

# Curso Desenvolvedor Java



# ***XSLT***

Permite transformar XML em outros formatos

Utiliza elementos XML do namespace

<http://www.w3.org/1999/XSL/Transform>

Acessa os elementos e atributos do XML por nome

Oferece controle de fluxo para o processamento

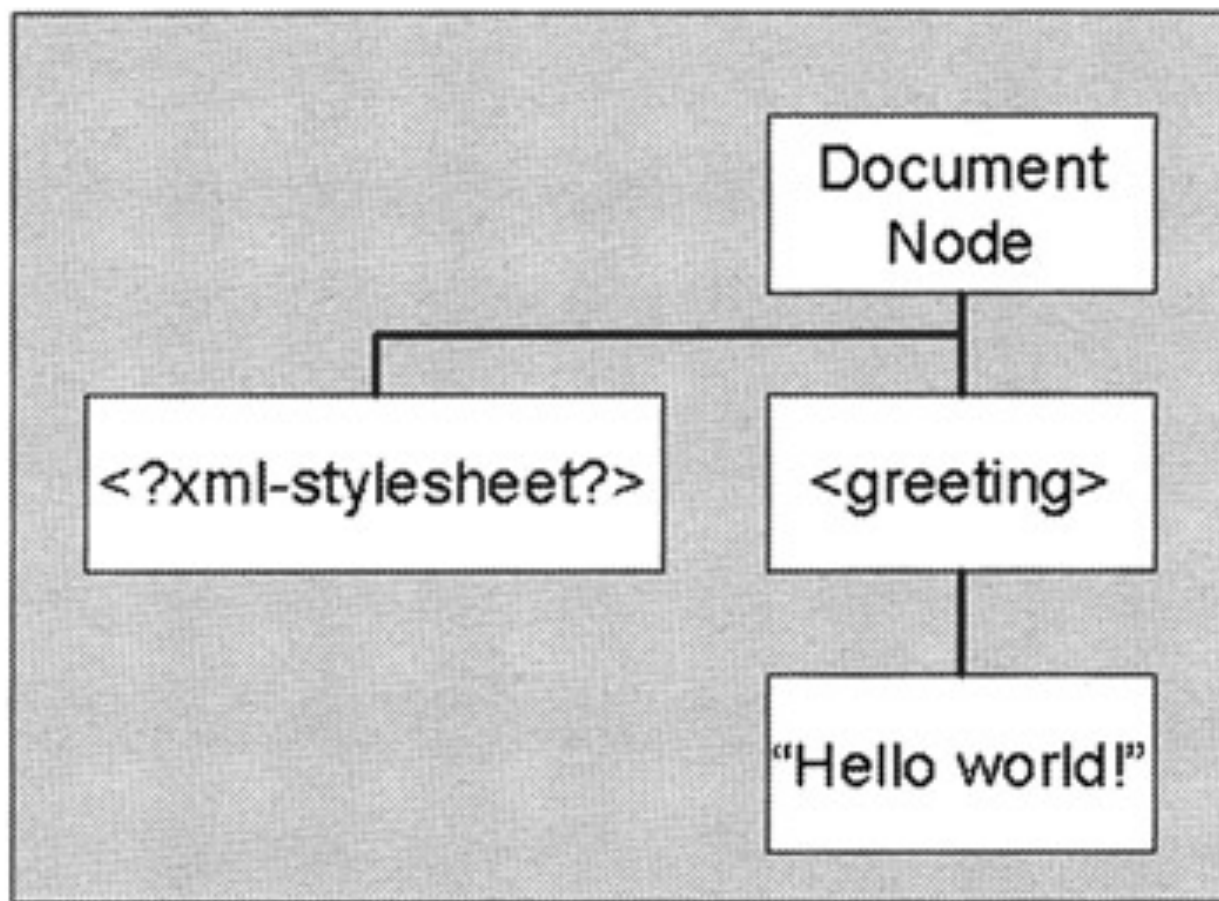
Utilizado quando há necessidade de separação entre dados e a forma como estes são apresentados

Também utilizado na transmissão de dados entre aplicações

## Um Exemplo de XML

```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/xsl" href="hello.xsl"?>  
<greeting>Hello, word!</greeting>
```

## A Representação dos Elementos XML



## Saída pretendida

```
<html>
  <head>
    <title>Today's greeting'</title>
  </head>
  <body>
    <p>Hello, word!</p>
  </body>
</html>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <head>
        <title>Today's greeting'</title>
      </head>
      <body>
        <p><xsl:value-of select="greeting" /></p>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

A aplicação de estilo em java acontece com a utilização de uma instância de **javax.xml.transform.Transformer**

Exemplo da obtenção da instância de Transformer

```
TransformerFactory factory = TransformerFactory.newInstance();  
Transformer t = factory.newTransformer(new StreamSource(style));
```



A transformação de XML ocorre a partir da chamada do método transform da instância de Transformer.

```
t.transform(new StreamSource(source), new StreamResult(System.out));
```

Existe um conjunto de anotações criados para permitir o processamento de XML através da API JAXB

As classes contendo as anotações são geradas a partir da DTD ou XML Schema que descreve um documento XML

O comando utilizado para a produção das classes Java chama-se “**xjc**”

Encontra-se no diretório “**bin**” da instalação do JDK

Sua sintaxe é:

```
xjc -p nome_do_pacote_java -dtd nome_do_documento_dtd
```

Exemplo de anotações geradas nas classes pelo **xjc**

```
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name = "", propOrder = {
    "prenome",
    "sobrenome"
})
@XmlRootElement(name = "nome")
public class Nome {

    @XmlElement(required = true)
    protected String prenome;
    @XmlElement(required = true)
    protected String sobrenome;
```

Após a geração das classes pelo xjc é possível carregar em objetos as informações contidas em XML através da utilização da API JAXB

Também é possível destes objetos gerar um novo XML

O processo de leitura de XML para gravação de objetos chama-se: Unmarshal

O processo de leitura de Objetos para gravação de XML chama-se: Marshal

Para podermos utilizar Marshal ou Unmarshal necessitamos da referência de um contexto JAXB

## Exemplo de criação um contexto JAXB

```
JAXBContext ctx =  
    JAXBContext.newInstance(Xml2.Pessoas.class);
```

## Exemplo de Unmarshal

```
Unmarshaller u = ctx.createUnmarshaller();  
Pessoas pessoas = (Pessoas)u.unmarshal(  
    new File("src/Xml2/hobby.xml"));
```

## Exemplo de Marshal

```
Marshaller m = ctx.createMarshaller();  
m.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);  
m.marshal(pessoas, System.out);
```