

Função Open

segunda-feira, 12 de setembro de 2022 15:15

Se você tiver um sistema que tenha configurações que precisem ser salvas em um arquivo podemos usar a função open.

Modo básico:

```
file = open("Arquivo.txt", 'w+')
```

Quando usamos o **w** indicamos que queremos um arquivo para escrever e quando usamos o **+** que queremos também ler esse arquivo

w+

para escrever dentro desse arquivo podemos usar a função **write**

Sempre que finalizamos um arquivo precisamos usar o **close** para finalizar o arquivo.

```
file = open("Arquivo.txt", 'w+')
file.write("Geronimo\n")
file.write("Morais\n")
file.write("Lima\n")
file.write("Neto")
file.close()
```

Quando queremos ler o arquivo todo usamos a função **read()**

```
file = open("Arquivo.txt", 'w+')
file.write("Geronimo\n")
file.write("Morais\n")
file.write("Lima\n")
file.write("Neto")

print("Lendo linhas: ")
print(file.read())
file.close()
```

desse modo o único problema é que o cursor está no final de

file.write("Neto") que foi a ultima escrita do arquivo.

Quando queremos que o cursor suba para o início, primeira escrita do arquivo usamos a função **seek()**

```
file = open("Arquivo.txt", 'w+')
file.write("Geronimo\n")
file.write("Morais\n")
file.write("Lima\n")
file.write("Neto")

file.seek(0,0)
print("lendo linhas: ")
print(file.read())
file.close()
```

Dessa forma quando usamos os parâmetros **seek(0,0)** - o primeiro é nada de qual índice que o seek vai iniciar no caso G de Geronimo

se usar o **seek(2,0)**

irá começar o r de ronimo

Podemos também usar a função **readline()** com print para ler linha por linha

Diferença entre **read()** e **readline()**

```
file = open("Arquivo.txt", 'w+')
file.write("Geronimo\n")
file.write("Morais\n")
file.write("Lima\n")
file.write("Neto")

print("Lendo linhas: ")
file.seek(0,0)
print(file.read())
file.close()
```

Na função **read()** retorna todas as strings do arquivo para o print e no

readline() ler linha por linha.

a readline pega também os \n pula de linha e podemos usar o end para subir.

```
file = open("Arquivo.txt", 'w+')
file.write("Geronimo\n")
file.write("Morais\n")
file.write("Lima\n")
file.write("Neto")

print("Lendo linhas: ")
file.seek(0,0)
print(file.readline(),end='')
print(file.readline(),end='')
print(file.readline(),end='')
print(file.readline(),end='')
file.close()
```

Existe a função **readlines()** que joga tudo dentro de uma lista, podendo usar o

for tanto na variavel que recebe a função open quanto a função

variavelopen.readlines()

Exemplo:

```

file = open("Arquivo.txt", 'w+')
file.write("Geronimo\n")
file.write("Morais\n")
file.write("Lima\n")
file.write("Neto")

print("Lendo linhas: ")
file.seek(0,0)
print(file.readlines())

print("#####")
file.seek(0,0)
for i in file.readlines():
    print(i,end=' ')

print("Separando...")
print("#####")
file.seek(0,0)
for i in file:
    print(i,end="")
file.close()

```

Usando os 3 métodos

Algumas pessoas usam o bloco try para criação de arquivos

Exemplo:

```

try:
    file = open("ABC.txt", "w+")
    file.write('linha1')
    file.seek(0)
    print(file.read())
finally:
    file.close()

```

O jeito mais correto de se usar a criação de arquivos é usando o gerenciador de contexto **with** que funciona da seguinte forma:

```

with open("abc.txt", "w+") as file:
    file.write("Geronimo\n")
    file.write("Morais\n")
    file.write("Lima\n")
    file.write("Neto")

    file.seek(0,0)
    print(file.read())

```

no **with** não precisamos fechar o arquivo com o close.

alterando w+

Quando usamos o r ao inves do w estamos indicando que queremos ler apenas o que tem dentro do arquivo já quando usamos o a ao inves do w ou r estamos queremos adicionar coisas ao arquivo sem apagar o que já temos.

Exemplo:

```
with open("abc.txt", "a+") as file:
    file.write("Nova linha\n")
    file.seek(0,0)
    print(file.read())
```

Usando o a ele pega o arquivo e adiciona mais coisas dentro dele contando com o que já existe lá.

Exemplo:

Arquivo antes de adicionar:

```
Geronimo
Morais
Lima
```

Arquivo pos adicionar:

```
with open("abc.txt", "a+") as file:
    file.write("Nova linha\n")
    file.seek(0,0)
    print(file.read())
```

```
Geronimo
Morais
Lima
Nova linha
```

Quando usamos o w+ ele vai apagar e só vai escrever o que está ali dentro do código.

Podemos então com isso criar um dicionário em json jogar dentro de um arquivo e depois recuperar como dicionário novamente

```
import json

d1 = {
    'pessoa1': {
        'nome': 'Geronimo',
        'idade': 18
    },
    'pessoa2': {
        'nome': 'Eunice',
        'idade': 20
    }
}

d1_json = json.dumps(d1)

with open('abc.txt', 'w+') as file:
    file.write(d1_json)
```

A função `dumps()` do `json` converte algo para `json`.

Lendo o dicionário jogado dentro do arquivo `abc.txt` e convertendo novamente em dicionário

Exemplo:

```
import json

with open("abc.txt", 'r') as file:
    d1_json = file.read()
    d1_json = json.loads(d1_json)
```

A função `loads` converte novamente o dicionário escrito para dicionário novamente.

Dando um `for` dentro do dicionário que veio do arquivo:

```
import json

with open("abc.txt", 'r') as file:
    d1_json = file.read()
    d1_json = json.loads(d1_json)

for k, v in d1_json.items():
    for v1, v2 in v.items():
        print(v1, v2)
```