

PYTHON PARA JORNALISTAS DE DADOS

Dicionário e mais *for loop*

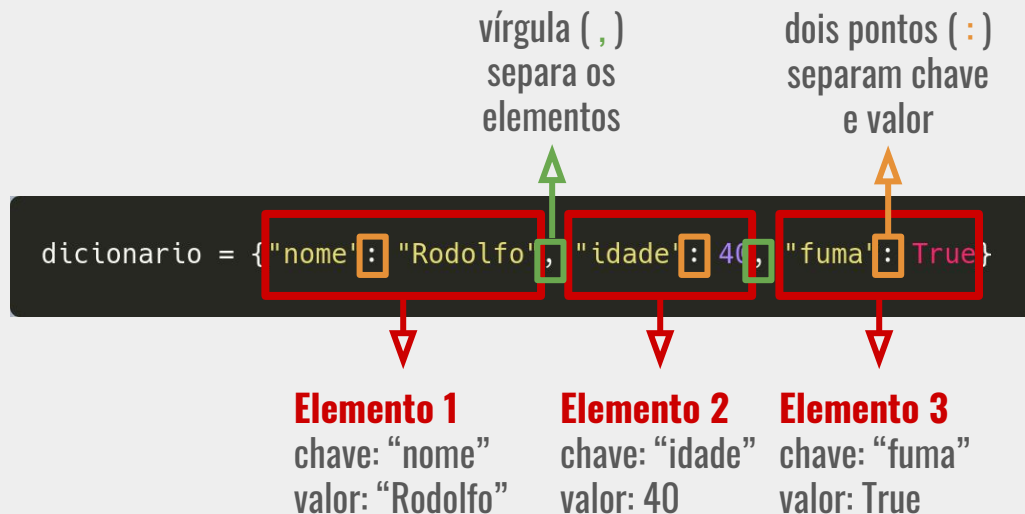
Rodolfo Viana
MBA em Jornalismo de Dados
20 de agosto de 2021



Dicionários | Coleção de dados, mas não só...

Assim como temos `list`, `tuple`, e `set` para “coleccionar” dados, existe `dict`, ou **dicionário**. Mas ele não é apenas uma coleção de dados: cada elemento de um dicionário guarda *chave* e *valor*, ou **par chave-valor**.

Um dicionário é feito usando os símbolos `{` e `}`, ou com a função `dict()`.



Dicionários | Coleção de dados, mas não só...

Isso muda bastante a forma de trabalharmos com coleções: se antes o usual era ter listas ou tuplas com dados de um mesmo tipo...



```
produtos = ["maçã", "uva", "laranja"]  
preco_kg = [12.89, 21.19, 9.99]
```

...agora podemos ter uma **coleção com diversos tipos de dados**...



```
fruta = {"produto": "maçã", "preco_kg": 12.89}
```

...ou até mesmo uma **coleção de coleções**, como **lista de dicionários** (algo bem comum, aliás).



```
frutas = [  
    {"produto": "maçã", "preco_kg": 12.89},  
    {"produto": "uva", "preco_kg": 21.19},  
    {"produto": "laranja", "preco_kg": 9.99}  
]
```

Dicionários | Atividade

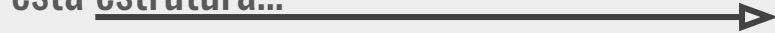
Monte um dicionário numa variável com seu nome. Ele deve conter as seguintes chaves (com os respectivos valores):

- nome
- sobrenome
- idade
- cidade_nascimento
- tem_pet
- quantidade_pet

Ao terminar, salve o arquivo.

Dicionários | Posição, adição e remoção

Agora você tem um arquivo que segue esta estrutura...



Mas como localizar um valor? E adicionar ou remover outro? Simples: **basta usar a chave!**

```
eu = {  
    "nome": "Rodolfo",  
    "sobrenome": "Viana",  
    "idade": 40,  
    "cidade_nascimento":  
    "Marília",  
    "tem_pet": False,  
    "quantidade_pet": 0  
}
```

Localização

```
eu["idade"]  
# 40
```

Adição

```
eu["profissao"] = "Cientista de dados"  
# OUTPUT:  
# eu = {  
#     "nome": "Rodolfo",  
#     "sobrenome": "Viana",  
#     "idade": 40,  
#     "cidade_nascimento": "Marília",  
#     "tem_pet": False,  
#     "quantidade_pet": 0,  
#     "profissao": "Cientista de dados"  
# }
```

Remoção

```
del eu["cidade_nascimento"]  
# OUTPUT:  
# eu = {  
#     "nome": "Rodolfo",  
#     "sobrenome": "Viana",  
#     "idade": 40,  
#     "tem_pet": False,  
#     "quantidade_pet": 0,  
#     "profissao": "Cientista de dados"  
# }
```

Dicionários | Atividade

No dicionário que você criou, adicione as seguintes chaves (com os respectivos valores):

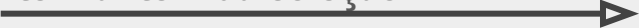
→ signo

→ nome_mae

Ao terminar, copie o dicionário e envie pelo chat.

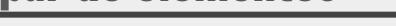
For loop com dicionários | Iterando chaves e valores

Na aula anterior vimos **for loop**. É quando temos uma coleção de dados e executamos uma operação item a item da coleção.



```
lista = ["Antonio", "João", "Pedro"]
for n in lista:
    print(n.upper())
# OUTPUT:
# ANTONIO
# JOÃO
# PEDRO
```

E agora que estamos trabalhando com dicionários, como fica? Afinal, **um item é composto de um par de elementos: chave e valor**.



```
eu = {
    "nome": "Rodolfo",
    "sobrenome": "Viana",
    "idade": 40
}
for i in eu:
    print(i)
# OUTPUT:
# nome
# sobrenome
# idade
```

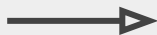
Reparem que *for loop* convencional não retorna os valores; apenas as chaves

For loop com dicionários | Iterando chaves e valores

Para iterar sobre chaves e valores:

1. distinguimos chave e valor no *for*,
2. usamos o método `.items()`

```
eu = {  
    "nome": "Rodolfo",  
    "sobrenome": "Viana",  
    "idade": 40  
}
```



```
for i in eu:  
    print(i)  
# OUTPUT:  
# nome  
# sobrenome  
# idade
```

```
for k, v in eu.items():  
    print(v)  
# OUTPUT:  
# Rodolfo  
# Viana  
# 40
```

Usei `k` para *key* e `v` para *value*, e pedi para imprimir apenas os `v`

For loop com dicionários | Iterando chaves e valores

Trabalhando chave e valor de maneira distinta com `.items()`, conseguimos fazer qualquer operação. Exemplo:

```
prods = {'maçã': 0.40, 'laranja': 0.35, 'banana': 0.25}
for k, v in prods.items():
    print("Com desconto de 10%, {} custa {}".format(k, round(v * 0.9, 2)))
```

Tomei um dicionário de produtos com o preço normal e retornei cada produto e o cálculo de 10% de desconto no preço

Tente reproduzir no seu computador.

Dicionários | Listas de dicionários

Temos **lista de dicionários**.
Como acessar os dados?

```
alunos = [  
    {  
        nome: 'Rodolfo',  
        sobrenome: 'Viana',  
        idade: 40,  
        cidade_nascimento: 'Marília',  
        tem_pet: False,  
        quantidade_pet: 0,  
    },  
    {  
        nome: 'Fernanda',  
        sobrenome: 'Torres',  
        idade: 21,  
        cidade_nascimento: 'São Paulo',  
        tem_pet: True,  
        quantidade_pet: 2,  
    },  
    {  
        nome: 'Roman',  
        sobrenome: 'Polanski',  
        idade: 30,  
        cidade_nascimento: 'Brasília',  
        tem_pet: False,  
        quantidade_pet: 0,  
    }  
]
```

1

Como **cada dicionário é um elemento da lista**,
usamos colchetes e a
posição na lista: `alunos[0]`,
`alunos[2]`, `alunos[-1]`...

```
print(alunos[0])  
# OUTPUT  
# {'nome': 'Rodolfo',  
#  'sobrenome': 'Viana', 'idade':  
#  40, 'cidade_nascimento':  
#  'Marília', 'tem_pet': False,  
#  'quantidade_pet': 0}
```

2

...e, no dicionário que
queremos, buscamos a chave
desejada:

```
alunos[0]["nome"],  
alunos[2]["idade"]...
```

```
print(alunos[0]["idade"])  
# OUTPUT  
# 40
```

For loop com lista de dicionários | Mais iterações

```
alunos = [  
    {  
        nome: 'Rodolfo',  
        sobrenome: 'Viana',  
        idade: 40,  
        cidade_nascimento: 'Marília',  
        tem_pet: False,  
        quantidade_pet: 0,  
    },  
    {  
        nome: 'Fernanda',  
        sobrenome: 'Torres',  
        idade: 21,  
        cidade_nascimento: 'São Paulo',  
        tem_pet: True,  
        quantidade_pet: 2,  
    },  
    {  
        nome: 'Roman',  
        sobrenome: 'Polanski',  
        idade: 30,  
        cidade_nascimento: 'Brasília',  
        tem_pet: False,  
        quantidade_pet: 0,  
    }  
]
```

Temos **lista de dicionários**. Como fazer *for loop*?

```
for a in alunos:  
    for k, v in a:  
        print("A chave {} tem o valor {}".format(k, v))
```

Como colocar numa lista separada apenas os nomes de quem tem mais de 25 anos?

```
acima = list()  
for a in alunos:  
    if a["idade"] > 25:  
        acima.append(a["nome"])
```

Entenda mais | Material complementar

Para assistir

- *Encontrando valores em listas com dicionários em Python*, em Otávio Miranda [[link](#)]
- *Percorrendo listas, tuplas, dicionários e conjuntos*, em Marcos Castro [[link](#)]

Para ler

- *Dictionaries in Python*, em Real Python [inglês][[link](#)]
- *How to Iterate Through a Dictionary in Python*, em Real Python [inglês][[link](#)]
- *Loops e estruturas de repetição no Python*, em Python Academy [[link](#)]

Documentação

- *Comandos for* [[link](#)]