

**Educação
Profissional
Paulista**

Técnico em
**Ciência de
Dados**

Variáveis e tipos de dados

Dicionários

Aula 2

[DADOS]ANO1C2B2S12A2

Exposição



Objetivos da Aula

- Introduzir conceito de operação de adição, remoção e atualização em dicionários Python.



Competências da Unidade (Técnicas e Socioemocionais)

- Ser proficiente em linguagens de programação para manipular e analisar grandes conjuntos de dados;
- Usar técnicas para explorar e analisar dados, aplicar modelos estatísticos, identificar padrões, realizar inferências e tomar decisões baseadas em evidências;
- Colaborar, efetivamente, com outros profissionais, como cientistas de dados e engenheiros de dados;
- Trabalhar em equipes multifuncionais colaborando com colegas, gestores e clientes.



Recursos Didáticos

- Recurso audiovisual para exibição de vídeos e imagens;
- Acesso ao laboratório de informática e/ou internet;
- Software Anaconda/Jupyter Notebook instalado, ou similar.



Duração da Aula

50 minutos

Relembrando

Dicionário é uma estrutura de dados que armazena diferentes tipos de dados e funciona como um mapeamento, ou seja, temos uma chave e um valor associado a essa chave. Para encontrar algo no dicionário, basta procurar pela chave que você quer.

dicionario = {chave: valor}

A **chave**, no geral, é **string**; já o valor é qualquer tipo de dado.

Operações: adição e remoção de elementos

Dicionários em Python permitem adicionar novos pares chave-valor e remover elementos existentes.

```
# Criando um dicionário inicial
meu_dicionario = {'nome': 'João', 'idade': 25, 'cidade': 'São Paulo'}

# Acesso a elementos
nome_da_pessoa = meu_dicionario['nome']
print(f"Nome da pessoa: {nome_da_pessoa}")

# Adição de um novo par chave-valor
meu_dicionario['profissao'] = 'Programador'

# Remoção de um par chave-valor
del meu_dicionario['cidade']

# Exibindo o dicionário após as operações
print("Dicionário atualizado:", meu_dicionario)
```

Nome da pessoa: João
Dicionário atualizado: {'nome': 'João', 'idade': 25, 'profissao': 'Programador'}

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Operações: atualização

Os valores associados às chaves podem ser atualizados, alterando-se diretamente o valor correspondente.

```
# Criando um dicionário inicial  
meu_dicionario = {'nome': 'João', 'idade': 25, 'cidade': 'São Paulo'}
```

```
# Acesso a elementos  
nome_da_pessoa = meu_dicionario['nome']  
print(f"Nome da pessoa: {nome_da_pessoa}")
```

```
# Adição de um novo par chave-valor  
meu_dicionario['profissao'] = 'Programador'
```

```
# Atualização de um valor existente  
meu_dicionario['nome'] = 'Carlos'  
meu_dicionario['profissao'] = 'Estudante'  
meu_dicionario['idade'] = 21
```

```
# Exibindo o dicionário após as operações  
print("Dicionário atualizado:", meu_dicionario)
```

Nome da pessoa: João

Dicionário atualizado: {'nome': 'Carlos', 'idade': 21, 'cidade': 'São Paulo', 'profissao': 'Estudante'}

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Métodos úteis

Já vimos que o Python fornece métodos como `keys()`, `values()`, e `items()` para obter listas de chaves, valores e pares chave-valor, respectivamente.

```
# Criando um dicionário
meu_dicionario = {'nome': 'Carlos', 'idade': 25, 'cidade': 'São Paulo', 'profissao': 'Estudante'}

# Exibindo todas as chaves e valores
print("Chaves e Valores:")
for chave, valor in meu_dicionario.items():
    print(f"{chave}: {valor}")

# Exibindo apenas as chaves
print("\nChaves:")
for chave in meu_dicionario.keys():
    print(chave)

# Exibindo apenas os valores
print("\nValores:")
for valor in meu_dicionario.values():
    print(valor)
```

Chaves e Valores:
nome: Carlos
idade: 25
cidade: São Paulo
profissao: Estudante

Chaves:
nome
idade
cidade
profissao

Valores:
Carlos
25
São Paulo
Estudante

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exemplos

Como criar um dicionário que armazena informações de contato?

```
1 # Dicionário representando informações de um contato
2 contato = {
3     'nome': 'Ana Silva',
4     'telefone': '123-456-7890',
5     'email': 'ana@email.com',
6     'idade': 30,
7     'cidade': 'São Paulo'
8 }
9
10 # Acesso às informações do contato
11 print(f"Nome: {contato['nome']}")
12 print(f"Telefone: {contato['telefone']}")
13 print(f"Email: {contato['email']}")
14 print(f"Idade: {contato['idade']}")
15 print(f"Cidade: {contato['cidade']}")
16
```

Nome: Ana Silva
Telefone: 123-456-7890
Email: ana@email.com
Idade: 30
Cidade: São Paulo

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exemplos

Como criar uma lista com nomes e notas dos alunos?

```
# Dicionário de notas dos alunos  
notas_alunos = {'Alice': 85, 'Bob': 92, 'Charlie': 78, 'Diana': 95}  
  
# Utilizando um loop para exibir as notas de cada aluno  
for aluno, nota in notas_alunos.items():  
    print(f"{aluno}: {nota} pontos")
```

```
Alice: 85 pontos  
Bob: 92 pontos  
Charlie: 78 pontos  
Diana: 95 pontos
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exemplos

Será que conseguimos colocar um dicionário dentro do dicionário?

```
# Dicionário aninhado representando informações sobre livros
biblioteca = {
    'livro1': {'titulo': 'Aventuras Fantásticas', 'autor': 'João Silva'},
    'livro2': {'titulo': 'Código Mestre', 'autor': 'Maria Oliveira'},
    'livro3': {'titulo': 'Noite Sombria', 'autor': 'Carlos Souza'}
}

# Acesso às informações de um livro específico
livro_id = 'livro2'
print(f"Título: {biblioteca[livro_id]['titulo']}")
print(f"Autor: {biblioteca[livro_id]['autor']}")
```

```
Título: Código Mestre
Autor: Maria Oliveira
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.



Vamos
fazer uma
atividade

Atividade: Times de futebol

Confira as orientações para a atividade ao lado:

 **25 minutos**

 **Em grupo**

- 1** Faça uma **pesquisa, na sala de aula**, com todos os estudantes anotando o **time de futebol** de preferência.
- 2** Crie um **dicionário** com o nome **times_futebol**, usando, como chave, o nome do time de futebol e, como valor, a quantidade de estudantes que torcem por aquele time.
- 3** Agora, crie um dicionário **quantidade_times**, cuja chave é a quantidade e o valor, o nome do time.
- 4** Acesse o time **“Palmeiras”** nos dois dicionários.

Ao finalizar a atividade, envie pelo AVA (Ambiente Virtual de Aprendizagem) arquivo com extensão .ipynb.



O que nós
aprendemos
hoje?

© Getty Images

Hoje desenvolvemos:

- 1 Conhecimento sobre operações com dicionário em Python, tais como: adição, remoção e atualização.
- 2 Compreensão sobre métodos como `keys()`, `values()`, e `items()` que o Python oferece para obter listas de chaves, valores e pares chave-valor.
- 3 Exercício com aplicação prática dos conceitos estudados sobre operações com dicionário em Python



Saiba mais

Confira outras **operações de dicionários** acessando o link abaixo:

SILVEIRA, G. *Python collections parte 2: conjuntos e dicionários. Mais operações de dicionários*. Alura, 2023. Disponível em:
<https://cursos.alura.com.br/course/python-collections-conjuntos-e-dicionarios/task/53515>. Acesso em: 13 mar. 2024.

Caso queira, também é possível **rever o conceito** de dicionário neste outro link:

COSTA, M. *Python para data science: primeiros passos. Dicionário*. Alura, 2024. Disponível em:
<https://cursos.alura.com.br/course/python-data-science-primeiros-passos/task/122400>. Acesso em: 13 mar. 2024.

Referências da aula

COSTA, M. *Python para data science: primeiros passos*. Dicionário. Alura, 2024. Disponível em: <https://cursos.alura.com.br/course/python-data-science-primeiros-passos/task/122400>. Acesso em: 13 mar. 2024.

MENEZES, N. N. C. *Introdução à programação com Python: algoritmos e lógica de programação para iniciantes*. São Paulo: Novatec, 2019.

SILVEIRA, G. *Python collections parte 2: conjuntos e dicionários*. Mais operações de dicionários. Alura, 2023. Disponível em: <https://cursos.alura.com.br/course/python-collections-conjuntos-e-dicionarios/task/53515>. Acesso em: 13 mar. 2024.

Identidade Visual: imagens © Getty Images

**Educação
Profissional
Paulista**

Técnico em
**Ciência de
Dados**