



Pontifícia Universidade Católica de Campinas  
Escola Politécnica

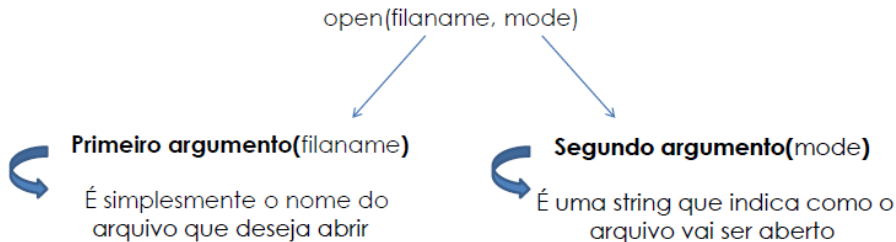
Manipulação de Arquivos  
Estudos Avançados de Banco de Dados

José Guilherme Picolo  
[jose.picolo@puc-campinas.edu.br](mailto:jose.picolo@puc-campinas.edu.br)

# Arquivos

Uma das tarefas mais comuns que realizamos no dia-a-dia é a manipulação de dados gravados em **arquivos**.

O primeiro processo natural que temos que realizar para fazer qualquer operação sobre um arquivo é abri-lo. Para abrir um arquivo usando Python, usaremos a função `open()`. A função retorna um objeto de arquivo e é mais comumente usado com dois argumentos:



Exemplos de modos diferentes para abrir um arquivo são mostrados na tabela a seguir:

Caractere	Significado
'r'	abre para leitura (padrão)
'w'	abre para escrita, truncando o arquivo primeiro (removendo tudo o que estiver contido no mesmo)
'x'	abre para criação exclusiva, falhando caso o arquivo exista
'a'	abre para escrita, anexando ao final do arquivo caso o mesmo exista
'b'	binary mode
't'	modo texto (padrão)
'+'	aberto para atualização (leitura e escrita)

## JavaScript Object Notation (JSON)

- É um padrão para a serialização de objetos de dados.
- É independente de linguagem de programação:
  - JSON usa a sintaxe de JavaScript para descrever objetos de dados, mas os parsers e bibliotecas para JSON existem em várias linguagens de programação diferentes.
- JSON é hierárquico, auto descritivo e fácil de entender.
- Permite definir vetores de elementos.

## JavaScript Object Notation (JSON) - Exemplo

```
{ "Livros":
  [
    { "ISBN":"ISBN-0-13-713526-2",
      "Preço":85,
      "Edição":3,
      "Título":"A First Course in Database Systems",
      "Autores":[ { "Nome":"Jeffrey", "Sobrenome":"Ullman"},
                   { "Nome":"Jennifer", "Sobrenome":"Widom"} ] }
    ,
    { "ISBN":"ISBN-0-13-815504-6",
      "Preço":100,
      "Nota":"Compre também o livro 'A First Course' e faça um excelente negócio!",
      "Título":"Database Systems: The Complete Livro",
      "Autores":[ { "Nome":"Hector", "Sobrenome":"Garcia-Molina"},
                   { "Nome":"Jeffrey", "Sobrenome":"Ullman"},
                   { "Nome":"Jennifer", "Sobrenome":"Widom"} ] }
  ],
  "Revistas":
  [
    { "Título":"National Geographic",
      "Mês":"Janeiro",
      "Ano":2009 }
    ,
    { "Título":"Newsweek",
      "Mês":"Fevereiro",
      "Ano":2009 }
  ]
}
```

## JavaScript Object Notation (JSON)

A sintaxe do JSON é um subconjunto da sintaxe para a definição de objetos de Javascript:

- **Valores simples** – são valores do tipo **string** (delimitados por aspas), **número** (inteiro ou em ponto flutuante) e **boolean**.
- **Objetos** – são conjuntos de pares do tipo **nome:valor**, separados por vírgula e delimitados por chaves ('{' e '}').
- **Vetores** – são listas ordenadas de valores separados por vírgulas e delimitados por colchetes ('[' e ']')
- Um **valor** pode ser um valor simples, um objeto ou um vetor.

## JavaScript Object Notation (JSON)

**Dados** em JSON são escritos como pares do tipo **nome:valor**.

Um par consiste em um nome de campos (delimitado por aspas), seguido por dois pontos e um valor.

`"nome" : "Ana"`

## JavaScript Object Notation (JSON)

A definição de um **objeto** é delimitada por '{' e '}'  
Um objeto pode conter vários pares **nome:valor**.

```
{ "nome" : "Ana" , "sobrenome" : "Gomes" }
```



## JavaScript Object Notation (JSON)

**Vetores** em JSON são delimitados por '[' e ']

Um vetor pode conter vários valores:

```
{ "empregados": [  
  { "nome": "Paulo", "sobrenome": "Costa" },  
  { "nome": "Ana", "sobrenome": "Gomes" },  
  { "nome": "Pedro", "sobrenome": "Neves" } ] }
```

## JavaScript Object Notation (JSON)

Podemos ter:

- Vetores vazios;
- Vetores que contém valores simples e objetos aos mesmo tempo.

```
{ "Autores": [ { "Nome": "Jeffrey", "Sobrenome": "Ullman" },  
               "Jennifer Widom" ] }
```

ou

```
{ "Autores": [ ] }
```

## XML (Extensible Markup Language)

O objeto básico em XML é o documento XML que contém uma estrutura hierárquica baseado em dois conceitos:

- Elementos
- Atributos – Fornecem informações adicionais a um elemento.

note.xml

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

## XML (Extensible Markup Language)

As tags do XML não são pré-definidas, ou seja, no exemplo anterior não são um padrão do XML, elas foram definidas pelo autor do documento. Em XML o autor deve definir as tags e a estrutura do documento.

## CSV (Comma-separated values)

CSV é uma abreviação que significa Comma-separated values (valores separados por vírgula). É um arquivo de apenas uma planilha que contém dados separados por um sinal de pontuação, como vírgula, ponto e vírgula ou ponto.

```
1,3,4.0,964981247  
1,6,4.0,964982224  
1,47,5.0,964983815  
1,50,5.0,964982931  
1,70,3.0,964982400  
1,101,5.0,964980868  
1,110,4.0,964982176
```