

## Escrevendo em um arquivo

### Transcrição

Uma funcionalidade que ainda nos atrapalha no nosso jogo é a palavra secreta, que atualmente está **fixa**. Se queremos que a palavra seja diferente, devemos modificá-la no código.

A nossa ideia é ler palavras de um arquivo de texto, e dentre elas escolhemos uma palavra aleatoriamente, que será a palavra secreta do jogo

### Escrita de um arquivo

Para **abrir um arquivo**, o Python possui a função *built-in* `open`. Ela recebe dois parâmetros: o primeiro é o nome do arquivo a ser aberto, e o segundo parâmetro é o modo que queremos trabalhar com esse arquivo, se queremos **ler** ou **escrever**. O modo é passado através de uma string: `"w"` para **escrita** e `"r"` para **leitura**.

No nosso jogo, faremos a leitura de um arquivo, mas vamos ver antes no terminal do Python 3 como funciona a escrita:

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

Vale lembrar que o `w` **sobrescreve o arquivo**, se o mesmo existir. Se só quisermos **adicionar conteúdo** ao arquivo, utilizamos o `a`.

Agora que temos o arquivo, como escrevemos nele? Basta chamar no arquivo a função `write`, passando para ela o que queremos escrever no arquivo:

```
>>> arquivo.write("banana")
6
>>> arquivo.write("melancia")
8
```

O retorno dessa função é o número de caracteres que adicionamos no arquivo.

### Fechando um arquivo

Quando estamos trabalhando com arquivos, devemos nos preocupar em fechá-lo. Assim como abrimos um arquivo, devemos fechá-los, chamando a função `close`:

```
>>> arquivo.close()
```

Após isso, podemos verificar o conteúdo do arquivo, ele foi criado na mesma pasta em que o comando para a abrir o console do Python 3 foi executado. Ao verificar o seu conteúdo, vemos:

```
bananamelancia
```

As palavras foram escritas em uma mesma linha. Mas como escrever em uma nova linha?

## Escrevendo palavras em novas linhas

A primeira coisa que devemos fazer é abrir o arquivo novamente, dessa vez utilizando o `a`, de *append*:

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

Podemos escrever novamente no arquivo, mas dessa vez com a preocupação de criar uma nova linha após o que vamos escrever. Para representar uma nova linha em código, adicionamos o `\n` ao final do que queremos escrever:

```
>>> arquivo.write("morango\n")
8
>>> arquivo.write("manga\n")
6
```

Ao fechar o arquivo e verificar novamente o seu conteúdo, vemos:

```
bananamelanciamorango
manga
```

A palavra **morango** ainda ficou na mesma linha, mas como especificamos na sua adição que após a palavra deverá ter uma quebra de linha, a palavra **manga** foi adicionada abaixo, em uma nova linha.

Por fim, vamos mover esse arquivo para dentro do nosso projeto, e ajeitar as suas palavras, quebrando as linhas. Ele ficará assim:

```
banana
melancia
morango
manga
```