INTRO TO PYTHON FOR FINANCE

Arrays

Adina Howe
Instructor

Installing packages

```
pip3 install package_name_here
```

```
pip3 install numpy
```

Importing packages

```
import numpy
```

NumPy and Arrays

```
import numpy

my_array = numpy.array([0, 1, 2, 3, 4])

print(my_array)

[0, 1, 2, 3, 4]

print(type(my_array))

<class 'numpy.ndarray'>
```

The array() function takes a list as its input.

Using an alias

```
import package_name
package_name.function_name(...)
```

```
import numpy as np
my_array = np.array([0, 1, 2, 3, 4])
print(my_array)

[0, 1, 2, 3, 4]
```

Why use an array for financial analysis?

- Arrays can handle very large datasets efficiently
  - Computationally-memory efficient
  - Faster calculations and analysis than lists
  - Diverse functionality (many functions in Python packages)

What's the difference?

NUMPY ARRAYS

```
my_array = np.array([3, 'is', True])

print(my_array)

['3' 'is' 'True']
```

Lists can contain only a single data type.
All the elements are converted to a string.

LISTS

```
my_list = [3, 'is', True]

print(my_list)

[3, 'is', True]
```

Numpy will automatically convert all the
elements to the most compatible type.

Array operations

ARRAYS

```
import numpy as np

array_A = np.array([1, 2, 3])
array_B = np.array([4, 5, 6])

print(array_A + array_B)

[5 7 9]
```

LISTS

```
list_A = [1, 2, 3]
list_B = [4, 5, 6]

print(list_A + list_B)

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

Array indexing

```
import numpy as np

months_array = np.array(['Jan', 'Feb', 'March', 'Apr', 'May'])

print(months_array[3])

Apr

print(months_array[2:5])

['March' 'Apr' 'May']
```

Array slicing with steps

```
import numpy as np

months_array = np.array(['Jan', 'Feb', 'March', 'Apr', 'May'])

print(months_array[0:5:2])

['Jan' 'March' 'May']
```

INTRO TO PYTHON FOR FINANCE

Let's practice!

INTRO TO PYTHON FOR FINANCE

Two Dimensional Arrays

Adina Howe
Instructor

Two-dimensional arrays

```
import numpy as np

months = [1, 2, 3]
prices = [238.11, 237.81, 238.91]
```

```
cpi_array = np.array([months, prices])

print(cpi_array)

[[   1.      2.      3.  ]
 [ 238.11  237.81  238.91]]
```

pass in a list of two lists as the input

Array Methods

```
print(cpi_array)
[[   1.      2.      3.  ]
 [ 238.11  237.81  238.91]]
```

`.shape` gives you dimensions of the array

```
print(cpi_array.shape)

(2, 3)
```

`.size` gives you total number of elements in the array

```
print(cpi_array.size)

6
```

Array Functions

```
import numpy as np

prices = [238.11, 237.81, 238.91]
prices_array = np.array(prices)
```

`np.mean()` calculates the mean of an input

```
print(np.mean(prices_array))

238.27666666666667
```

`np.std()` calculates the standard deviation of an input

```
print(np.std(prices_array))

0.46427960923946671
```

The arange() function

`numpy.arange()` creates an array with start, end, step

```
import numpy as np

months = np.arange(1, 13)

print(months)

[ 1  2  3  4  5  6  7  8  9 10 11 12]

months_odd = np.arange(1, 13, 2)

print(months_odd)

[ 1  3  5  7  9 11]
```

The transpose() function

`numpy.transpose()` switches rows and columns of a numpy array

```
print(cpi_array)

[[   1.      2.      3.   ]
 [ 238.11  237.81  238.91]]

cpi_transposed = np.transpose(cpi_array)

print(cpi_transposed)

[[   1.    238.11]
 [   2.    237.81]
 [   3.    238.91]]
```

Array Indexing for 2D arrays

```
print(cpi_array)

[[   1.      2.      3.  ]
 [ 238.11  237.81  238.91]]

# row index 1, column index 2
cpi_array[1, 2]

238.91

# all row slice, third column
print(cpi_array[:, 2])

[   3.    238.91]
```

INTRO TO PYTHON FOR FINANCE

Let's practice!

INTRO TO PYTHON FOR FINANCE

Using Arrays for Analyses

Adina Howe
Instructor

Indexing Arrays

```
import numpy as np

months_array = np.array(['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun'])

indexing_array = np.array([1, 3, 5])

months_subset = months_array[indexing_array]
print(months_subset)

['Feb' 'Apr' 'Jun']
```

More on indexing arrays

```
import numpy as np

months_array = np.array(['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun'])

negative_index = np.array([-1, -2])

print(months_array[negative_index])

['Jun' 'May']
```

Boolean arrays

```
import numpy as np

months_array = np.array(['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun'])

boolean_array = np.array([True, True, True, False, False, False])

print(months_array[boolean_array])

['Jan' 'Feb' 'Mar']
```

The boolean array subsets only the elements where the index is True.

More on Boolean arrays

```
prices_array = np.array([238.11, 237.81, 238.91])

# Create a Boolean array
boolean_array = (prices_array > 238)

print(boolean_array)

[ True False  True]

print(prices_array[boolean_array])

[ 238.11  238.91]
```

INTRO TO PYTHON FOR FINANCE

Let's practice!