Educação Profissional Paulista

Técnico em Ciência de Dados

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Programação aplicada à Ciência de Dados

Estrutura de controle de fluxo

Aula 3

Código da aula: [DADOS]ANO1C2B2S10A3

Secretaria da Educação
SÃO PAULO
GOVERNO DO ESTADO

Exposição

Objetivo da aula

- Aplicar 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 à internet;
- Software Anaconda/Jupyter Notebook instalado ou similar.

Duração da aula

50 minutos.

Exposição

Função - Resumo

Uma função é um bloco de código reutilizável que realiza uma tarefa específica.

```
def nome_da_funcao(parametros):
    # Corpo da função
    # Realiza alguma tarefa
    return resultado
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook

- Promove a **reutilização** de código.
- Contribui para a **organização** e a **modularidade**.
- Facilita a **compreensão** do código.

Exposição

Criando funções

Outros exemplos de como criar uma função:

```
1  def minha_primeira_funcao():
2      print('Olá Mundo')
```

```
1  # chamar a função pelo nome para executá-la
2  minha_primeira_funcao()
```

```
Olá Mundo
```

```
1  def bom_dia():
2      print('Bom dia!')
```

```
1  bom_dia()
```

```
Bom dia!
```

```
1  def exemplo_funcao_simples():
2      print('criando uma função com o "def"')
```

```
1  exemplo_funcao_simples()
```

```
criando uma função com o "def"
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Parâmetros

Será que conseguimos personalizar?

```python
def funcao_ola(nome):
    print(f'Olá, meu nome é {nome}.')
```

```python
funcao_ola("Alice")
```

```
Olá, meu nome é Alice.
```

```python
funcao_ola("Renata")
```

```
Olá, meu nome é Renata.
```

```python
funcao_ola("Tiago")
```

```
Olá, meu nome é Tiago.
```

```python
def horario(hora):
    if hora < 12:
        print('Bom dia')
    elif hora < 18:
        print('Boa tarde')
    else:
        print('Boa noite')
```

```python
horario(10)
```

```
Bom dia
```

```python
horario(15)
```

```
Boa tarde
```

```python
horario(19)
```

```
Boa noite
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Ordem dos parâmetros

A ordem dos parâmetros importa?

```python
def exemplo(nome, idade):
    print(f"Nome: {nome}, Idade: {idade}")
```

```python
exemplo('Alice', 25)
```

```
Nome: Alice, Idade: 25
```

```python
exemplo(25, 'Alice')
```

```
Nome: 25, Idade: Alice
```

```python
exemplo(nome="Alice", idade=25)
```

```
Nome: Alice, Idade: 25
```

```python
exemplo(idade=25, nome="Alice")
```

```
Nome: Alice, Idade: 25
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Parâmetros e *return*

Qual é a diferença entre as funções?

```
1 def soma(a, b):
2     print(a + b)
```

```
1 soma(2, 9)
```

```
11
```

```
1 soma(7, 8)
```

```
15
```

```
1 soma(10, 15)
```

```
25
```

```
1 resultado = soma(10, 15)
2 print('O resultado é: ', resultado)
```

```
25
O resultado é:  None
```

```
1 def soma(a, b):
2     return a + b
```

```
1 soma(2, 9)
```

```
11
```

```
1 soma(7, 8)
```

```
15
```

```
1 soma(10, 15)
```

```
25
```

```
1 resultado = soma(10, 15)
2 print('O resultado é: ', resultado)
```

```
O resultado é:  25
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Parâmetros e *return*

```python
def soma(a, b):
    resultado = a + b
    print("print do resultado dentro da função: ", resultado) # apenas mostra na tela o valor de uma variável
    return resultado  # retorna um resultado/saída da função


resultado = soma(2, 3)

media = resultado / 2
print(f'A média é {media}')
```

```
print do resultado dentro da função:  5
A média é 2.5
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Chamar uma função como argumento

```
1  def soma_dois_valores(a, b):
2      resultado = a + b
3      return resultado
4
5  res = soma_dois_valores(2, 3)
6
7  def media_dois_valores(resultado):
8      media = resultado / 2
9      return media
10
11 print(media_dois_valores(res))
```

```
2.5
```

```
1  media_dois_valores(soma_dois_valores(2,3))
```

```
2.5
```

Elaborado especialmente para o curso com a ferramenta Jupyter Notebook.

Exposição

Exercícios

1. Faça uma função que recebe um número e imprima seu **dobro**.

2. Faça uma função que recebe o valor do raio de um **círculo** e retorna o valor do comprimento de sua circunferência **C = 2*pi*r**.

3. Crie uma função chamada **concatenar_palavras**, que receba duas *strings* como parâmetros e retorna a concatenação dessas duas *strings*, separadas por um espaço.

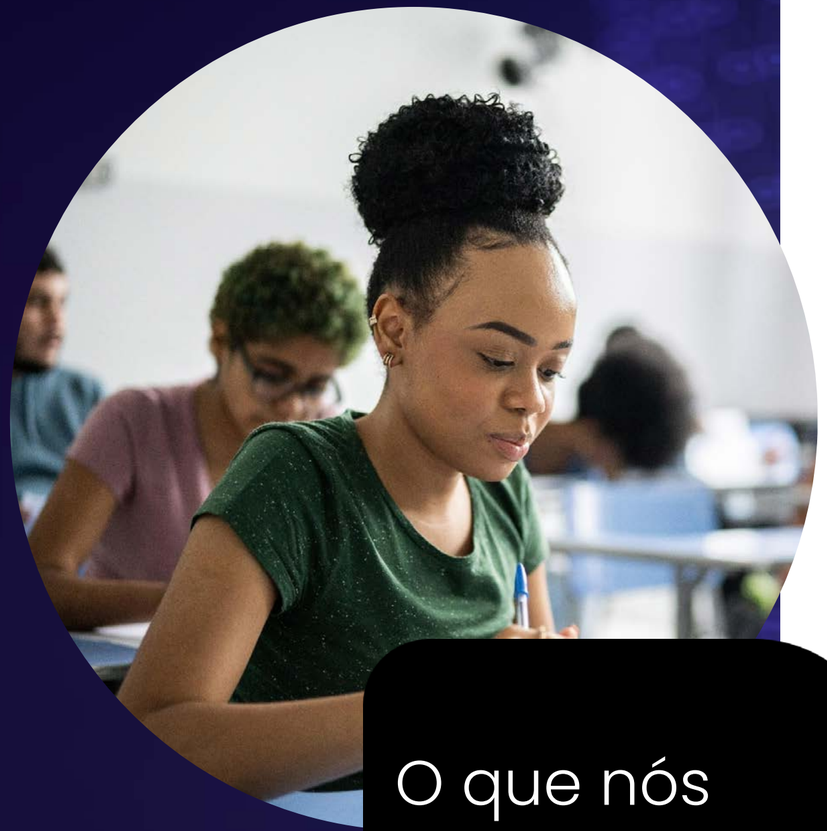
Exposição

Exercícios

4. Crie uma função chamada **verificar_par** que receba um número como parâmetro e retorne **True**, se o número for par, e **False**, em caso contrário.

5. Crie uma função chamada **calcular_media** que receba três números como parâmetros e retorne a **média aritmética** desses números.

6. Faça uma função para cada **operação matemática básica** (soma, subtração, multiplicação e divisão). As funções devem receber dois números e retornar o resultado da operação.



© Getty Images

O que nós **aprendemos hoje?**

Hoje desenvolvemos:

1 Aplicações práticas de funções em Python.

2 Resoluções de exercícios sobre funções.

Saiba mais

Entenda mais sobre *built-in functions* e funções:

ALURA. *Python para Data Science*: trabalhando com funções, estruturas de dados e exceções. Disponível em: https://cursos.alura.com.br/course/python-data-science-funcoes-estruturas-dados-excecoes/task/125896. Acesso em: 28 fev. 2024.

Referências da aula

Identidade visual: Imagens © Getty Images.

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