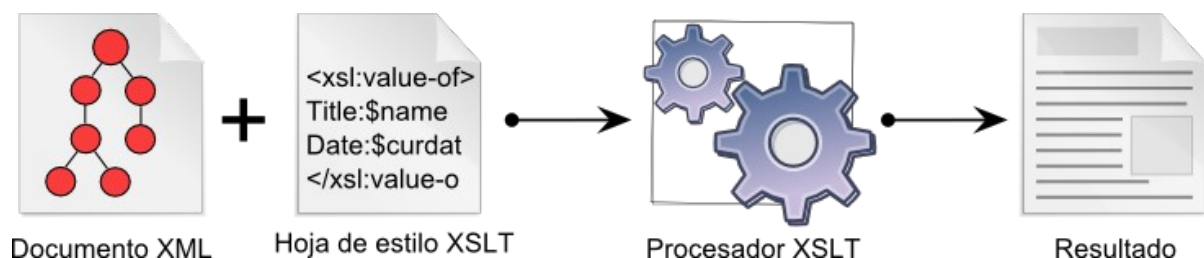
# XSLT: XSL Transformazioak

Askotan, XML dokumentu baten formatua aldatu behar izaten da:

- Adibidez, bi erakundek datu berberak gordetzeko XML-a erabili arren, izen desberdineko edo ordena desberdineko etiketak erabil ditzakete.
- Edo beste kasu batzuetan, hainbat dokumentutatik datorren informazioa bateratzeko beharrezkoa izan daiteke beste XML dokumentu bat sortzea, hala nola, artikulu baten eta bezero baten datuak bateratuta erakusten dituen eskaeren dokumentu berri bat.

XSLT (XSL Transformazioak) adierazpenezko programazio-lengoaia bat da, XML dokumentuetatik abiatuta, hauek eraldatzeko, haien egitura aldatzeko, dokumentuak fusionatzeko edo datuak iragazteko diseinatutako lengoaia da XSLT.

Jatorrizko dokumentuak XML izan behar badu ere, helmugak edozein formatu izan dezake: XML, HTML, testu laua..., irudi honek erakusten duen bezala:



- XML dokumentua emaitza sortuko denaren hasierako dokumentua da.
- XSLT estilo-orria programaren kodea duen dokumentua da, hau da, hasierako dokumentuari aplikatuko zaizkion eraldaketa-arauak jasotzen dituena.
- XSLT prozesadorea hasierako dokumentuari XSLT estilo-orrian sartutako transformazio-arauak aplikatzen dizkion eta azken dokumentua sortzen duen programa da.
- Programa gauzatzearen emaitza beste formato batean sortutako dokumentu berri bat da.

XML dokumentu bati XSLT estiloko hainbat orri aplika dakizkioke emaitza desberdinak lortzeko, eta XSLT estilo orri bera XML dokumentu desberdinetan aplikatu daiteke ere.

XSLT estilo-orri bat XML dokumentu bat da, eta gutxienez honako etiketa hauek ditu:

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

- <?xml> adierazpena, edozein XML dokumentukoa.
- <xsl:stylesheet> estilo-orriaren erro etiketa da, eta dituen atributuek dagokion [bertsioa](#) eta [izenen espazioa](#) adierazten dute.

## XML dokumentuak XSLT estilo orriekin lotzea

`<?xml-stylesheet?>` instrukzioaren bidez XML dokumentu bat XSLT estilo-orri batekin modu iraunkorrean lotu daiteke.

`<?xml-stylesheet?>` instrukzioa dokumentuaren hasieran doa, XML adierazpenaren ondoren.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml:stylesheet type="text/xsl" href="xsl-dokumentua.xsl"?>
```

## XSLT dokumentu baten prozesamendua

XSLT-ak zuhaitz moduan eraldatu beharreko dokumentua hartzen du sarreratzat, erro (/) elementutik prozesatzen hasten da, eta nodoetatik jaisten dihoa, bata bestearen atzetik.

XSLT dokumentu bat txantilo (template) multzo bat besterik ez da. Prozesamendua nodo jakin batera iristen denean *template*-ek zer irteera sortu behar den zehazten dute.

Prozesamendua honela egiten da:

- XSLT interpreteak une bakoitzean eraldatu beharreko XML dokumentuaren zuhaitzean nodo bat du, eta irteera bat dago prozesamenduaren emaitza bidaltzeko.
- Errotik hasten da prozesatzen, une horretan duen nodoarekin bat datorren match-a duen *template*-aren baten bila.
  - `<xsl:template>`: elementua txantiloak sortzeko erabiltzen da, eta honen match atributoak txantilo bat XML elementu batekin lotzeko erabiltzen da.
- *Template* bat aurkitzekotan, honen barruko jarraibideak exekutatzen dira, funtsean bi izan daitezkeenak (nahiz eta hizkuntzan askoz gehiago egon):
  - `<xsl:apply-templates>`: XSLT interpreteak zuhaitzean zehar nibel bat gehiago jaitsi behar duela adierazten du, eta oraingo nodoaren seme-alaba guztiak prozesatu. Etiketa honen `select` atributuaren bidez, seme-alaben artean zein nodo-multzoa prozesatu nahi den zehaztu daiteke ere.
  - `<xsl:value-of>`: XSLT interpreteak `select` atributuaren bidez zehaztutako nodoaren balioa irteerara bidali behar duela esan nahi du.

`match` eta `select` atributoetan Xpath estandar-aren adierazpenak jartzen dira. Adibidez, `match="/"` dokumentu osoa hautatzen du.

### Template lehenetsiak

XSLT-n ez da beharrezkoa *template*-rik idaztea jatorrizko XML dokumentuko nodo guztiak prozesatzeko; izan ere, *template* lehenetsi batzuk aplikatzen dira *template*-a definituta ez duten nodoetarako.

- Seme-alabak dituzten nodoetan interpretearen portaera lehenetsia uneko nodoaren seme-alaba guztiak prozesatzeko jaitea da.
- Edukia dituzten nodoetan interpretearen portaera lehenetsia daukaten testua inprimatzea da, `<xsl:value-of select="."/>` adierazpena sartu izan balitz bezala.

Aurreko teoriaren fitxategiko ereduarekin jarraituz (liburuak.xml):

Hurrengo adibidean, txantiloirik ez dagoenez, interpretea nodoen seme-alaba guztietatik jeisten dihoa eta irteerak `<izenburua>` eta `<egilea>` nodoen barne edukia hartzen du.

XSLT	Eraitza
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/T ransform" version="1.0"&gt; &lt;/xsl:stylesheet&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt;  Hobbita J.R.R. Tolkien  Erastunen jauna J.R.R. Tolkien  Maus Art Spiegelman</pre>

*Template*-a hutsik badago, prozesadoreak nodo horretarara heltzean, *template*-a prozesatu eta ez du irteerarik sortuko nodo horretarako azken dokumentuan.

XSLT	Eraitza
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/T ransform" version="1.0"&gt;   &lt;xsl:template match="/egilea"&gt;   &lt;/xsl:template&gt; &lt;/xsl:stylesheet&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt;  Hobbita  Erastunen jauna  Maus</pre>

*Template*-a hutsik badago, era berean, prozesadoreak ez du zuhaitzean nibelaz jeitsiko ezta seme-alabak prozesatuko ere.

*Template* hutsa erro-nodoan aplikatzen bada, prozesadoreak ez du irteerarik sortzen azken dokumentuan, ezta nodorik zeharkatzen ere.

XSLT	Eraitza
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/T ransform" version="1.0"&gt;   &lt;xsl:template match="/"&gt;   &lt;/xsl:template&gt; &lt;/xsl:stylesheet&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt;</pre>

`<xsl:apply-templates>` instrukzioa erabiliz gero, elementu erroaren seme-alaba diren nodo guztiak prozesatuko dira, eta horrela, lehenetsitako funzioamenduari jarraituz, interpretearen irteeran azken nodoetako testuak agertuko dira berriro.

XSLT	Eraitza
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/T ransform" version="1.0"&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt;  Hobbita J.R.R. Tolkien  Erastunen jauna</pre>

<pre>&lt;xsl:template match="/"&gt;   &lt;xsl:apply-templates/&gt; &lt;/xsl:template&gt; &lt;/xsl:stylesheet&gt;</pre>	<p>J.R.R. Tolkien</p> <p>Maus</p> <p>Art Spiegelman</p>
--	---

`<xsl:apply-templates>` erabiliz select atributoarekin batera, interpreteak prozesatzen jarraitu behar duen nodoaren seme-alabak defini daitezke.

Hurrengo adibidean, interpretea liburu bat aztertzen ari denean, `<egilea>` seme-alaba nodoak soilik interpretatzen jarraituko du horrela adierazi zaiolako.

XSLT	Emitza
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/T ransform" version="1.0"&gt;   &lt;xsl:template match="liburua"&gt;     &lt;xsl:apply-templates       select="egilea"/&gt;   &lt;/xsl:template&gt; &lt;/xsl:stylesheet&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt;</pre> <p>J.R.R. Tolkien</p> <p>J.R.R. Tolkien</p> <p>Art Spiegelman</p>

Hurrengo adibidean emitza bera lortzen da, baina beste era batean eginda.

Behin interpretea liburu nodoa aztertzen ari denean, txantiloiak adierazten dio `<egilea>` seme-alaba nodoaren edukia erakusteko, baina ez da beste *template*-rik aplikatzen eta prozesadoreak ez du mailaz jaisten; beraz, ez dira kontuan hartuko gainerako seme-alabak ezta ondorengoak ere.

XSLT	Emitza
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/T ransform" version="1.0"&gt;   &lt;xsl:template match="liburua"&gt;     &lt;xsl:value-of="egilea"/&gt;   &lt;/xsl:template&gt; &lt;/xsl:stylesheet&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt;</pre> <p>J.R.R. Tolkien</p> <p>J.R.R. Tolkien</p> <p>Art Spiegelman</p>

## Atributuak sortu

Demagun atributu bat duen etiketa bat sortu nahi dela, eta atributua jatorrizko dokumentuan beste etiketa bat zela.

Adibidez, *template* hau (ez da zuzena) `<liburua>` etiketa bihurtzen saiatzen da, izenburuaren izena atributu moduan duen beste etiketa batean.

```
<xsl:template match="liburua">
  <liburua izena="<xsl:value-of select="izenburua"/>" />
</xsl:template>
```

Aurreko tenplatearekin dagoen arazoa da XMLa ez dagiela ondo eratua, etiketa bat ezin baitu beste baten barruan "txertatuta" egon. Hau dela eta, kasu horietarako hautapena adierazteko modu alternatibo bat dago, adierazpena giltzen artean jartzea.

```
<xsl:template match="liburua">
  <liburua izena="{izenburua}"/>
</xsl:template>
```

Hurrengo ereduan, adibideko XML-aren informazio bera duen XML bat lortu nahi da, baina beste ordenean eta beste etiketa egitura batekin informazioa biltzen duenak.

XML	XSLT
<pre>&lt;?xml version="1.0" encoding="UTF-8"&gt; &lt;liburuak&gt;   &lt;liburua argitaratzea="1937"&gt;     &lt;egilea&gt;       &lt;izena&gt;J.R.R. Tolkien&lt;/izena&gt;       &lt;jaiotza&gt;03/01/1892&lt;/jaiotza&gt;     &lt;/egilea&gt;     &lt;izena&gt;Hobbita&lt;/izena&gt;   &lt;/liburua&gt;   &lt;liburua argitaratzea="1954"&gt;     &lt;egilea&gt;       &lt;izena&gt;J.R.R. Tolkien&lt;/izena&gt;       &lt;jaiotza&gt;03/01/1892&lt;/jaiotza&gt;     &lt;/egilea&gt;     &lt;izena&gt;Erastunen jauna&lt;/izena&gt;   &lt;/liburua&gt;   &lt;liburua argitaratzea="1991"&gt;     &lt;egilea&gt;       &lt;izena&gt;Art Spiegelman&lt;/izena&gt;       &lt;jaiotza/&gt;     &lt;/egilea&gt;     &lt;izena&gt;Maus&lt;/izena&gt;   &lt;/liburua&gt; &lt;/liburuak&gt;</pre>	