

# Programação Orientada a Serviços

**Prof. Diego Cirilo**

**Aula 10: Parsers**

# *Parsers*

- *Parsers* são utilizados para o processamento de documentos.
- Permitem acessar os dados de um XML/JSON em um programa.

# XML DOM

- *Documento Object Model*
- Padrão da W3C para processamento de documentos XML.
- Permite a leitura e edição de documentos XML de forma programática.
- Existem alternativas como o SAX (*Simple API for XML*).
- Identifica a estrutura em árvore, com elementos e nós, e consegue navegar na hierarquia.

# XML DOM em Python

- O XML DOM faz parte da biblioteca padrão Python
- A versão mais básica é chamada de `minidom`
- `from xml.dom.minidom import parse`
- `dom = parse("arquivo.xml")`

# Métodos XML DOM

- `documentElement` , retorna o elemento raiz
- `tagName` , nome do elemento
- `getElementByTagName("nomeDaTag")` , retorna NodeList
- `getAttribute("attr")` retorna atributo
- `firstChild` , `lastChild` , `childNodes[0]` , retorna os "filhos" de um elemento.
- `nodeValue` retorna conteúdo do elemento.
- ...

# Exemplo

```
<biblioteca>
  <livro categoria="ficção">
    <título>Harry Potter</título>
    <autor origem="Inglaterra">J.K. Rowling</autor>
    <ano>2005</ano>
  </livro>
  <livro categoria="não-ficção">
    <título>Sapiens</título>
    <autor origem="Israel">Yuval Noah Harari</autor>
    <ano>2014</ano>
  </livro>
  <livro categoria="ficção">
    <título>Vinte Mil Léguas Submarinas</título>
    <autor origem="França">Júlio Verne</autor>
    <ano>1869</ano>
  </livro>
</biblioteca>
```

# Exemplo

```
from xml.dom.minidom import parse

dom = parse("biblioteca.xml")

# Elemento raiz do XML (biblioteca)
biblioteca = dom.documentElement

# Recebe uma lista dos elementos com tag "livro"
livros = biblioteca.getElementsByTagName('livro')

# Acessa as informações de cada livro
for livro in livros:
    categoria = livro.getAttribute('categoria')
    elemento_titulo = livro.getElementsByTagName('título')[0]
    titulo = elemento_titulo.firstChild.nodeValue
    elemento_autor = livro.getElementsByTagName('autor')[0]
    origem = elemento_autor.getAttribute('origem')
    autor = elemento_autor.firstChild.nodeValue
    elemento_ano = livro.getElementsByTagName('ano')[0]
    ano = elemento_ano.firstChild.nodeValue

    print("Categoria:", categoria)
    print("Título:", titulo)
    print(f'Autor: {autor} ({origem})')
    print("Ano:", ano)
    print("---\n")
```

# Tarefa

- Usando a biblioteca `xml.dom.minidom` do Python, escreva um programa que faça o *parse* do arquivo `cardapio.xml` das tarefas anteriores.
- O programa deve apresentar ao usuário um Menu com os IDs e nomes dos pratos e perguntar qual prato o usuário deseja saber mais detalhes.
- Ao digitar o ID e apertar enter, o programa deve imprimir todas as informações do prato.
- Exemplo da saída no próximo slide.
- **DICA**



# Exemplo

```
$ python cardapio.py
```

```
1 - Feijoada
```

```
2 - Bauru
```

```
3 - Cuscuz Recheado
```

```
Digite o id do prato para saber mais: 3
```

```
Nome: Cuscuz Recheado
```

```
Descrição: Prato típico nordestino
```

```
Ingredientes:
```

```
    Cuscuz
```

```
    Carne de sol desfiada
```

```
    Queijo de Coalho
```

```
    Coentro
```

```
    Cebola
```

```
Preço: R$10,00
```

```
Calorias: 1000kcal
```

```
Tempo de preparo: 30 minutos.
```

# Parsers para JSON

- O JSON é suportado nativamente no Python
- A biblioteca `json` fornece, entre outros, os métodos:
  - `load` e `loads`, de um arquivo ou string, respectivamente.
  - `dump` e `dumps`, para um arquivo ou string, respectivamente.
- A conversão básica é entre um `dict` para um Objeto JSON e vice-versa.

# Equivalência Python-JSON

Python	JSON
dict	Object
list	Array
tuple	Array
str	String
int	Number
float	Number
True	true
False	false
None	null

# Exemplos

```
import json

# Importar de um arquivo
with open('data.json') as json_file:
    parsed_data = json.load(json_file)

# Importar de uma string
json_string = '{"key": "value", "array": [1, 2, 3]}'
parsed_data = json.loads(json_string)

# Escrever em um arquivo:
data = {
    "name": "John",
    "age": 30,
    "city": "New York"
}

with open("data.json", "w") as json_file:
    json.dump(data, json_file)

# Escrever em uma string:
json_string = json.dumps(data)
```

# Exercício

- Baseado nos exemplos das aulas anteriores, faça o *parse* do arquivo XML da imobiliária para um arquivo JSON equivalente.

# Exercício

- Já com o arquivo JSON da imobiliária, faça um programa interativo para visualização dos dados dos imóveis, no mesmo padrão do exercício anterior (do cardápio).
- O programa deve apresentar ao usuário um Menu com um índice(ID) dos imóveis e perguntar qual imóvel o usuário deseja saber mais detalhes.
- Ao digitar o ID e apertar enter, o programa deve imprimir todas as informações do imóvel, em uma formatação legível.

# Dúvidas?

