

XML – eXtensible Markup Language

Prof. Fellipe Aleixo – fellipe.aleixo@ifrn.edu.br

XML

- Projetada para transportar e armazenar dados (estruturados)
- Importante conhecer e fácil de entender
- Exemplo de documento XML:

```
<?xml version="1.0"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

HTML vs. XML

HTML

- Criada para exibir dados/informações
- Foco – como os dados irão aparecer
- *Tags* predefinidas
- Podem haver tags unitárias, ex.: <P> e

XML

- Criada para transportar e armazenar dados
- Foco – o que os dados significam
- Tags “inventadas”
- Todas as *tags* precisam ter início e fim (
)

Exemplo:

Tag de início → <html>

Tag de fim → </html>

Importância do XML

- Hoje,
 - XML é tão importante para a Web quanto HTML foi para a início da Web
 - XML é a “ferramenta” mais comum para a transmissão de dados entre todos os tipos de aplicações

Aplicações do XML

- Separar os dados do HTML
- Simplifica o compartilhamento de dados
- Simplifica do transporte dos dados
- Simplifica as mudanças de plataforma
- Possibilita diferentes formas de acesso aos dados

Aplicações do XML

- Criar novas “linguagens” para a Internet, como por exemplo:
 - XHTML
 - WSDL – descrever serviços *Web*
 - WAP e WML – linguagens de marcação para dispositivos móveis
 - RSS – para sistemas de notícias
 - RDF e OWL – para descrever recursos e ontologias
 - SMIL – para descrever multimídia para a *Web*

Estrutura

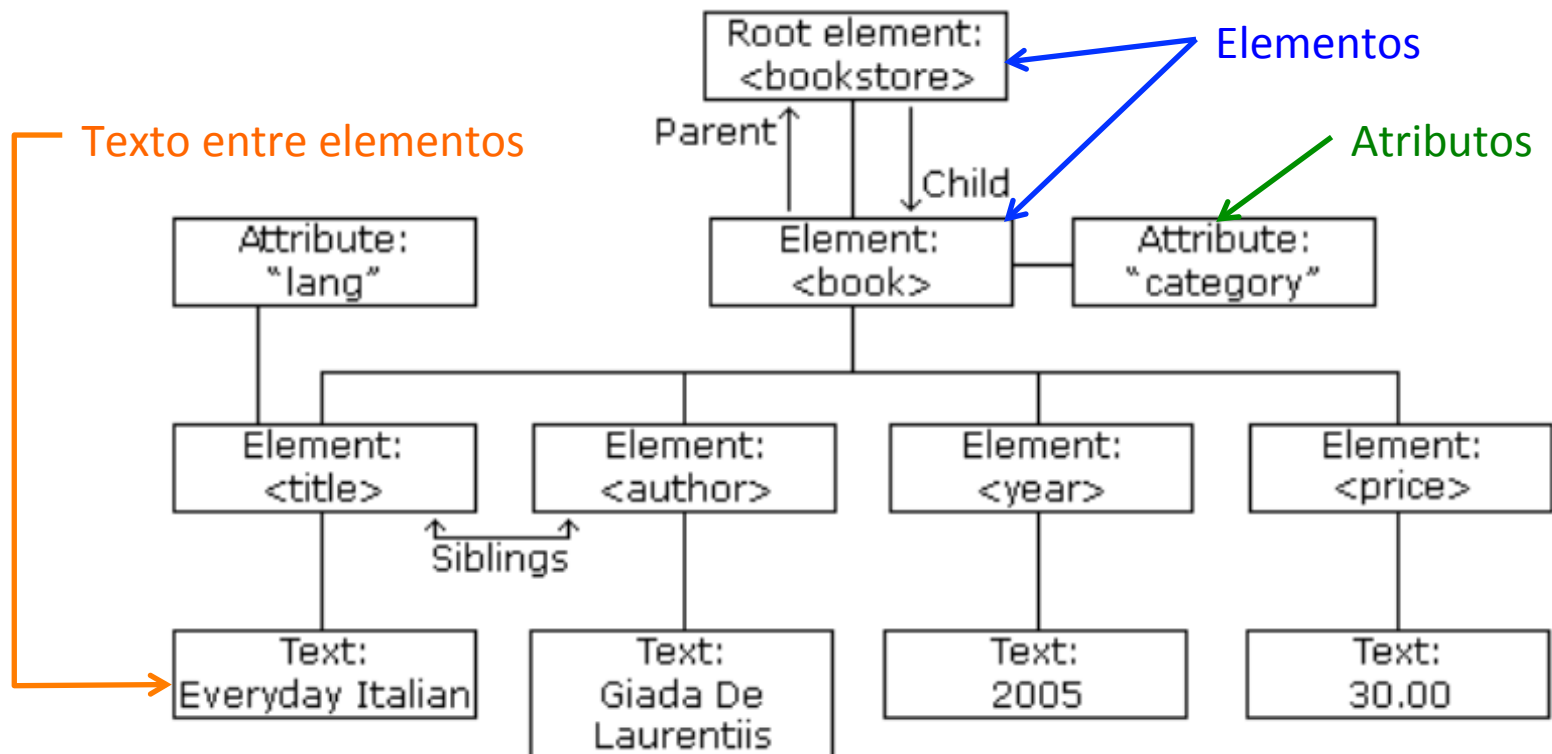
- Documentos XML formam uma estrutura de árvore – com um elemento como “raiz”, interligada a outros elementos, até os elementos terminadores ou “folhas”

```
<?xml version="1.0"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

- Primeiro elemento – indica versão do XML e codificação utilizada
- Elemento raiz = “note” → este documento trata-se de um “aviso”
- Os próximos quatro elementos são “filhos” (propriedades) da raiz

Estrutura de Árvore

- Um documento XML deve conter um elemento raiz, como “pai” de todos os demais elementos
- Exemplo:



Estrutura de Árvore

```
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

Tags XML

- *As tags XML*
 - são sensíveis ao caso
 - diferencia maiúsculas e minúsculas
 - precisam ser corretamente aninhadas
 - Exemplo:

```
<b><i>This text is bold and italic</b></i>
```

- Funciona no HTML, mas incorreto segundo a definição XML
- Um documento XML só pode ter um elemento raiz
- Valores de atributos deve vir entre “aspas duplas”

Tags XML

- Caracteres especiais:

<	<	menor que
>	>	maior que
&	&	“e” comercial
'	'	apostrofe
"	"	aspas duplas

- Comentários em XML:

```
<!-- Este é um comentário -->
```

Elementos XML

- Tudo o que é definido entre uma *tag* de início e uma de fim
- Um elemento pode conter:
 1. Outros elementos
 2. Texto
 3. Atributos
 4. Ou uma coleção dos itens acima
- Os nomes dos elementos devem seguir as seguintes regras:
 - Podem conter letras, números e outros caracteres
 - Não podem iniciar com um número ou caractere de pontuação
 - Não podem iniciar com as letras “xml” (ou “XML”, ou “Xml”, etc.)
 - Não podem conter espaços

Melhores Práticas para Nomear Elementos

- Use nomes claros e auto-descritivos
 - Pode fazer uso do sublinha “_”
 - Exemplo: `<primeiro_nome>` ou `<último_nome>`
- Nomes devem ser curtos e simples
 - Bom exemplo: `<book_title>`
 - Mal exemplo: `<the_title_of_the_book>`
- Evite o uso do hífen “-”
- Evite o uso do “.”
- Evite o uso dos “:”

Por que Extensível?

- Inicialmente podemos ter:

```
<note>  
<to>Tove</to>  
<from>Jani</from>  
<body>Don't forget me this weekend!</body>  
</note>
```

- Depois podemos adicionar algumas informações:

```
<note>  
<date>2008-01-10</date>  
<to>Tove</to>  
<from>Jani</from>  
<heading>Reminder</heading>  
<body>Don't forget me this weekend!</body>  
</note>
```

XML Bem Formado

- Um XML com a sintaxe correta é um XML “bem formado”
- Um XML “válido” é um XML bem formado, o qual está conforme as regras de um *Document Type Definition*, ou DTD

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<!DOCTYPE note SYSTEM "Note.dtd">  
<note>  
<to>Tove</to>  
<from>Jani</from>  
<heading>Reminder</heading>  
<body>Don't forget me this weekend!</body>  
</note>
```

- A declaração DOCTYPE referencia um DTD externo

DTD

- O propósito de um DTD é definir a estrutura de um documento XML

```
<!DOCTYPE note  
[  
  <!ELEMENT note (to,from,heading,body)>  
  <!ELEMENT to (#PCDATA)>  
  <!ELEMENT from (#PCDATA)>  
  <!ELEMENT heading (#PCDATA)>  
  <!ELEMENT body (#PCDATA)>  
]>
```


XML Schema

- A W3C suporta um documento (baseado em XML) alternativo ao DTD, chamado de XML Schema (extensão XSD)

```
<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Formatando Exibição de XML com CSS

- É possível usar CSS para formatar a visualização de um XML
 - Exemplo:
 - [Um catálogo de CDs em XML](#)
 - [O arquivo CSS](#)
 - [O catálogo formatado pelo CSS](#)
 - Formatar arquivos com XML com CSS **NÃO** é uma forma comum
 - Para esta finalidade a W3C recomenda o uso de XSTL

Formatando Exibição de XML com XSTL

- XSTL é o mecanismo de folhas de estilo próprio para XML
 - Mais sofisticado do que o CSS
- XSLT – *eXtensible Stylesheet Language Transformations*
 - Pode ser usado para transformar XML em HTML
 - Exemplo:
 - <http://www.w3schools.com/xml/tryxslt.asp?xmlfile=simple&xsltfile=simple>

UTILIZAÇÕES DO XML NA IMPLEMENTAÇÃO DE SISTEMAS WEB

XML nos Sistemas Web

- Um desenvolvedor de sistemas para a Web irá encontrar XML nas seguintes ocasiões:
 1. Armazenamento e transporte de informações
 2. Definição e comunicação com serviços Web
 - WSDL e SOAP
 3. Na forma de descritores de implantação de aplicações
 - Ex.: “web.xml” para aplicações Web em Java
 4. Na forma de descritores dos serviços de um contêiner
 - Ex.: “faces-config.xml” para configuração do JSF
 5. Na forma de descritores de mapeamento objeto relacional
 - Ex.: “persistence.xml” da JPA – *Java Persistence API*

Exemplo de “web.xml”

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/
xml/ns/j2ee/web-app_2_4.xsd"
  version="2.4">

  <display-name>HelloWorld Application</display-name>
  <description>
    This is a simple web application with a source code organization
    based on the recommendations of the Application Developer's Guide.
  </description>

  <servlet>
    <servlet-name>HelloServlet</servlet-name>
    <servlet-class>examples.Hello</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>HelloServlet</servlet-name>
    <url-pattern>/hello</url-pattern>
  </servlet-mapping>

</web-app>
```

Outro Exemplo de “web.xml”

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
3 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
5
6 <context-param>
7   <param-name>debug</param-name>
8   <param-value>>false</param-value>
9 </context-param>
10
11 <session-config> <!-- 10 minutes -->
12   <session-timeout>10</session-timeout>
13 </session-config>
14
15 <servlet>
16   <display-name>Servlet1</display-name>
17   <servlet-name>Servlet1</servlet-name>
18   <servlet-class>test.Servlet1</servlet-class>
19   <init-param>
20     <param-name>sleep-time-in-seconds</param-name>
21     <param-value>10</param-value>
22   </init-param>
23   <load-on-startup>1</load-on-startup>
24 </servlet>
25
26 <servlet-mapping>
27   <servlet-name>Servlet1</servlet-name>
28   <url-pattern>/Servlet1</url-pattern>
29 </servlet-mapping>
30
31 <env-entry>
32   <description>admin email</description>
33   <env-entry-name>adminEmail</env-entry-name>
34   <env-entry-value>admin@example.x</env-entry-value>
35 </env-entry>
```

Exemplo de “web.xml”

```
37 <resource-ref>
38   <res-ref-name>HRDS</res-ref-name>
39   <res-type>javax.sql.DataSource</res-type>
40   <mapped-name>jdbc/__default</mapped-name>
41 </resource-ref>
42
43 <filter>
44   <display-name>Filter1</display-name>
45   <filter-name>Filter1</filter-name>
46   <filter-class>Filter1</filter-class>
47 </filter>
48
49 <filter-mapping>
50   <filter-name>Filter1</filter-name>
51   <url-pattern>/Filter1</url-pattern>
52 </filter-mapping>
53
54 <filter-mapping>
55   <filter-name>Filter1</filter-name>
56   <servlet-name>Servlet1</servlet-name>
57   <dispatcher>REQUEST</dispatcher>
58   <dispatcher>FORWARD</dispatcher>
59   <dispatcher>INCLUDE</dispatcher>
60   <dispatcher>ERROR</dispatcher>
61 </filter-mapping>
62
63 <listener>
64   <listener-class>Listener1</listener-class>
65 </listener>
66
67 <welcome-file-list>
68   <welcome-file>index.html</welcome-file>
69   <welcome-file>index.htm</welcome-file>
70   <welcome-file>index.jsp</welcome-file>
71   <welcome-file>default.html</welcome-file>
72   <welcome-file>default.htm</welcome-file>
73   <welcome-file>default.jsp</welcome-file>
74 </welcome-file-list>
75
76 </web-app>
```



```

<?xml version='1.0' encoding='UTF-8'?>
<faces-config xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
facesconfig_1_2.xsd"
version="1.2">
  <managed-bean>
    <managed-bean-name>login</managed-bean-name>
    <managed-bean-class>br.com.urubatan.jsfjpassec.Login</managed-bean-class>
    <managed-bean-scope>session</managed-bean-scope>
  </managed-bean>
  <managed-bean>
    <managed-bean-name>mdata</managed-bean-name>
    <managed-bean-class>br.com.urubatan.jsfjpassec.SomeData</managed-bean-class>
    <managed-bean-scope>request</managed-bean-scope>
    <managed-property>
      <property-name>loginOk</property-name>
      <property-class>java.lang.Boolean</property-class>
      <value>#{login.loginOk}</value>
    </managed-property>
  </managed-bean>
  <navigation-rule>
    <from-view-id>/login.jsp</from-view-id>
    <navigation-case>
      <from-outcome>login</from-outcome>
      <to-view-id>/login.jsp</to-view-id>
      <redirect/>
    </navigation-case>
    <navigation-case>
      <from-outcome>secpage</from-outcome>
      <to-view-id>/secureView.jsp</to-view-id>
      <redirect/>
    </navigation-case>
  </navigation-rule>
</faces-config>

```

Exemplo do “faces-config.xml”

Exemplo de “persistence.xml”

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="1.0" xmlns="http://java.sun.com/xml/ns/persistence">
  <persistence-unit name="nomeDaPersistenceUnit">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <properties>
      <property name="hibernate.dialect" value="org.hibernate.dialect.MySQLDialect" />
      <property name="hibernate.connection.url" value="jdbc:mysql://localhost:3306/database" />
      <property name="hibernate.connection.driver_class" value="com.mysql.jdbc.Driver" />
      <property name="hibernate.connection.password" value="" />
      <property name="hibernate.connection.username" value="root" />
      <property name="hibernate.hbm2ddl.auto" value="create-drop" />
    </properties>
  </persistence-unit>
</persistence>
```