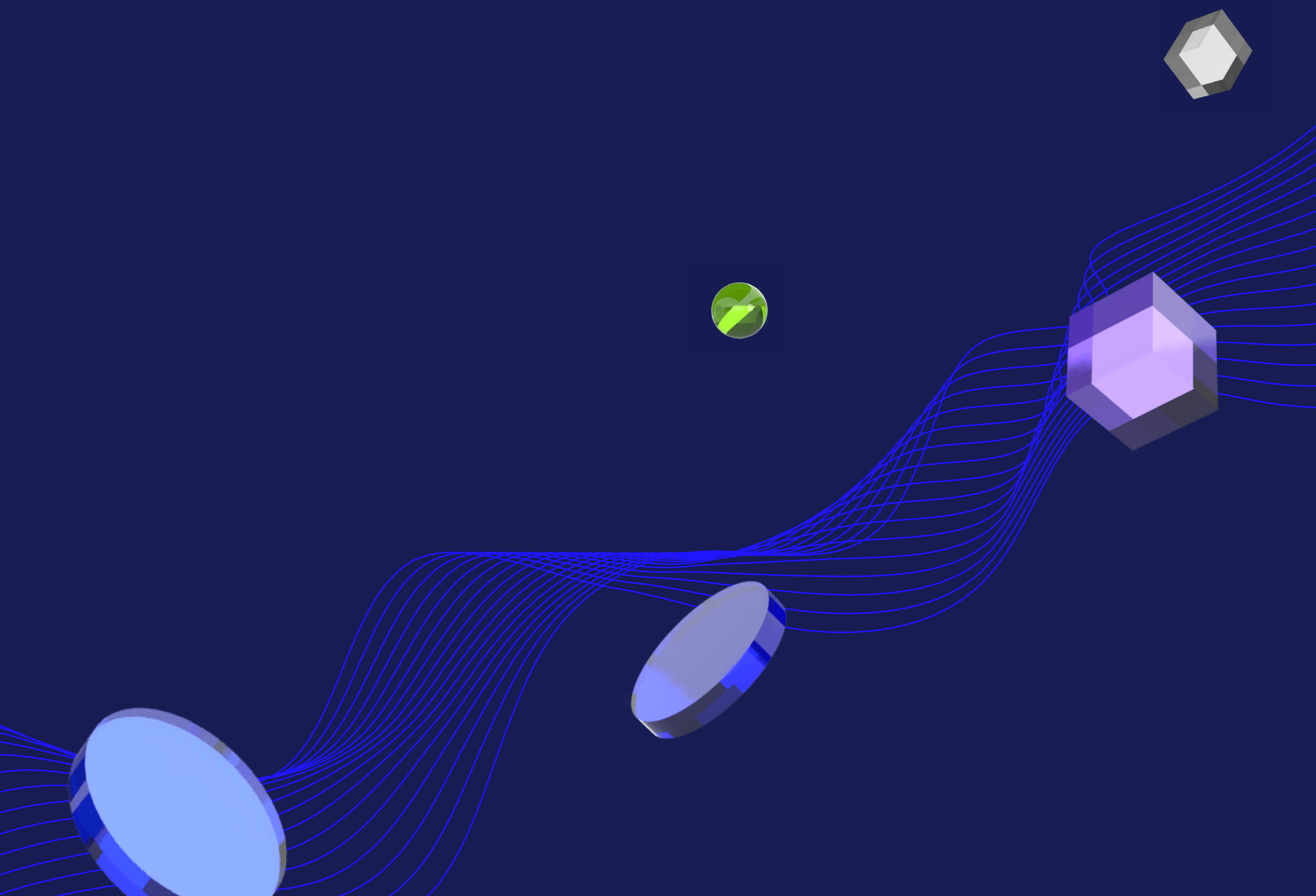
INDIVIDUAL LEARNERS

UDACITY

SCHOOL OF DATA SCIENCE

# Programming for Data Science with Python

Nanodegree Program Syllabus



# Overview

The Programming for Data Science with Python Nanodegree program offers learners the opportunity to learn the most important programming languages used by data scientists today. Get started with the fascinating field of data science and learn Python, SQL, terminal, and Git with the help of experienced instructors. Learners will emerge prepared to tackle real world data analysis problems.

### Built in collaboration with:

MODE

# Program information

### Estimated Time

3 months at 10hrs/week*

### Skill Level

Beginner

### Prerequisites

A well-prepared learner should have the ability to perform basic operations on your computer like opening files and folders, opening applications, and copying/pasting. Learners should also be able to read, write, and listen in English.

### Required Hardware/Software

Learners need access to the internet and a 64-bit computer.

*The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 5-10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

Course 1

# Introduction to SQL

Learn SQL fundamentals such as JOINs, aggregations, and subqueries. Learn how to use SQL to answer complex business problems.

Course Project

## Investigate a Database

In this project, learners will work with a relational database while working with PostgreSQL. They'll complete the entire data analysis process, starting by posing a question, running appropriate SQL queries to answer questions, and finishing by sharing findings.

Lesson 1

### Basic SQL

- Write common SQL commands including SELECT, FROM, and WHERE.
- Use logical operators like LIKE, AND, and OR.

Lesson 2

### SQL Joins

- Write JOINs in SQL, as you are now able to combine data from multiple sources to answer more complex business questions.
- Understand different types of JOINs and when to use each type.

Lesson 3

### SQL Aggregations

- Write common aggregations in SQL including COUNT, SUM, MIN, and MAX.
- Write CASE and DATE functions, as well as work with NULLs.

Lesson 4

### Advanced SQL Queries

- Use subqueries, also called CTEs, in a number of different situations.
- Use other window functions including RANK, NTILE, LAG, LEAD new functions along with partitions to complete complex tasks.

Course 2

# Introduction to Python Programming

Learn Python programming fundamentals such as data structures, variables, loops, and functions. Learn to work with data using libraries like NumPy and Pandas.

Course Project

## Explore US Bikeshare Data

Learners will use Python to answer analytical questions about bikeshare trip data collected from three US cities. They will write code to collect the data, compute descriptive statistics, and create an interactive experience in the terminal that presents the answers to your questions.

Lesson 1

### Why Python Programming

- Gain an overview of what you'll be learning and doing in the course.
- Understand why you should learn programming with Python.

Lesson 2

## Data Types & Operators

- Represent data using Python's data types: integers, floats, booleans, strings, lists, tuples, sets, dictionaries, compound data structures.
- Perform computations and create logical statements using Python's operators: arithmetic, assignment, comparison, logical, membership, and identity.
- Declare, assign, and reassign values using Python variables.
- Modify values using built-in functions and methods.
- Practice whitespace and style guidelines.

Lesson 3

## Control FLow

- Write conditional expressions using if statements and boolean expressions to add decision making to your Python programs.
- Use for and while loops along with useful built-in functions to iterate over and manipulate lists, sets, and dictionaries.
- Skip iterations in loops using break and continue.
- Condense for loops to create lists efficiently with list comprehensions.

Lesson 4

## Functions

- Define your own custom functions.
- Create and reference variables using the appropriate scope.
- Add documentation to functions using docstrings.
- Define lambda expressions to quickly create anonymous functions.
- Use iterators and generators to create streams of data.

Lesson 5

## Scripting

- Install Python 3 and set up your programming environment.
- Run and edit Python scripts.
- Interact with raw input from users.
- Identify and handle errors and exceptions in your code.
- Open, read, and write to files.
- Find and use modules in Python Standard Library and third-party libraries.
- Experiment in the terminal using a Python interpreter.

Lesson 6

### Numpy

- Create, access, modify, and sort multidimensional NumPy arrays (ndarrays).
- Load and save ndarrays.
- Use slicing, boolean indexing, and set operations to select or change subsets of an ndarray.
- Understand difference between a view and a copy of ndarray.
- Perform element-wise operations on ndarrays.
- Use broadcasting to perform operations on ndarrays of different sizes.

Lesson 7

### Pandas

- Create, access, and modify the main objects in Pandas, Series, and DataFrames.
- Perform arithmetic operations on Series and DataFrames.
- Load data into a DataFrame.
- Deal with Not a Number (NaN) values.

Course 3

# Introduction to Version Control

Learn how to use version control and share work with other people in the data science industry.

Course Project

## Post Your Work on Github

Learn the important tools that all programmers use. First, get an introduction to working in the terminal. Next, learn to use Git and Github to manage versions of a program and collaborate with others on programming projects. Learners will post two different versions of a Jupyter Notebook capturing learnings from the course, and add commits to their project Git repository.

Lesson 1

### Shell Workshop

- The Unix shell is a powerful tool for developers of all sorts. Get a quick introduction to the basics of using it on your computer.

Lesson 2

### Purpose & Terminology

- Learn why developers use version control and discover ways you use version control in your daily life.
- Get an overview of essential Git vocabulary.
- Configure Git using the command line.

Lesson 3

### Create a Git Repo

- Create your first Git repository with git init.
- Copy an existing Git repository with git clone.
- Review the current state of a repository with the powerful git status.

Lesson 4

### Review a Repo's History

- Review a repo's commit history git log.
- Customize git log's output using command line flags in order to reveal more (or less) information about each commit.
- Use the git show command to display just one commit.

Lesson 5

### Add Commits to a Repo

- Master the Git workflow and make commits to an example project.
- Use git diff to identify what parts of a file have been changed in a commit.
- Learn how to mark files as "untracked" using .gitignore.

Lesson 6

### Tagging, Branching & Merging

- Explore tagging, branching, and merging.
- Organize your commits with tags and branches.
- Jump to particular tags and branches using git checkout.
- Learn how to merge together changes on different branches and crush those pesky merge conflicts.

Lesson 7

## Undoing Changes

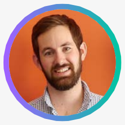
- Learn how and when to edit or delete an existing commit.
- Use git commit’s -amend flag to alter the last commit.
- Use git reset and git revert to undo and erase commits.

# Meet your instructors.

## Josh Bernhard

**Data Scientist at Nerd Wallet**

Josh has been sharing his passion for data for nearly a decade at all levels of university, and as lead data science instructor at Galvanize. He's used data science for work ranging from cancer research to process automation.



## Derek Steer

**CEO at Mode**

Derek is the CEO of Mode Analytics. He developed an analytical foundation at Facebook and Yammer and is passionate about sharing it with future analysts. He authored SQL School and is a mentor at Insight Data Science.

## Juno Lee

**Curriculum Lead at Udacity**

Juno is the curriculum lead for the School of Data Science. She has been sharing her passion for data and teaching, building several courses at Udacity. As a data scientist, she built recommendation engines, computer vision and NLP models, and tools to analyze user behavior.

## Richard Kalehoff

**Course Developer**

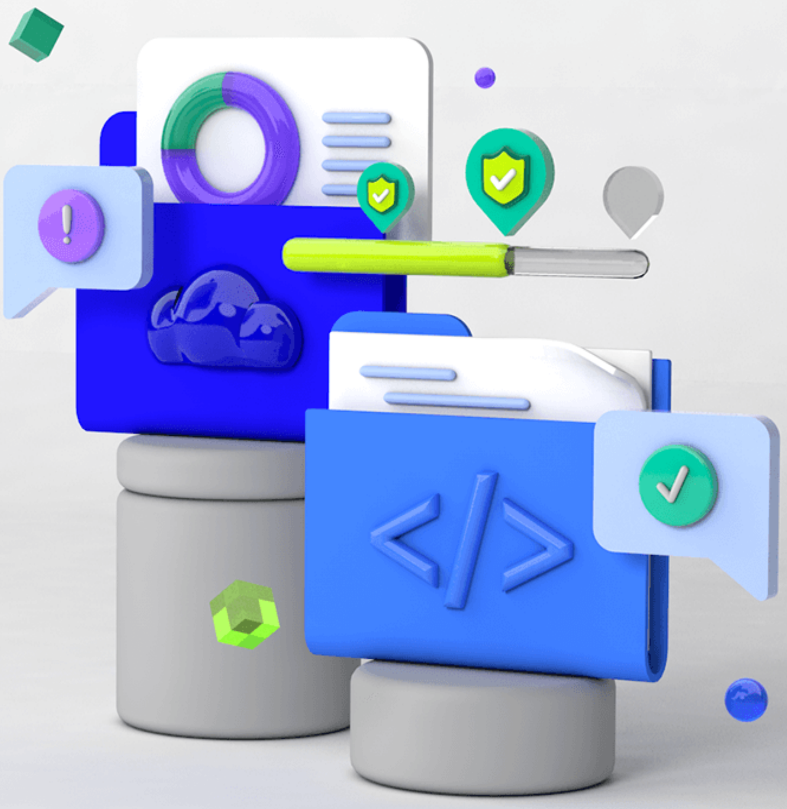
Richard is a course developer with a passion for teaching. He has a degree in computer science, and first worked for a nonprofit doing everything from front end web development, to backend programming, to database and server management.

### Karl Krueger

**Command Line Instructor**

Before joining Udacity, Karl was a site reliability engineer (SRE) at Google for eight years, building automation and monitoring to keep the world’s busiest web services online.

# Udacity’s learning experience



### Hands-on Projects

Open-ended, experiential projects are designed to reflect actual workplace challenges. They aren’t just multiple choice questions or step-by-step guides, but instead require critical thinking.

### Quizzes

Auto-graded quizzes strengthen comprehension. Learners can return to lessons at any time during the course to refresh concepts.

### Knowledge

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover how to solve the challenges that you encounter.

### Custom Study Plans

Create a personalized study plan that fits your individual needs. Utilize this plan to keep track of movement toward your overall goal.

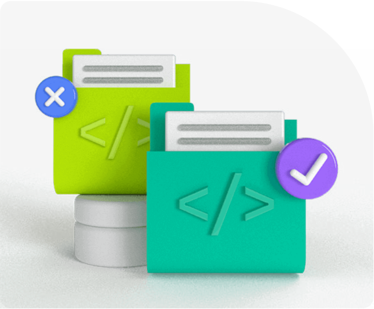
### Workspaces

See your code in action. Check the output and quality of your code by running it on interactive workspaces that are integrated into the platform.

### Progress Tracker

Take advantage of milestone reminders to stay on schedule and complete your program.

# Our proven approach for building job-ready digital skills.



### Experienced Project Reviewers

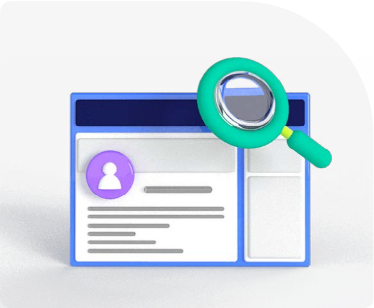
## Verify skills mastery.

- Personalized project feedback and critique includes line-by-line code review from skilled practitioners with an average turnaround time of 1.1 hours.
- Project review cycle creates a feedback loop with multiple opportunities for improvement—until the concept is mastered.
- Project reviewers leverage industry best practices and provide pro tips.

### Technical Mentor Support

## 24/7 support unblocks learning.

- Learning accelerates as skilled mentors identify areas of achievement and potential for growth.
- Unlimited access to mentors means help arrives when it's needed most.
- 2 hr or less average question response time assures that skills development stays on track.



### Personal Career Services

## Empower job-readiness.

- Access to a Github portfolio review that can give you an edge by highlighting your strengths, and demonstrating your value to employers.*
- Get help optimizing your LinkedIn and establishing your personal brand so your profile ranks higher in searches by recruiters and hiring managers.

### Mentor Network

## Highly vetted for effectiveness.

- Mentors must complete a 5-step hiring process to join Udacity's selective network.
- After passing an objective and situational assessment, mentors must demonstrate communication and behavioral fit for a mentorship role.
- Mentors work across more than 30 different industries and often complete a Nanodegree program themselves.

*Applies to select Nanodegree programs only.

UDACITY

Learn more at

www.udacity.com/online-learning-for-individuals →

12.16.22 | V1.0