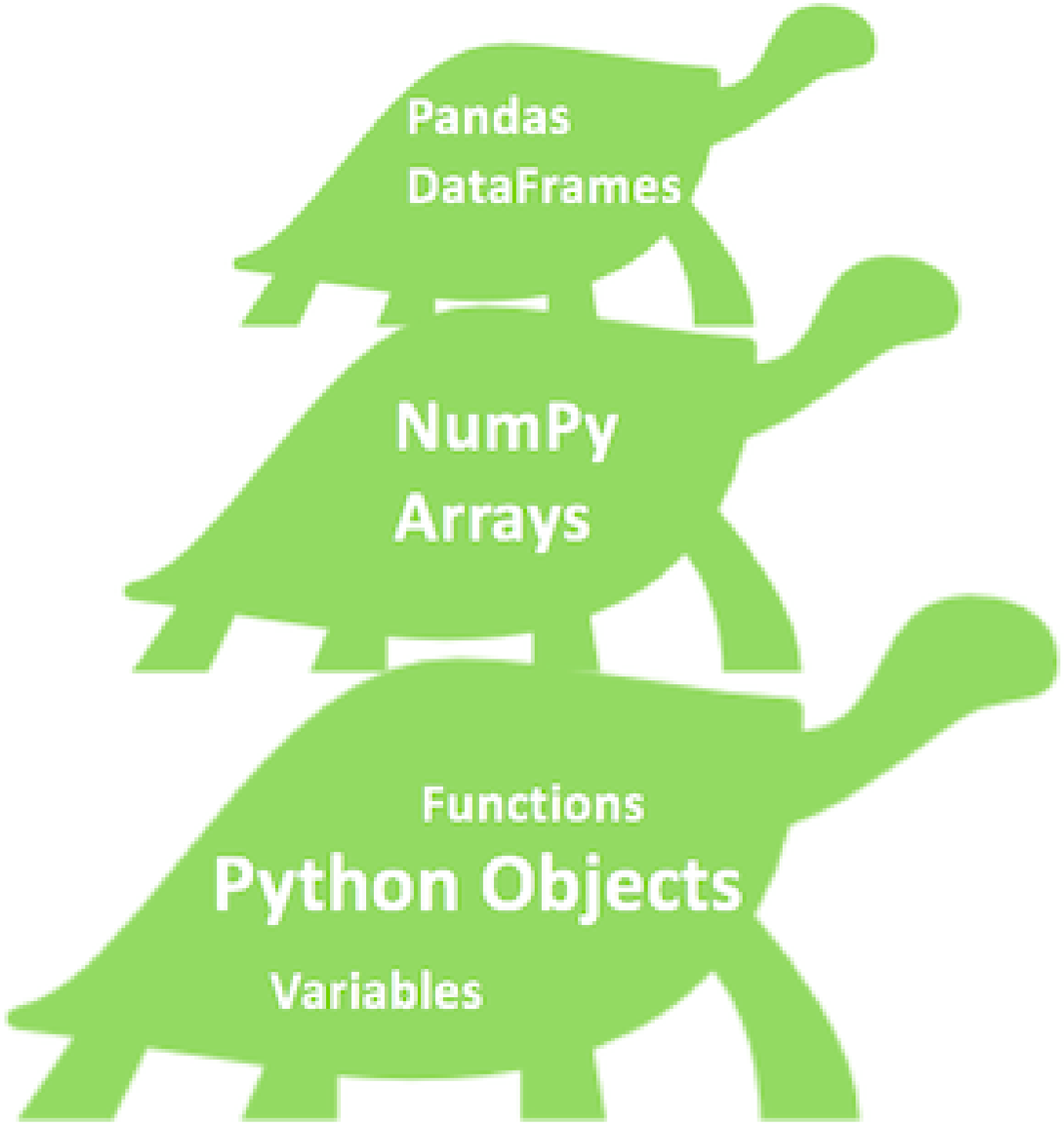
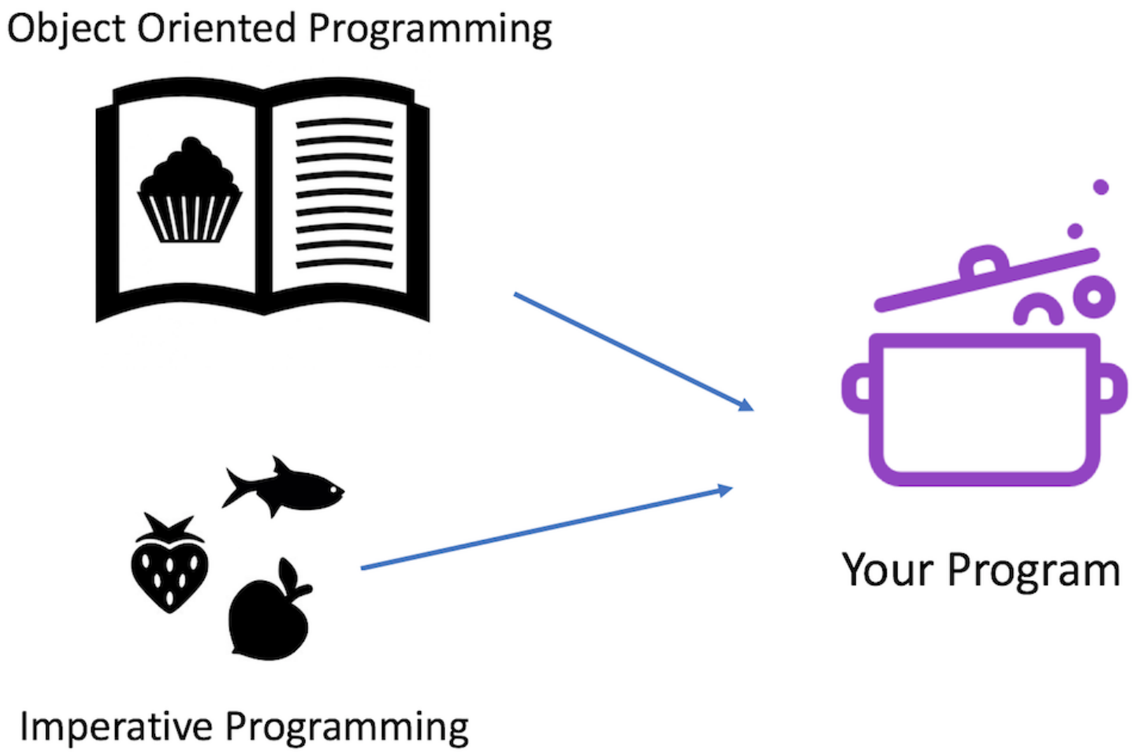
OBJECT ORIENTED PROGRAMMING IN PYTHON

Intro to Object Oriented Programming in Python

Vicki Boykis
Senior Data Scientist


Pandas
DataFrames
NumPy
Arrays
Functions
Python Objects
Variables

What's Object-Oriented Programming? (OOP)



- A way to build flexible, reproducible code
- Developing building blocks to developing more advanced modules and libraries

Imperative Style and OOP Style

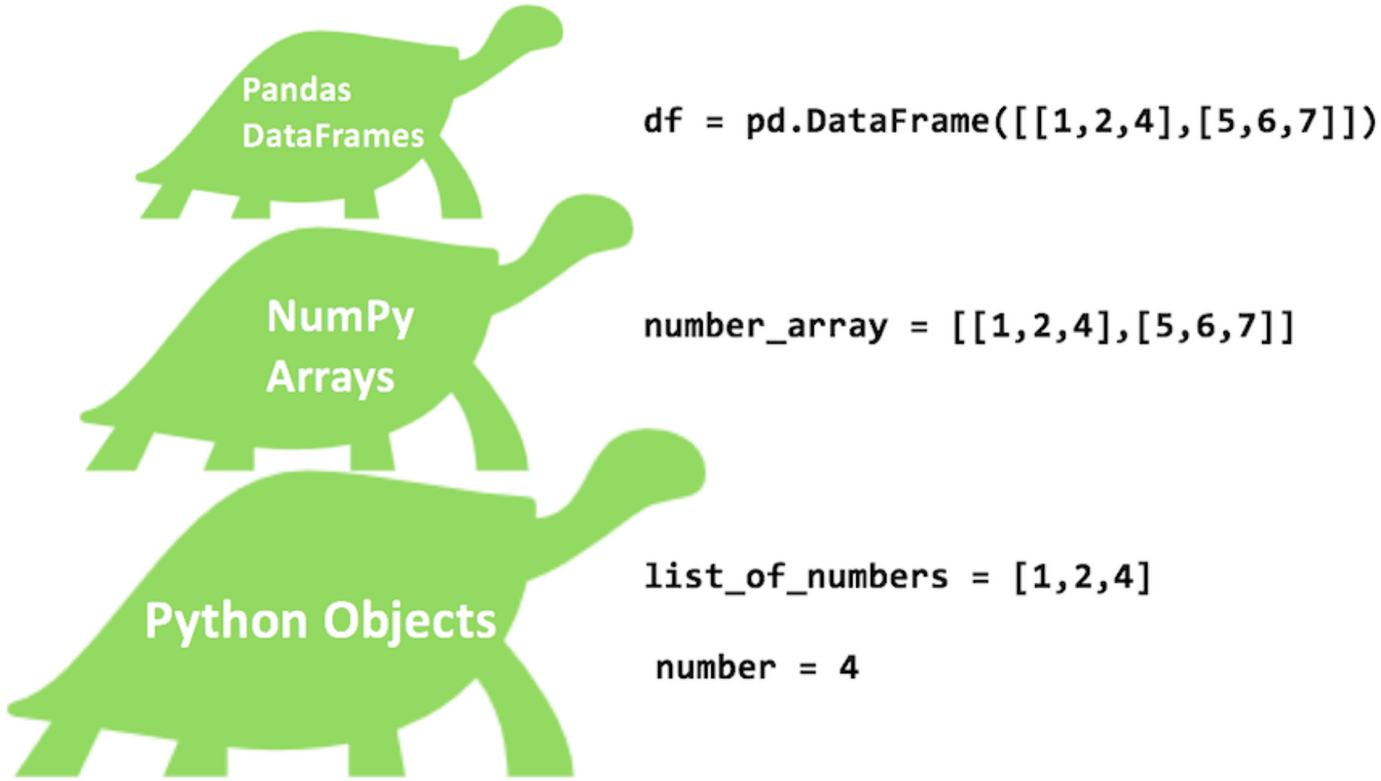
IMPERATIVE

```
our_list = [1,2,3]

for item in our_list:
    print(f"Item {item}")
```

OOP

```
class PrintList:

    def __init__(self,numberlist):
        self.numberlist = numberlist

    def print_list(self):
       for item in self.numberlist:
            print(f"Item {item}")

A = PrintList([1,2,3])
A.print_list()
```

All Python libraries work together




```
df = pd.DataFrame([[1,2,4],[5,6,7]])
```

```
number_array = [[1,2,4],[5,6,7]]
```

```
list_of_numbers = [1,2,4]
```

```
number = 4
```

OBJECT ORIENTED PROGRAMMING IN PYTHON

Let's get started!

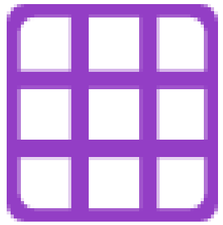
OBJECT ORIENTED PROGRAMMING IN PYTHON

Introduction to NumPy Internals

Vicki Boykis
Senior Data Scientist

What's NumPy?

NumPy is a package for scientific computing in Python.



- Uses matrices and vectors as data structures
- Perfect for data science, where data is laid out in table-like formats

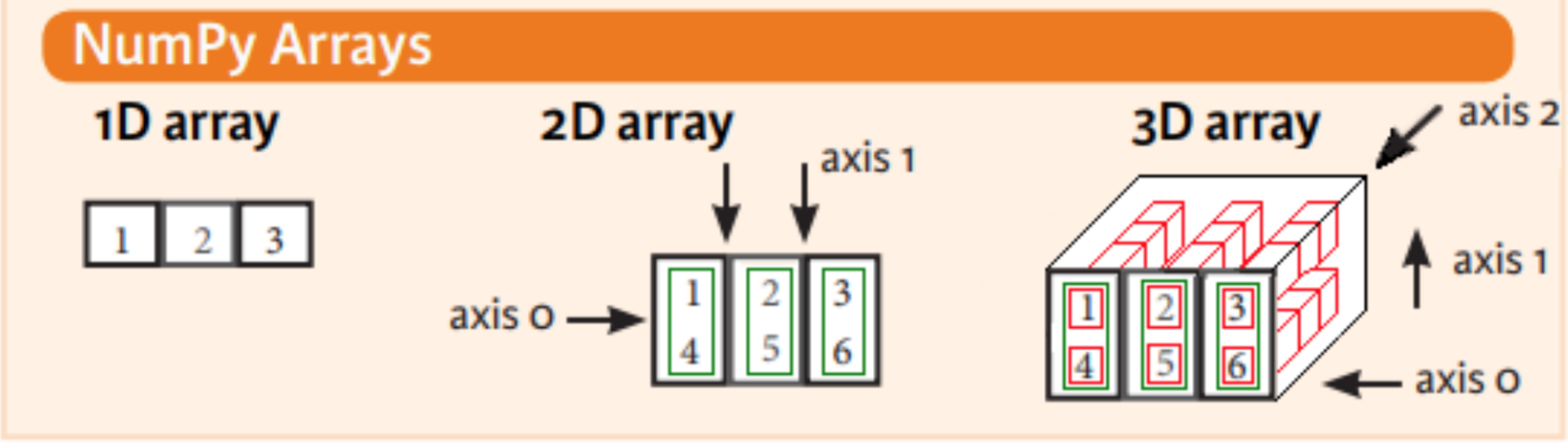
NumPy as a building block to Pandas

pandas.Series ¶

class pandas.**Series**(*data=None*, *index=None*, *dtype=None*, *name=None*, *copy=False*, *fastpath=False*) [source]

One-dimensional ndarray with axis labels (including time series).

Creating NumPy arrays



Source: DataCamp

NumPy Array example

Example:

```
import numpy as np

our_array = np.array([2,3,4])
print(our_array)

[2 3 4]
```

```
print(type(our_array))

<type 'numpy.ndarray'>
```

Creating Multi-Dimensional Arrays

Example 1:

```
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14]])
```

Example 2:

```
array([6, 7, 8])
```

OBJECT ORIENTED PROGRAMMING IN PYTHON

Let's practice!

OBJECT ORIENTED PROGRAMMING IN PYTHON

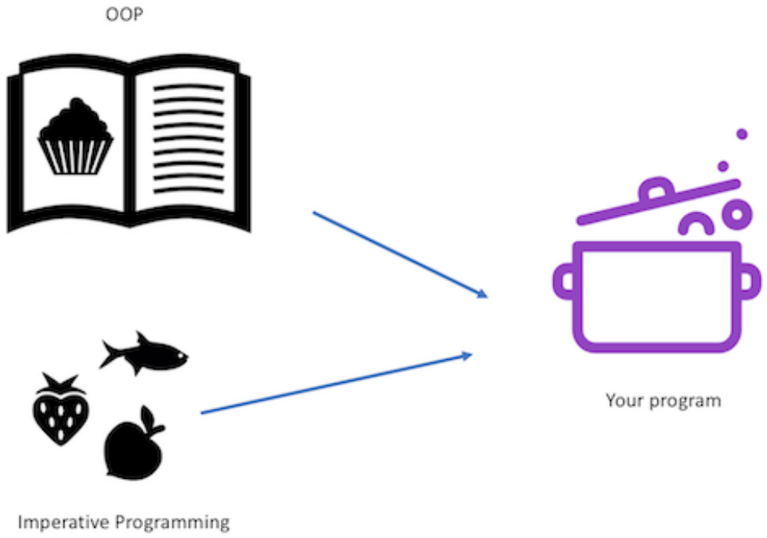
Introduction to Objects and Classes

Introduction to Classes

Vicki Boykis

What is a class?

A reuseable chunk of code that has methods and variables.

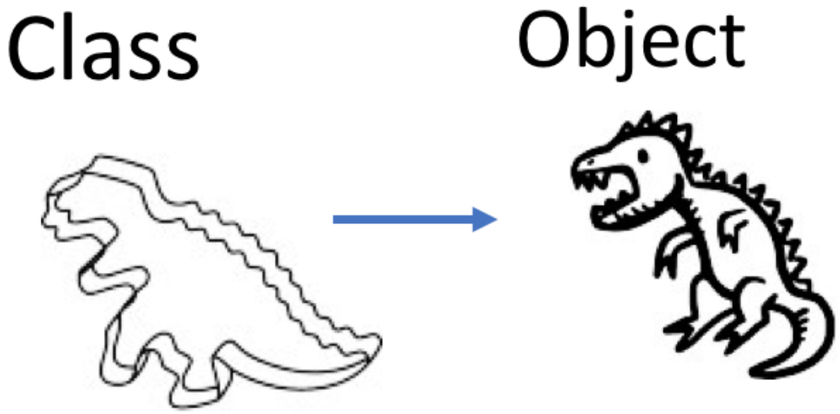



```
class PrintList:

    def __init__(self,numberlist):
        self.numberlist = numberlist

    def print_list(self):
       for item in self.numberlist:
            print(f"Item {item}")

A = PrintList([1,2,3])
A.print_list()
```

OOP Vocabulary

Imperative	OOP
Variable	Attribute/Field
Function	Method

A Class is a template for an object



Declaring a Class

Declaring a class

```
class Dinosaur:
 pass
```

```
# Used in Python 3, with/without parentheses
class Dinosaur():
  pass

# Used in Python 2
class Dinosaur(object):
  pass
```

An object is an instance of a class.

```
Tyrannosaurus = Dinosaur()
```

OBJECT ORIENTED PROGRAMMING IN PYTHON

Let's practice!