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.

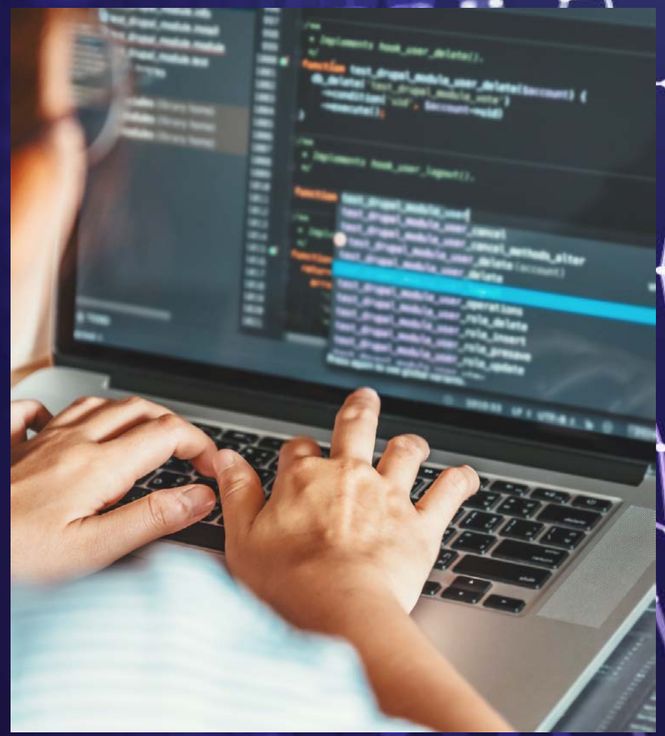
Recursos Didáticos

- 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