

Criar e acessar atributos

Transcrição

Nosso XML já cresceu um pouco, contendo informações de preço e mais de um produto. Mas e se quisermos incluir também a moeda em que a venda foi realizada? Perceba que a moeda é mais uma informação sobre os dados da venda, como uma meta-informação. Pensando nesse cenário, existe outra forma de adicionarmos dados a um arquivo XML, que é criando atributos.

Por exemplo, na tag `<venda>`, podemos ter um atributo `moeda` cujo valor será `real`.

```
<?xml version="1.0" encoding="UTF-8"?>
<venda moeda="real">
    <formaDePagamento>Cartão</formaDePagamento>
    <produtos>
        <produto>
            <nome>Livro de xml</nome>
            <preco>29.90</preco>
        </produto>
        <produto>
            <nome>Livro de O.O. java</nome>
            <preco>29.90</preco>
        </produto>
    </produtos>
</venda>
```

Sendo assim, nem sempre criaremos tags para adicionarmos conteúdos. Porém, até o momento só aprendemos a pegar conteúdo armazenado em tags, e precisamos de uma forma de acessar também os atributos. Voltando ao nosso `Sistema.java`, vamos supor que queremos exibir a moeda antes de exibirmos os produtos. Para isso, usaremos o `document` retornado pelo nosso `builder.parse()`. Mas o que exatamente esse objeto representa?

A tag mais externa do XML é a `<venda>`. Chamamos esse tipo de tag de "root element", ou "elemento raiz". O `document` é justamente uma referência para esse elemento raiz, e, a partir dele, podemos chamar o `getDocumentElement()` para nos retornar a própria `<venda>`. Associaremos então essa chamada a uma variável local `venda`.

A partir da venda, poderemos acessar o atributo `moeda` com `getAttribute()` e armazenar o seu retorno em uma string `moeda`. Por fim, podemos imprimir essa variável na tela.

```
Element venda = document.getDocumentElement();
String moeda = venda.getAttribute("moeda");
System.out.println(moeda);
```

Feito isso, teremos como retorno:

real

Nome:Livro de xml

Preço:29.9

Nome:Livro de O.O. java

Preço:29.9