UDACITY

NANODEGREE PROGRAM SYLLABUS

Intermediate Python

UDACITY

Overview

Python is the future of computer programming. It's the language that powers machine learning and AI, two technologies that are on the forefront of digital transformation. Plus, it is an essential backend language for web application development. Learners who take this class will be prepared to work in a number of different high growth fields that are experiencing significant demand for talent.

This Nanodegree program teaches intermediate-level skills for programming with the Python language. The training wheels will come off and learners will free analyze data and build the backend of web applications themselves. They'll acquire techniques like Python objects, object-oriented programming, debugging and control flow. This course is ideal for developers interested in using Python to build more complex algorithms with greater capabilities (i.e. image resizing, document templates, word counts, name entity recognition on a webpage, etc.) in preparation for a variety of different roles spanning fields like data science, AI and software engineering.

Estimated Time:
2 months at
10hrs/week*

Prerequisites:
- Write and run basic programming scripts
- Read basic Python syntax
- Basic familiarity with python and object-oriented programming

Flexible Learning:
Self-paced

Need Help?
udacity.com/advisor
Discuss this program with an enrollment advisor.

*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 10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

UDACITY

Course 1: Advanced Python Topics

Python is a particularly powerful programming language, and the features it provides permit programmers to produce practical programs. Learn how to elevate your Python language abilities, mastering a myriad of modern subject matter. Framed by fundamentals, you'll first find a foundation in Python's methods to describe data. You'll dig deeper into functions and functional design, and create strategies for solving problems. You'll investigate the ins-and-outs of objects and object-based design, obtaining order from the interconnected ideas captured within class objects and instance objects. Finally, you'll have an opportunity to fuse Python with external files, culminating in the creation of complete codebases that can crunch numbers or protect the planet from peril.

Project 1
Near-Earth Objects

Python can take you out of this world, and this project is no different! You will produce a program that can inspect and query close approaches of near-Earth objects — moments in the past (or future) at which an asteroid (or comet-like object in space) passes quite close to Earth. You'll read 200 years of data from CSV and JSON files into Python models, and build a database capable of answering questions such as "when does Halley's comet pass by Earth?" and "what are the next ten close approaches of big, hazardous asteroids whose orbit takes them exceptionally close to Earth." Finally, you'll save your results back to CSV or JSON files. By completing this project, you'll demonstrate an ability to represent data in Python, transform that data using principles of function and object-based design, and connect to external data sources.

LEARNING OUTCOMES		
LESSON ONE	**Representing Data**	• Evaluate intrinsic or prescribed characteristics of structured data. • Understand Python's approach to objects, names and namespaces. • Explore fundamental types, such as booleans, numbers and text. • Explore collections, such as lists, tuples, strings, dictionaries and sets.
LESSON TWO	**Functions and Functional Programming**	• Trace the details of function execution. • Create simple function interfaces using advanced arguments types, including keyword arguments and variadic arguments. • Create functional programs, using map/filter, lambdas, iterators and generators. • Create decorators, high-level tools to transform functional behavior.

UDACITY

LEARNING OUTCOMES		
LESSON THREE	**Object-Oriented Programming**	• Trace the details of instantiation and attribute resolution on class objects and instance objects. • Create classes with custom methods, including initializers and decorated properties. • Analyze object-based design patterns, including polymorphism (through magic methods) and inheritance. • Handle and produce errors (builtin or custom) to process or signal failure.
LESSON FOUR	**File I/O**	• Understand the principles of files and file systems, in order to open files for reading or writing. • Create programs that can read data from or write data to a plain text file. • Create programs that can read or write JSON data. • Create programs that can read or write CSV data.
LESSON FIVE	**Project: Near-Earth Objects**	• Build a database to inspect and query properties of close approaches of near-Earth objects by reading data into Python, transforming the data with functional and object-based design principles, and saving the results back to a file.

UDACITY

Course 2: Large Codebases with Libraries

Python can be used to develop very large systems to solve complex problems. Learn how you can write, structure and extend your code to be able to support developing these large systems at scale. Understand how you can leverage open source libraries to quickly add advanced functionality to your code and how you can package your code into libraries of your own. Apply Object Oriented Programming to ensure that your code remains modular, clear and understandable. Honing these skills is the foundation for building codebases that are maintainable and efficient as they grow to tens of thousands of lines.

Project 2
Meme Generator

Python is well suited for solving both web and data problems. You will build a service that demonstrates an understanding of both of these domains. First, you'll import quote data from many different data types (PDF, DOCX, CSV, TXT). Then, you'll demonstrate an understanding of the Strategy Object design pattern to write clean, modular code to handle these different file types. Then, you'll resize images and overlay the quotes onto the resized graphics. Finally, you'll practice making your service available for others to use as a command line utility and as a deployable web service.

LEARNING OUTCOMES

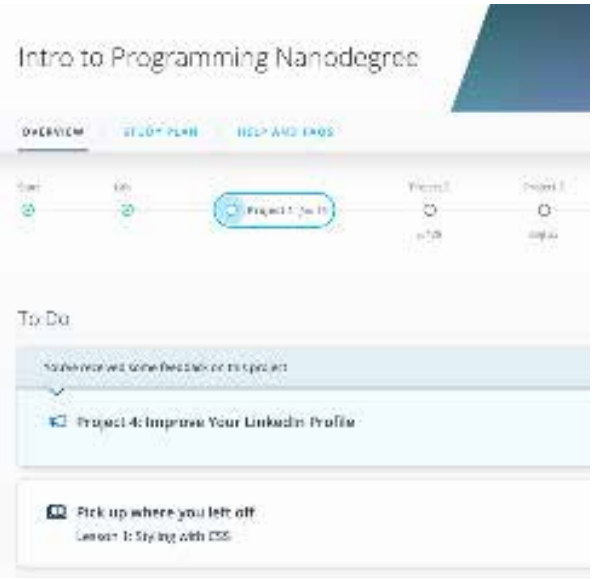
LESSON ONE	**Foundations**	• Review PEP standards to write clear, compliant code. • Practice implementing Object Oriented Programming in python. • Understand core pythonic principles to write code that can scale.
LESSON TWO	**Building Modules**	• Understand how you can write modular code building blocks to reuse functional units of code. • Learn advanced Object Oriented Programming concepts including Inheritance and Abstraction.
LESSON THREE	**Using Libraries**	• Install and use open source libraries to solve complex problems. • Explore the common use cases of open source libraries available on the Python Package Index (PyP). • Learn how to use Virtual Environments to maintain clear dependency states during development. • Expand on Object Oriented design using the advanced Strategy Object design pattern.

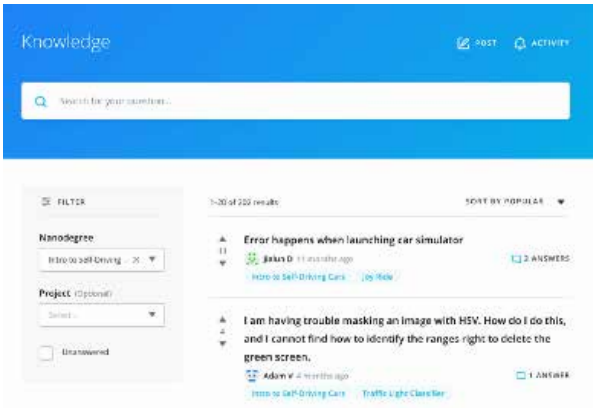
UDACITY

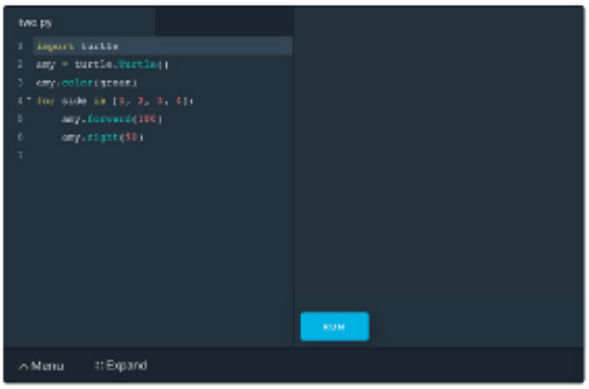
LEARNING OUTCOMES		
LESSON FOUR	**Python in Systems**	• Design complex systems of code that communicate across the operating system interfaces. • Understand how you can create Command Line tools using your Python scripts. • Learn how to consume other Command Line tools within your Python scripts.
LESSON FIVE	**Python for Web**	• Connect your code to systems that expand beyond a single computer (the internet). • Learn how to download and use data from web services using requests. • Understand the basics of backend development by making a Python service available from the web using Flask.
LESSON SIX	**Project: Meme Generator**	• Build a meme generator that overlays quotes on images by applying advanced Pythonic Object Oriented Principles and using complex libraries to interact with command line tools to process data and make your service available on the web.

UDACITY

Our Classroom Experience







REAL-WORLD PROJECTS

Build your skills through industry-relevant projects. Get personalized feedback from our network of 900+ project reviewers. Our simple interface makes it easy to submit your projects as often as you need and receive unlimited feedback on your work.

KNOWLEDGE

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students and discover in real-time how to solve the challenges that you encounter.

STUDENT HUB

Leverage the power of community through a simple, yet powerful chat interface built within the classroom. Use Student Hub to connect with your technical mentor and fellow students in your Nanodegree program.

WORKSPACES

See your code in action. Check the output and quality of your code by running them on workspaces that are a part of our classroom.

QUIZZES

Check your understanding of concepts learned in the program by answering simple and auto-graded quizzes. Easily go back to the lessons to brush up on concepts anytime you get an answer wrong.

CUSTOM STUDY PLANS

Work with a mentor to create a custom study plan to suit your personal needs. Use this plan to keep track of your progress toward your goal.

PROGRESS TRACKER

Stay on track to complete your Nanodegree program with useful milestone reminders.

UDACITY

Learn with the Best

Sam Redmond

CO-FOUNDER, HEDRON VISION

Sam holds a BS in Math and MS in CS from Stanford, where he created CS 41: The Python Programming Language and lectured for four years. Currently, he's cofounding Hedron Vision, developing headset-free holographic display technologies.

Gabriel Ruttner

CTO ,URSA & TECH ADVISOR
FOR START-UPS

Gabe is the CTO at Ursa & Tech Advisor for Start-Ups. Gabe has expertise in building cloud-based machine learning and natural language processing services at early stage tech companies. He holds technical degrees from Cornell University and Stony Brook University

UDACITY

All Our Nanodegree Programs Include:

EXPERIENCED PROJECT REVIEWERS

REVIEWER SERVICES

- Personalized feedback & line by line code reviews
- 1600+ Reviewers with a 4.85/5 average rating
- 3 hour average project review turnaround time
- Unlimited submissions and feedback loops
- Practical tips and industry best practices
- Additional suggested resources to improve

TECHNICAL MENTOR SUPPORT

MENTORSHIP SERVICES

- Questions answered quickly by our team of technical mentors
- 1000+ Mentors with a 4.7/5 average rating
- Support for all your technical questions

PERSONAL CAREER SERVICES

CAREER SUPPORT

- Resume support
- Github portfolio review
- LinkedIn profile optimization

UDACITY

FAQs + Contact Info

WHY SHOULD I ENROLL?

Python is consistently ranked as one of the most popular and in-demand programming languages and its importance in high growth fields like Machine Learning and Data Science indicates that won't change any time soon. The Intermediate Python Nanodegree program gives you the tools to power your career in these and other exciting fields that are reshaping industries such as Financial Services, Automotive and Telecommunications. With each lesson, you'll be more equipped to leverage the capabilities of Python and streamline the functionality of applications that perform complex tasks, such as classifying files, data mining a webpage and more.

WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

As noted above, developers with Python skills are in demand in a variety of roles and industries. Specific roles that professionals with python experience typically hold Include Data Engineer, QA Engineer, Full-stack Developer, Back-end Developer, Web Developer, Data Analyst and more.

HOW DO I KNOW IF THIS PROGRAM IS RIGHT FOR ME?

This program is designed for anyone who wants to further develop their python skills to build more complex projects and algorithms with greater capabilities (i.e. image resizing, document templates, word counts, name entity recognition on a webpage, etc.) in preparation for a variety of roles in Software Engineering, Data Science, AI and beyond. If you want to become proficient in Python to grow in your current role or transition to a new one, the Intermediat Python Nanodegree Program is a great fit for you.

ENROLLMENT AND ADMISSION

DO I NEED TO APPLY? WHAT ARE THE ADMISSION CRITERIA?

There is no application. This Nanodegree program accepts everyone, regardless of experience and specific background.

WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

A well prepared student can:

- Understand the basics of object-oriented programming.
- Read basic Python syntax, including using white space in Python.
- Distinguish between object types like integers and strings in scripts.
- Use Python to build basic algorithms for simple programs (i.e. a one-dimensional game like rock, paper, scissors) and scripts that automate common tasks (i.e. renaming files).
- Write and run basic programming scripts in a terminal that include function definitions and loops.

FAQs + Contact Info cont.

IF I DO NOT MEET THE REQUIREMENTS TO ENROLL, WHAT SHOULD I DO?

Udacity offers numerous programs that can help you prepare for the Intermediate Python Nanodegree Program. The **Intro to Programming Nanodegree program** provides a thorough introduction to Python fundamentals alongside courses in HTML, CSS and JavaScript. Our free **Introduction to Python Programming** course is a great place to learn Python fundamentals, and the **Programming Languages** course provides an overview of the concepts that underpin many languages.

TUITION AND TERM OF PROGRAM

HOW IS THIS NANODEGREE PROGRAM STRUCTURED?

The Intermediate Python Nanodegree program is comprised of content and curriculum to support two projects. We estimate that students can complete the program in two months, working five to ten hours per week. Each project will be reviewed by the Udacity reviewer network. Feedback will be provided, and if you do not pass the project, you will be asked to resubmit the project until it passes.

HOW LONG IS THIS NANODEGREE PROGRAM?

You will have access to this Nanodegree program for as long as your subscription remains active. The estimated time to complete this program can be found on the webpage and in the syllabus, and is based on the average amount of time we project that it takes a student to complete the projects and coursework. See the **Terms of Use** and **FAQs** for other policies regarding the terms of access to our Nanodegree programs.

CAN I GET A REFUND?

Please see the Udacity Program **Terms of Use** and **FAQs** for policies on enrollment in our programs.

SOFTWARE AND HARDWARE - WHAT DO I NEED FOR THIS PROGRAM?

WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?

There are no software and version requirements to complete this Nanodegree program. All coursework and projects can be done via Student Workspaces in the Udacity online classroom.