

PROGRAMAÇÃO EM PYTHON



PROGRAMAÇÃO EM PYTHON



Por meio de uma parceria com o SENAI-SP, a Prefeitura de Santana de Parnaíba está disponibilizando um curso focado em Python, com o objetivo de preparar os alunos para o desenvolvimento de programas e aplicações.



PROGRAMAÇÃO EM PYTHON



Plano de Aula

- Conteúdo:**
- Constantes;
 - Listas;
 - Atividades;

Início:

As informações deste conteúdo visam compreender o conteúdo do curso.



PROGRAMAÇÃO EM PYTHON



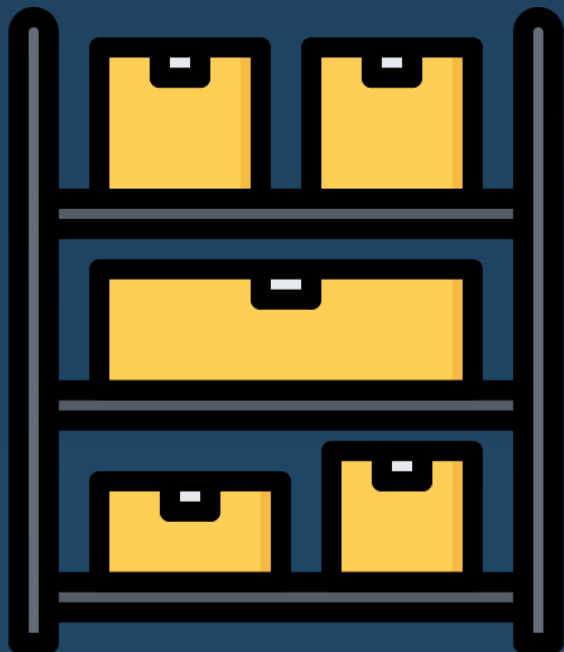
Objetivo

Capacitar profissionais para desenvolver aplicações em linguagem Python, por meio de técnicas de programação, seguindo boas práticas, procedimentos e normas.

PROGRAMAÇÃO EM PYTHON



Constantes



Definição

Constantes são valores que, uma vez atribuídos, **não devem ser modificados durante a execução de um programa.**

Diferença em relação às variáveis

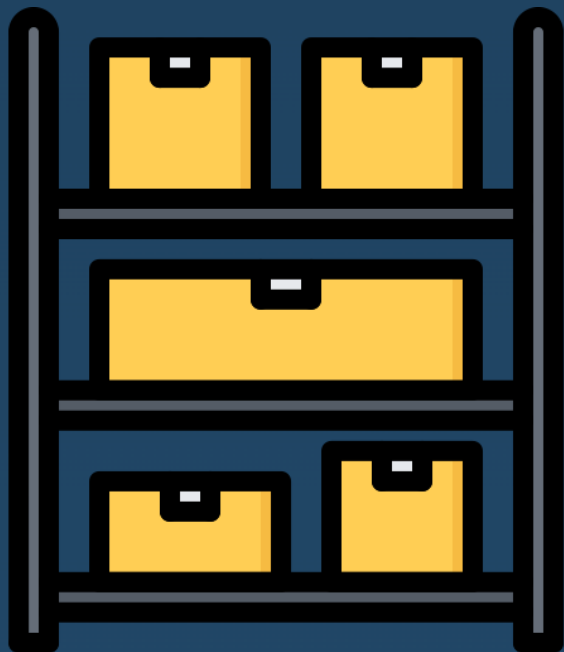
Enquanto **variáveis** podem ter seus valores alterados, **as constantes permanecem fixas no programa.**

No entanto, Python possui uma particularidade: a linguagem não oferece uma palavra-chave ou mecanismo nativo para garantir a imutabilidade de valores.

PROGRAMAÇÃO EM PYTHON



Constantes



O uso de constantes melhora a legibilidade, manutenção e segurança do código.

- **Legibilidade:** Nomes descritivos evitam "números mágicos" e tornam o código mais claro.
- **Prevenção de erros:** Indicar que um valor não deve mudar reduz a chance de modificações acidentais.
- **Reutilização:** Constantes podem ser usadas em várias partes do código sem necessidade de lembrar valores exatos.
- **Documentação implícita:** Nomes significativos ajudam a entender o propósito do valor sem comentários adicionais.

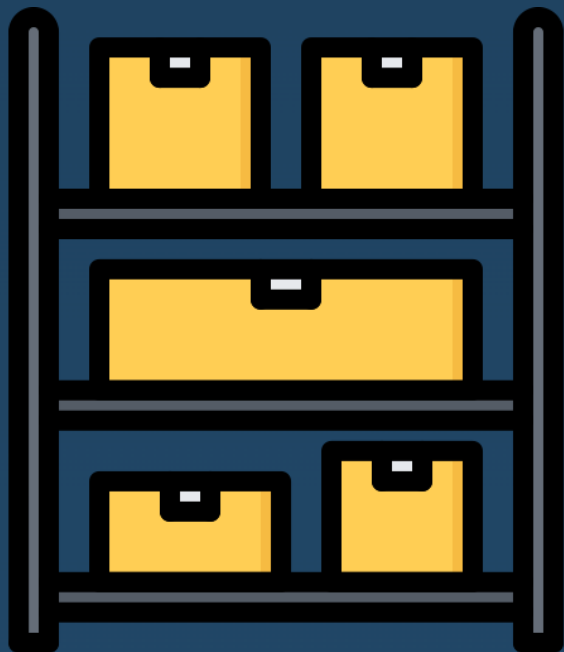


Mesmo que seja possível reatribuir um valor a uma "constante", fazer isso vai contra as boas práticas e pode gerar confusão no código.

PROGRAMAÇÃO EM PYTHON



Constantes



Convenções para Definir Constantes

A prática mais comum é utilizar nomes em letras maiúsculas e separar as palavras por sublinhados.

```
PI = 3.14159  
TAXA_DE_CAMBIO = 5.50  
NOME_DO_APLICATIVO = "Meu Programa"
```

Atribuição Inicial



As constantes devem ser definidas no início do programa ou do módulo, garantindo que seu valor seja facilmente identificado e alterado, se necessário, em um único local.

PROGRAMAÇÃO EM PYTHON



Constantes

Vamos
realizar
juntos!

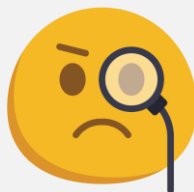


Convenções para Definir Constantes

A prática mais comum é utilizar nomes em letras maiúsculas e separar as

```
# Definição da constante para a taxa de câmbio
DOLAR = 5.50

valor_em_dolar = float(input("Digite o valor em dólares: "))
valor_em_real = valor_em_dolar * DOLAR
print(f"O valor convertido em reais é: R$ {valor_em_real:.2f}")
```



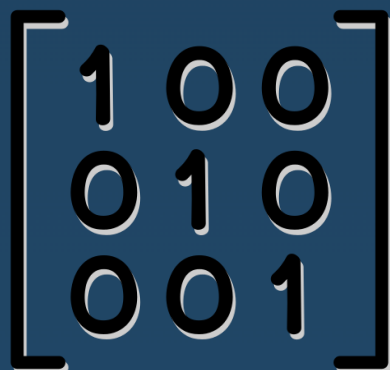
```
{valor_em_real:.2f}
```

`:.2f` → Significa que o número deve ser formatado como float (f) e exibido com duas casas decimais (.2).

PROGRAMAÇÃO EM PYTHON



Listas



Definição

Listas são **uma das estruturas de dados mais fundamentais e versáteis da linguagem.**

Elas permitem **armazenar** coleções ordenadas de itens e **oferecem uma variedade de operações para manipulação e acesso aos dados.**

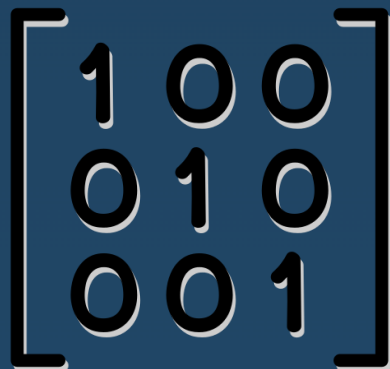
```
numeros = [1, 2, 3, 4, 5]
nomes = ["Joaquim", "Maria", "Ana"]

print(numeros)
print(nomes)
```

PROGRAMAÇÃO EM PYTHON



Listas



Características das Listas

- **Acessibilidade:** Os itens da lista são acessados através de um número chamado de índice, começando do 0.

```
nomes = ["Joaquim", "Maria", "Ana"]  
  
print(nomes[0])  
print(nomes[1])  
print(nomes[2])
```

- **Mutáveis:** É possível alterar, adicionar ou remover elementos após a criação da lista.
- **Heterogêneas:** Podem conter elementos de tipos diferentes, como números, strings, booleanos, até outras listas.

```
lista_mista = [1, "dois", 3.0, True]
```

PROGRAMAÇÃO EM PYTHON



Listas

Vamos
realizar
juntos!



```
nomes = ["Joaquim", "Maria", "Ana"]
```

```
print(nomes[0]) # Joaquim
```

```
print(nomes[1])
```

```
print(nomes[2])
```

```
nomes[0] = "João" # estava Joaquim
```

```
print(nomes[0]) # alterado para João
```

```
print(nomes[1])
```

```
print(nomes[2])
```

PROGRAMAÇÃO EM PYTHON



Listas

Vamos
realizar
juntos!



```
nomes = ["Joaquim", "Maria", "Ana"]
```

```
nomes.append("João")
```

```
nomes.append("Joana")
```

```
print(nomes[0])
```

```
print(nomes[1])
```

```
print(nomes[2])
```

```
print(nomes[3])
```

```
print(nomes[4])
```

PROGRAMAÇÃO EM PYTHON



Listas



Operações Comuns com Listas

- **Adicionando elementos:**
 - **append(item):** Adiciona um item ao final da lista.
 - **insert(índice, item):** Insere um item em uma posição específica.
- **Modificando elementos:**
 - ```
nomes = ["Joaquim", "Maria", "Ana"]
nomes[0] = "João"
```
- **Removendo elementos:**
  - **del lista[índice]:** Remove o elemento no índice especificado.
  - **remove(valor):** Remove a primeira ocorrência de um valor específico.
  - **pop(índice):** Remove e retorna o elemento no índice especificado; se omitido, remove o último.
  - **clear():** Remove todos os elementos, esvaziando a lista.

# PROGRAMAÇÃO EM PYTHON



## Listas

Vamos  
realizar  
juntos!



## Parte 01

```
1 # Lista inicial
2 nomes = ["Joaquim", "Maria", "Ana"]
3 print("Lista inicial:", nomes)
4
5 # Adicionando elementos
6 nomes.append("Carlos") # Adiciona ao final
7 print("Após append:", nomes)
8
9 nomes.insert(1, "Fernanda") # Insere "Fernanda" na posição 1
10 print("Após insert:", nomes)
11
12 # Modificando elementos
13 nomes[2] = "Paulo" # Modifica o elemento no índice 2
14 print("Após modificação:", nomes)
15
```

# PROGRAMAÇÃO EM PYTHON



## Listas

Vamos  
realizar  
juntos!



## Parte 02

```
16 # Removendo elementos
17 del nomes[3] # Remove o elemento no índice 3
18 print("Após del:", nomes)
19
20 nomes.remove("Maria") # Remove a primeira ocorrência de "Maria"
21 print("Após remove:", nomes)
22
23 removido = nomes.pop(2) # Remove e retorna o elemento no índice 2
24 print(f"Após pop (removido '{removido}'):", nomes)
25
26 nomes.clear() # Esvazia a lista
27 print("Após clear:", nomes)
28
```

# PROGRAMAÇÃO EM PYTHON



## Apresentação da situação de aprendizagem



### Programa Cadastro - FABPROG

Imagine que você precisa criar uma agenda de contatos para organizar e facilitar a comunicação com pessoas do seu convívio. Seu desafio é desenvolver um programa interativo que permita ao usuário realizar as seguintes operações: cadastro, consulta e exclusão de dados.

#### O programa deve apresentar o seguinte menu:

- [1] **Cadastrar pessoa:** Permite inserir o nome de uma nova pessoa na agenda.
- [2] **Listar pessoas:** Exibe todos os nomes cadastrados até o momento.
- [3] **Excluir pessoa:** Solicita o nome da pessoa a ser removida e a exclui da lista, se existir.
- [0] **Sair:** Encerra o programa.

#### Objetivo:

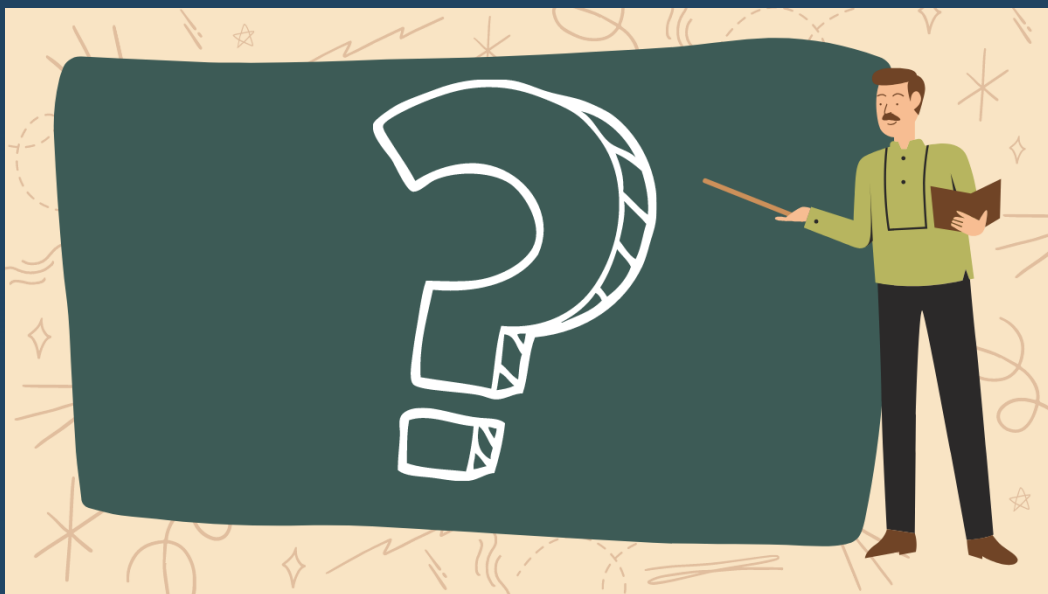
Desenvolver a lógica de um sistema de cadastro simples, aplicando os conceitos de laços de repetição e manipulação de listas.



# PROGRAMAÇÃO EM PYTHON



## Questões Mediadoras



## Sobre Constantes

- O que diferencia uma constante de uma variável?
- Por que adotamos a convenção de escrever constantes em letras maiúsculas?
- Como a definição de constantes pode contribuir para a legibilidade e manutenção do código?

## Sobre Listas

- Quais são as vantagens e desvantagens de utilizar listas em determinados contextos de programação?
- Quais métodos comuns você conhece para manipulação de listas e quando utilizá-los?
- Quais estratégias você pode utilizar para iterar sobre os elementos de uma lista e quais são as implicações de cada uma?

## REFERÊNCIAS

ALVES, William Pereira. Lógica de programação de computadores. São Paulo: Érica, 2012.

ASCENCIO, Ana F. G.; CAMPOS, Edilene A. V. Fundamentos da programação de computadores. 3. ed. São Paulo: Pearson, 2012.

FARRELL, Joyce. Lógica e design de programação. São Paulo: Cengage Learning, 2010.

MANZANO, J. A. N. G.; OLIVEIRA, J. F. Algoritmos: Lógica para desenvolvimento de programação de computadores. 27. ed. rev. São Paulo: Érica, 2014.

MEDINA, Marco; FERTIG, Cristina. Algoritmos e programação. Teoria e prática. São Paulo: Novatec, 2005.

PEREIRA, Silvio do Lago. Algoritmos e lógica de programação em C: Uma abordagem didática. São Paulo: Érica, 2010.

## Importante:

Os conteúdos disponibilizados são específicos para este curso/turma, a divulgação ou reprodução do material para outras pessoas/organização não é autorizada.



[daniel.petinice@sp.senai.br](mailto:daniel.petinice@sp.senai.br)



Daniel Tadeu Petinice



Instrutor de Formação Profissional III

