

JSON e Python



José Victor Viriato e Lana Rossato



<https://github.com/jviriato/json-seminario>

1.

TRANSFORMAR XML PARA JSON

Transformações sem perder o conteúdo, usando Python

- » Utilizaremos a biblioteca **xmljson**.
- » Ela converte arquivos xml em dicionários do Python (árvores, como JSON) e vice versa.



Instalando a biblioteca

```
$ sudo pip3 install xmljson
```

- » XML pode ser convertido em estrutura de dado
- » Existem várias convenções para a conversão XML → JSON
- » Nesse exemplo usaremos a **Badger Fish**

Outros convenções existentes:

- » Abdera
- » Parker
- » GData
- » Cobra
- » Yahoo

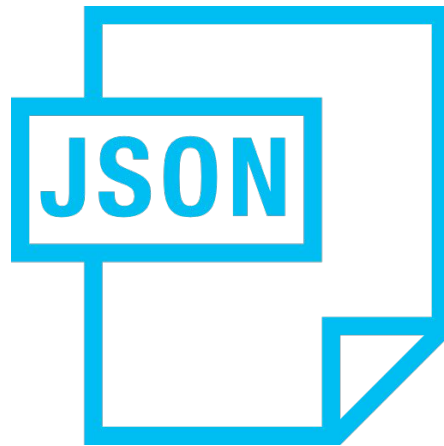
BadgerFish popula @xmlns em todos os elementos onde o namespace está ativo, não apenas nos elementos onde é declarado. Isso pode ser “overkill” para quem não liga muito para namespaces.

~apresentar o código~

2.

VALIDAR COM JSON SCHEMA

- » Atualmente no draft 7
- » Existe a biblioteca **jsonschema** (draft 7) para validar
- » Para gerar schemas, existe a **genjson** (draft 4)



Instalando a biblioteca

```
$ sudo pip3 install jsonschema
```

Usando a biblioteca para validar JSON

```
$ jsonschema -i nota2.json nota.schema
```

Se tudo der certo, não tem output nenhum

```
$ jsonschema -i nota1.json nota.schema
```

Caso ocorra um erro, é avisado

```
$ jsonschema -i nota6.json nota.schema
```

```
$ S.N: 'S.N' is not of type 'number'
```

Para arrumar o erro anterior, alteramos o schema para que **nro** aceite, além de números, aceitar também strings.

```
"type": "number"
```

```
"type": ["number", "string"]
```


3.

CONSULTAS EM JSON

Mostrando uma lista de produtos ordenada por preço

~apresentar o código~

4.

TRANSFORMAÇÕES EM JSON

Criando uma página HTML a partir de JSON

- » Para criar uma página em HTML, utilizaremos a biblioteca Jinja2
- » É uma ferramenta de templates para Python
- » Variáveis são mapeadas desta maneira:
`<title>{{ titulo }}</title>`

- » Bastante útil para trabalhar com JSON!
- » Como exemplo, criamos um template HTML de cupom fiscal com algumas variáveis.

~apresentar o código~

Obrigado! 