

Trabalhando com Dictionary

Antes de tudo, é bom deixar claro que um **Dictionary NÃO** é uma **LISTA**, embora ambos armazenam uma coleção de dados!

Isso porque, uma **lista** se trata de uma coleção de dados ordenada, ou seja, cada item possui um índice numérico. Já o **dictionary** se trata de uma coleção de dados **NAO** ordenada, onde cada valor é acessado por uma chave.

Dito isso, então um **dictionary** é uma estrutura de dados que armazena **pares de chave e valor**, e você usa a **chave** para acessar o **valor** correspondente.

```
1 # Introducao aos dictionarys
2 aluno = {'nome': 'Ana', 'idade': 16, 'cidade': 'SP'}
3
4 print(type(aluno))
5 print(aluno)

<class 'dict'>
{'nome': 'Ana', 'idade': 16, 'cidade': 'SP'}
```

Obtendo itens com GET

Um recurso muito legal com o **Dictionary** ou **Dic**, é o uso do **get** para obter o valor de um item no **dic**, como podemos ver a seguir:

```
aluno = {'nome': 'Ana', 'idade': 16, 'cidade': 'SP'}
print(aluno.get('nome'))

Ana
```

E claro, podemos obter o valor através do índice da chave, que a forma convencional!

```
aluno['nome']
```

Porém a grande diferença, é que dessa forma se a chave não existir, o **python** irá retornar um erro, já com a função **get** não retornara valor nenhum e sem erro, e mais, podemos customizar uma mensagem de retorno também!

```
print(aluno.get('nome2', 'chave nao encontrada'))

chave nao encontrada
```

Looping no Dic

E claro, podemos sim trabalhar com **loops** em nossos **Dic**, até porque como já foi dito antes, **dictionary** se trata também de uma coleção!

```
alunoDic = {'nome': 'Ana', 'idade': 16, 'cidade': 'SP'}

for aluno in alunoDic:
    print(aluno)

nome
idade
cidade
```

E ao utilizar o **for**, veja que foi printado as chaves do **dic**!! Isso porque, por padrão como se trata de um **Dic** o **for** utiliza a função **keys()** por debaixo dos panos!!

```
for aluno in alunoDic.keys():  
    print(aluno)
```

Dito isso, para obter somente os valores do **Dic**, usaremos a função **values()**.

```
for aluno in alunoDic.values():  
    print(aluno)
```

```
Ana  
16  
SP
```

Agora se quisermos listar os valores junto com a chave correspondentes, podemos utilizar a função **items()**.

```
for aluno in alunoDic.items():  
    print(aluno)
```

```
('nome', 'Ana')  
('idade', 16)  
('cidade', 'SP')
```

Veja que quando utilizamos a função **items()**, será retornado uma lista de **Tuple**, onde terá as informações de chave e valor! E como se trata de **Tuple**, logo podemos declarar **2 variáveis** no próprio **for**, que será a **chave** e o **valor** correspondente!

```
for key, value in alunoDic.items():  
    print(f'{key} - {value}')
```

```
nome - Ana  
idade - 16  
cidade - SP
```