

Next ?

MCDB 170

Where are we now with Python?

- Variables
- Conditionals
- Loops
- Functions
- Data containers
- Classes
- Modules
- Packages

- Python: General programming language, NOT domain specific
 - Web development
 - Game programming
 - Desktop application
 - System programming
 - Hardware control (Robotics, DIY projects, Embedded system, etc)
 - Scientific computing
 - Etc...
 - Ton of library
 - Most scientific fields are moving to Python/R
- Matlab: Domain specific
 - Matrix computation → Easier for scientific computation, but lacks libraries

Some differences between Python and Matlab

	Python	Matlab
Variable	References, except for the native types (int, float, complex, string)	Values
Loop	for while, while-else	for-end while-end
Conditional	if, if-else, if-elif-else	if-end, if-elseif-end, if-elseif-else-end
Package management	import	Everything is accessible all the time
Code block	indentation	'end' and parantheses
Indexing	3:7 does not include 7	3:7 includes 7
Array	You must learn numpy. Manipulation is a bit cumbersome	Everything is matrix and operations are intuitive
Broadcasting	There is certain rules	Just works
Code blocking	Indentation	Needs "end" or parentheses
Other syntaxes	Sometimes quirky for scientific computation	Relatively simple, but not very powerful

	Python	Matlab
OOP	Fully supported	Add on (clunky)
Libraries	Maybe the 2 nd best for any scientific field	Lacking a lot
Difficulty	Very easy to learn	Easier for numerical computing. Not worth the effort as a general programming language
Fast?	OK	A bit faster for numerical analysis
Cool new features (e.g., AI)	Very good	OK
Should I learn?	Yes for everyone	Depends

Courses at MCDB

MCDB 108c

Dr. Matthieu Louis

Computational & Systems Biology

→ with Matlab

Description

Models of Biochemical and Cellular Systems. Introductory systems–biology approach to model the design and the function of biological systems. Students will develop an intuition about physical concepts that are fundamental to discuss how biological organisms acquire and process information from the environment. Those concepts and tools will cover probabilities and basic dynamical systems theory. Students will build models of processes of increasing complexity, ranging from viral dynamics, bacterial resistance to drugs, the maintenance of homeostatic equilibrium (trp operon), biological oscillators (mitotic clock) and genetic switches underlying cellular decisions (bacteriophage lambda and lac operon).

MCDB 172

Dr. Max Wilson

Computational & Systems Biology

→ with Python

Description

An introduction to mathematical models and computer simulations used to describe and understand time varying biological systems. Learning Objectives: Survey mathematical methods for describing the dependence on time of biological phenomena. Illustrate how to construct mathematical models to gain insights into complex biological systems. Develop working knowledge of a python code base that enables future evaluation of common classes of models applied to the study of biological dynamics.

Courses at Statistics → with R

PSTAT 10. Principles of Data Science with R

(5) STAFF

Prerequisite: *Math 2B or 3B with a minimum grade of C or better.*

Fundamentals of programming for data science using R. Descriptive statistics, distributions and graphics in R. Relational database management systems including the relational model, relational algebra, database design principles and data manipulation using SQL. An introduction to the concept of big data.

PSTAT 100. Data Science Concepts and Analysis

(4) STAFF

Prerequisite: *PSTAT 120A; CS 9 or CS 16; and Math 4A, all with letter grade C or better.*

Recommended Preparation: *PSTAT 120B recommended, but can be taken concurrently.*

Overview of data science key concepts and the use of tools for data retrieval, analysis, visualization, and reproducible research. Topics include an introduction to inference and prediction, principles of measurement, missing data, and notions of causality, statistical traps, and concepts in data ethics and privacy. Case studies illustrate the importance of domain knowledge.

Online Courses

The screenshot shows the Coursera interface for the 'Bioinformatics Specialization' course. The header includes the Coursera logo, an 'Explore' dropdown, a search bar with the placeholder 'What do you want to learn?', and links for 'Online Degrees' and 'Find your New Career'. The breadcrumb trail reads 'Browse > Health > Health Informatics'. The course title 'Bioinformatics Specialization' is prominently displayed, followed by a description: 'Journey to the Frontier of Computational Biology. Master bioinformatics software and computational approaches in modern biology.' The course has a 4.4-star rating from 921 reviews. The instructor is Pavel Pevzner, with a link to '+2 more instructors'. A white button labeled 'Enroll for Free' indicates the course 'Starts Mar 9' and that 'Financial aid available'. At the bottom, it states '49,057 already enrolled'. The background is a solid brown color.

Bioinformatics Specialization
Journey to the Frontier of Computational Biology. Master bioinformatics software and computational approaches in modern biology.
★★★★★ 4.4 921 ratings
Pavel Pevzner +2 more instructors
Enroll for Free
Starts Mar 9
Financial aid available
49,057 already enrolled

No specific language

The screenshot shows the Coursera interface for the 'Data Science: Statistics and Machine Learning Specialization' course. The header is identical to the first screenshot. The breadcrumb trail reads 'Browse > Data Science > Data Analysis'. The course title 'Data Science: Statistics and Machine Learning Specialization' is prominently displayed, followed by a description: 'Journey to the Frontier of Computational Biology. Master bioinformatics software and computational approaches in modern biology.' The course has a 4.6-star rating from 518 reviews. The instructor is Brian Caffo, PhD, with a link to '+2 more instructors'. A blue button labeled 'Enroll for Free' indicates the course 'Starts Mar 9' and that 'Financial aid available'. At the bottom, it states '29,784 already enrolled'. The background is a solid blue color.

Data Science: Statistics and Machine Learning Specialization
Journey to the Frontier of Computational Biology. Master bioinformatics software and computational approaches in modern biology.
★★★★★ 4.6 518 ratings
Brian Caffo, PhD +2 more instructors
Enroll for Free
Starts Mar 9
Financial aid available
29,784 already enrolled


with R

About Python

<https://openbookproject.net/thinkcs/python/english3e/index.html>

How to Think Like a Computer Scientist: Learning with Python 3 »

How to Think Like a Computer Scientist



Learning with Python 3 (RLE)

Version date: October 2012

by Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers

(based on 2nd edition by Jeffrey Elkner, Allen B. Downey, and Chris Meyers)

Corresponding author: p.wentworth@ru.ac.za

Source repository is at <https://code.launchpad.net/~thinkcs-py-rle-team/thinkcs-py/thinkcs-py3-rle>

For offline use, download a zip file of the html or a pdf version (the pdf is updated less often) from <http://www.ict.ru.ac.za/Resources/cspw/thinkcs-py3/>

- [Search Page](#)
- [Copyright Notice](#)
- [Foreword](#)
- [Preface](#)
- [Preface-3](#) *This Rhodes Local Edition (RLE) of the book*
- [Contributor List](#)
- [Chapter 1](#) *The way of the program*
- [Chapter 2](#) *Variables, expressions, and statements*
- [Chapter 3](#) *Hello, little turtles!*
- [Chapter 4](#) *Functions*
- [Chapter 5](#) *Conditionals*
- [Chapter 6](#) *Fruitful functions*

<http://www.openbookproject.net/books/pythonds/index.html>

RSI Problem Solving with Algorithms and Data Structures Chapters ▾

Problem Solving with Algorithms and Data Structures using Python

By Brad Miller and David Ranum, Luther College (as remixed by Jeffrey Elkner)

- 1. Introduction
 - 1.1. Objectives
 - 1.2. Getting Started
 - 1.3. What Is Computer Science?
 - 1.4. What Is Programming?
 - 1.5. Why Study Data Structures and Abstract Data Types?
 - 1.6. Why Study Algorithms?
 - 1.7. Review of Basic Python
 - 1.8. Getting Started with Data
 - 1.8.1. Built-in Atomic Data Types
 - 1.8.2. Built-in Collection Data Types
 - 1.9. Input and Output
 - 1.9.1. String Formatting
 - 1.10. Control Structures
 - 1.11. Exception Handling
 - 1.12. Defining Functions
 - 1.13. Object-Oriented Programming in Python: Defining Classes
 - 1.13.1. A **Fraction** Class
 - 1.13.2. Inheritance: Logic Gates and Circuits
 - 1.14. Summary
 - 1.15. Key Terms
 - 1.16. Discussion Questions
 - 1.17. Programming Exercises
- 2. Analysis
 - 2.1. Objectives
 - 2.2. What Is Algorithm Analysis?
 - 2.3. Big-O Notation

What should be your NEXT language?

- Web development → HTML5, Java script, Ruby, Java
- Game programming → C#, C++
- Desktop application → Java, C++, C#
- System programming → Go, C/C++
- Hardware control (Robotics, DIY projects, Embedded system, etc) → C/C++, Rust
- Scientific computing → **R**, **Python**, *Matlab*, Julia, C/C++