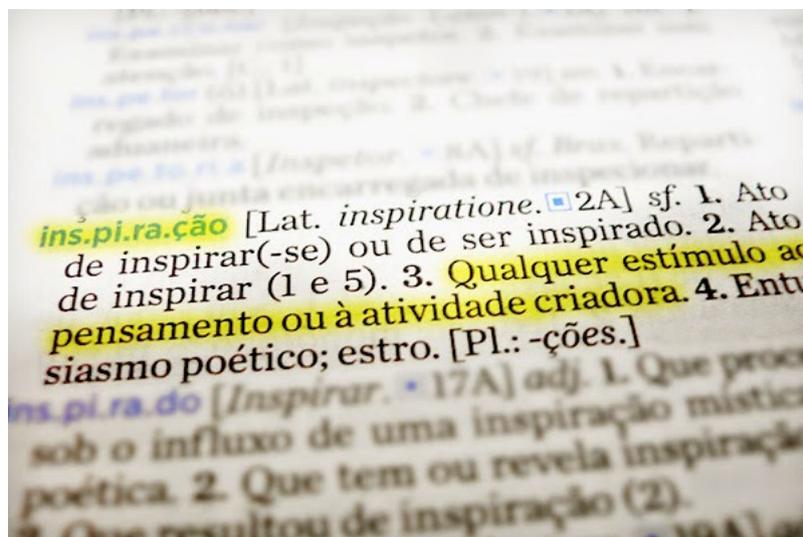


# 1 Dicionários

Como vimos anteriormente, **listas** e **tuplas** permitem o armazenamento de um conjunto de dados através de uma única variável. Em comum, possuem o fato de que ambas são indexadas a partir de zero. A maior diferença entre elas é que listas são mutáveis e tuplas são imutáveis.

**Dicionários** também permitem o armazenamento de um conjunto de dados em Python. Assim como nas lista, é possível modificar, remover ou adicionar elementos à lista. Entretanto, dicionários são indexados a partir de uma “palavra” ao invés de um número. Para cada palavra, existe uma definição (como em um dicionário):



**Figura 1:** Palavras e definições em um dicionário.

Em Python, a palavra é chamada de **chave** (*key*) e sua definição é chamada de **valor** (*value*). Os valores não são enumerados e não possuem uma ordem específica na memória. Um primeiro item inserido num dicionário não será armazenado necessariamente na mesma posição todas as vezes em que o programa for executado.

Como exemplo, vamos imaginar uma agenda telefônica que salva nome e número de telefone de vários contatos. A forma mais fácil de armazená-la seria através de uma lista de tuplas, sendo que cada elemento contém um nome e seu respectivo telefone. Ex:

```
1 lista = [ ("Enzo", "(27)99900-1122"),
2           ("Valentina", "(27)99900-1133"),
3           ("Thor", "(27)99900-1144") ]
```

Para adicionarmos ler os dados de um novo contato e armazená-lo na lista, podemos fazer:

```
1 nome = input("Digite o nome do novo usuario: ")
2 telefone = input("Digite o telefone: ")
3 lista.append( (nome, telefone) )
```

Dada uma variável *nome* contendo o nome de uma pessoa, precisamos percorrer a lista se quisermos imprimir o telefone de *nome*:

```
1 def getTel(nome, lista):
2     for (n, c) in lista:
3         if n == nome:
4             return c
5     return None
6
7 print(getTel("Valentina", lista))
```

Já num dicionário, isso não é necessário. Um dicionário é criado utilizando “{” e “}”, e as chaves são separadas de seus valores por “:”.

```
1 agenda = { "Enzo" : "(27)99900-1122",
2            "Valentina" : "(27)99900-1133",
3            "Thor" : "(27)99900-1144" }
```

Para adicionar um novo elemento, basta atribuir um valor a uma chave inexistente:

```
1 agenda["Ana Julia"] = "(27)99900-1155"
2
3 nome = input("Digite o nome do novo usuario: ")
4 telefone = input("Digite o telefone: ")
5 agenda[nome] = telefone
```

Se a chave não existia, será criada uma nova chave com o valor atribuído a ela. Caso já existisse uma chave igual, seu valor será substituído. As chaves são sempre únicas e sem repetições, pois os valores são indexados apenas pelas chaves. Para remover uma chave:

```
1 del agenda["Enzo"]
```

Para acessar o valor de uma chave específica:

```
1 print(agenda["Valentina"])
```

Para verificar se uma chave pertence ao dicionário:

```
1 if "Enzo" in agenda:
2     print("O telefone de Enzo eh", agenda["Enzo"])
3 else:
4     print("Telefone nao cadastrado.")
```

Para obter o nome de todas as chaves:

```
1 print(agenda.keys())
```

Para obter os valores:

```
1 print(agenda.values())
```

Para obter a quantidade de elementos armazenados:

```
1 tamanho = len(agenda)
```

O dicionário também pode ser criado inicialmente sem nenhum elemento:

```
1 pauta = {}
```

Os valores em um dicionário podem ser de qualquer tipo, inclusive tuplas ou listas:

```
1 pauta["Valentina"] = [9.5, 10.0]
2 pauta["Enzo"] = []
3 pauta["Enzo"].append(2.5)
4 pauta["Enzo"].append(3.1)
5 pauta["Enzo"].append(8.0)
6
7 print(pauta)
```

A saída do código acima seria:

```
{'Enzo': [2.5, 3.1, 8.0], 'Valentina': [9.5, 10.0]}
```

Por fim, também podemos iterar sobre os elementos de um dicionário:

```
1 for nome in pauta:
2     media = sum(pauta[nome]) / len(pauta[nome])
3     print("A media de", nome, "eh: ", media)
```

A saída do código acima seria:

```
A media de Enzo eh: 4.533333333333333
A media de Valentina eh: 9.75
```