# Dicionários em Python - Explicação Completa

Em Python, um dicionário (dict) é uma estrutura de dados que armazena pares chave-valor.   
Ele é útil quando você precisa acessar dados por um identificador único (a chave), em vez   
de um índice numérico como em listas.  
  
📌 Criando um dicionário  
Você pode criar um dicionário usando {} ou a função dict():  
  
# Criando um dicionário com {}  
dados = {"nome": "Ana", "idade": 25, "cidade": "São Paulo"}  
  
# Criando com dict()  
dados2 = dict(nome="Carlos", idade=30, cidade="Rio de Janeiro")  
  
print(dados)  
print(dados2)  
  
Saída:  
{'nome': 'Ana', 'idade': 25, 'cidade': 'São Paulo'}  
{'nome': 'Carlos', 'idade': 30, 'cidade': 'Rio de Janeiro'}  
  
📌 Acessando valores  
Você pode acessar um valor pela chave:  
  
print(dados["nome"]) # Saída: Ana  
  
Se a chave não existir, dict.get() evita erro e permite definir um valor padrão:  
  
print(dados.get("email", "Não informado")) # Saída: Não informado  
  
📌 Adicionando e modificando valores  
Para adicionar ou modificar valores, basta usar a chave:  
  
dados["email"] = "ana@email.com" # Adiciona  
dados["idade"] = 26 # Modifica  
  
print(dados)  
  
Saída:  
{'nome': 'Ana', 'idade': 26, 'cidade': 'São Paulo', 'email': 'ana@email.com'}  
  
📌 Removendo itens  
del dados["cidade"] # Remove um item específico  
dados.pop("email") # Remove e retorna o valor removido  
dados.popitem() # Remove o último item  
  
print(dados)  
  
📌 Percorrendo um dicionário  
Você pode iterar sobre as chaves, valores ou ambos:  
  
# Percorrendo as chaves  
for chave in dados:  
 print(chave, "->", dados[chave])  
  
# Percorrendo os valores  
for valor in dados.values():  
 print(valor)  
  
# Percorrendo chave e valor  
for chave, valor in dados.items():  
 print(f"{chave}: {valor}")  
  
📌 Verificando se uma chave existe  
if "idade" in dados:  
 print("A chave 'idade' existe no dicionário.")  
  
📌 Tamanho do dicionário  
print(len(dados)) # Retorna o número de pares chave-valor  
  
Os dicionários são extremamente poderosos e usados em diversas situações, como   
armazenamento de dados estruturados, JSON e contagem de elementos.