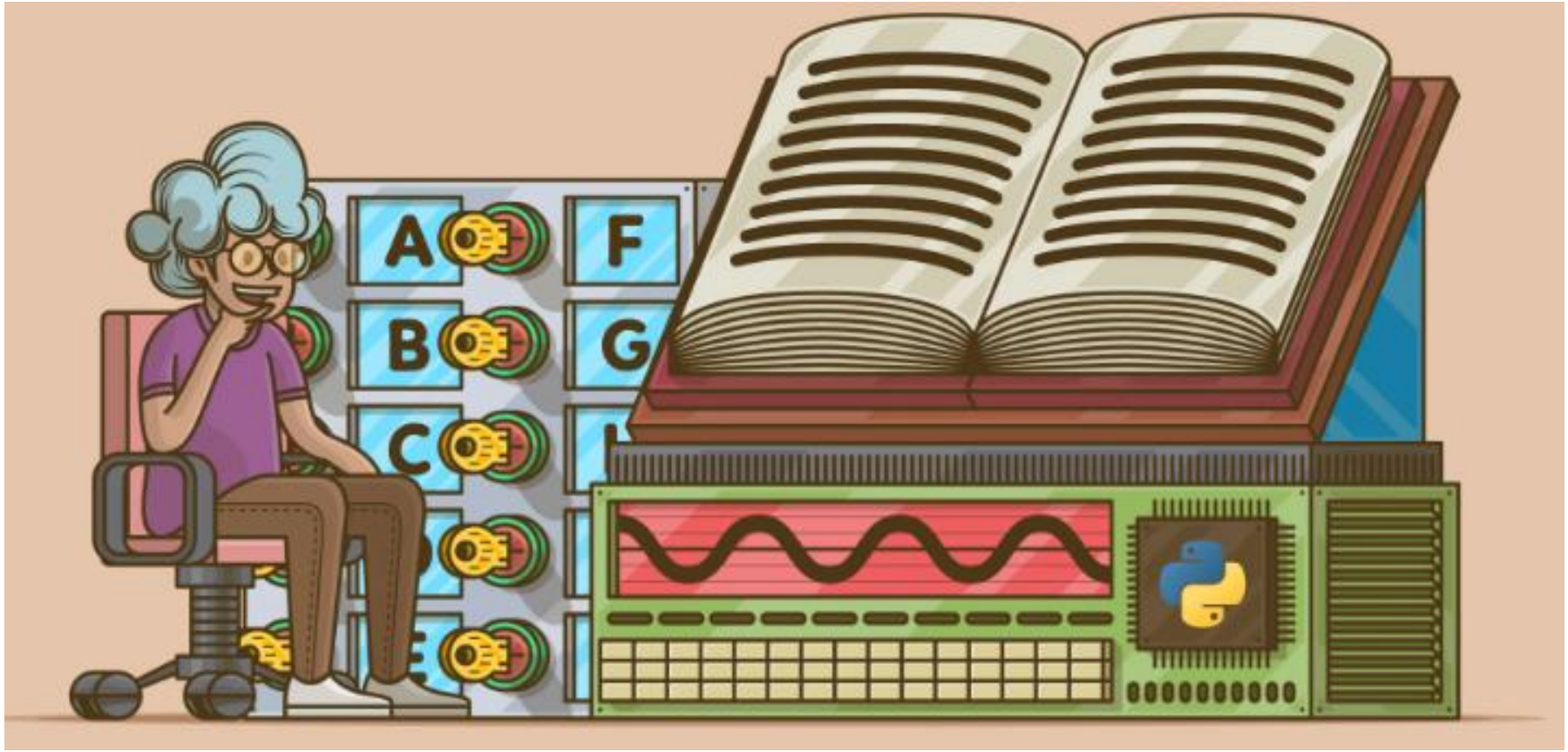
LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO



LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

Dicionários.

Em Python, um dicionário é uma estrutura de dados que armazena pares chave-valor. Cada elemento do dicionário consiste em uma chave e o valor associado a essa chave. As chaves em um dicionário são únicas, ou seja, não podem existir chaves duplicadas.

Para criar um Dicionário em Python utilizamos a {}.

LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

Dicionários.

A sintaxe de um dicionário em Python é bastante simples e direta. Você define um dicionário usando chaves **{}** e especifica os pares chave-valor separados por vírgulas. A forma básica é a seguinte:

```
dicionario = {
        "key":"value,
        "cor":"laranja",
        "ano" : 2024,
}
```

A **CHAVE** É UM IDENTIFICADOR ASSOCIADO A UM VALOR ESPECÍFICO.

O **VALOR** REFERE-SE À INFORMAÇÃO ASSOCIADA A UMA DETERMINADA CHAVE.

DICIONÁRIOS PODEM ARMAZENAM UMA VARIEDADE DE TIPOS DE DADOS, COMO STRINGS, NÚMEROS, LISTAS, DICIONÁRIOS, ENTRE OUTROS.

LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

Dicionários.

Para acessar um valor em um dicionário em Python, você utiliza a sintaxe de colchetes **[]**. A chave correspondente ao valor desejado é colocada entre os colchetes.

```
comidasFavoritas = {
        “comida”: “Lanche”,
        “suco”: “Laranja”,
        “refrigerante” : “Coca-Cola”
}
```

```
#Acessando todo o Dicionário
print(comidasFavoritas)
#Resultado: {'comida': 'Lanche', 'suco': 'Laranja', 'refrigerante': 'Coca-Cola'}
```

LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

Dicionários.

Para acessar os itens de um dicionário você utiliza o método **items**.

```
comidasFavoritas = {
        “comida”: “Lanche”,
        “suco”: “Laranja”,
        “refrigerante” : “Coca-Cola”
}
```

```
21 #Acessando os itens do Dicionario
22 print(comidasFavoritas.items())
23 #Resultado: ([('comida', 'Lanche'), ('suco', 'Laranja'), ('refrigerante', 'Coca-Cola')])
24
```

LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

Dicionários.

Para acessar as chaves de um dicionário você utiliza o método **keys**.

```
comidasFavoritas = {
        “comida”: “Lanche”,
        “suco”: “Laranja”,
        “refrigerante” : “Coca-Cola”
}
```

```
#Acessando os itens do Dicionario
print(comidasFavoritas.keys())
#Resultado: ['comida', 'suco', 'refrigerante']
```

LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

Dicionários.

Para acessar os valores de um dicionário você utiliza o método **values**.

```
comidasFavoritas = {
        “comida”: “Lanche”,
        “suco”: “Laranja”,
        “refrigerante” : “Coca-Cola”
}
```

```
12 #Acessando os valores do Dicionario
13 print(comidasFavoritas.values())
14 #Resultado: (['Lanche', 'Laranja', 'Coca-Cola'])
```

LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

Dicionários.

Adicionando, alterando e removendo itens.

```
comidasFavoritas = {
        “comida”: “Lanche”,
        “suco”: “Laranja”,
        “refrigerante” : “Coca-Cola”
}
```

```python
#Adicionando itens em um Dicionário
comidasFavoritas['sobremesa'] = 'Torta Holandesa'

#Alterando itens de um Dicionário
comidasFavoritas['suco'] = 'Maracuja'

#Deletando itens em um Dicionário
del comidasFavoritas['sobremesa']
```

LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

Dicionários.

Buscando itens dentro do dicionário.

```
comidasFavoritas = {
        “comida”: “Lanche”,
        “suco”: “Laranja”,
        “refrigerante” : “Coca-Cola”
}
```

```
#Adicionando itens em um Dicionário
comidasFavoritas['sobremesa'] = 'Torta Holandesa'

#Alterando itens de um Dicionário
comidasFavoritas['suco'] = 'Maracuja'

#Deletando itens em um Dicionário
del comidasFavoritas['sobremesa']
```

LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

Tuplas, Listas e Dicionários.

As **listas** são usadas quando a ordem dos elementos é importante e quando é necessário modificar a coleção após a criação.

As **tuplas** são úteis quando se quer uma coleção imutável, por exemplo, para definir coordenadas.

Os **dicionários** são ideais para associar valores a chaves e fazer buscas rápidas por essas chaves.

LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

Tuplas, Listas e Dicionários.

Qual é mais utilizado?

Lista é geralmente a estrutura mais utilizada por sua flexibilidade e versatilidade. A mutabilidade das listas torna-as ideais para a maioria das situações em que a coleção de elementos pode mudar durante o ciclo de vida do programa.

Dicionário também é amplamente usado, especialmente quando a associação de valores a chaves únicas é necessária, como na manipulação de grandes volumes de dados e buscas rápidas.

LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

Tuplas, Listas e Dicionários.

Qual é mais utilizado?

Tupla é utilizada em situações específicas onde a imutabilidade é desejada. Seu uso é menos comum em comparação com listas e dicionários, mas ainda é fundamental em contextos específicos, como chaves de dicionários ou retornos múltiplos de funções.

VAMOS PRATICAR.

EXEMPLOS E ATIVIDADES DE DICIONÁRIOS