Educação Profissional Paulista

Técnico em
Ciência de Dados

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Variáveis e tipos de dados

Dicionários

Aula 1

[DADOS]ANO1C2B2S12A1

Exposição

Objetivos da Aula

- Introduzir o conceito de dicionários em 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

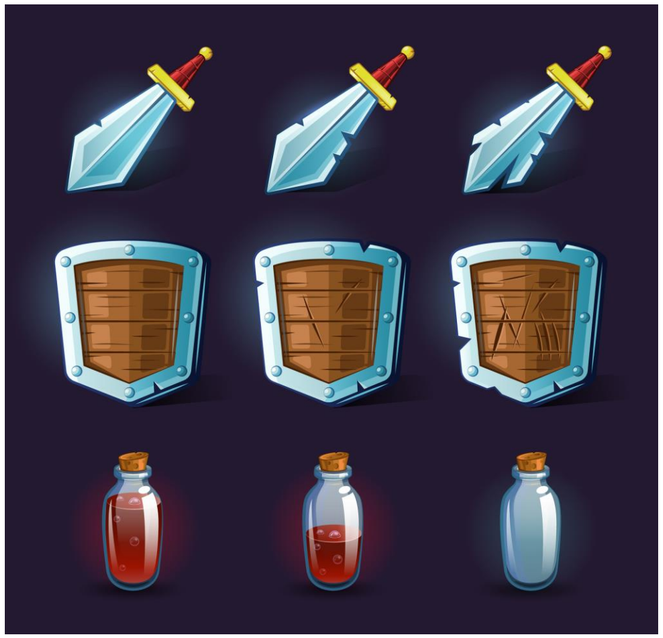
- Recurso audiovisual para exibição de vídeos e imagens;
- Acesso ao laboratório de informática e/ou internet;
- Software Anaconda/Jupyter Notebook instalado, ou similar.

Duração da Aula

50 minutos

Exposição

Motivação: jogo de RPG - Inventário de jogador



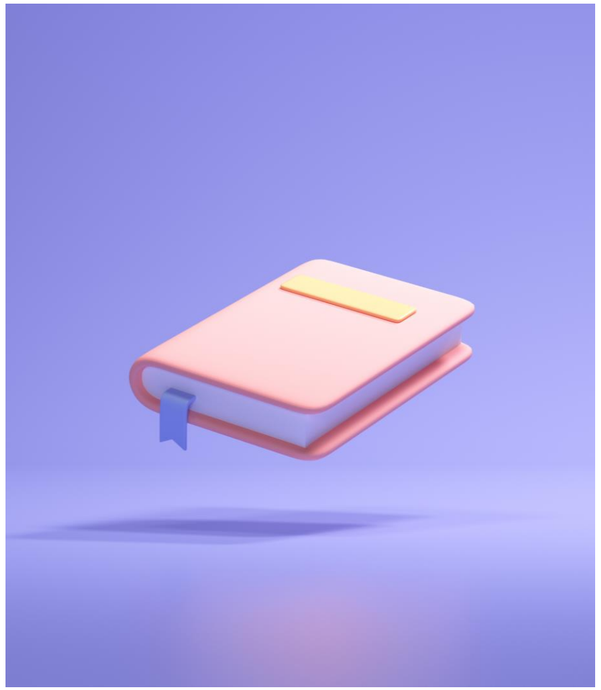
© Getty Images

Imagine que você está desenvolvendo um jogo de **RPG (*Role-playing game*) online**, em que os jogadores têm inventários para armazenar itens como armas, poções, armaduras etc. Cada jogador pode ter um **dicionário de inventário**, no qual as chaves representam os tipos de itens, e os valores representam a **quantidade** de cada item.

Que estrutura de dados em Python você usaria para armazenar os dados?

Exposição

Dicionário



© Getty Images

Um **dicionário em Python** é uma **estrutura de dados** que permite **armazenar pares chave-valor**. Cada chave deve ser única e associada a um valor específico.

Os dicionários são conhecidos por sua capacidade de associar chaves a valores, permitindo que você **recupere** facilmente o valor associado a uma determinada chave.

Essa recuperação facilitada é o principal diferenciador dos dicionários em relação a outras estruturas de dados, aumentando a performance do código na consulta de informações.

Exposição

Dicionário

A sintaxe básica de um dicionário inclui chaves - **{ }** - para delimitar os elementos e usar pares chave-valor separados por dois pontos :

```
meu_dicionario = {'chave1': 'valor1', 'chave2': 'valor2', ...}
```

```
print(type(meu_dicionario))
<class 'dict'>
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Exemplo

De acordo com a definição, qual dos exemplos abaixo é um dicionário?

```
# Exemplos

exemplo1 = [1, 2, 3, 4]
exemplo2 = {1, 2, 3}
exemplo3 = {'key': 'value', 'name': 'John'}
exemplo4 = (10, 20, 30)
exemplo5 = 5.5
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Criando um dicionário

1. Sintaxe de chaves e valores:

```
: # Criando um dicionário vazio
meu_dicionario = {}

# Adicionando pares chave-valor
meu_dicionario['nome'] = 'João'
meu_dicionario['idade'] = 25
meu_dicionario['cidade'] = 'São Paulo'

print(meu_dicionario)
```

```
{'nome': 'João', 'idade': 25, 'cidade': 'São Paulo'}
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Criando um dicionário

2. Usando a função dict():

```
# Criando um dicionário usando a função dict()
outro_dicionario = dict(nome='Maria', idade=30, cidade='Campinas')

print(outro_dicionario)
```

```
{'nome': 'Maria', 'idade': 30, 'cidade': 'Campinas'}
```

3. Atribuindo dicionários diretamente:

```
# Criando um dicionário diretamente
dicionario_direto = {'a': 1, 'b': 2, 'c': 3}

print(dicionario_direto)
```

```
{'a': 1, 'b': 2, 'c': 3}
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Acessando dicionários

1. Acesso direto

Os elementos do dicionário podem ser acessados através de suas chaves. O acesso é feito utilizando a notação de colchetes: **[]**.

```
meu_dicionario = {'nome': 'João', 'idade': 25, 'cidade': 'São Paulo'}

# Acessando o valor associado à chave 'idade'
idade = meu_dicionario['idade']
print(idade)
```

```
25
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Acessando dicionários

2. Método get()

O método **get**() permite acessar um valor com uma chave e fornece um valor padrão se a chave não existir.

```
cidade = meu_dicionario.get('cidade', 'Cidade não especificada')
print(cidade)

# Caso a chave não exista
profissao = meu_dicionario.get('profissao', 'Não especificada')
print(profissao)
```

```
São Paulo
Não especificada
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Acessando dicionários

3. Iterando pelas chaves

Você pode iterar pelas chaves e acessar os valores correspondentes:

```
for chave in meu_dicionario:
    valor = meu_dicionario[chave]
    print(f'{chave}: {valor}')
```

```
nome: João
idade: 25
cidade: São Paulo
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Acessando dicionários

4. Método keys() e iteração

O método **keys**() retorna uma visão das chaves do dicionário, que pode ser usada para iterar:

```
for chave in meu_dicionario.keys():
    valor = meu_dicionario[chave]
    print(f'{chave}: {valor}')
```

```
nome: João
idade: 25
cidade: São Paulo
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Acessando dicionários

5. Método items()

O método **items**() retorna pares chave-valor como tuplas e pode ser usado em um loop:

```
for chave, valor in meu_dicionario.items():
    print(f'{chave}: {valor}')
```

```
nome: João
idade: 25
cidade: São Paulo
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Como criar e acessar o inventário do jogo de RPG?

Imagine que você está desenvolvendo um jogo de **RPG (*Role-playing game*) online**, em que os jogadores têm inventários para armazenar itens como armas, poções, armaduras etc. Cada jogador pode ter um **dicionário de inventário**, no qual as chaves representam os tipos de itens e os valores representam a **quantidade** de cada item. Que estrutura de dados em Python você usaria para armazenar os dados?

```python
# Dicionário de Inventário de Jogador
inventario_jogador = {'armas': 3, 'poções': 5, 'armaduras': 2, 'moedas': 100}

# Acesso aos itens do inventário
armas_do_jogador = inventario_jogador['armas']
poções_do_jogador = inventario_jogador['poções']
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Vamos fazer um **quiz**

Qual é a maneira correta de acessar o valor associado à chave 'idade' em um dicionário chamado dados?

- dados(idade)
- dados['idade']
- dados.valor('idade')
- dados.take('idade')

Vamos fazer um **quiz**

Qual é a maneira correta de acessar o valor associado à chave 'idade' em um dicionário chamado dados?

- ✗ dados(idade)
- ✓ dados['idade']
- ✗ dados.valor('idade')
- ✗ dados.take('idade')

FEEDBACK GERAL DA ATIVIDADE

Em Python, você acessa o valor associado a uma chave em um dicionário usando colchetes []. A sintaxe correta para acessar o valor associado à chave 'idade' em um dicionário chamado dados é **dados['idade']**.

Registro

Vamos fazer um **quiz**

Qual das seguintes opções é uma maneira válida de criar um dicionário em Python?

- `novo_dicionario = {1, 2, 3, 4}`
- `novo_dicionario = dict(1='um', 2='dois', 3='três')`
- `novo_dicionario = {'nome': 'Alice', 'idade': 25, 'cidade': Santos'}`
- `novo_dicionario = dict(['nome', 'Alice', 'idade', 25, 'cidade', Santos'])`

Registro

Vamos fazer um **quiz**

Qual das seguintes opções é uma maneira válida de criar um dicionário em Python?

✗ `novo_dicionario = {1, 2, 3, 4}`

✗ `novo_dicionario = dict(1='um', 2='dois', 3='três')`

✓ `novo_dicionario = {'nome': 'Alice', 'idade': 25, 'cidade': Santos'}`

✗ `novo_dicionario = dict(['nome', 'Alice', 'idade', 25, 'cidade', Santos'])`

FEEDBACK GERAL DA ATIVIDADE

A alternativa correta é **`novo_dicionario = {'nome': 'Alice', 'idade': 25, 'cidade':'Santos'}`**. Essa é a sintaxe correta para criar um dicionário em Python usando chaves e pares chave-valor.

Registro

Vamos fazer um **quiz**

O que é um dicionário em Python?

- Uma sequência ordenada de elementos.
- Uma coleção imutável de pares chave-valor.
- Uma estrutura de dados que armazena apenas números inteiros.
- Uma coleção mutável e indexada de pares chave-valor.

Registro

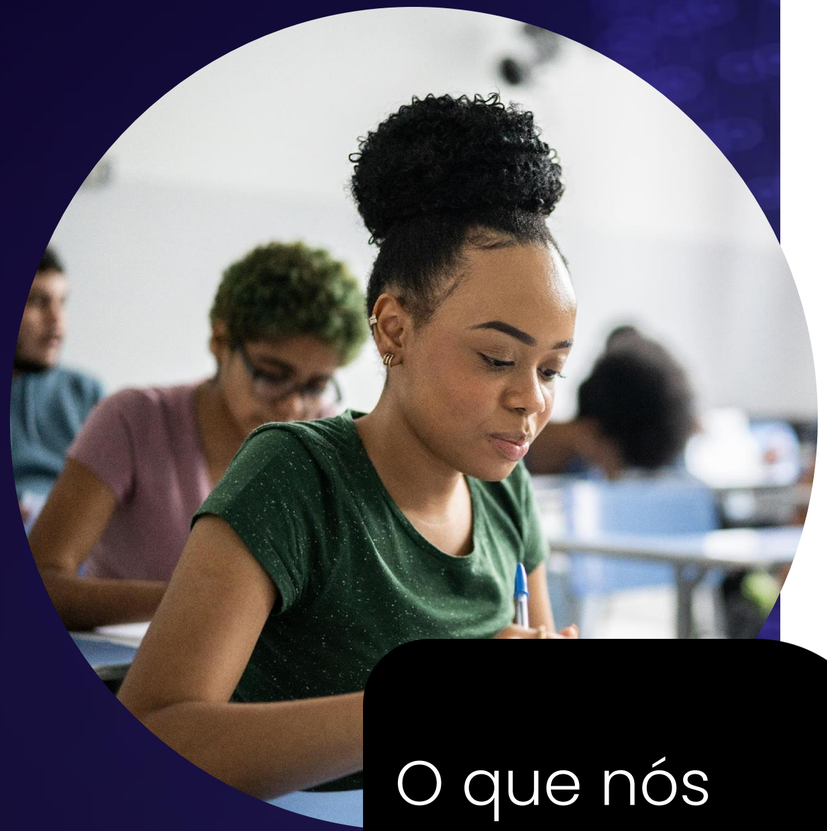
Vamos fazer um **quiz**

O que é um dicionário em Python?

- ✗ Uma sequência ordenada de elementos.
- ✗ Uma coleção imutável de pares chave-valor.
- ✗ Uma estrutura de dados que armazena apenas números inteiros.
- ✓ Uma coleção mutável e indexada de pares chave-valor.

FEEDBACK GERAL DA ATIVIDADE

Em Python, um dicionário é uma estrutura de dados que permite armazenar pares únicos de chave-valor, mutável (pode ser modificado após a criação) e indexado (os valores podem ser acessados por meio das chaves).



© Getty Images

O que nós **aprendemos hoje?**

Hoje desenvolvemos:

1. Conhecimento sobre o conceito de dicionário em Python.

2. Experiência prática de criação e acesso a dicionários em Python.

Saiba mais

Quer saber mais sobre dicionário em Python? Clique nos links abaixo e se aprofunde neste conceito.

SILVEIRA, G. *Python* collections *parte 2*: conjuntos e dicionários. Alura, 2023. Disponível em: https://cursos.alura.com.br/course/python-collections-conjuntos-e-dicionarios/task/53514. Acesso em: 13 mar. 2024.

ORESTES, Y. Python: Trabalhando com dicionários. Alura, 2018. Disponível em: https://www.alura.com.br/artigos/trabalhando-com-o-dicionario-no-python. Acesso em: 13 mar. 2024.

Referências da aula

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

ORESTES, Y. Python: Trabalhando com dicionários. Alura, 2018. Disponível em: https://www.alura.com.br/artigos/trabalhando-com-o-dicionario-no-python. Acesso em: 13 mar. 2024.

SILVEIRA, G. *Python* collections *parte 2*: conjuntos e dicionários. Alura, 2023. Disponível em: https://cursos.alura.com.br/course/python-collections-conjuntos-e-dicionarios/task/53514. Acesso em: 13 mar. 2024.

Educação Profissional Paulista

Técnico em
Ciência de Dados

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

S12 – Aula 1 – Quiz

Condições de conclusão

Ver

Qual é a maneira correta de acessar o valor associado à chave 'idade' em um dicionário chamado dados?

- dados.take('idade')
- dados(idade)
- dados.valor('idade')
- dados['idade']

Qual das seguintes opções é uma maneira válida de criar um dicionário em Python?

- novo_dicionario = {'nome': 'Alice', 'idade': 25, 'cidade': Santos'}
- novo_dicionario = dict(1='um', 2='dois', 3='três')
- novo_dicionario = {1, 2, 3, 4}
- novo_dicionario = dict(['nome', 'Alice', 'idade', 25, 'cidade', Santos'])

O que é um dicionário em Python?

- Uma estrutura de dados que armazena apenas números inteiros.
- Uma coleção imutável de pares chave-valor.
- Uma coleção mutável e indexada de pares chave-valor.
- Uma sequência ordenada de elementos.

Disciplina
Programação Aplicada a Ciência de Dados 2º Bimestre

Curso
Técnico em Ciência de Dados

Ano letivo
2025

Retornar ao Sumário

Educação Profissional Paulista

Técnico em
Ciência de Dados

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

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

Exposição

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.

Exposição

Operações: adição e remoção de elementos

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

```python
# 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.

Exposição

Operações: atualização

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

```python
# 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.

```python
# 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.

Exposição

Exemplos

Como criar um dicionário que armazena informações de contato?

```python
# Dicionário representando informações de um contato
contato = {
    'nome': 'Ana Silva',
    'telefone': '123-456-7890',
    'email': 'ana@email.com',
    'idade': 30,
    'cidade': 'São Paulo'
}

# Acesso às informações do contato
print(f"Nome: {contato['nome']}")
print(f"Telefone: {contato['telefone']}")
print(f"Email: {contato['email']}")
print(f"Idade: {contato['idade']}")
print(f"Cidade: {contato['cidade']}")

```

```
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.

Exposição

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.

Exposição

Exemplos

Será que conseguimos colocar um dicionário dentro do dicionário?

```python
# 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.

Registro

Vamos fazer uma **atividade**

Atividade: Times de futebol

Confira as orientações para a atividade ao lado:

25 minutos

Em grupo

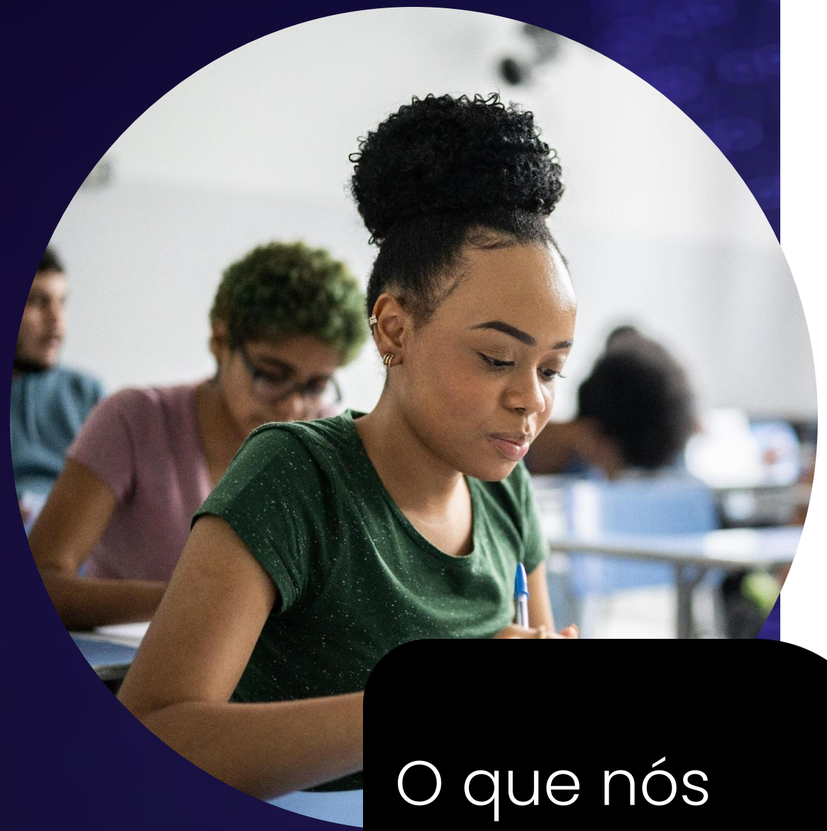
1 Faça uma **pesquisa, na sala de aula,** com todos os estudantes anotando o **time de futebol** de preferência.

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.



© Getty Images

O que nós **aprendemos hoje?**

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: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.

Educação Profissional Paulista

Técnico em
Ciência de Dados

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

S12 – Aula 2 – Registro

Atividade: Times de futebol

Para fazer a atividade, confira as orientações a seguir:

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

Condições de conclusão

Fazer um envio

Resumo das Avaliações

Turmas separadas: 293566972 | 2ª SERIE BT MANHA ANUAL | 99 | JOAO CRUZ PROF

Oculto para estudantes	Não
Participantes	43
Enviado	0
Precisa ser avaliado	0

Disciplina
Programação Aplicada a Ciência de Dados 2º Bimestre

Curso
Técnico em Ciência de Dados

Ano letivo
2025

Retornar ao Sumário

Educação Profissional Paulista

Técnico em
Ciência de Dados

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Variáveis e tipos de dados

Dicionários

Aula 3

[DADOS]ANO1C2B2S12A3

Exposição

Objetivos da Aula

- Conhecer outros métodos de dicionários e aplicar os seus conceitos em 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

Exposição

Métodos de dicionário

clear(): remove todos os itens do dicionário.

```
meu_dicionario = {'a': 1, 'b': 2, 'c': 3}
meu_dicionario.clear()
```

copy(): retorna uma cópia rasa do dicionário.

```
dicionario_original = {'a': 1, 'b': 2, 'c': 3}
dicionario_copia = dicionario_original.copy()
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Métodos de dicionário

get(chave, valor_default): retorna o valor associado à chave, ou um valor padrão, caso a chave não exista.

```
meu_dicionario = {'a': 1, 'b': 2, 'c': 3}
valor = meu_dicionario.get('b', 0)
```

pop(chave, [valor_default]): remove a chave do dicionário e retorna seu valor. Se a chave não existir, pode retornar um valor padrão, se esse for fornecido.

```
meu_dicionario = {'a': 1, 'b': 2, 'c': 3}
valor = meu_dicionario.pop('b')
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Exercícios

1. Crie um dicionário cujas chaves são os meses do ano e os valores são a duração (em dias) de cada mês.

Obs.: os meses devem ser escritos com letras minúsculas.

2. Imprima as chaves seguidas dos seus valores para dicionário criado no exercício 1.

3. Imprima como no formato abaixo:

```
Janeiro - 31 dias
Fevereiro - 28 dias
Março - 31 dias
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Exercícios

4. Crie um **dicionário** para as seguintes relações de fruta e preço:
   'maçã': 2.5,
   'banana': 1.0,
   'uva': 3.0,
   'morango': 4.5,
   'laranja': 2.0,
   'abacaxi': 5.0,
   'kiwi': 3.5,
   'melancia': 6.0,
   'pêssego': 4.0,
   'manga': 3.8

Exposição

Exercícios

5. Qual o valor do **abacaxi**? Mostre o valor da chave “abacaxi”.

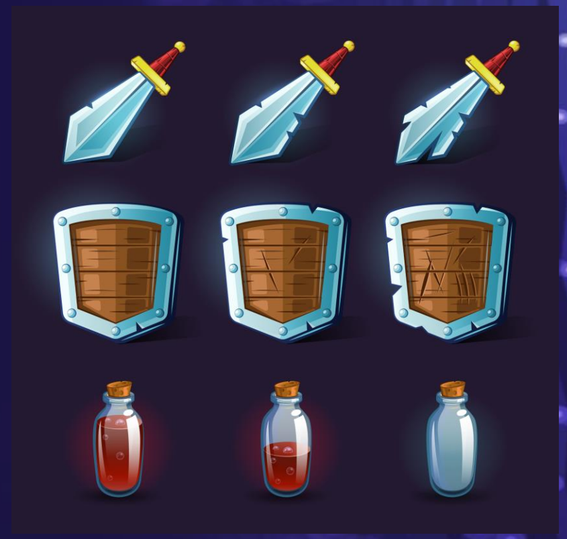
6. Qual o valor do **melão**? Mostre o valor da chave “melão”.

7. O preço do **kiwi** foi alterado para 4. Altere o valor da chave “kiwi” no dicionário do exercício anterior para 4.

8. O estoque de manga acabou. Apague a chave “manga” do dicionário de frutas.

9. Acabou de chegar no estoque a goiaba ao preço de 2.8. Insira a chave “goiaba” com valor 2.8.

Vamos fazer uma **atividade**



© Getty Images

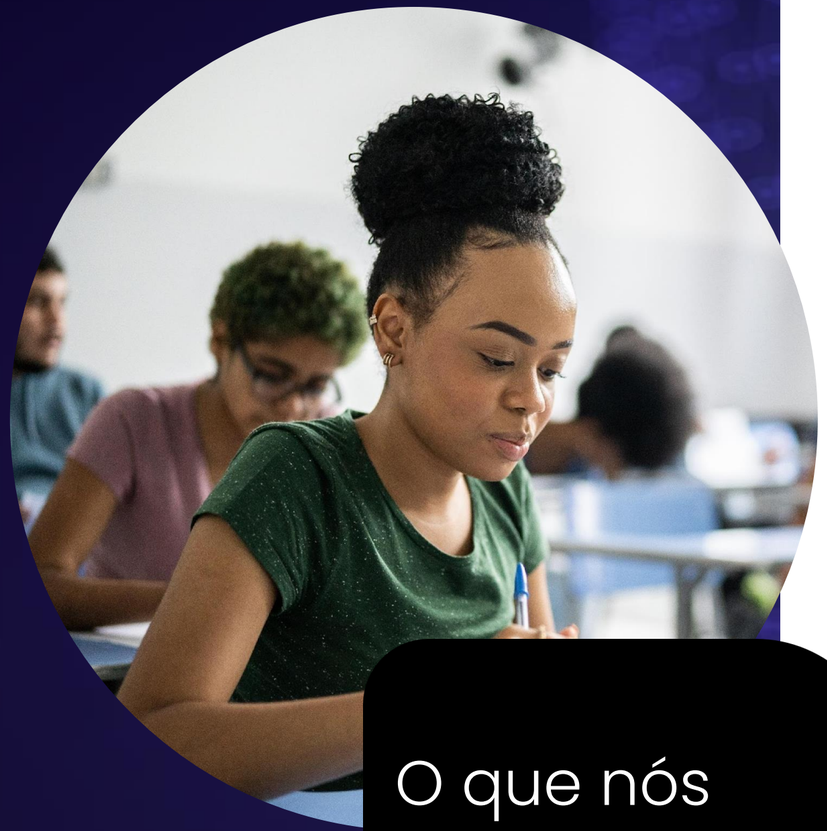
Inventário do jogo de RPG

Imagine que você está jogando um jogo de **RPG (*role-playing game*),** em que os jogadores têm **inventários para armazenar itens** como armas, poções, armaduras etc. Você conseguiu acessar o dicionário do seu perfil e quer fazer algumas alterações para trapacear no jogo.

inventario_jogador = {'armas': 3, 'poções': 5, 'armaduras': 2, 'moedas': 100, energia: [2, 1, 5]}

a) No dicionário acima, aumente o número de moedas para 500.
b) Imprima o valor de armas que você tem.
c) Apague a chave “armaduras”.
d) Insira ‘armaduras_poderosas': 10.
e) Acesse e substitua a energia 1 por 3.

Ao final da atividade, envie arquivo com extensão .ipynb pelo meio que o professor indicar



© Getty Images

O que nós **aprendemos hoje?**

Hoje desenvolvemos:

1. Compreensão de outros métodos de dicionários em Python, seus conceitos e funções.

2. Exercícios práticos contemplando os diferentes métodos de dicionários.

3. Atividade prática para criação de dicionário, com inserção de novos dados e alteração de informações.

Saiba mais

Quer aprender mais sobre **conceitos de dicionários**? No link abaixo é possível encontrar mais informações sobre o tema:
COSTA, M. *Python para* data science: primeiros passos. Aprofundando em dicionários. Alura, 2024. Disponível em: https://cursos.alura.com.br/course/python-data-science-primeiros-passos/task/122401. Acesso em: 13 mar. 2024.

Gostaria de entender como funcionam **listas em dicionários**? Veja mais neste link:
COSTA, M. *Python para* data science: primeiros passos. Para saber mais: listas em dicionários. Alura, 2024. Disponível em: https://cursos.alura.com.br/course/python-data-science-primeiros-passos/task/123727. Acesso em: 13 mar. 2024.

Referências da aula

COSTA, M. *Python para* data science: primeiros passos. Aprofundando em dicionários. Alura, 2024. Disponível em: https://cursos.alura.com.br/course/python-data-science-primeiros-passos/task/122401. Acesso em: 13 mar. 2024.

COSTA, M. *Python para* data science: primeiros passos. Para saber mais: listas em dicionários. Alura, 2024. Disponível em: https://cursos.alura.com.br/course/python-data-science-primeiros-passos/task/123727. 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.

Identidade Visual: imagens © Getty Images

Educação Profissional Paulista

Técnico em
Ciência de Dados

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO