01

Encontrando letras

Transcrição

Já temos o laço do jogo pronto, enquanto o usuário não acertou a palavra secreta e não se enforcou, ele continua jogando. Agora falta implementar a lógica do jogo.

Capturando a entrada do usuário

Ao jogar, o usuário deve digitar uma letra, então devemos pedi-la para ele. Já fizemos isso no treinamento anterior, basta utilizar a função `input` para capturar a entrada do usuário:

```
while (not acertou and not enforcou):

    chute = input("Qual letra? ")

    print("Jogando...")
```

Com o chute do usuário em mãos, devemos procurar se essa letra existe dentro da palavra secreta. Mas como fazer isso?

Encontrando a posição de uma letra em uma string

A palavra secreta é uma string, tipo `str`, e nesse tipo há uma função em que podemos procurar algo dentro da string. Essa função é a `find`, ao passar o chute do usuário para ela, a mesma retorna a posição na palavra em que a letra se encontra, começando da posição **0** (que representa a primeira letra). Por exemplo:

```
>>> palavra = "banana"
>>> palavra.find("b")
0
>>> palavra.find("n")
2
```

Caso a letra não exista na string, nos é retornado o número **-1**:

```
>>> palavra = "banana"
>>> palavra.find("z")
-1
```

Mas podemos reparar em um problema da função, pois ao pesquisar pela letra **n**, foi retornado o número **2**, mas a letra **n** também existe na posição **4** da palavra, porém a função não faz isso, ela só retorna uma única posição, a primeira que ela encontrar. Por isso não vamos utilizar essa função. Então o que faremos?

Iterando sobre a palavra

Resolveremos isso **iterando sobre a palavra secreta**. Faremos um laço em cima de cada letra, no nosso caso, na primeira iteração teremos a letra **b**, na segunda teremos a letra **a**, depois a **n** e assim até terminar a palavra. Com a letra em

mãos, basta comparar com o chute do usuário.

Mas aí entramos em outra questão: como fazemos um laço em cima de uma palavra?

A palavra é uma sequência de letras, logo podemos utilizar o laço `for` ! Por exemplo:

```
>>> palavra = "banana"
>>> for letra in palavra:
...     print(letra)
...
b
a
n
a
n
a
```

Faremos isso no nosso jogo. Para cada letra da palavra, comparamos com o chute do usuário, e se o chute for igual à letra, podemos imprimi-lo:

```
while (not acertou and not enforcou):

    chute = input("Qual letra? ")

    for letra in palavra_secreta:
        if (chute == letra):
            print(chute)

    print("Jogando...")
```

Agora, ao executar o programa e procurar pela letra **a**, vemos o seguinte resultado:

```
********************************
***Bem vindo ao jogo da Forca!***
********************************
Qual letra? a
a
a
a
Jogando...
```

Podemos melhorar ainda mais, exibindo a posição do chute na palavra, assim como a função `find` fazia.

Exibindo a posição da letra

Temos que fazer isso na mão, então, fora do `for` , vamos criar a variável `index` , com o valor 0. E dentro do `for` , vamos incrementar essa variável a cada iteração:

```
while (not acertou and not enforcou):
```

```
    chute = input("Qual letra? ")

    index = 0
    for letra in palavra_secreta:
        if (chute == letra):
            print(chute)
        index = index + 1

    print("Jogando...")
```

Com a letra e a posição em mãos, vamos imprimi-los, utilizando a interpolação de strings:

```
while (not acertou and not enforcou):

    chute = input("Qual letra? ")

    index = 0
    for letra in palavra_secreta:
        if (chute == letra):
            print("Encontrei a letra {} na posição {}".format(letra, index))
        index = index + 1

    print("Jogando...")
```

Ao executar novamente o nosso jogo:

```
*********************************
***Bem vindo ao jogo da Forca!***
*********************************
Qual letra? n
Encontrei a letra n na posição 2
Encontrei a letra n na posição 4
Jogando...
```

Mas se digitarmos uma letra em maiúsculo, por exemplo a letra **A**, a mesma não é encontrada na palavra. O ideal é não fazermos essa diferenciação entre letras minúsculas e maiúsculas. Vamos resolver esse problema no próximo capítulo :)