Educação Profissional Paulista

Técnico em Ciência de Dados

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Estrutura de Controle de Fluxo

Enumeração de Iteráveis

Aula 1

[DADOS]ANO1C2B2S11A1

Exposição

Objetivos da Aula

Introduzir o conceito de parâmetros-padrão nas funções.

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.

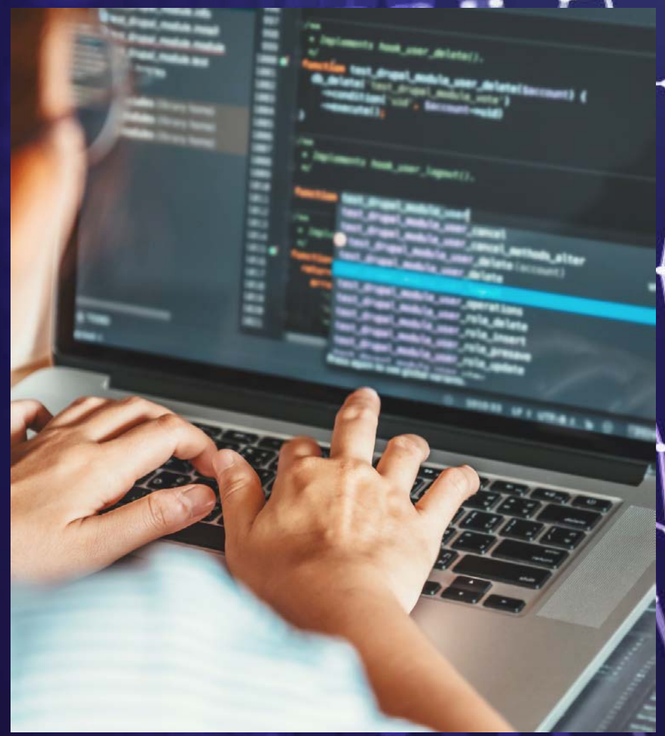
Recursos Didáticos

- 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



© Getty Images

Motivação: enumerando conquistas em um jogo

Imagine que você está jogando um jogo eletrônico emocionante, repleto de desafios e conquistas!

Você precisa enfrentar diferentes níveis, derrotar monstros e coletar itens valiosos.

Nisso, temos **3 conquistas** ordenadas na lista para seguir:

conquistas = ['Primeira Vitória', 'Derrotou o Chefe Final', 'Coletou o Tesouro Secreto’]

Como podemos gerar as conquistas no código abaixo?

```
1 conquistas = ['Primeira Vitória', 'Derrotou o Chefe Final', 'Coletou o Tesouro Secreto']
2
3 # to do
```

```
Conquista 1: Primeira Vitória
Conquista 2: Derrotou o Chefe Final
Conquista 3: Coletou o Tesouro Secreto
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Enumerando conquistas em um jogo

Uma alternativa é:

```
:
1 conquistas = ['Primeira Vitória', 'Derrotou o Chefe Final', 'Coletou o Tesouro Secreto']
2 indice = 1  # Valor inicial do índice
3
4 for valor in conquistas:
5     print(f'Índice: {indice}, Valor: {valor}')
6     indice += 1
```

```
Índice: 1, Valor: Primeira Vitória
Índice: 2, Valor: Derrotou o Chefe Final
Índice: 3, Valor: Coletou o Tesouro Secreto
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Porém, ela ainda não está no formato correto.

Exposição

Enumerando conquistas em um jogo

```
:  1 conquistas = ['Primeira Vitória', 'Derrotou o Chefe Final', 'Coletou o Tesouro Secreto']
   2 indice = 1  # Valor inicial do índice
   3
   4 for conquista in conquistas:
   5     print(f'Conquista {indice}: {conquista}')
   6     indice += 1
```

```
Conquista 1: Primeira Vitória
Conquista 2: Derrotou o Chefe Final
Conquista 3: Coletou o Tesouro Secreto
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Agora sim! E já pensou como ficaria se usássemos função?

Exposição

Enumerando conquistas em um jogo

```python
def imprimir_conquistas(conquistas, start=1):
    indice = start
    for conquista in conquistas:
        print(f'Conquista {indice}: {conquista}')
        indice += 1

# Exemplo de uso:
conquistas = ['Primeira Vitória', 'Derrotou o Chefe Final', 'Coletou o Tesouro Secreto']
imprimir_conquistas(conquistas)

```

```
Conquista 1: Primeira Vitória
Conquista 2: Derrotou o Chefe Final
Conquista 3: Coletou o Tesouro Secreto
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Agora é o momento de fazer alguns testes!

Exposição

Funções

```python
def imprimir_conquistas(conquistas, start=1):
    indice = start
    for conquista in conquistas:
        print(f'Conquista {indice}: {conquista}')
        indice += 1
```

```
: 1 conquistas = ['Primeira Vitória', 'Derrotou o Chefe Final', 'Coletou o Tesouro Secreto']
  2
  3 imprimir_conquistas(conquistas)
```

```
Conquista 1: Primeira Vitória
Conquista 2: Derrotou o Chefe Final
Conquista 3: Coletou o Tesouro Secreto
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Reflita

Qual o nome desta função? Quantos parâmetros ela tem e quais são eles?

Onde está o parâmetro start na linha 3 ao lado?

Exposição

Funções – parâmetros-padrão (ou opcionais)

O que é **parâmetro-padrão**?

Parâmetro-padrão (ou opcional) é um **valor atribuído** a um parâmetro de uma função no momento de sua definição.

Esse valor é utilizado caso o chamador da função não forneça um valor correspondente ao chamar a função.

Sua utilização acontece quando o argumento correspondente não é especificado durante a chamada da função.

Vamos compreender melhor na prática, com o exemplo a seguir.

Exposição

Funções – parâmetros-padrão

Os parâmetros-padrão são úteis quando você deseja fornecer um **comportamento-padrão** para uma função e permitir que o chamador substitua esse comportamento se necessário.

Tome nota

Parâmetros proporcionam flexibilidade, tornando as funções mais versáteis.

```python
def saudacao(nome, mensagem='Olá'):
    print(f'{mensagem}, {nome}!')

# Exemplos de uso:
saudacao('João')            # Saída: Olá, João!
saudacao('Maria', 'Oi')     # Saída: Oi, Maria!

```

```
Olá, João!
Oi, Maria!
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

No exemplo acima, a mensagem é um parâmetro com um valor-padrão ('Olá'). Se você chamar a função sem fornecer um valor para mensagem, o valor-padrão será utilizado.

Exposição

Funções – parâmetros-padrão

Voltando ao nosso exemplo das conquistas no jogo eletrônico:

```
def imprimir_conquistas(conquistas, start=1):
    indice = start
    for conquista in conquistas:
        print(f'Conquista {indice}: {conquista}')
        indice += 1
```

Qual o resultado para:

```
1 conquistas = ['Primeira Vitória', 'Derrotou o Chefe Final', 'Coletou o Tesouro Secreto']
2
3 imprimir_conquistas(conquistas)
4 imprimir_conquistas(conquistas, start=1)
5 imprimir_conquistas(conquistas, start=10)
6 imprimir_conquistas(start=1)
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Funções – parâmetros-padrão

```
1 conquistas = ['Primeira Vitória', 'Derrotou o Chefe Final', 'Coletou o Tesouro Secreto']
2
3 imprimir_conquistas(conquistas)
4 print('\n')
5 imprimir_conquistas(conquistas, start=1)
6 print('\n')
7 imprimir_conquistas(conquistas, start=10)
8 print('\n')
9 imprimir_conquistas(start=1)
```

```
Conquista 1: Primeira Vitória
Conquista 2: Derrotou o Chefe Final
Conquista 3: Coletou o Tesouro Secreto


Conquista 1: Primeira Vitória
Conquista 2: Derrotou o Chefe Final
Conquista 3: Coletou o Tesouro Secreto


Conquista 10: Primeira Vitória
Conquista 11: Derrotou o Chefe Final
Conquista 12: Coletou o Tesouro Secreto



---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[12], line 9
      7 imprimir_conquistas(conquistas, start=10)
      8 print('\n')
----> 9 imprimir_conquistas(start=1)

TypeError: imprimir_conquistas() missing 1 required positional argument: 'conquistas'
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Funções – parâmetros-padrão

Em resumo, sobre o tema, compreende-se que:

- São parâmetros que têm um valor-padrão predefinido;
- Se não se fornecer um valor para esses parâmetros durante a chamada, o valor-padrão será utilizado;
- Permitem que você forneça valores-padrão para argumentos, tornando a função mais flexível.

Em Python, os parâmetros-padrão devem ser declarados após os parâmetros posicionais. Por exemplo:

```
1  # Correto
2  def exemplo(param1, param2=10, param3='abc'):
3      # corpo da função
4
5  # Incorreto - Gera um erro de sintaxe
6  def exemplo(param1=5, param2, param3='abc'):
7      # corpo da função
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Registro

Vamos fazer um **quiz**

O que acontece se você não fornecer um valor para um parâmetro opcional durante a chamada da função?

- O programa gera um erro de sintaxe.
- O valor-padrão na função é automaticamente usado.
- A função solicita ao usuário que forneça um valor.
- O parâmetro recebe um valor nulo por padrão.

Registro

Vamos fazer um **quiz**

O que acontece se você não fornecer um valor para um parâmetro opcional durante a chamada da função?

- ✗ O programa gera um erro de sintaxe.
- ✓ O valor-padrão na função é automaticamente usado.
- ✗ A função solicita ao usuário que forneça um valor.
- ✗ O parâmetro recebe um valor nulo por padrão.

FEEDBACK GERAL DA ATIVIDADE

Parâmetros opcionais em Python são preenchidos automaticamente com valores-padrão caso não sejam especificados. Isso adiciona versatilidade e flexibilidade ao código, pois evita a necessidade de definir valores para parâmetros comuns.

Registro

Vamos fazer um **quiz**

Como você define um parâmetro com um valor-padrão em Python?

- Atribuindo o valor-padrão ao parâmetro na criação da função.
- Colocando o valor-padrão entre colchetes após o nome da função.
- Usando a palavra-chave optional antes do nome do parâmetro.
- Não é possível definir valores-padrão em Python.

Registro

Vamos fazer um **quiz**

Como você define um parâmetro com um valor-padrão em Python?

- ✓ **Atribuindo o valor-padrão ao parâmetro na criação da função.**
- ✗ **Colocando o valor-padrão entre colchetes após o nome da função.**
- ✗ **Usando a palavra-chave optional antes do nome do parâmetro.**
- ✗ **Não é possível definir valores-padrão em Python.**

FEEDBACK GERAL DA ATIVIDADE

Em Python, caso não sejam especificados, os parâmetros opcionais são preenchidos automaticamente com valores-padrão. Isso adiciona versatilidade e flexibilidade ao código, pois evita a necessidade de definir valores para parâmetros comuns.

Registro

Vamos fazer um **quiz**

Qual é a principal finalidade dos parâmetros opcionais em funções?

- Forçar o usuário a fornecer valores específicos durante a chamada da função.
- Tornar a função mais flexível, permitindo valores-padrão para argumentos.
- Reduzir a legibilidade do código.
- Impedir a reutilização da função em diferentes contextos.

Registro

Vamos fazer um **quiz**

Qual é a principal finalidade dos parâmetros opcionais em funções?

- ✗ Forçar o usuário a fornecer valores específicos durante a chamada da função.
- ✓ Tornar a função mais flexível, permitindo valores-padrão para argumentos.
- ✗ Reduzir a legibilidade do código.
- ✗ Impedir a reutilização da função em diferentes contextos.

FEEDBACK GERAL DA ATIVIDADE

A principal finalidade dos parâmetros opcionais em funções em Python é proporcionar flexibilidade ao programador. Se um valor não é dado para um parâmetro opcional, o padrão é usado, simplificando a chamada da função.

© Getty Images

O que nós **aprendemos hoje?**

Hoje desenvolvemos:

1. A compreensão sobre o conceito de parâmetros-padrão, também conhecidos como opcionais, de uma função.

2. A percepção da importância de atribuir diretamente o valor-padrão ao parâmetro na criação da função em Python.

3. O entendimento sobre a capacidade dos parâmetros opcionais de proporcionarem flexibilidade, ou seja, tornarem as funções mais versáteis.

Saiba mais

Já ouviu falar em desempacotamento no Python? Aproveite o ensinamento sobre parâmetros-padrão e conheça sobre o assunto! Siga a seguinte referência:

MATHEUS, Y. Entendendo o desempacotamento no Python. **Alura**, 15 nov. 2018. Disponível em: https://www.alura.com.br/artigos/entendendo-o-desempacotamento-no-python. Acesso em: 7 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.

Educação Profissional Paulista

Técnico em Ciência de Dados

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

S11 – Aula 1 – Quiz

Condições de conclusão

Ver

O que acontece se você não fornecer um valor para um parâmetro opcional durante a chamada da função?

- O valor-padrão na função é automaticamente usado.
- O parâmetro recebe um valor nulo por padrão.
- A função solicita ao usuário que forneça um valor.
- O programa gera um erro de sintaxe.

Como você define um parâmetro com um valor-padrão em Python?

- Não é possível definir valores-padrão em Python.
- Colocando o valor-padrão entre colchetes após o nome da função.
- Usando a palavra-chave optional antes do nome do parâmetro.
- Atribuindo o valor-padrão ao parâmetro na criação da função.

Qual é a principal finalidade dos parâmetros opcionais em funções?

- Reduzir a legibilidade do código.
- Impedir a reutilização da função em diferentes contextos.
- Forçar o usuário a fornecer valores específicos durante a chamada da função.
- Tornar a função mais flexível, permitindo valores-padrão para argumentos.

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

Estrutura de Controle de Fluxo

Enumeração de Iteráveis

Aula 2

[DADOS]ANO1C2B2S11A2

Exposição

Objetivos da Aula

Introduzir o conceito da função embutida enumerate.

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.

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

Enumerando conquistas em um jogo

```python
def imprimir_conquistas(conquistas, start=1):
    indice = start
    for conquista in conquistas:
        print(f'Conquista {indice}: {conquista}')
        indice += 1

conquistas = ['Primeira Vitória', 'Derrotou o Chefe Final', 'Coletou o Tesouro Secreto']
imprimir_conquistas(conquistas)
```

```
Conquista 1: Primeira Vitória
Conquista 2: Derrotou o Chefe Final
Conquista 3: Coletou o Tesouro Secreto
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Enumerar é algo que faremos muito e existe uma forma bem mais simples de fazer isso. Vamos conhecer?

Exposição

Enumerate

O enumerate é uma função embutida do Python usada para iterar simultaneamente sobre os **índices e os elementos** de uma sequência (como uma lista, tupla ou string).

Ele retorna um objeto enumerado, que consiste em pares de índice e valor.

A sintaxe básica do enumerate é a seguinte:

```
enumerate(iteravel, start=0)
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

- **iteravel:** A sequência que você deseja iterar.
- **start (opcional):** O valor inicial do índice. O padrão é 0.

Exposição

Enumerate

Exemplo 1:

```
1  frutas = ['maçã', 'banana', 'laranja']
2
3  for indice, valor in enumerate(frutas):
4      print(f'Índice: {indice}, Valor: {valor}')
5
```

```
Índice: 0, Valor: maçã
Índice: 1, Valor: banana
Índice: 2, Valor: laranja
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Tome nota

Após a palavra **for**, existem duas variáveis. Uma representa o índice do iterável, e a outra o valor do iterável.

Exposição

Enumerate

Exemplo 2:

```python
frase = "Python é incrível!"

for indice, letra in enumerate(frase, start=1):
    print(f'Índice: {indice}, Letra: {letra}')
```

```
Índice: 1, Letra: P
Índice: 2, Letra: y
Índice: 3, Letra: t
Índice: 4, Letra: h
Índice: 5, Letra: o
Índice: 6, Letra: n
Índice: 7, Letra:
Índice: 8, Letra: é
Índice: 9, Letra:
Índice: 10, Letra: i
Índice: 11, Letra: n
Índice: 12, Letra: c
Índice: 13, Letra: r
Índice: 14, Letra: í
Índice: 15, Letra: v
Índice: 16, Letra: e
Índice: 17, Letra: l
Índice: 18, Letra: !
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Enumerate

```
[17]:
1  numero = 42
2
3  for indice, valor in enumerate(numero, start=1):
4      print(f'Índice: {indice}, Valor: {valor}')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[17], line 3
      1 numero = 42
----> 3 for indice, valor in enumerate(numero, start=1):
      4     print(f'Índice: {indice}, Valor: {valor}')

TypeError: 'int' object is not iterable
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Tome nota

Neste exemplo, a variável número **não é uma sequência iterável** e, por isso, não foi possível usar a função enumerate.

Exposição

Enumerate

Observe o código abaixo. A sequência iterável é uma tupla e também funciona com o enumerate.

```
1  pontos_cardeais = ('Norte', 'Sul', 'Leste', 'Oeste')
2
3  for indice, ponto_cardeal in enumerate(pontos_cardeais, start=1):
4      print(f'Índice: {indice}, Ponto Cardeal: {ponto_cardeal}')
```

```
Índice: 1, Ponto Cardeal: Norte
Índice: 2, Ponto Cardeal: Sul
Índice: 3, Ponto Cardeal: Leste
Índice: 4, Ponto Cardeal: Oeste
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Mais adiante, o conceito de **tupla** será abordado.

Exposição

Enumerate

O objeto iterável é um dicionário e também funciona com o enumerate, como se nota no código abaixo:

```
1  pessoas = {'João': 25, 'Maria': 30, 'Pedro': 28}
2
3  for indice, (nome, idade) in enumerate(pessoas.items(), start=1):
4      print(f'Índice: {indice}, Nome: {nome}, Idade: {idade} anos')
```

```
Índice: 1, Nome: João, Idade: 25 anos
Índice: 2, Nome: Maria, Idade: 30 anos
Índice: 3, Nome: Pedro, Idade: 28 anos
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Mais adiante o conceito de **dicionário** também será abordado.

Exposição

Vamos treinar

1. Dada a lista de cores, utilize a função enumerate para imprimir cada cor junto com o seu índice.

cores = ['vermelho', 'verde', 'azul', 'amarelo', 'roxo']

```
1  cores = ['vermelho', 'verde', 'azul', 'amarelo', 'roxo']
2
3  for indice, cor in enumerate(cores, start=1):
4      print(f'Índice: {indice}, Cor: {cor}')
```

```
Índice: 1, Cor: vermelho
Índice: 2, Cor: verde
Índice: 3, Cor: azul
Índice: 4, Cor: amarelo
Índice: 5, Cor: roxo
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Vamos treinar

2. Dada a lista de cores, utilize a função enumerate para imprimir cada cor junto com o índice começando em 6.

cores = ['vermelho', 'verde', 'azul', 'amarelo', 'roxo']

```
1  cores = ['vermelho', 'verde', 'azul', 'amarelo', 'roxo']
2
3  for indice, cor in enumerate(cores, start=6):
4      print(f'Índice: {indice}, Cor: {cor}')
```

```
Índice: 6, Cor: vermelho
Índice: 7, Cor: verde
Índice: 8, Cor: azul
Índice: 9, Cor: amarelo
Índice: 10, Cor: roxo
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Vamos treinar

3. Dada a string abaixo, use a função enumerate para imprimir cada caractere junto com o seu índice.

frase = "Adoro estudar!"

```
1  frase = "Adoro estudar!"
2
3  for indice, caractere in enumerate(frase):
4      print(f'Índice: {indice}, Caractere: {caractere}')
5
```

```
Índice: 0, Caractere: A
Índice: 1, Caractere: d
Índice: 2, Caractere: o
Índice: 3, Caractere: r
Índice: 4, Caractere: o
Índice: 5, Caractere:
Índice: 6, Caractere: e
Índice: 7, Caractere: s
Índice: 8, Caractere: t
Índice: 9, Caractere: u
Índice: 10, Caractere: d
Índice: 11, Caractere: a
Índice: 12, Caractere: r
Índice: 13, Caractere: !
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

© Getty Images

O que nós **aprendemos hoje?**

Hoje desenvolvemos:

1. O entendimento da definição de enumerate de iteráveis e seu impacto em iterar simultaneamente sobre os índices e elementos de uma sequência.

2. O reconhecimento da ação de enumerate por meio de exemplos práticos.

3. A concepção de sintaxe básica do enumerate.

Saiba mais

Você conhece a função built-in como o enumerate?

Dê o primeiro passo para conhecer melhor essa função no curso Python para Data Science. Clique no link abaixo:

ALURA. **Python para Data Science**: primeiros passos. Disponível em: https://www.alura.com.br/curso-online-python-data-science-primeiros-passos . Acesso em: 7 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.

Educação Profissional Paulista

Técnico em Ciência de Dados

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Educação Profissional Paulista

Técnico em Ciência de Dados

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Estrutura de Controle de Fluxo

Enumeração de Iteráveis

Aula 3

[DADOS]ANO1C2B2S11A3

Exposição

Objetivos da Aula

Aplicar de forma prática os conceitos de funções 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.

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.

Vamos fazer uma **atividade**

Atividade: O dobro

15 min

Em grupo

1. Crie uma lista de números.

2. Utilize a função enumerate para criar uma nova lista, na qual cada elemento é o dobro do valor original.

números = [1, 3, 5, 7, 9]

3. Envie o código de programação para o AVA (Ambiente Virtual de Aprendizagem).

Exposição

Exercícios de fixação

1. **Você está desenvolvendo um programa de inventário para uma loja.**

   Crie uma função chamada imprimir_produtos que aceite uma lista de produtos e imprima cada produto com seu índice.

   produtos = ['Computador', 'Mouse', 'Teclado']

2. **Os alunos de uma escola têm notas em diferentes disciplinas.**

   Crie uma função chamada calcular_media que aceite uma lista de notas e retorne a média.

   notas_alunos = [8, 7, 5, 9, 6]

Exposição

Exercícios de fixação

3. **Os funcionários de uma empresa têm salários diferentes.**

   Crie uma função chamada aumentar_salarios que aceite uma lista de salários e retorne uma nova lista na qual cada salário foi aumentado em 10%.

   salarios = [3000, 5000, 7000]

4. **Os alunos de uma turma têm alturas diferentes.**

   Utilize enumerate para imprimir a altura de cada aluno junto com seu índice.

   alturas_alunos = [160, 175, 150, 180]

© Getty Images

O que nós **aprendemos hoje?**

Hoje desenvolvemos:

1 Aplicações práticas da função embutida enumerate.

2 A resolução de exercícios práticos, baseados em casos do cotidiano, como forma de fixação do conteúdo sobre a função enumerate.

Saiba mais

Para saber mais sobre built-in functions e funções, acesse o curso focado em Python:

ALURA. Python para Data Science: trabalhando com funções, estruturas de dados e exceções. Disponível em: https://www.alura.com.br/curso-online-python-data-science-funcoes-estruturas-dados-exceções. Acesso em: 7 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.

Educação Profissional Paulista

Técnico em Ciência de Dados

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

S11 – Aula 3 – Registro

Atividade: O dobro

1. Crie uma lista de números.

1. Utilize a função enumerate para criar uma nova lista, na qual cada elemento é o dobro do valor original. **números = [1, 3, 5, 7, 9]**

1. Envie o código de programação para o AVA (Ambiente Virtual de Aprendizagem).

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

Estrutura de Controle de Fluxo

Enumeração de Iteráveis

Aula 4

[DADOS]ANO1C2B2S11A4

Exposição

Objetivos da Aula

Aplicar conceitos de funções e a função embutida enumerate.

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.

Recursos Didáticos

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

Duração da Aula

50 minutos.

Exposição

Exercícios de fixação

1. **Você está criando um jogo e tem uma lista de personagens.**

   Crie uma função chamada verificar_vida que aceite uma lista de pontos de vida e imprima se cada personagem está vivo ou morto (vida > 0).

   pontos_vida_personagens = [100, 0, 50, 75]

2. **Você tem uma lista de temperaturas em graus Celsius.**

   Crie uma função chamada converter_para_fahrenheit que aceite a lista de temperaturas e retorne uma nova lista com as temperaturas convertidas para Fahrenheit.

   temperaturas_celsius = [20, 25, 30]

   Dica:

   C = (F – 32) X 5/9

Exposição

Exercícios de fixação

3. **Os alunos de uma turma têm notas em diferentes disciplinas.**

   Crie uma função chamada verificar_aprovacao que aceite uma lista de notas e imprima se cada aluno foi aprovado ou reprovado (nota >= 6).

   notas_alunos = [7, 5, 8, 4, 6]

4. **Você está desenvolvendo um sistema de reservas para um restaurante.**

   Crie uma função chamada mostrar_mesas_disponiveis que aceite uma lista de mesas reservadas e imprima as mesas disponíveis (índices não reservados).

   mesas_reservadas = [2, 4, 6]

Exposição

Exercícios de fixação

5. **Você está criando um sistema de gerenciamento de tarefas.**

   Crie uma função chamada imprimir_tarefas que aceite uma lista de tarefas e imprima cada tarefa com seu índice.

   tarefas = ['Estudar Python', 'Fazer compras', 'Enviar e-mails']

6. **Você tem uma lista de produtos e seus preços.**

   Crie uma função chamada calcular_total que aceite a lista de preços e retorne o preço total dos produtos.

   precos = [10.5, 5.2, 8.0, 12.99]

Exposição

Exercícios de fixação

7. **Você tem uma lista de palavras.**

   Crie uma função chamada contar_letras que aceite a lista de palavras e imprima o número de letras em cada palavra junto com seu índice.

   palavras = ['python', 'exemplo', 'programacao’]

8. **Você está desenvolvendo um programa meteorológico e precisa converter as temperaturas de uma lista de graus Celsius para Kelvin.**

   Crie uma função chamada converter_para_kelvin que aceite a lista de temperaturas em graus Celsius e imprima cada temperatura convertida para Kelvin, junto com seu índice.

   Dica: A fórmula de conversão de Celsius para Kelvin é K = C + 273.15, em que K é a temperatura em Kelvin e C é a temperatura em graus Celsius.

   temperaturas_celsius = [25, 30, 15, 10]

Vamos fazer uma **atividade**

15 minutos

Conversão

Você está aprimorando seu programa meteorológico e deseja **criar uma função** chamada **converter_temperaturas**

Essa função aceita uma lista de temperaturas em Kelvin e outra lista de temperaturas em Fahrenheit.

A função deve imprimir cada temperatura convertida para Celsius, junto com seu índice.

Dica

Confira na próxima tela dicas úteis para a realização da atividade proposta!

Vamos fazer uma **atividade**

Conversão

Dicas:

- Para converter de Kelvin para Celsius, subtraia 273.15 da temperatura em Kelvin.
- Para converter de Fahrenheit para Celsius, use a fórmula:

C = (F – 32) X 5/9, onde

C é a temperatura em Celsius e

F é a temperatura em Fahrenheit.

```
12 # Testando a função
13 temperaturas_kelvin = [300, 310, 290, 280]
14 temperaturas_fahrenheit = [68, 86, 59, 50]
15 converter_temperaturas(temperaturas_kelvin, temperaturas_fahrenheit)
16
```

```
Índice: 1, Temperatura - Celsius (Kelvin): 26.85 °C
Índice: 1, Temperatura - Celsius (Fahrenheit): 20.00 °C
Índice: 2, Temperatura - Celsius (Kelvin): 36.85 °C
Índice: 2, Temperatura - Celsius (Fahrenheit): 30.00 °C
Índice: 3, Temperatura - Celsius (Kelvin): 16.85 °C
Índice: 3, Temperatura - Celsius (Fahrenheit): 15.00 °C
Índice: 4, Temperatura - Celsius (Kelvin): 6.85 °C
Índice: 4, Temperatura - Celsius (Fahrenheit): 10.00 °C
```

© Getty Images

O que nós **aprendemos hoje?**

Hoje desenvolvemos:

1 O entendimento de como são as aplicações de funções em Python.

2 A resolução de exercícios práticos, baseados em casos do cotidiano, como forma de fixação do conteúdo sobre a função enumerate.

Saiba mais

Para saber mais sobre built-in functions e funções, acesse o curso focado em Python:

ALURA. **Python para Data Science**: trabalhando com funções, estruturas de dados e exceções. Disponível em: https://www.alura.com.br/curso-online-python-data-science-funcoes-estruturas-dados-excecoes. Acesso em: 7 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.

Educação Profissional Paulista

Técnico em Ciência de Dados

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO