NumPy Syntax

Indexing NumPy elements using conditionals

NumPy elements can be indexed using conditionals. The syntax to filter an array using a conditional is `array_name[conditional]`. The returned array will contain only the elements for which the conditional evaluates to `True`.

```
numbers = np.array([-5, 4, 0, 2, 3])
positives = numbers[numbers > 0]
print(positives)
# array([4, 2, 3])
```

NumPy element-wise logical operations

NumPy Arrays support element-wise logical operations, returning new Arrays populated with `False` or `True` based on their evaluation.

```
numbers = np.array([-5, 4, 0, 2, 3])
is_positive = numbers > 0
print(is_positive)
# array([False, True, False, True,
True], dtype=bool)
```

Creating NumPy Arrays from files

NumPy Arrays can be created from data in CSV files or other delimited text by using the `np.genfromtxt()` method.
The named parameter `delimiter` is used to determine the delimiting character between values.

```
imported_csv =
np.genfromtxt("filename.txt",
delimiter=",")
```

codecademy

NumPy Arrays

NumPy (short for "Numerical Python") is a Python module used for numerical computing, creating arrays and matrices, and performing very fast operations on those data structures. The core of NumPy is a multidimensional Array object.
The NumPy `.array()` method is used to create new NumPy Arrays.

```
# Import the NumPy module, aliased as
'np'
import numpy as np

# NumPy Arrays can be created with a
list
my_array = np.array([1, 2, 3, 4])
```

Accessing NumPy Array elements by index

Individual NumPy Array elements can be accessed by index, using syntax identical to Python lists:
`array[index]` for a single element, or `array[start:end]` for a slice, where `start` and `end` are the starting and ending indexes for the slice.
Nested Arrays or elements can be accessed by adding additional comma-separated parameters.

```
matrix = np.array([[1, 2, 3],
[4, 5, 6],
[7, 8, 9]])

print(matrix[0, 0])
# 1

print(matrix[1,:])
# array([4, 5, 6])
```

NumPy element-wise arithmetic operations

NumPy Arrays support element-wise operations, meaning that arithmetic operations on arrays are applied to each value in the array.
Multiple arrays can also be used in arithmetic operations, provided that they have the same lengths.

```
odds = np.array([1, 3, 5, 7, 9])
evens = odds + 1
print(evens)
# array([2, 4, 6, 8, 10])

array1 = np.array([1, 2, 3])
array2 = np.array([4, 3, 2])
new_array = array1 + array2
print(new_array)
# array([5, 5, 5])
```

Print Share

codecademy