**Sobre dicionários - Python**

**Vejam diariamente informações em** [**https://www.youtube.com/@PythonCoding**](https://www.youtube.com/@PythonCoding)

# Nível 1: Criação Básica de Dicionário

# Criando um dicionário simples

pessoa = {

    "nome": "Alice",

    "idade": 30,

    "cidade": "Nova York"

}

print(pessoa)

# Saída: {'nome': 'Alice', 'idade': 30, 'cidade': 'Nova York'}

# Nível 2: Acessando e Modificando Valores

# Acessando valores

print(pessoa["nome"])  # Saída: Alice

# Modificando valores

pessoa["idade"] = 31

print(pessoa["idade"])  # Saída: 31

# Nível 3: Adicionando e Removendo Pares Chave-Valor

# Adicionando um novo par chave-valor

pessoa["email"] = "alice@example.com"

print(pessoa)

# Saída: {'nome': 'Alice', 'idade': 31, 'cidade': 'Nova York', 'email': 'alice@example.com'}

# Removendo um par chave-valor

del pessoa["cidade"]

print(pessoa)

# Saída: {'nome': 'Alice', 'idade': 31, 'email': 'alice@example.com'}

# Nível 4: Métodos de Dicionário

# Obtendo todas as chaves

print(pessoa.keys())  # Saída: dict\_keys(['nome', 'idade', 'email'])

# Obtendo todos os valores

print(pessoa.values())  # Saída: dict\_values(['Alice', 31, 'alice@example.com'])

# Obtendo todos os pares chave-valor

print(pessoa.items())  # Saída: dict\_items([('nome', 'Alice'), ('idade', 31), ('email', 'alice@example.com')])

# Usando o método get()

print(pessoa.get("nome"))  # Saída: Alice

print(pessoa.get("cidade", "Não Encontrado"))  # Saída: Não Encontrado

# Usando o método pop()

email = pessoa.pop("email")

print(email)  # Saída: alice@example.com

print(pessoa)  # Saída: {'nome': 'Alice', 'idade': 31}

# Nível 5: Compreensões de Dicionário

# Compreensão de dicionário

quadrados = {x: x\*x for x in range(6)}

print(quadrados)

# Saída: {0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

**Explicações:**

**Nível 1:**

* Este nível demonstra a criação de um dicionário básico. Dicionários são criados usando chaves {} e os pares chave-valor são separados por dois pontos :.
* pessoa é um dicionário que armazena informações sobre uma pessoa, com chaves como "nome", "idade" e "cidade".

**Nível 2:**

* Dicionários permitem o acesso a valores usando a chave correspondente dentro de colchetes [].
* pessoa["nome"] recupera o valor associado à chave "nome".
* Os valores podem ser modificados atribuindo um novo valor a uma chave. pessoa["idade"] = 31 atualiza a idade.

**Nível 3:**

* Novos pares chave-valor podem ser adicionados a um dicionário usando atribuição. pessoa["email"] = "alice@example.com" adiciona uma nova chave de email.
* A palavra-chave del remove pares chave-valor. del pessoa["cidade"] remove o par chave-valor da cidade.

**Nível 4:**

* **.keys()**: Retorna um objeto de exibição de todas as chaves no dicionário.
* **.values()**: Retorna um objeto de exibição de todos os valores no dicionário.
* **.items()**: Retorna um objeto de exibição de todos os pares chave-valor como tuplas.
* **.get(chave)**: Retorna o valor associado à chave, ou None se a chave não for encontrada.
* **.get(chave, padrão)**: Retorna o valor associado à chave, ou o valor padrão especificado se a chave não for encontrada.
* **.pop(chave)**: Remove e retorna o valor associado à chave. Lança um KeyError se a chave não for encontrada.

**Nível 5:**

* **Compreensão de Dicionário:** Fornece uma forma concisa de criar dicionários com base em estruturas de dados existentes.
* quadrados = {x: x\*x for x in range(6)} cria um dicionário onde as chaves são números de 0 a 5 e os valores são seus quadrados.