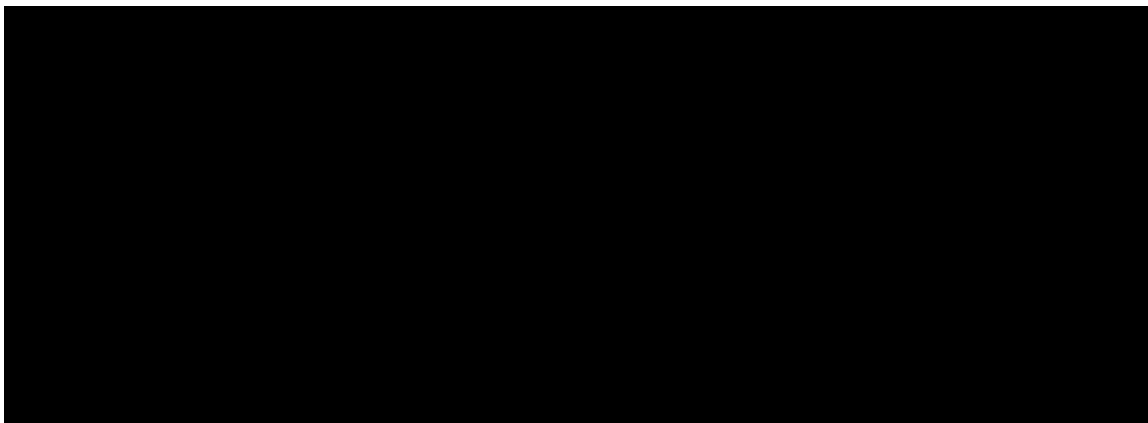


DICIONÁRIOS

O CONCEITO DE DICIONÁRIOS NO PYTHON

Nesse vídeo, a expert [Patrícia Pampanelli](#), Deep Learning Solutions Architect da NVIDIA, apresenta os principais conceitos sobre dicionários no Python.



00:00



MANIPULAÇÃO DE DADOS COM DICIONÁRIOS

O objetivo desta aula é entender o conceito da estrutura de dados de dicionários. Esta estrutura é utilizada na linguagem Python para armazenamento e manipulação de dados. A seguir vocês podem encontrar um exemplo de criação de dicionário:

```
entretenimento = {'séries': 'The Crown' }
```

CONCEITO DE DICIONÁRIOS

Os dicionários são estruturas de dados em Python utilizadas para mapear **valores arbitrários (ou chaves)** em **valores**. No exemplo abaixo, podemos observar esses conceitos. O dicionário a seguir é composto de 3 pares de **chave : valor**.

DICIONÁRIOS

```
chave_2 : valor_2,  
'chave_3' : valor_3  
}
```

CHAVES

As chaves de um dicionário deve ser representada por um valor **imutável**.

As chaves de um dicionário em Python podem ser dos seguintes tipos: inteiros (*integer*), valores reais (*float*), texto (*string*) ou verdadeiro ou falso (*boolean*). Seguem alguns exemplos destes dicionários:

INTEIROS

```
my_dict = {  
    2: 'dois',  
    3: 'três',  
    16: 'dezesseis'  
}
```

STRING

```
entretenimento = {  
    'filmes': 'Soul',  
    'séries': 'The Crown',  
    'livros': 'O Pequeno Príncipe'  
}
```

VALORES

Os valores de um dicionário podem ser de diversos tipos (inteiros, valores reais, texto, booleanos) e até mesmo outros dicionários e listas.

Esses dois últimos vamos exemplificar a seguir:

LISTAS

```
primary_colors = {  
    'red': [255, 0, 0],  
    'green': [0, 255, 0],  
}
```

DICIONÁRIOS

DICIONÁRIOS

```
entretenimento = {  
    'filmes': {  
        'action': 'Fast and Furious',  
        'romance': 'Love is blind'  
    },  
    'séries': 'The Crown',  
    'livros': 'O Pequeno Príncipe'  
}
```

O exemplo apresentado acima é o que chamamos de dicionário com dois níveis. Para acessar o valor do segundo nível, você precisa utilizar a chave do primeiro nível ('filmes') e, posteriormente, a chave do segundo nível. Desta forma, tanto o acesso como a inclusão de novos valores funcionam de forma bastante semelhante aos dicionários que vimos até este momento:

Imprimindo o filme de ação:

```
print(entretenimento['filmes']['action'])
```

Inserindo um novo gênero de filme:

```
entretenimento['filmes']['pixar'] = 'Soul'
```

VERIFICANDO SE UMA CHAVE FAZ PARTE DO DICIONÁRIO (IN OU NOT IN)

Para verificar se uma chave faz parte do dicionário ou não você pode utilizar **in** ou **not in** da mesma forma que você faz para as listas:

```
primary_colors = {  
    'red': [255, 0, 0],  
    'green': [0, 255, 0],  
    'blue': [0, 0, 255]  
}
```

DICIONÁRIOS

ACESSANDO OS ELEMENTOS DE UM DICIONÁRIO

Agora Patrícia apresenta as formas de acessar os valores de um dicionário utilizando a linguagem Python.



03:43



ACESSANDO OS ELEMENTOS DE UM DICIONÁRIO (ATRAVÉS DA CHAVE OU DA FUNÇÃO GET)

Para acessar os elementos de um dicionário você pode utilizar a própria chave como forma de acesso:

```
entretenimento = {  
    'filmes': 'Soul',  
    'séries': 'The Crown',  
    'livros': 'O Pequeno Príncipe'  
}  
print(entretenimento['filmes']) # Irá imprimir 'Soul'
```

Para acessar os elementos associados a uma chave você também pode utilizar a função **get** passando como parâmetro uma chave:

DICIONÁRIOS

```
        'series': 'The Crown',  
        'livros': 'O Pequeno Príncipe'  
    }  
    print(entretenimento.get('filmes')) # Irá imprimir 'Soul'
```

INSERINDO ELEMENTOS EM UM DICIONÁRIO

As operações de inserção em um dicionário são bastante comuns durante a manipulação de dados com dicionários.

Você pode inserir novos pares de **chave** e **valor** da seguinte forma:

```
entretenimento = {  
    'filmes': 'Soul',  
    'séries': 'The Crown',  
    'livros': 'O Pequeno Príncipe'  
}  
entretenimento['cantoras'] = 'Taylor Swift'
```

O resultado dessa inserção é:

```
entretenimento = {  
    'filmes': 'Soul',  
    'séries': 'The Crown',  
    'livros': 'O Pequeno Príncipe',  
    'cantoras': 'Taylor Swift'  
}
```

REMOVENDO ELEMENTOS DE UM DICIONÁRIO (FUNÇÃO DEL)

Além do acesso e da inserção de novos pares de **chave** e **valor** em um dicionário, você pode remover uma **chave** e os seus respectivos valores:

```
entretenimento = {  
    'filmes': 'Soul',  
    # ...  
}
```

DICIONÁRIOS

```
del entretenimento['livros']
```

Após a remoção, você terá um dicionário da seguinte forma:

```
entretenimento = {  
    'filmes': 'Soul',  
    'séries': 'The Crown'  
}
```

AVANÇAR

DICIONÁRIOS

CONCLUSÃO



Parabéns! Que bom que você chegou até aqui.

Agora você conhece os dicionários no Python.

Revise os aprendizados desta aula e avalie se você já tem segurança para realizar essas ações.

CHECKLIST

Ao final desta aula, você deve sentir segurança para:

- Entender conceito de “**Chave: Valor**”
- Alterar o valor de um dicionário passando uma **chave**
- Criar dicionário através de **dict()**
- Criar dicionário através de **{}**
- Adicionar **pares de valores** ao dicionário
- Acessar **elementos** no dicionário através da chave

DICIONÁRIOS

Caso ainda tenha dúvidas, não hesite em voltar aos conteúdos e buscar ajuda, esses fundamentos são super importantes para você construir uma base de trabalho sólida.

Esta aula ficará disponível para você rever quando quiser.

[Notebook utilizado no vídeo](#)

Até breve!

CONCLUIR