

# Intro to Programming Nanodegree Syllabus

---



## Contact Info

While going through the program, if you have questions about anything, you can reach us at [support@udacity.com](mailto:support@udacity.com). For help from Udacity Mentors and your peers visit the Udacity Classroom.

## Nanodegree Program Info

**Version:** 4.0.0

**Length of Program:** 227 Days\*

*\* 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.*

## Part 1: Learn to Code

Welcome to the Introduction to Programming Nanodegree program. This is your first step on your journey to become a programmer. Get started early with html and feedback from our reviewers!

### Project: Getting Started with HTML

For this project, you will submit your very first programming file containing HTML code. HTML is the coding language for building websites. We recommend taking notes from this section and using your notes as the content for your HTML file. This project is not graded. Our reviewers will provide you with helpful suggestions and advice for learning in this program.

### Supporting Lessons

Lesson	Summary
The World Wide Web	Get acquainted with how the web works! Learn about your browser, the internet, and a filetype called HTML.
HTML Basics	HTML stands for "Hypertext Markup Language." HTML is the language used to create webpages. Learn the basics of HTML to create a simple web page!

## Part 2: Make a Stylish Web Page

For the project, you'll use HTML and CSS to make a stylish web page on any topic. You will apply your knowledge of HTML Document Structure to your html file and then create custom CSS styling based on your personal preferences.

### Project: Make a Web Page

For the project, you'll use HTML and CSS to make a stylish web page on any topic. You will apply your knowledge of HTML Document Structure to your html file and then create custom CSS styling based on your personal preferences.

### Supporting Lessons

Lesson	Summary
<b>Getting Focused</b>	Prepare for this project by getting into a programming mindset. Make sure you are set up for success!
<b>Creating a Structured Document</b>	Learn about HTML page structure, visual styling, and box design. Work with HTML structure tags to organize your contents.
<b>Work Session: HTML Structure</b>	Create an HTML file for your notes and examine the differences between structured and unstructured code.
<b>Adding CSS for Style</b>	Learn how to add CSS, divs, spans, classes, and ids to your page. CSS allows you to create the aesthetics that all web users enjoy when browsing the web.
<b>Work Session: CSS Practice</b>	Review the previous lesson and add style to your notes. Once you've practiced enough, you'll be ready to create your own stylish website for your project!

## Part 3: Code Your Own Quiz

You will use the Python programming language to build your own quiz. You will use a fill-in-the blank style to create a quiz that can even be used as a study tool to help you remember important vocabulary.

### Project: Code Your Own Quiz

Complete and submit the final project: "Code Your Own Quiz". This is a a fill-in-the-blanks quiz based on concepts you've learned from your Mad Libs generator.

### Supporting Lessons

Lesson	Summary
<b>Getting Set Up with Python</b>	Get your computer set up to start programming in Python on your computer!
<b>Introduction to Serious Programming</b>	Write your first computer program and learn about programming grammar.
<b>Work Session: Basic Debugging</b>	Complete a couple debugging exercises to understand the importance of syntax.
<b>Variables &amp; Strings</b>	Learn how to store values in Python variables, and work with text as Python strings.
<b>Work Session: String Manipulation</b>	Begin working on your Mad Libs generator by practicing how to manipulate variables and strings.
<b>Input -&gt; Function -&gt; Output</b>	Learn how to use a function to take some input and generate some output based on the contained programming rules within the function.
<b>Work Session: Print vs Return</b>	Practice using functions while learning more about print and return statements.
<b>Control Flow: If Statements &amp; While Loops</b>	Learn how to manage the flow of a computer program using Boolean values, if statements, and while loops.
<b>Deep Debugging</b>	Get acquainted with five key debugging strategies to help you address problems in your code.
<b>Work Session: Mad Libs Generator</b>	Use the skills you've learned so far to continue developing your Mad Libs generator. You now know how to implement control flow and loops. You also know how to properly debug your code!
<b>Structured Data: Lists &amp; For Loops</b>	Learn how to structure data in Python lists and then iterate over the elements with for loops.
<b>How to Solve Problems</b>	Practice problem-solving techniques by breaking down large problems into smaller ones.
<b>Work Session: Mad Libs Continued</b>	Finish the Mad Libs generator by putting together everything you've learned so far!

## Part 4: Create a Movie Website

You will write code to store a list of your favorite movies, including box art imagery and a movie trailer URL. You will then use your code to generate a static web page allowing visitors to browse their movies and watch the trailer.

### Project: Create a Movie Trailer Website

You will write code to store a list of your favorite movies, including box art imagery and a movie trailer URL. You will then use your code to generate a static web page allowing visitors to browse their movies and watch the trailer.

## Supporting Lessons

Lesson	Summary
<b>Getting Started</b>	Get a quick introduction to this section, your next instructor, and what you'll be working on to build your "Create a Movie Website" project at the end.
<b>Mini-Project: Take a Break</b>	In this Mini-Project, figure out how to use Python libraries to create a program that reminds you to take a break from working.
<b>Mini-Project: Secret Message</b>	In this Mini-Project, figure out how to use Python libraries and documentation to decode a secret message within files on your computer.
<b>Mini-Project: Draw Turtles</b>	Learn how to use classes to call on libraries like Turtle to draw cool shapes in Python.
<b>Mini-Project: Send a Text</b>	In this Mini-Project, use the Twilio API to make a program that sends a text message to your phone.
<b>Mini-Project: Profanity Editor</b>	In this mini-project, you will write a program that checks for profanity in a message and replaces curse words.
<b>Movie Website Creation</b>	Make a movie website using the concept of classes in Python as well as your understanding of HTML & CSS.
<b>Advanced Class Making</b>	Learn about advanced ideas in Object Oriented Programming like class variables, inheritance, reusing methods, and method overloading.

## Part 5: Discover Your Path

In this section, there is no project submission. Instead, you will explore a quick overview of the vast world of programming. After this section, you'll have a better understanding of different options you have as a programmer. This will help guide you in your final project for this program.

## Part 6: Front-End Development

Learn all about JavaScript and start writing your code immediately using the JavaScript console. Learn enough to program an awesome memory game for your final project.

## Project: Memory Game

The Memory Game Project is all about demonstrating your mastery of HTML, CSS, and JavaScript. You'll build a complete browser-based card matching game (also known as Concentration).

### Supporting Lessons

Lesson	Summary
<b>What is JavaScript?</b>	Learn the history of JavaScript and start writing your code immediately using the JavaScript console.
<b>Data Types &amp; Variables</b>	Learn to represent real-world data using JavaScript variables, and distinguish between the different data types in the language.
<b>Conditionals</b>	Learn how to add logic to your JavaScript programs using conditional statements.
<b>Loops</b>	Harness the power of JavaScript loops to reduce code duplication and automate repetitive tasks.
<b>Functions</b>	Dive into the world of JavaScript functions. Learn to harness their power to streamline and organize your programs.
<b>Arrays</b>	Learn how to use Arrays to store complex data in your JavaScript programs.
<b>Objects</b>	Meet the next JavaScript data structure: the Object. Learn to use it to store complex data alongside Arrays.
<b>The Basics: the DOM, \$, and Selectors</b>	Enter the exciting, interactive world of the DOM! Learn how to use JavaScript and jQuery to select and manipulate HTML elements on a page.
<b>The Tricks: DOM Manipulation</b>	Now that you've seen how to select elements using jQuery, learn to manipulate the DOM and change a webpage dynamically.
<b>Event Listeners with jQuery</b>	Dive into the interactive world of DOM events using jQuery's Event Listeners! Learn how to use jQuery to respond to clicks, keyboard input, scrolling, and other common DOM events.

## Part 7: Back-End Development

In this project, you'll practice your SQL skills by building a reporting tool that summarizes data from a large database.

### Project: Logs Analysis Project

In this project, you'll practice your SQL skills by building a reporting tool that summarizes data from a large

database.

## Supporting Lessons

Lesson	Summary
<b>Intro to Back-End Development</b>	Georgia introduces the Back End path and some tips and tricks for learning about Relational Databases.
<b>Data and Tables</b>	Learn the principles behind relational data organization: tables, queries, aggregations, keys, and joins.
<b>Elements of SQL</b>	Start learning SQL by using the select and insert statements to read and write data in database tables.
<b>Python DB-API</b>	Learn the Python database API, and apply your knowledge to fix common bugs that arise in database-backed web services.
<b>Deeper Into SQL</b>	Create your own database tables using normalized table design, using keys to declare relationships between tables; then apply these relationships to draw conclusions from data.

## Part 8: Data Analysis

Choose one of Udacity's curated datasets, perform an investigation, and share your findings.

### Project: Investigate a Dataset

Choose one of Udacity's curated datasets, perform an investigation, and share your findings.

## Supporting Lessons

Lesson	Summary
<b>Data Analyst Path Intro</b>	Georgia introduces the front end path and some tips and tricks for learning data analysis.
<b>Setting up Anaconda</b>	Learn to use Anaconda for managing packages and development environments.
<b>Setting up Jupyter Notebook</b>	Learn to use Jupyter notebooks for creating narrative documents containing code, text, and images.
<b>Dictionaries</b>	Get an awareness of dictionaries, an important data structure in Python that you are going to encounter in the rest of this path.
<b>Data Analysis Process</b>	Go through the data analysis process from start to finish using a dataset about Udacity student engagement.
<b>NumPy and Pandas for 1D Data</b>	Learn how to NumPy and Pandas to write clean, concise, and fast data analysis code for one-dimensional data.
<b>NumPy and Pandas for 2D Data</b>	Learn more features of NumPy and Pandas you can use to analyze two-dimensional data.

## Part 9: iOS Development

Create an iPhone app that records a conversation with you and a friend and plays it back to make you sound like a chipmunk or Darth Vader!

### Project: Pitch Perfect

Create an iPhone app that records a conversation with you and a friend and plays it back to make you sound like a chipmunk or Darth Vader!

### Supporting Lessons

## Lesson

## Summary

### Introduction to iOS

Georgia introduces the front end path and some tips and tricks for learning iOS programming

### Swift Variables and Types

Learn about one of the most basic building blocks of nearly every programming language — variables. Variables give developers the ability to store information that can be used to control the behavior

### Strings

Learn how to represent a sequence of characters, or in other words, a string! Also, learn how to manipulate strings using their powerful built-in features.

### If Statements

Learn how to control when and how Swift code is executed by using `_if_` statements. Also, learn about the counterpart to if statements called `_else_` statements.

### Functions

Functions give developers the ability to bundle up code so that it can be reused multiple times. Learn how to define and use them.

### Introduction to Xcode

Introduction to Swift and Xcode, taking the first steps in becoming an iOS developer.

### AutoLayout and Buttons

Learn Storyboard, AutoLayout, UIButtons and UILabels. Start developing the Pitch Perfect App and creating the user interface.

### ViewController and Multiple Views

View Controllers, View Controller Lifecycle, and Multiple Views. Learn how to react to touch events and change the user interface based on those events.

### Delegation and Recording

Introduction to AVAudioRecorder, Delegation, and programmatic segues. Setup audio recording in the Pitch Perfect App

### Playback and Effects

Setup audio playback including rate, pitch, echo, and reverb. Learn about UIStackViews and class extensions.

## Part 10: Android Development

Build your first Android app - a design for a local business that could be used as a business card.

### Project: Build a Single Screen App

Build a design for a local business that could be used as a mobile business card.

### Supporting Lessons



## Lesson

## Summary

---

### Introduction to Android

Georgia introduces the android path and some tips and tricks for learning Android programming.

---

### Building Layouts

Learn how to place layouts on a page to create images, buttons, and text on a phone screen

---

### Building Layouts Continued

Learn how grouping layouts together allows for more creative and original designs.

---

### Practice Set: Building Layouts

Download Android Studio and get your first designs onto an actual phone.

---



Udacity

Generated Sun Aug 25 08:59:43 PDT 2019