UDACITY
FOR ENTERPRISE

THE SCHOOL OF PROGRAMMING AND DEVELOPMENT

Intermediate Python

NANODEGREE SYLLABUS

Overview

Intermediate Python Nanodegree Degree

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.

Program Information

TIME
2 months at 10hrs/week*

LEVEL
Practitioner

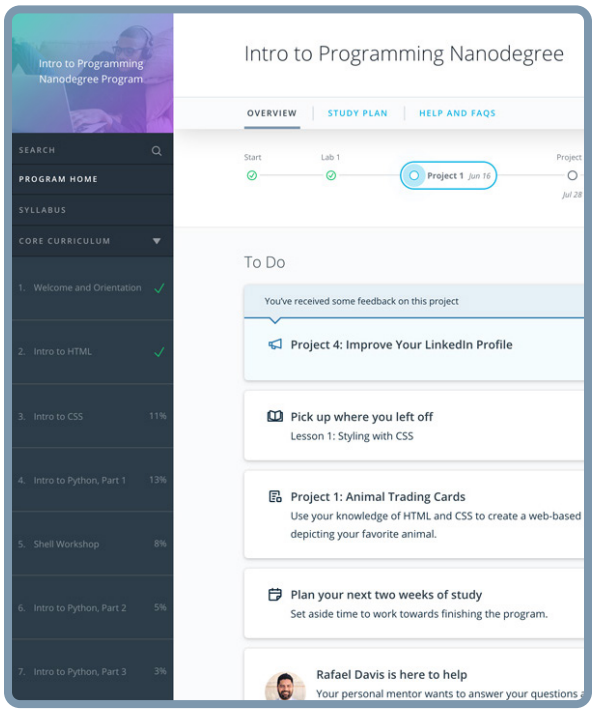
PREREQUISITES
Write and run basic programming scripts. Read basic Python syntax. Basic familiarity with Python and object-oriented programming.

HARDWARE/SOFTWARE REQUIRED
Access to the internet and a 64-bit computer.

LEARN MORE ABOUT THIS NANODEGREE
Contact us at enterpriseNDs@udacity.com.

*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.

Our Classroom Experience



REAL-WORLD PROJECTS

Learners build new skills through industry-relevant projects and receive personalized feedback from our network of 900+ project reviewers. Our simple user interface makes it easy to submit projects as often as needed and receive unlimited feedback.

KNOWLEDGE

Answers to most questions can be found with Knowledge, our proprietary wiki. Learners can search questions asked by others and discover in real-time how to solve challenges.

LEARNER HUB

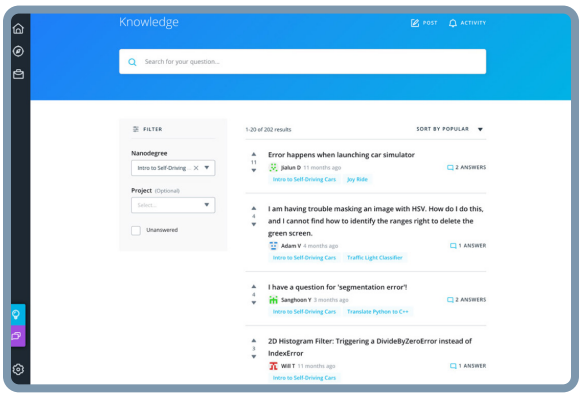
Learners leverage the power of community through a simple, yet powerful chat interface built within the classroom. Learner Hub connects learners with their technical mentor and fellow learners.

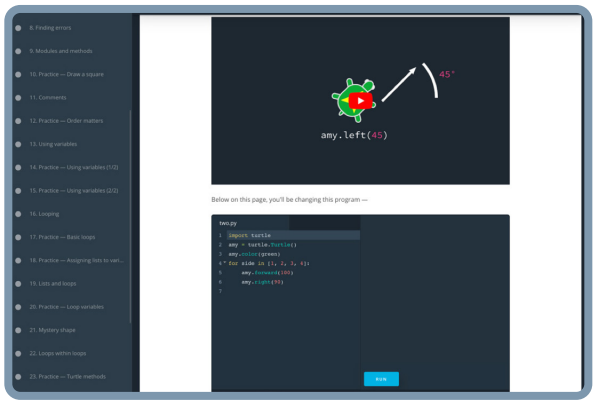
WORKSPACES

Learners can check the output and quality of their code by testing it on interactive workspaces that are integrated into the classroom.

QUIZZES

Understanding concepts learned during lessons is made simple with auto-graded quizzes. Learners can easily go back and brush up on concepts at anytime during the course.





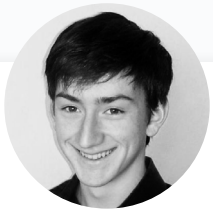
CUSTOM STUDY PLANS

Mentors create a custom study plan tailored to learners' needs. This plan keeps track of progress toward learner goals.

PROGRESS TRACKER

Personalized milestone reminders help learners stay on track and focused as they work to complete their Nanodegree program.

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

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: 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.

LESSON TITLE	LEARNING OUTCOMES
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.

Course 1: Advanced Python Topics, cont.

LESSON TITLE	LEARNING OUTCOMES
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.
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.
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.
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.

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: 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.

LESSON TITLE	LEARNING OUTCOMES
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.
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.
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.

Course 2: Large Codebases with Libraries, cont.

LESSON TITLE	LEARNING OUTCOMES
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.
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.
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.

Our Nanodegree Programs Include:

Pre-Assessments

Our in-depth workforce assessments identify your team's current level of knowledge in key areas. Results are used to generate custom learning paths designed to equip your workforce with the most applicable skill sets.

Dashboard & Progress Reports

Our interactive dashboard (enterprise management console) allows administrators to manage employee onboarding, track course progress, perform bulk enrollments and more.

Industry Validation & Reviews

Learners' progress and subject knowledge is tested and validated by industry experts and leaders from our advisory board. These in-depth reviews ensure your teams have achieved competency.

Real World Hands-on Projects

Through a series of rigorous, real-world projects, your employees learn and apply new techniques, analyze results, and produce actionable insights. Project portfolios demonstrate learners' growing proficiency and subject mastery.

Our Review Process

Real-life Reviewers for Real-life Projects

Real-world projects are at the core of our Nanodegree programs because hands-on learning is the best way to master a new skill. Receiving relevant feedback from an industry expert is a critical part of that learning process, and infinitely more useful than that from peers or automated grading systems. Udacity has a network of over 900 experienced project reviewers who provide personalized and timely feedback to help all learners succeed.

Vaibhav
UDACITY LEARNER

"I never felt overwhelmed while pursuing the Nanodegree program due to the valuable support of the reviewers, and now I am more confident in converting my ideas to reality."

now at
CODING VISIONS INFOTECH

All Learners Benefit From:

- Line-by-line feedback for coding projects
- Industry tips and best practices
- Advice on additional resources to research
- Unlimited submissions and feedback loops

How it Works

Real-world projects are integrated within the classroom experience, making for a seamless review process flow.

- Go through the lessons and work on the projects that follow
- Get help from your technical mentor, if needed
- Submit your project work
- Receive personalized feedback from the reviewer
- If the submission is not satisfactory, resubmit your project
- Continue submitting and receiving feedback from the reviewer until you successfully complete your project

About our Project Reviewers

Our expert project reviewers are evaluated against the highest standards and graded based on learners' progress. **Here's how they measure up to ensure your success.**

900+

Expert Project Reviewers

Are hand-picked to provide detailed feedback on your project submissions.

1.8M

Projects Reviewed

Our reviewers have extensive experience in guiding learners through their course projects.

3

Hours Average Turnaround

You can resubmit your project on the same day for additional feedback.

4.85 /5

Average Reviewer Rating

Our learners love the quality of the feedback they receive from our experienced reviewers.

UDACITY
FOR ENTERPRISE

2440 W El Camino Real, #101
Mountain View, CA 94040, USA - HQ

For more information visit: www.udacity.com/enterprise