

Arquivos

Introdução à Programação
SI1

Conteúdo

- Arquivos
 - Conceitos
 - Operações
 - Métodos
 - Exercícios

Arquivos

- **Entrada** e **saída** são operações de **comunicação** de um programa com o **mundo externo**
- Essa comunicação se dá usualmente através de *arquivos*
- Em Python, um arquivo pode ser lido/escrito através de um objeto da classe `file`

Abrindo Arquivos

- `open(name, mode)`
 - `name`: nome do arquivo a abrir
 - `mode`: (opcional) modo de abertura – string contendo
 - `r` : leitura (default)
 - `w` : escrita (se o arquivo existir terá o conteúdo apagado)
 - `a` : escrita a partir do final (não apaga conteúdo)
 - `r+` : indica leitura e escrita
 - `b` : binário (`rb`, `wb`, `ab`)

Exemplo

- O comando `open` retorna um objeto do tipo *file* (arquivo)
- O objeto *file* é usado para operações de entrada e saída:

```
>>> arq = open ("teste.txt", "w")
>>> arq.write ("Oi")
>>> arq.close ()
>>> arq = open ("teste")
>>> x = arq.read()
>>> x
'Oi'
```

Open

```
>>> f = open('texto.txt', 'w')  
>>> f.write('Ola Mundo!')
```

- Não devemos esquecer de fechar o arquivo:

```
>>> f.close()
```

Métodos *Read*, *Write* e *Close*

- `read(num)`
 - Lê `num` bytes do arquivo e os retorna numa string
 - Se `num` não é especificado, todos os bytes desde o ponto atual até o fim do arquivo são retornados

Métodos *Read*, *Write* e *Close*

- `write(string)`
 - Escreve `string` no arquivo
 - Devido ao uso de buffers, a escrita pode não ser feita imediatamente
 - Use o método `flush()` ou `close()` para assegurar a escrita física
- `close()`
 - Termina o uso do arquivo para operações de leitura e escrita

Exemplo *Read*

```
>>> f = open('texto.txt', 'r')
>>> ler = f.read()
>>> f.close()
>>> print ler
Ola Mundo!
```

- Metodo `read()` usado sem nenhum argumento, mostrará tudo que esta no aquivo

Exemplo *Read*

- Se quisermos ler apenas os três primeiros caracteres (bytes), podemos usar da seguinte maneira:

```
>>> f = open('texto.txt') # r é
    default
>>> ler = f.read(3)
>>> f.close()
>>> print ler
Ola
```

Exemplo *Read*

- e caso queira ler o restante, o `read()` começará de onde parou:

```
>>> restante = f.read()
```

```
>>> print restante
```

```
Mundo!
```

Fim de Linha

- Arquivos de texto são divididos em linhas usando caracteres especiais
- Python usa sempre `\n` para separar linhas
 - Leitura ou escrita de arquivo aberto em modo **texto**
 - Em modo **binário**, entretanto, a conversão **não é feita**

Lendo e Escrevendo Linhas

- `readline(n)`
 - Se `n` não é especificado, retorna exatamente uma linha lida do arquivo
 - Caso contrário, lê uma linha, mas busca no máximo `n` caracteres pelo final de linha
 - Se o tamanho da linha é zero significa que o `final do arquivo` foi atingido

Lendo e Escrevendo Linhas

- `readlines()`
 - Retorna o restante do conteúdo do arquivo em uma lista de strings

Lendo e Escrevendo Linhas

- `writelines(lista)`
 - Escreve a `lista` (ou qualquer seqüência) de strings, uma por uma no arquivo
 - Caracteres de final de de linha *não são* acrescentados

Exemplos

- adicionar mais alguma coisa no arquivo do exemplo:

```
>>> f = open('texto.txt', 'a')
```

```
>>> f.write('\nOla Python')
```

```
>>> f.close()
```

- o 'a' (append) é usado para adicionar texto sem apagar o que já havia no arquivo
- '\n' é usado pra pular uma linha, lembrando que tudo é string (este caractere fica visível)

Exemplos

```
>>> f = open('texto.txt', 'r')
>>> linha1 = f.readline()
>>> linha2 = f.readline()
>>> f.close()
>>> print linha1
Ola Mundo!\n
>>> print linha2
Ola Python
```

Exemplos

```
>>> lista = ['Ola mundo\n', 'Ola Python\n', 'Ola UFRPE']
>>> f = open('texto.txt', 'w')
>>> f.writelines(lista)
>>> f = open('texto.txt', 'r')
>>> cont = f.readlines()
>>> print cont
['Ola mundo\n', 'Ola Python\n', 'Ola UFRPE']
```

Exemplos

```
>>> for linha in f:  
      print linha
```

Resumo

| operação | Interpretação |
|---|--|
| <code>output = open("teste.txt","w")</code> | Cria arquivo de saída ("w" significa gravação) |
| <code>input = open("teste.txt","r")</code> | Cria arquivo de entrada ("r" significa leitura) |
| <code>S = input.read()</code> | Lê o arquivo inteiro em uma única string |
| <code>S = input.read(N)</code> | Lê N byte (1 ou mais) |
| <code>S = input.readline()</code> | Lê a próxima linha |
| <code>L = input.readlines()</code> | Cria uma lista onde cada elemento é uma linha do arquivo |
| <code>output.write(S)</code> | Grava a string S no arquivo |
| <code>output.writelines(L)</code> | Grava no arquivo todas as strings da lista L |
| <code>output.close()</code> | Fechamento manual do arquivo |