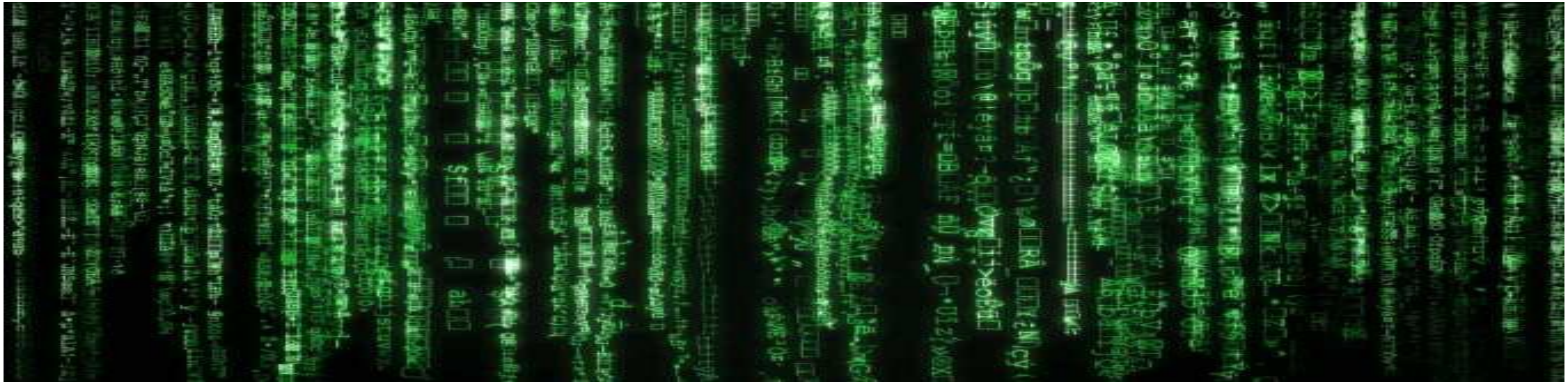

INF 110 Discovering Informatics

Arrays and Tables

NORTHERN ARIZONA UNIVERSITY

Arrays (numpy)

A sequence of the same type.

NumPy arrays differ from Python lists in two ways:

1. They aren’t part of the base Python distribution
    - You have to import numpy
2. All items in the list have to be the ***same data type.***
    - This differs from base Python lists where you can have strings, floats, integers all in one list

```
from datascience import *
import numpy as np

make_array(100, 200, 300)
make_array(1.0, 2.0, 3.0)
make_array("a", "b", "c")
```

Arrays (numpy)

Also:

The make_array() syntax is supplied by the datascience library, not base numpy.

If you're in numpy, you can't just use make_array(), you have to use the array constructor that comes with numpy.

```
from datascience import *
import numpy as np

make_array(100, 200, 300)
make_array(1.0, 2.0, 3.0)
make_array("a", "b", "c")
```

This is why we will load the datascience library in our import statements!

Arrays from ranges

Ranges allow us to create a sequence of values

It’s often easier to use a convenient function like “arange” to specify a list instead of manually specifying [0,1,2,3,4].

- Specifying np.arrange(5) will give you the same result.

```
from datascience import *
import numpy as np

np.arange(5)
np.arange(1, 10)
np.arange(1, 10, 2)
```

Arrays from ranges

There are several parametrizations for the *arange* function.

```
from datascience import *
import numpy as np

np.arange(5)
np.arange(1, 10)
np.arange(1, 10, 2)
```

With one argument, that's the number of elements that will be in the list.

With two, the first is the start value and the second is the end value.

- But it's exclusive: it won't be included in the array.

With three, it entails a ***start***, ***end***, and ***step***.

Element from a slice

We can retrieve an element using an index (or get) operation

```
letters = make_array("a", "b", "c", "d", "e", "f")
```

```
letters[1]
```

Here, the syntax is the same as with Python lists: bracket syntax (and 0-indexed)

What will this index retrieval result in?

Arrays from slices

We can also access multiple items at the same time.

```
letters = make_array("a", "b", "c", "d", "e", "f")

letters[1:4]
```

We use bracket syntax, but with two arguments separated by a colon.
Here, we are asking for elements 1 through 4.

What will this index retrieval result in?

Live Code Examples of Arrays

Task: Create some arrays and ranges that illustrate the construction process and options

Learning Outcomes

- Creating arrays with make_array()
- Creating ranges with np.arange()
- Pulling individual elements from sequences
- Creating arrays with slicing

Table Structure

- A Table is a sequence of labeled columns
- Each row represents one individual
- Data within a column represents one attribute of the individuals

Name	**Code**	**Area (m2)**
California	CA	163696
Nevada	NV	110567

Tables aren’t part of base Python, so you have to import the datascience package.

Table Structure

- A Table is a sequence of labeled columns
- Each row represents one individual
- Data within a column represents one attribute of the individuals

Name	**Code**	**Area (m2)**
California	CA	163696
Nevada	NV	110567

Column

Table Structure

- A Table is a sequence of labeled columns
- Each row represents one individual
- Data within a column represents one attribute of the individuals

Name	**Code**	**Area (m2)**
Californ	CA	163696
Nevada	NV	110567

Label

Table Structure

- A Table is a sequence of labeled columns
- Each row represents one individual
- Data within a column represents one attribute of the individuals

Name	Code	Area (m2)
California	CA	163696
Nevada	NV	110567

Row

Live Code Creating a table in Excel

Task: Create a table in Excel that describes some favorite food options for each student

Learning Outcomes

- Creating a table with labels, columns, and rows
- Sorting columns in Excel
- Using the CSV file format

"If these are available in Excel, why would I want to learn Python?"

Tables in Python are objects that contain the same data as a table you might work with in Excel but with some differences:

- Bigger – Excel has a row limit
- Faster – Have you tried opening an excel sheet with 1M rows?
- Manipulation with methods
- Support scientific work flows
- Tables in Python give us ***scalability*** and ***accessibility***.

Some Table Operations

- `t.select(label)` - constructs a new table with just the specified columns
    - **If you have 5 columns but you only want 3**
- `t.drop(label)` - constructs a new table in which the specified columns are omitted
- `t.sort(label)` - constructs a new table with rows sorted by the specified column
- `t.where(label, condition)` - constructs a new table with just the rows that match the condition

Live Code Getting Help

Task: Python has a lot of built-in documentation use “?” and “dir” to inspect the methods in the Table class.

Learning Outcomes

- Learn about methods in the Table class
- Become comfortable looking up documentation in Jupyter

Live Code Working with Tables

Task: Use the table you built in Excel earlier in Python

Learning Outcomes

- Loading tables
- Sorting columns
- Adding new columns

end