Overview

Bookmark this page

Iteration

Introduction

In this lesson we will explore functional programming in more depth, We will look at the lambda special form, which can be used in conjunction with map() and filter() and also in comprehensions. We will consider the debates as to lambda's effectiveness by looking at its history and the comments of Python's creator Guido van Rossum.

Where we had been avoiding the topic up until now, at long last we investigate iterators and iterables, core constructs in Python 3. Both terms are sprinkled throughout the Python programming literature and the distinction between them can be subtle.

Finally we look at generators, powerful programming constructs that are lazy by nature (in the functional programming sense of lazy) and thus can produce infinite sequences without exhausting the resources of real-world computers.

Learning Objectives

Upon successful completion of this lesson, you will be able to:

- use the lambda special form to define an anonymous function.
- use lambda expressions with map and $fi < er$ and in comprehensions.
- articulate the difference between an iterator and an iterable.
- use $yield$ to create a generator.

New Words, Concepts, and Tools

- Anonymous function
- Lambda
- Iterator
- Iterable
- Generator
- yield

Prerequisites

To be successful in this lesson you will need to be very familiar with the following concepts, covered earlier in the class:

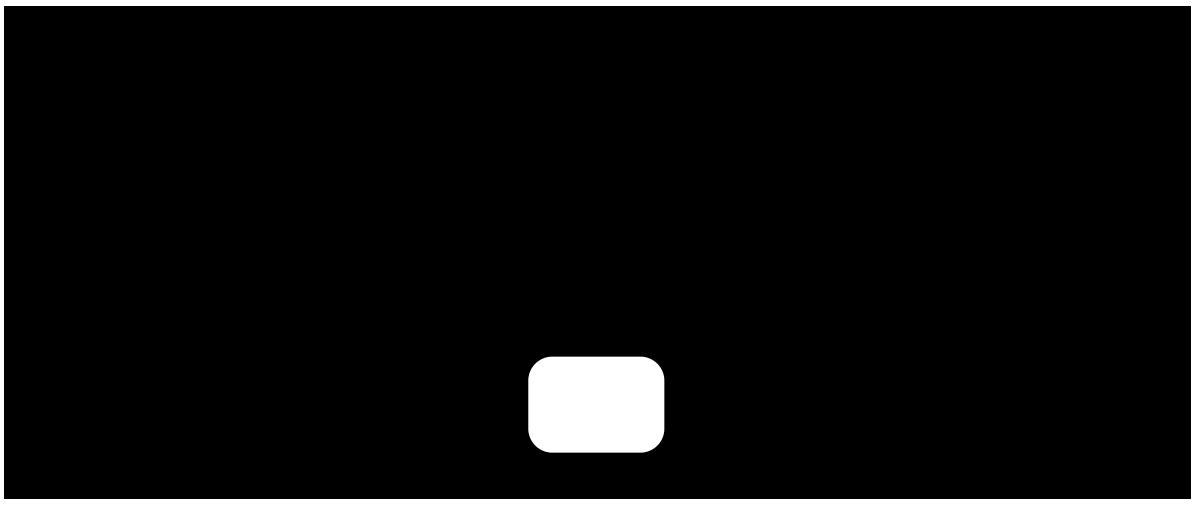
1. Familiar with classes, objects and inheritance.
2. Able to code simple OO program using these concepts.

Before you Start

Read the following articles and watch the Functional Programming Video in preparation for this lesson.

Functional Programming, Why

Start of transcript. Skip to the end.



Functional programming with its idioms,

its focus and data

transformations rather than on control flow,

its obsession with state and state management,

why does any of this matter?

Why are we studying this, functional programming as a paradigm?

Good question.

In a certain sense, Python doesn't make

Video

Download video file

Transcripts

Download SubRip (.srt) file

Download Text (.txt) file

Required Reading

- Small functions and the lambda expression
  https://docs.python.org/dev/howto/functional.html?highlight=lambda#small-functions-and-the-lambda-expression
- Iterators
  https://docs.python.org/3/glossary.html#term-iterator
  https://docs.python.org/3/library/stdtypes.html#iterator-types
  https://docs.python.org/dev/howto/functional.html#iterators
- Itertools
  https://docs.python.org/3/library/itertools.html
  https://pymotw.com/3/itertools/index.html
- What exactly are Python's iterator, iterable, and iteration protocols?
  https://stackoverflow.com/questions/9884132/what-exactly-are-pythons-iterator-iterable-and-iteration-protocols
- Generators
  https://wiki.python.org/moin/Generators

Optional Reading

Functional Programming

- Lott, S. (2015) Chapter 5. Higher-order Functions. Using Python lambda forms. In Functional Python Programming.
- Lott, S. (2015) Chapter 3. Functions, Iterators, and Generators. In Functional Python Programming.
- The Iterator Protocol: How For Loops Work in Python
  http://treyhunner.com/2016/12/python-iterator-protocol-how-for-loops-work/
- Chapter 14 in Fluent Python by Luciano Ramalho
  http://www.learningpython.com/2009/02/23/iterators-iterables-and-generators-oh-my/
- PEP 255 — Simple Generators
  https://www.python.org/dev/peps/pep-0255/
- https://maryrosecook.com/blog/post/a-practical-introduction-to-functional-programming
- https://www.dataquest.io/blog/introduction-functional-programming-python/

Iterators

- https://anandology.com/python-practice-book/iterators.html
- https://www.programiz.com/python-programming/iterator
- https://stackoverflow.com/questions/2776829/difference-between-pythons-generators-and-iterators

Itertools

- https://medium.com/@mr_rigden/a-guide-to-python-itertools-82e5a306cdf8
- https://docs.python.org/3.1/library/itertools.html

Programming paradigms

- https://en.wikipedia.org/wiki/Comparison_of_programming_paradigms
- https://dev.to/ericnormand/programming-paradigms-and-the-procedural-paradox

‹ Previous Next ›

Refactoring

- http://blog.thedigitalcatonline.com/blog/2017/07/21/refactoring-with-test-in-python-a-practical-example/
- Classic book: https://martinfowler.com/books/refactoring.html

Suggested Workflow

- Explore the "Before you Start" readings and video
- Work through the lesson content pages
- Watch the required videos
- Submit your assignment

At the End of the Lesson

What material in this lesson do you still feel unclear about? Please use the discussion forum for this lesson to describe any concepts or ideas you've struggled with this week. Your instructor will address concepts that seem problematic for several students. You may also want to address issues raised by other students or ask for their help.