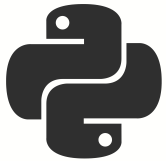
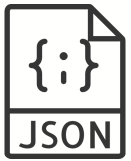
Seminário
Python com JSON

Deivis Costa Pereira e
William Felipe de Almeida





github.com/oscaruno/json-seminario

1. Transformação XML -> JSON

- Para transformar XML em JSON utilizamos a biblioteca **lxml**;
- Esta biblioteca permite carregar o XML em formato de árvore e manipulá-lo (XQuery, XPath, XSLT)
- Aplicou-se um arquivo XSLT, que realiza a XML->JSON, sobre o arquivo de entrada

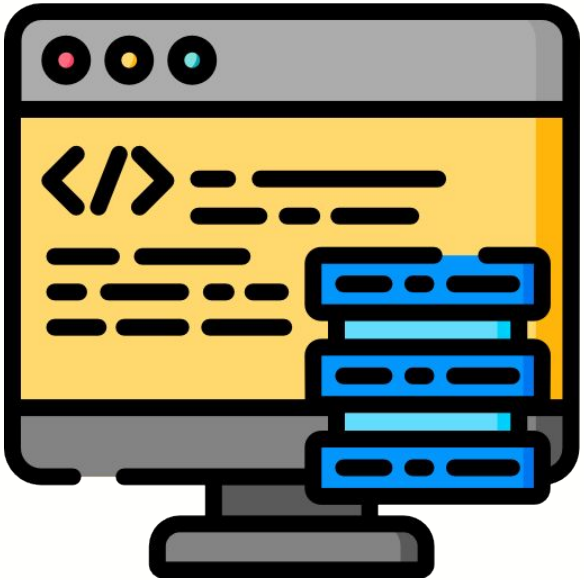
```
$ sudo pip3 install lxml
```

2. Validação - JSON Schema

- Para a validação do JSON, utilizamos a biblioteca **jsonschema**;
- Utiliza-se a função ***validate()***, a qual recebe o JSON a ser validado e o seu Schema.

```
$ sudo pip3 install jsonschema
```

● LIVE



3. Consultas em JSON

- Para as *queries* utilizamos a biblioteca **objectpath** e as expressões ***lambda*** do Python;

- A ObjectPath é uma linguagem de query para dados semi estruturados(JSON), que incorpora elementos do JSONPath, GraphQL e SQL, além de contar com funções incorporadas para: vetores, *strings*, Datas, conversão e aritmética.

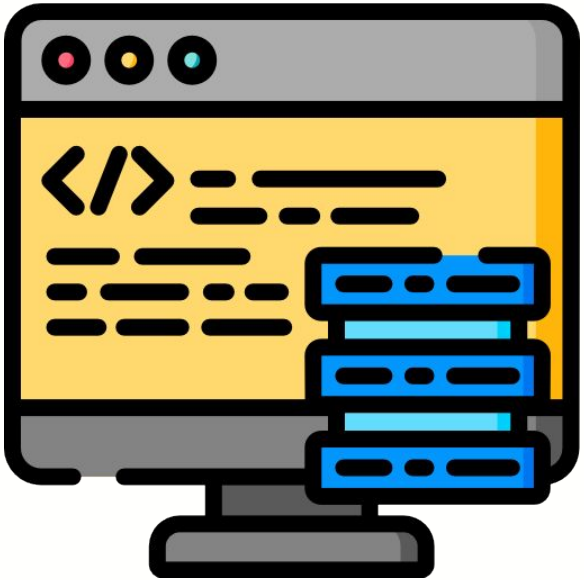
```
$ sudo pip3 install objectpath
```

JSONPath

JSONPath implementada na **objectpath**, baseada na proposta de Stefan Goessner em 2007.

XPath	JSONPath	Descrição
/	$	Objeto/elemento raiz
.	@	Objeto/elemento atual
/	. or []	Operador para filho
..	n/a	Operador para pai
//	..	Descendente recursivo. JSONPath pega essa sintaxe da E4X.
*	*	Coringa. Todos os objetos/elementos.
@	n/a	Acesso à atributo. Estruturas JSON não tem atributos.
[]	[]	Operador de seleção

● LIVE



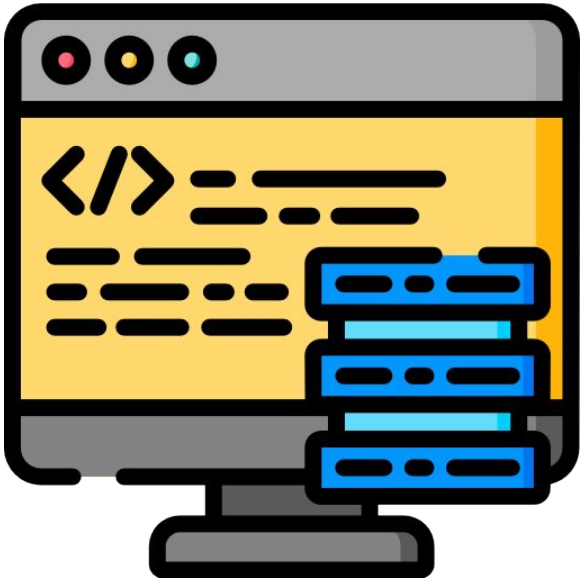
4. Transformação JSON -> HTML

- Para gerar o conjunto das páginas HTML usamos a biblioteca **jinja2** e o framework **Flask**;
- A Jinja2 é uma ferramenta de criação de *templates* para Python;

```
<p>{{ oc.resourceData }}</p>
```

```
$ sudo pip3 install flask
```

● LIVE



Dúvidas?

Seminário
Python com JSON

Deivis Costa Pereira e
William Felipe de Almeida