



AI Programming with Python Nanodegree Syllabus

Programming Skills, Linear
Algebra, Neural Networks



UDACITY
FOR ENTERPRISE



Table of Contents

Before You Start	3
Contact Info	3
Nanodegree Program Info	3
Supporting Lesson Content: Introduction to Python	4
Supporting Lesson Content: Numpy, Pandas, and Matplotlib	5
Supporting Lesson Content: Linear Algebra Essentials	5
Supporting Lesson Content: Neural Networks	6
Project: Image Classifier	7





Before You Start

Prerequisites: Formal prerequisites include basic knowledge of algebra and calculus. Basic programming knowledge will help to quickly pick up AI's essential coding concepts. Additional skills recommended include, basic calculus knowledge, such as how to calculate derivatives, basic computer skills like managing files, navigating the Internet, and running programs, and basic algebra and programming knowledge.

Educational Objectives: This program will teach you all the tools needed to succeed in your journey into the world of AI.

Contact Info

While going through the program, if you have questions about anything, you can reach us at enterprise-support@udacity.com. For help from Udacity Mentors and your peers, visit the Udacity Classroom.

Nanodegree Program Info

This program covers all the fundamental building blocks you will need to master to become an AI practitioner. Specifically, we will focus on programming skills with Python, linear algebra, and even dive into neural networks and deep learning.

One of our main goals at Udacity is to help you create a portfolio. Building a project is one of the best ways both to test the skills you've acquired and to demonstrate your newfound abilities. At the end of this Nanodegree program, you'll have the opportunity to prove your skills by building an image classifier.

In the sections below, you'll find a detailed description of the project along with the course material that presents the skills required to complete the project.

LENGTH OF PROGRAM*: 4 months

FREQUENCY OF CLASSES: Self-paced

INSTRUCTIONAL TOOLS AVAILABLE: Video lectures, Text instructions, Quizzes, Study Groups, Knowledge, Project Reviews

*This is a self-paced program and the length is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. Actual hours may vary.

Supporting Lesson Content: Introduction to Python:

Lesson Title	Learning Outcomes
WHY PYTHON PROGRAMMING	<ul style="list-style-type: none">• Learn why we program.• Prepare for the course ahead with a detailed topic overview.• Understand how programming in Python is unique
DATA TYPES AND OPERATORS	<ul style="list-style-type: none">• Understand how data types and operators are the building blocks programming in Python• Use the following data types: integers, floats, booleans, strings, lists, tuples, sets, dictionaries• Use the following operators: arithmetic, assignment, comparison, logical, membership, identity
CONTROL FLOW	<ul style="list-style-type: none">• Implement decision making in your code with conditionals• Repeat code with for and while loops.• Exit a loop with break and skip an iteration of a loop with continue• Use helpful built-in functions like zip and enumerate.• Construct lists in a natural way with list comprehensions.
FUNCTIONS	<ul style="list-style-type: none">• Write your own functions to encapsulate a series of commands• Understand variable scope, i.e., which parts of a program variables can be referenced from• Make functions easier to use with proper documentation• Use lambda expressions, iterators, and generators
SCRIPTING	<ul style="list-style-type: none">• Write and run scripts locally on your computer• Work with raw input from users• Read and write files, handle errors, and import local scripts• Use modules from the Python standard library and from third-party libraries• Use online resources to help solve problems
LAB	<ul style="list-style-type: none">• Learn how to use a pre-trained image classifier to write a script that identifies dog breeds!

Supporting Lesson Content: Numpy, Pandas, and Matplotlib

Lesson Title	Learning Outcomes
ANACONDA	<ul style="list-style-type: none">Learn how to use Anaconda to manage packages and environments for use with Python
JUPYTER NOTEBOOKS	<ul style="list-style-type: none">Learn how to use Jupyter Notebooks to create documents combining code, text, images, and more
NUMPY BASICS	<ul style="list-style-type: none">Learn the value of NumPy and how to use it to manipulate data for AI problems
PANDAS BASICS	<ul style="list-style-type: none">Learn to use Pandas to load and process data for machine learning problems
MATPLOTLIB BASICS	<ul style="list-style-type: none">Learn how to use Matplotlib to choose appropriate plots for one and two variables based on the types of data you have.
LAB	<ul style="list-style-type: none">Use NumPy to mean normalize an ndarray and separate it into several smaller ndarrays. Use Pandas to plot and get statistics from stock data.

Supporting Lesson Content: Linear Algebra Essentials

Lesson Title	Learning Outcomes
LINEAR ALGEBRA ESSENTIALS	<ul style="list-style-type: none">Take a sneak peek into the beautiful world of Linear Algebra and learn why it is such an important mathematical tool
VECTORS	<ul style="list-style-type: none">Learn about the basic building block of Linear Algebra
LINEAR COMBINATION	<ul style="list-style-type: none">Learn how to scale and add vectors and how to visualize it
LINEAR TRANSFORMATION AND MATRICES	<ul style="list-style-type: none">What is a linear transformation and how is it directly related to matrices? Learn how to apply the math and visualize the concept.

Supporting Lesson Content: Linear Algebra Essentials (Continued)

Lesson Title	Learning Outcomes
LINEAR ALGEBRA IN NEURAL NETWORKS	<ul style="list-style-type: none">Take a peek into the world of Neural Networks and see how it related directly to Linear Algebra!
LABS	
VECTORS LAB	<ul style="list-style-type: none">Learn how to graph 2D and 3D vectors
LINEAR COMBINATION LAB	<ul style="list-style-type: none">Learn how to computationally determine a vector's span and solve a simple system of equations
LINEAR MAPPING LAB	<ul style="list-style-type: none">Learn how to solve problems computationally using vectors and matrices

Supporting Lesson Content: Neural Networks

Lesson Title	Learning Outcomes
INTRODUCTION TO NEURAL NETWORKS	<ul style="list-style-type: none">Acquire a solid foundation in deep learning and neural networks. Implement gradient descent and backpropagation in Python.
TRAINING NEURAL NETWORKS	<ul style="list-style-type: none">Learn about techniques of how to improve their training of a neural network; such as: early stopping, regularization and dropout.
DEEP LEARNING WITH PYTORCH	<ul style="list-style-type: none">Learn how to use PyTorch for building deep learning models.

Project 1: Image Classifier

The deep learning revolution is said to have begun in 2012 when a group headed by Prof. Geoff Hinton applied deep learning to the ImageNet visual object classification challenge and demonstrated superior performance to all previous techniques. The ImageNet data set contains 14 million images belonging to 20,000 object classes. Since then deep neural networks have been demonstrated to match human performance in classifying images across many domains, which a few years ago would have sounded like science fiction. You will now learn how to do it yourself by building a Python application that can train an image classifier on a dataset, then predict new images using the trained model.



Learn more at www.udacity.com/enterprise