

NumPy arrays

Presenter: Steve Baskauf
steve.baskauf@vanderbilt.edu



Jean & Alexander Heard
LIBRARIES

CodeGraf landing page

- vanderbi.it/codegraf

Vectorized computation

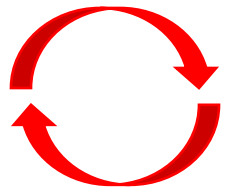


Jean & Alexander Heard
LIBRARIES

Procedural approach: for loop

```
price_list = [3.89, 14.78, 20.01, 99.62, 0.47]
```

```
euro_conversion_factor = 0.93
```



```
euro_list = []
```

```
for price in price_list:
```

```
    euro_list.append(price * euro_conversion_factor)
```

```
print(euro_list)
```

NumPy library

```
import numpy as np
```

This is the convention for import and should be followed.

Vectorized computation approach

A numpy **ndarray** object is designed for **vectorized computation**.

```
import numpy as np
price_array = np.array([3.89, 14.78, 20.01, 99.62, 0.47])
euro_conversion_factor = 0.93

euro_array = price_array * euro_conversion_factor

print(euro_array)
```

Comparison of approaches

- The **procedural** approach requires the operation to be applied to each item **one at a time**.
- The **vectorized** approach automatically applies the operation to **all elements** of the array.
- Operations on NumPy arrays are carried out **10 to 100 times faster** than on "normal" Python lists.

Dimensions and indexing



Jean & Alexander Heard
LIBRARIES

One vs. two dimensions

- One dimension is like a list:

```
[ 3.89 14.78 20.01 99.62 0.47]
```

- Two dimensions are like a table:

```
[[34506 35446 40190 43824 46456]  
 [45369 46894 43901 44870 45978]  
 [21554 28745 34369 43593 53982]]
```

- More than two dimensions are possible

Indexing

- `prices[3]`

```
[3.6177, 13.7454, 18.6093, 92.6466, 0.4371]
```

- `annual_sales[1, 2]`

```
[[34506 35446 40190 43824 46456]  
 [45369 46894 43901 44870 45978]  
 [21554 28745 34369 43593 53982]]
```

- Displayed as [row, column] with zero-based indexing

Array-based calculations

where () function

- The `np.where()` function applies a boolean test to all array elements.
- Similar to `if` function in Excel.

```
np.where(trucks >= 45000, 'good year', 'bad year')
```

Value if True Value if False

- Return value is an array of the same size with pairwise substitution.

where () function


`np.where(trucks >= 45000, 'good year', 'bad year')`

Value if True Value if False

trucks: [45369 46894 43901 44870 45978]

True True False False True

return value: ['good year' 'good year' 'bad year' 'bad year' 'good year']



- Return value is an array of the same size with **pairwise substitution**.

Image examples



Jean & Alexander Heard
LIBRARIES

An image can be considered a 2D array

- Each pixel is an array element
- The grayscale value ranges from 0=black to 255=white

```
[ [156 157 160 ... 152 152 152]
  [156 157 159 ... 152 152 152]
  [158 157 156 ... 152 152 152]
  ...
  [121 123 126 ... 121 113 111]
  [121 123 126 ... 121 113 111]
  [121 123 126 ... 121 113 111] ]
```

Remote Support for Teaching and Research Needs



Access to digital collections 24/7



Skype consultations with your
subject librarian



Ask a Librarian: an easy way to
submit a question via email



Live chat available from the
Library home page

NEED HELP? ASK A LIBRARIAN!

<https://www.library.vanderbilt.edu/ask-librarian.php>

Jean & Alexander Heard
LIBRARIES