Para declarar um array bidimensional em Python, você pode usar listas aninhadas. Aqui estão alguns exemplos de como criar e manipular um array bidimensional.

**Declaração de um array bidimensional com valores específicos**

python

Copiar código

# Array bidimensional 2x2 com valores específicos

meu\_array = [

[1, 2],

[3, 4]

]

# Acessando elementos do array bidimensional

print(meu\_array[0][0]) # Saída: 1

print(meu\_array[0][1]) # Saída: 2

print(meu\_array[1][0]) # Saída: 3

print(meu\_array[1][1]) # Saída: 4

**Declaração de um array bidimensional vazio ou com valores padrão**

**Inicialização com None**

python

Copiar código

# Array bidimensional 2x2 inicializado com None

meu\_array = [

[None, None],

[None, None]

]

# Atribuindo valores

meu\_array[0][0] = 'a'

meu\_array[0][1] = 'b'

meu\_array[1][0] = 'c'

meu\_array[1][1] = 'd'

# Imprimindo o array

for linha in meu\_array:

print(linha)

**Inicialização com zero**

python

Copiar código

# Array bidimensional 2x2 inicializado com zero

meu\_array = [

[0, 0],

[0, 0]

]

# Atribuindo valores

meu\_array[0][0] = 5

meu\_array[0][1] = 10

meu\_array[1][0] = 15

meu\_array[1][1] = 20

# Imprimindo o array

for linha in meu\_array:

print(linha)

**Criação dinâmica de um array bidimensional**

Você também pode criar um array bidimensional de forma dinâmica, especialmente útil quando você não conhece o tamanho do array de antemão.

**Compreensão de listas**

python

Copiar código

# Array bidimensional 2x2 inicializado com zero usando compreensão de listas

meu\_array = [[0 for \_ in range(2)] for \_ in range(2)]

# Atribuindo valores

meu\_array[0][0] = 5

meu\_array[0][1] = 10

meu\_array[1][0] = 15

meu\_array[1][1] = 20

# Imprimindo o array

for linha in meu\_array:

print(linha)

Esses exemplos mostram como você pode declarar, inicializar e manipular arrays bidimensionais em Python.