

NumPy - Indexação e fatiamento

O conteúdo do objeto ndarray pode ser acessado e modificado por indexação ou fatiamento, assim como os objetos contêineres integrados do Python.

Conforme mencionado anteriormente, os itens no objeto ndarray seguem o índice baseado em zero. Três tipos de métodos de indexação estão disponíveis – **acesso de campo**, **fatiamento básico** e **indexação avançada** .

O fatiamento básico é uma extensão do conceito básico do Python de fatiamento em n dimensões. Um objeto de fatia Python é construído fornecendo parâmetros **start**, **stop** e **step** à **função de fatia** integrada . Este objeto de fatia é passado para o array para extrair uma parte do array.

Exemplo 1

```
import numpy as np
a = np.arange(10)
s = slice(2,7,2)
print a[s]
```

[Demonstração ao vivo](#)

Sua saída é a seguinte -

```
[2 4 6]
```

No exemplo acima, um objeto **ndarray** é preparado pela função **arange()** . Em seguida, um objeto de fatia é definido com valores de início, parada e etapa 2, 7 e 2, respectivamente. Quando esse objeto de fatia é passado para o ndarray, uma parte dele começando com o índice 2 até 7 com um passo de 2 é fatiada.

O mesmo resultado também pode ser obtido fornecendo os parâmetros de fatiamento separados por dois pontos: (start:stop:step) diretamente ao objeto **ndarray** .

Exemplo 2

```
import numpy as np
a = np.arange(10)
```

[Demonstração ao vivo](#)

```
b = a[2:7:2]
print b
```

Aqui, obteremos a mesma saída -

```
[2 4 6]
```

Se for colocado apenas um parâmetro, será retornado um único item correspondente ao índice. Se um `:` for inserido na frente dele, todos os itens desse índice em diante serão extraídos. Se dois parâmetros (com `:` entre eles) forem usados, os itens entre os dois índices (não incluindo o índice de parada) com a etapa um padrão serão fatiados.

Exemplo 3

```
# slice single item
import numpy as np

a = np.arange(10)
b = a[5]
print b
```

[Demonstração ao vivo](#)

Sua saída é a seguinte -

```
5
```

Exemplo 4

```
# slice items starting from index
import numpy as np
a = np.arange(10)
print a[2:]
```

[Demonstração ao vivo](#)

Agora, a saída seria -

```
[2 3 4 5 6 7 8 9]
```



Reserve melhores imóveis por temporada em Puebla. ~
férias espaçosas e privadas com tudo o que precisa pa

Exemplo 5

```
# slice items between indexes
import numpy as np
a = np.arange(10)
print a[2:5]
```

[Demonstração ao vivo](#)

Aqui, a saída seria -

```
[2 3 4]
```

A descrição acima também se aplica ao ndarray multidimensional .



Samsung

Loja Oficial Samsung



Exemplo 6

```
import numpy as np
a = np.array([[1,2,3],[3,4,5],[4,5,6]])
print a
```

[Demonstração ao vivo](#)

```
# slice items starting from index
print 'Now we will slice the array from the index a[1:]'
print a[1:]
```

A saída é a seguinte -

```
[[1 2 3]
 [3 4 5]
 [4 5 6]]
```

Now we will slice the array from the index a[1:]



```
[[3 4 5]
 [4 5 6]]
```

O fatiamento também pode incluir reticências (...) para criar uma tupla de seleção do mesmo comprimento que a dimensão de uma matriz. Se reticências forem usadas na posição da linha, ela retornará um ndarray composto por itens nas linhas.

DE ANÚNCIOS



Saluda - Espaço para toda a família - Preço sem surpresas



Reserve alugueis por temporada incríveis em Saluda, (Muito, Na Vela @) ligamos famílias e serviços ao arred

Exemplo 7

```
# array to begin with
import numpy as np
a = np.array([[1,2,3],[3,4,5],[4,5,6]])
```

Demonstração ao vivo

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

```
# this returns array of items in the second column
print 'The items in the second column are:'
print a[...,1]
print '\n'
```

```
# Now we will slice all items from the second row
print 'The items in the second row are:'
print a[1,...]
print '\n'
```

```
# Now we will slice all items from column 1 onwards
print 'The items column 1 onwards are:'
print a[...,1:]
```

A saída deste programa é a seguinte -

```
Our array is:
[[1 2 3]
 [3 4 5]
 [4 5 6]]
```



The items in the second column are:

[2 4 5]

The items in the second row are:

[3 4 5]

The items column 1 onwards are:

[[2 3]

[4 5]

[5 6]]