

NumPy - Indexação Avançada

É possível fazer uma seleção de ndarray que seja uma sequência não tupla, objeto ndarray de tipo de dados inteiro ou booleano ou uma tupla com pelo menos um item sendo um objeto de sequência. A indexação avançada sempre retorna uma cópia dos dados. Contra isso, o fatiamento apresenta apenas uma visão.

Existem dois tipos de indexação avançada - **Inteira** e **Booleana**.

Indexação inteira

Este mecanismo ajuda a selecionar qualquer item arbitrário em um array com base em seu índice Ndimensional. Cada matriz inteira representa o número de índices nessa dimensão. Quando o índice consiste em tantas matrizes inteiras quanto as dimensões do ndarray de destino, ele se torna simples.

No exemplo a seguir, um elemento da coluna especificada de cada linha do objeto ndarray é selecionado. Conseqüentemente, o índice da linha contém todos os números das linhas e o índice da coluna especifica o elemento a ser selecionado.

Exemplo 1

```
import numpy as np
```

```
x = np.array([[1, 2], [3, 4], [5, 6]])  
y = x[[0,1,2], [0,1,0]]  
print y
```

[Demonstração ao vivo](#)

Sua saída seria a seguinte -

```
[1 4 5]
```

A seleção inclui elementos em (0,0), (1,1) e (2,0) da primeira matriz.

No exemplo a seguir, os elementos colocados nos cantos de uma matriz 4X3 são selecionados. Os índices de linha de seleção são [0, 0] e [3,3], enquanto os índices de coluna são [0,2] e [0,2].

Exemplo 2



```
import numpy as np
x = np.array([[ 0,  1,  2],[ 3,  4,  5],[ 6,  7,  8],[ 9, 10, 11]])

print 'Our array is:'
print x
print '\n'

rows = np.array([[0,0],[3,3]])
cols = np.array([[0,2],[0,2]])
y = x[rows,cols]

print 'The corner elements of this array are:'
print y
```

Demonstração ao vivo

A saída deste programa é a seguinte -

Our array is:

```
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

The corner elements of this array are:

```
[[ 0  2]
 [ 9 11]]
```

A seleção resultante é um objeto ndarray contendo elementos de canto.

A indexação avançada e básica pode ser combinada usando uma fatia (:) ou reticências (...) com uma matriz de índice. O exemplo a seguir usa fatia para linha e índice avançado para coluna. O resultado é o mesmo quando slice é usado para ambos. Mas o índice avançado resulta em cópia e pode ter layout de memória diferente.

Exemplo 3

```
import numpy as np
x = np.array([[ 0,  1,  2],[ 3,  4,  5],[ 6,  7,  8],[ 9, 10, 11]])

print 'Our array is:'
print x
print '\n'
```

Demonstração ao vivo



```
# slicing
z = x[1:4,1:3]

print 'After slicing, our array becomes:'
print z
print '\n'

# using advanced index for column
y = x[1:4,[1,2]]

print 'Slicing using advanced index for column:'
print y
```

O resultado deste programa seria o seguinte -

Our array is:

```
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

After slicing, our array becomes:

```
[[ 4  5]
 [ 7  8]
 [10 11]]
```

Slicing using advanced index for column:

```
[[ 4  5]
 [ 7  8]
 [10 11]]
```

Indexação de matriz booleana

Este tipo de indexação avançada é usado quando o objeto resultante é o resultado de operações booleanas, como operadores de comparação.

Exemplo 1

Neste exemplo, os itens maiores que 5 são retornados como resultado da indexação booleana.

```
import numpy as np
x = np.array([[ 0,  1,  2],[ 3,  4,  5],[ 6,  7,  8],[ 9, 10, 11]])

print 'Our array is:'
print x
print '\n'

# Now we will print the items greater than 5
print 'The items greater than 5 are:'
print x[x > 5]
```

O resultado deste programa seria -

Our array is:

```
[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

The items greater than 5 are:

```
[ 6  7  8  9 10 11]
```

Exemplo 2

Neste exemplo, os elementos NaN (Not a Number) são omitidos usando ~ (operador de complemento).

```
import numpy as np
a = np.array([np.nan, 1,2,np.nan,3,4,5])
print a[~np.isnan(a)]
```

Sua saída seria -

```
[ 1.  2.  3.  4.  5.]
```

Exemplo 3

O exemplo a seguir mostra como filtrar os elementos não complexos de uma matriz.

```
import numpy as np
a = np.array([1, 2+6j, 5, 3.5+5j])
print a[np.iscomplex(a)]
```

Aqui, a saída é a seguinte -

```
[2.0+6.j 3.5+5.j]
```