UDACITY
FOR ENTERPRISE

THE SCHOOL OF PROGRAMMING AND DEVELOPMENT

Intro to Programming

NANODEGREE SYLLABUS

Overview

Introduction to Programming Nanodegree Program

Learn the basics of programming through HTML, CSS, Python, and JavaScript. Get extensive practice with hands-on exercises and projects that demonstrate your grasp of coding fundamentals, and build confidence in your ability to think and problem-solve like a programmer.

Educational Objectives

A graduate of this program will be able to:

- Create basic web pages using HyperText Markup Language (HTML).
- Modify web page style with Cascading Style Sheets (CSS).
- Write Python scripts that use core programming concepts, including variables, functions, loops, classes, objects, data types, conditionals, and debugging.
- Run Unix shell commands and Python code from a Command-Line Interface (CLI).
- Access and manipulate files on your computer using Python code.
- Use Python to get and process data from a web-based Application Programming Interface (API).
- Write basic JavaScript scripts that demonstrate core elements of the language, including data types, variables, loops, functions, arrays, and objects.
- Use JavaScript and the Document Object Model (DOM) to create and modify web page content.

Program Information

ESTIMATED TIME
4 months
Study 10 hours/week

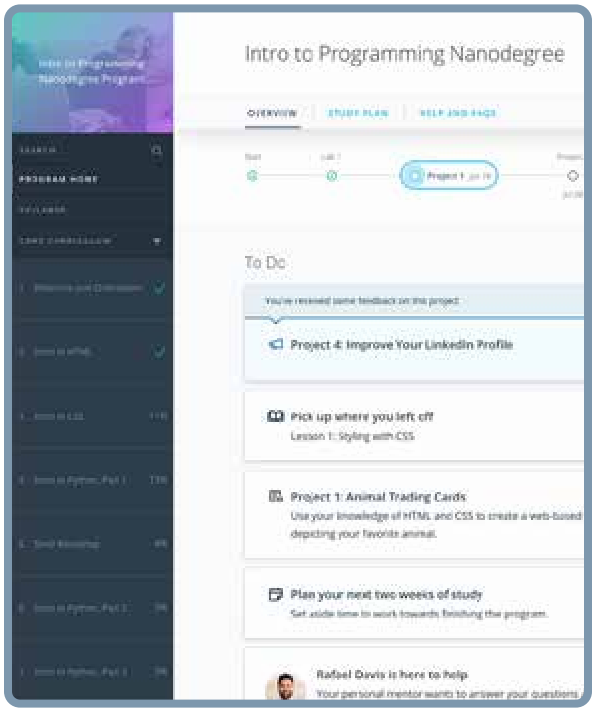
LEVEL
Foundational

PREREQUISITES
Basic computer skills, such as managing files, running programs, and using a web browser to navigate the Internet.

HARDWARE/SOFTWARE REQUIRED
There are no software and version requirements to complete this Nanodegree program. All coursework and projects can be completed via Student Workspaces in the Udacity online classroom. Udacity's basic tech requirements can be found at https://www.udacity.com/tech/requirements.

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

Our Classroom Experience


Intro to Programming Nanodegree

To Do

You've received some feedback on this project

Project 4: Improve Your LinkedIn Profile

Pick up where you left off
Lesson 1: Styling with CSS

Project 1: Animal Trading Cards
Use your knowledge of HTML and CSS to create a web-based depicting your favorite animal.

Plan your next two weeks of study
Set aside time to work towards finishing the program.

Rafael Davis is here to help
Your personal mentor wants to answer your questions

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

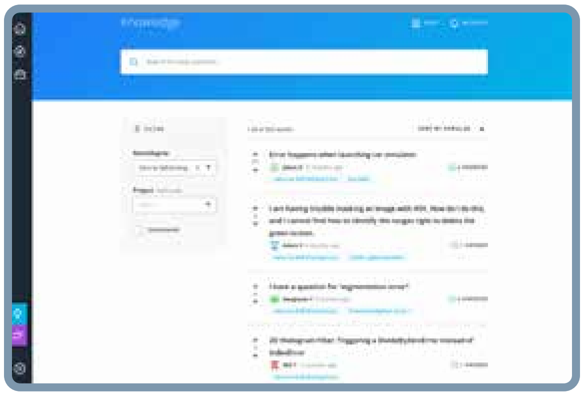
Learners leverage the power of community through a simple, yet powerful chat interface built within the classroom. Learner Hub connects learners with their technical mentor and fellow learners.

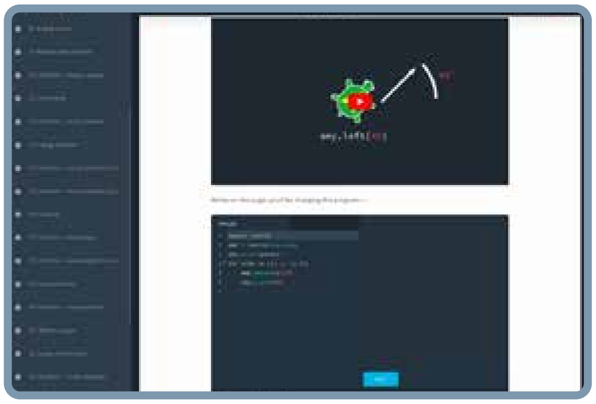
WORKSPACES

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.





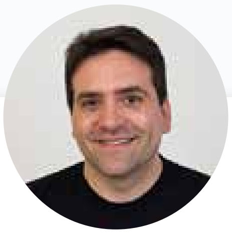
CUSTOM STUDY PLANS

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.

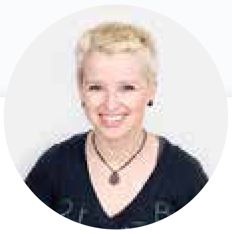
Learn with the Best



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.



Kelly Howard

INSTRUCTOR

Kelly is the Product Lead for the Web Development Nanodegree programs at Udacity.

Julia Van Cleve

INSTRUCTOR

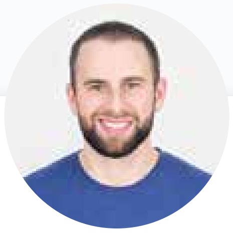
Julia is a Content Developer at Udacity and was previously a middle school math teacher in San Jose, CA. She also dabbled in freelance web development, designing websites for small businesses in the Bay Area.

Learn with the Best

Abe Feinberg

INSTRUCTOR

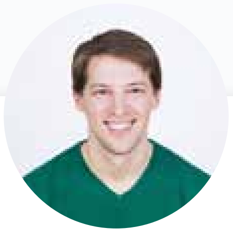
Abe is a science teacher and educational psychologist who loves learning and finding out how things work. He has a particular interest in using AI to optimize education, and his ultimate goal is to replace himself with a robot that can teach better than he can.



James Parkes

INSTRUCTOR

James received his degree in Computer Science and Mathematics, then went on to become a Udacity instructor in several programs. His personal mission is clear: to open the doors of opportunity for others by empowering them with excellent educational experiences.



Richard Kalehoff

INSTRUCTOR

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.

Course 1: Intro to Web Development

In this course you'll learn how to make basic web pages using HyperText Markup Language (HTML) and how to add style to your pages with Cascading Style Sheets (CSS). You'll begin by learning some basics about how the Web works, then build a very basic web page using only HTML, and finally explore how to add styles to your page with CSS. At the end of the course, you'll demonstrate your new skills by completing a project in which you create a web page that replicates a given design.

Project Animal Trading Cards

For this project, you'll use HTML and CSS to make Animal Trading Cards. You will apply your knowledge of HTML Document Structure to your html file and then create custom CSS styling based on your preferences. This project will demonstrate your understanding of linking CSS files in HTML files, implementing CSS classes to avoid repetition, as well create semantically organized HTML code.

LESSON TITLE	LEARNING OUTCOMES
THE WEB AND HTML	• Describe the fundamentals of how the web works • Edit web pages using a text editor and test work in the browser • Create HTML files that use elements and tags to provide the structure of a web page • Write fully qualified URL pathways by identifying each part of file path structures
LAB: BASIC HTML PAGE	• Demonstrate your understanding of HTML basics by creating a simple web page

LESSON TITLE	LEARNING OUTCOMES
STYLING WITH CSS	• Use CSS to change basic style properties, like the font, color, and border of a given element • Use CSS type and class selectors to apply style to specific subsets of HTML elements • Separate the style of a web page from its structure and semantics • Apply style in multiple ways, including via a separate, linked stylesheet • Recognize tree structures in HTML and CSS code • Modify the layout and resizing behavior of a web page using containers and the flexible box model (flexbox) • Use Developer Tools to inspect the elements of a web page

Course 2: Intro to Programming with Python I

Learn basic programming with Python, one of the most versatile and widely used programming languages! You'll first learn core programming concepts and fundamental Python syntax by writing code to make a virtual "turtle" robot draw colorful shapes on the screen. You'll then learn how to write Python functions, run Python from a Command-Line Interface (CLI), manipulate strings and lists, and refactor your code to improve its structure and make it more modular.

Project Adventure Game

Create a simple, interactive, text-based adventure game in Python, using modules, loops, conditionals, and functions. This project will demonstrate your ability to write correct Python syntax, practice with fundamental programming logic, refactor code using functions, and ultimately write a complete Python script that results in a working, playable game.

LESSON TITLE	LEARNING OUTCOMES
TURTLES AND CODE	• Use methods from Python's turtle module to draw simple geometric shapes • Define variables using assignment statements, and then use these variables in place of hard-coded constants • Work with fundamental data types, including integers, strings, and lists • Write compound statements and use indentation to indicate when code belongs to a given block • Import and use modules from the Python standard library • Use code comments to add basic documentation and activate/de-activate code • Adjust the sequential order of code to alter the flow of execution • Iterate over a data structure using a "for" loop • Use loop variables to generate a result that changes each time a loop runs and use nested loops in and correctly predict the number of times a nested loop will run • Recognize and fix common types of errors (including logical errors, syntax errors, and "usage" errors)

Nanodegree Program Overview

LESSON TITLE	LEARNING OUTCOMES
FUNCTIONS	• Generate a sequence of numbers using Python's range function and iterate over the sequence using a loop • Perform basic calculations on constants and variables with Python arithmetic operators • Define and call Python functions to create more modular, reusable code • Use parameters in function definitions to make functions more flexible and to avoid hard-coded values • Distinguish between global and local variable scope • Use a Python conditional "if" statement to alter execution flow depending on a given condition • Use the Python modulo operator to create repeating patterns • Use "return" statements when defining and calling Python functions • Use Python's "random" module to generate random numbers and select random items from a list • Use Python comparison operators to compare values
SHELL WORKSHOP	• Distinguish between a GUI vs CLI and why the latter is an important tool for a developer • Output text to the command line with the "echo" command • Use BASH shell variables to store and retrieve values • Navigate directories using the "cd" command • List the contents of a directory using the "ls" command. • Modify commands by adding options (such as the "-a" option in "ls -a") • Organize files and directories using "mkdir" and "mv" commands • Download files with the "curl" command • View files with the "cat" and "less" commands • Remove files and directories with the "rm" and "rmdir" commands

Nanodegree Program Overview

LESSON TITLE	LEARNING OUTCOMES
PYTHON AT HOME	• Use a Command-Line Interface (CLI) to run Python scripts • Run simple and compound statements in Python's interactive mode via the CLI • Use Python's input function to get and use input from the user via the CLI • Use Python's print function to output text to the CLI. • Use Python's print function to debug code • Distinguish between how a Python function's return value behaves when the function is run from a code file vs interactive mode • Recognize and fix type errors in Python • Use a Python traceback to trace through the relevant execution flow and identify the source of an error • Write a Python function and import it using interactive mode

Nanodegree Program Overview

LESSON TITLE	LEARNING OUTCOMES
STRINGS AND LISTS	• Distinguish between variables and literals and use complex strings in your code that contain punctuation and newline characters • Use Python's length (len) function to get the length of a string and Iterate (loop) over the individual characters in strings • Use string indexes to access the character at a given location and handle index errors on strings • Use slicing to access a substring within a larger string and concatenation to join multiple strings • Use formatted string literals (f-strings) to concatenate values into strings and format them • Convert between string data types and integer data types and write functions to perform basic string manipulation tasks • Use common string methods to retrieve and manipulate string data and use Boolean values with strings and perform operations and methods on lists • Distinguish the key differences between mutable data structures and immutable data structures and perform augmented assignment to update the value of a variable • Use while loops to iterate over strings and lists and Identify and create infinite loops • Interrupt or break out of loops when needed and find and manipulate substrings and use the join method to concatenate strings from a list

Nanodegree Program Overview

LESSON TITLE	LEARNING OUTCOMES
STYLE AND STRUCTURE	• Use a code linter to check whether code meets the conventions specified in the Python style guide (PEP 8) • Write multi-line strings in Python using a variety of techniques (triple quotes, escape characters, and implicit line joining) • Write a basic interactive Python program to meet specifications • Refactor a basic Python program to make the code easier to understand, more modular, and more flexible • Handle invalid user input in a basic interactive Python program • Use Python functions (instead of loops) to repeat a behavior repeatedly until a condition is met • Distinguish between global and local variable scope

Course 3: Intro to Programming with Python II

Advance your skills as a beginning programmer with Python—one of the most versatile and widely used programming languages! In this course, you will build on your understanding of fundamental Python and learn some more advanced skills, including how to work with files on your computer's disk, how to retrieve data using a web API, and how to use Object-Oriented Programming (OOP) to create your own classes, objects, and methods.

Project Rock Paper Scissors

In this project, you'll apply your Python and object-oriented programming skills to build a program that plays the game of Rock Paper Scissors. You'll build classes that represent the game and its players. You'll write computer players that follow various different strategies, as well as a human player class that lets a human play the game against the computer.

LESSON TITLE	LEARNING OUTCOMES
WORKING WITH FILES	• Distinguish between files and other forms of data stored in memory (such as variables) • List files in a directory and extract file names • Move and organize files and read text from a text file • Process text using string operations • Write text output to a file • Identify and fix common bugs in text processing

Nanodegree Program Overview

LESSON TITLE	LEARNING OUTCOMES
WEB APIS	• Use the BASH shell and the Python requests module to send requests to a web API • Use Python try/except blocks to handle exceptions • Recognize JSON as a standardized format for structuring data • Use Python dictionaries to structure data in key-value pairs • Use a Python loop to iterate over a list • Use a Python loop to iterate over a dictionary • Store and access nested elements of data structures within Python lists and dictionaries • Use a Python loop to iterate over data structures containing nested elements (e.g., lists within lists or dictionaries within dictionaries) • Use Python to retrieve data from another application via a web API
OBJECTS AND CLASSES	• Identify the key characteristics of classes and objects in Python • Use the “isinstance” and “type” functions to identify the type of a given object in Python • Define a Python class and use it to instantiate an object • Use the self parameter to access the current instance (i.e., object) of a given Python class • Use class-level variables to store values that are shared across all instances of a Python class • Use instance-level variables in Python to store values that apply only to a specific instance (object) of a class • Use Python initializers to set initial values for an object when it is instantiated • Differentiate between is-a and has-a relationships with Python classes and subclasses • Use the super method in Python to create subclasses

Course 4: Intro to Javascript

Learn the history of JavaScript and how it compares to Python programming. Understand how the DOM is formed, what Nodes and Elements are, and how to select items from the DOM. By the end, you'll write JavaScript code that allows the user to create a grid of squares representing their design, and apply colors to those squares to create a digital masterpiece.

Project Pixel Art Maker

For this project, you'll build a single-page web app that allows users to draw pixel art on a customizable canvas.

LESSON TITLE	LEARNING OUTCOMES
WHAT IS JAVASCRIPT?	• Understand the history of JavaScript and start writing your code immediately using the JavaScript console.
DATA TYPES & VARIABLES	• Learn to represent real-world data using JavaScript variables, and distinguish between the different data types in the language.
CONDITIONALS	• Learn how to add logic to your JavaScript programs using conditional statements.
LOOPS	• Harness the power of JavaScript loops to reduce code duplication and automate repetitive tasks.

Course 4: Intro to Javascript, cont.

LESSON TITLE	LEARNING OUTCOMES
FUNCTIONS	• Dive into the world of JavaScript functions. Learn to harness their power to streamline and organize your programs.
ARRAYS	• Learn how to use Arrays to store complex data in your JavaScript programs.
OBJECTS	• Meet the next JavaScript data structure: the Object. Learn to use it to store complex data alongside Arrays.
THE DOCUMENT OBJECT MODEL	• Understand how the DOM is formed, what Nodes and Elements are, and how to select items from the DOM.
CREATING CONTENT WITH JAVASCRIPT	• Use JavaScript and DOM methods to create new page content, update existing content, and delete content.
WORKING WITH BROWSER EVENTS	• Learn what an event is, how to listen for an event and respond to it, what data is included with an event, and the phases of an event.

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