

Compreensão de lista

Transcrição

Atualmente, a nossa palavra secreta é **banana**, que possui seis letras. Por isso a lista de palavras acertadas possui seis `_`. Mas se trocarmos a palavra secreta? Precisamos alterar a quantidade de `_` na lista, de acordo com o número de letras da nova palavra.

Meio trabalhoso, né? Então vamos tornar isso mais dinâmico.

Inicializando a lista de acordo com o número de letras da palavra

Queremos que a inicialização da lista de palavras acertadas seja dinâmica, baseada na quantidade de letras que houver na palavra secreta. Já sabemos implementar isso, podemos criar a lista vazia, e para cada letra na palavra secreta, adicionamos um `_` à ela:

```
palavra_secreta = "banana".upper()
letras_acertadas = []

for letra in palavra_secreta:
    letras_acertadas.append("_")
```

Perfeito, está funcionando! Mas há uma forma mais interessante de fazer isso.

Realizando um laço dentro da inicialização da lista

Quando inicializamos a lista, queremos inicializá-la com um caractere `_` para cada letra na palavra secreta. Só que com Python, podemos fazer isso diretamente, dentro da inicialização da lista:

```
palavra_secreta = "banana".upper()
letras_acertadas = ["_" for letra in palavra_secreta]
```

Essa funcionalidade do Python se chama **List Comprehension**, em português, *Compreensão de lista*.