

Lendo um arquivo

Transcrição

Ainda no terminal do Python 3, vamos ver o funcionamento da **leitura de um arquivo**. Como agora o arquivo **palavras.txt** está na pasta do projeto **jogos**, devemos executar o comando que abre o terminal do Python 3 na pasta do projeto.

Leitura de um arquivo

Vamos então abrir o arquivo no modo de leitura, basta passar o nome do arquivo e a letra **r** para a função **open**, como já vimos no vídeo anterior:

```
>>> arquivo = open("palavras.txt", "r")
```

Como abrimos o arquivo no modo de leitura, a função **write** não funciona. Para **ler o arquivo inteiro**, utilizamos a função **read**:

```
>>> arquivo.read()
'banana\nmelancia\nmorango\nmanga\n'
```

Mas se executarmos a função novamente:

```
>>> arquivo.read()
''
```

Nos é retornado uma string vazia. Por quê?

O arquivo é como um fluxo de linhas, que começa no início do arquivo, como se fosse o ponteiro. Ele vai descendo e lendo arquivo, após ler tudo, ele fica posicionado no final do arquivo, então quando chamamos a função **read()** novamente, não há mais conteúdo, pois ele todo já foi lido.

Ou seja, para ler o arquivo novamente, devemos fechá-lo e abri-lo novamente.

Lendo linha por linha do arquivo

Mas não queremos ler todo o conteúdo do arquivo, e sim ler linha por linha. Como já foi visto, um arquivo é um fluxo de linhas, uma sequência de linhas, então como é uma sequência, podemos fazer um **for** nela:

```
>>> arquivo = open("palavras.txt", "r")
>>> for linha in arquivo:
...     print(linha)
...
banana

melancia
```

morango

manga

Mas podemos reparar que existe uma linha entre cada fruta. Por que isso acontece? Para ver melhor, vamos ler somente uma linha do arquivo, com a função `readLine()` :

```
>>> arquivo = open("palavras.txt", "r")
>>> linha = arquivo.readline()
>>> linha
'banana\n'
```

Há um `\n` ao final da linha, porque a linha sabe que ao seu final deve ser feita uma nova linha. Mas anteriormente havíamos feito um `print` , que também quebra uma linha ao final da impressão, colocando também um `\n` ! Assim, são criadas duas novas linhas, por isso havia uma linha em branco entre as frutas.

Limpendo a linha

Como vimos, há um `\n` ao final de cada linha, de cada palavra, mas queremos somente a palavra. Já vimos como tirar espaços em branco no início e no fim da string, basta utilizar a função `strip()` , que também remove caracteres especiais, como o `\n` .

Sabendo disso tudo, podemos implementar a funcionalidade de leitura de arquivo no nosso jogo. Faremos isso no próximo vídeo.