

# Object-Oriented Programming in Python

**Level:** Advanced

**Duration:** 6 hours

Object-oriented programming is the dominant programming paradigm in Python and can be used to improve the structure of your data science code. Here, we will learn how to model real-world entities using classes, how to create class instances, and how to attach data and behaviour to these objects. The main ideas of object-oriented design (inheritance, polymorphism, encapsulation, abstraction) are covered, and you will learn how to extend existing classes from well-known data science packages.



## Course Outline

- **Object-oriented design:** An introduction to the aims and mindset of object-oriented programming, the most dominant programming paradigm in Python.
- **Classes in Python:** How to define classes, and how to use these to create instances.
- **Methods and attributes for instances and classes:** How to attach data and behaviour to classes and their instances.
- **Inheritance, Composition, Polymorphism:** A brief introduction to best practices for creating flexible object-oriented code.
- **Extending classes from well-known data science packages:** How to take existing classes and add new features to them.

# Learning Outcomes

## Session 1:

*By the end of session 1 participants will...*

- understand how to model real-world (and more abstract) objects using classes
- be able to define classes in Python
- understand how to make class instances
- understand instance methods and attributes

## Session 2:

*By the end of session 2 participants will...*

- understand how inheritance can help with code reuse
- be able to define class and static methods
- understand how to use composition to make classes that contain other objects
- understand how OOP and inheritance makes it easier to write polymorphic code

*This course does not include:*

- using object-relational mappings to interact with databases, see our [Intro to SQL with Python](#) course for this
- advanced OOP ideas (design patterns, metaclasses)
- unified modelling language or other graphical ways to represent how classes/instances interact
- a comparison of the programming paradigms that are available in Python
- an introduction to Pandas, Numpy or Matplotlib, see our [Intro to Python](#) course for this

## Prior Knowledge

The course will assume a basic understanding of popular Python libraries including NumPy, Pandas and Matplotlib, as well as common data types like Pandas DataFrames. The course will also make use of basic programming concepts like defining functions and running for loops. Completion of our [Intro to Python](#) and [Programming with Python](#) (or equivalent experience) would provide a suitable background.

# Attendee Feedback

## Contact

[hello@jumpingrivers.com](mailto:hello@jumpingrivers.com)