Python Programming 1 Course Checklist

** Windows computers must use a more recent operating system than xp.*

Practice all the programming commands and techniques covered in each video until you understand them, before moving on to the next session. The videos are progressive, building on previous concepts. Don't skip over things you don't like or find difficult or the course will become increasingly more difficult for you.

Remember, practice is how you will improve! Pause the videos to try the demonstrated technique on your computer, rewind if you need something explained again, and practice, practice, practice!

Charting Your Progress
Check the first box when you've watched the video. Check the second when you've thoroughly practiced **all the skills** introduced in the video and completed the suggested project. Enter dates instead of checks if you'd rather.

I've recorded solutions to each of the programming assignments. **DO NOT** watch these before you complete your own assignment—unless you are totally, absolutely stuck. You don't learn programming by looking at answers, but by working (and often struggling) to come up with the answers yourself. Trust me, it will get easier with each assignment you complete.

Viewing Solutions and Revising Your Programs
Compare my solution to yours. Are there things you did incorrectly or inefficiently? If so, revise your program using a more correct or efficient approach. You don't have to copy my version exactly, however if you took 15 lines to solve something that can be done in 3, there's a problem with your logic. Mark the third column in the checklist after watching the video that explains my programming solution and making any necessary changes to your code. Feel free to expand on the programming assignments, adding new features as desired.

Show Off Your Work!
Demonstrate your completed program to someone–or to lots of people! It's fun to share what you have made. Others may have suggestions for improving your work, and your understanding will be solidified by explaining to them what you have done.

Video Watched	Assignment Completed	Reviewed Solution & Made Necessary Changes	Video Topic and Assignment
		NA	**Lessons 1-4** **Intro** **Install Python 3 and the Wing IDE on Windows** **Install Python 3 and the Wing IDE on a Mac** **Using Python 3 on the Raspberry Pi** **Practice**: Install python and Wing (Mac/Win), be able to start the Wing software or locate Python 3 on the Pi. (Obviously you only need to watch the session that applies to your system)
		NA	**5 Basic Instructions on Using the Wing IDE (Mac/ Win only)** Create a folder to store your python programs, enter the Hello World program and run it, set the preferences in the Wing editor **Practice:** Practice video topics
		NA	**6 Variables in Python** Practice using the shell and the editor to assign values to variables and print them. Understand the difference between the comma and the + with string variables **Practice:** Practice video topics
			7 Getting Input From the User, Converting Variable Types Use the input command, convert between the string, float, and int data types, use type() to determine variable type **Assignment 01:** Variables, Input, and Print *Assignment details from here on, will be in the python-assignments.pdf file. They're also described at the end of each video in the Mission Impossible section. Assignment answers may be viewed in the assignment solution videos – after the student has completed the assignment.*
			8 Introduction to the If Statement Learn how an if statement executes code based on the true/false result of the comparison statement. **Assignment 02:** Simple If Statements
		NA	**9 Boolean Logic** Learn how to combine conditional statements using

			not, and, or to produce True and False values **Assignment:** (no programming assignment – this lesson is continued in session 10)
			10 Creating a Compound If Statement How to implement the boolean logic concepts in python code **Assignment 03:** If statements with boolean operators
			11 Using Nested If Statements How to place an if statement inside another if statement in python **Assignment 04:** Branching story using nested if statements to produce different outcomes
		NA	**12 Precedence of Math Operations** Learn how python determines the order to evaluate expressions and calculations **Assignment:** Practice video topics
			13 Boolean Operator Precedence Learn how python determines the order to evaluate boolean expressions, the in keyword, and isnumeric() **Assignment 05:** Simple Math Calculator
			14 Introduction to Python Lists Introduction to lists: creating, adding items, printing, using indexes/subscripts, slicing, sorting, and using the in operator. **Assignment 06:** Intro to Lists
		NA	**15 Creating a For Loop, the Wing Debugger** Introduction to for loops and how to use the Wing debugger to step through a program **Assignment:** Practice video topics
			16 More For Loop Practice, Loops & Lists More for loop practice and using for loops with lists **Assignment 07:** Student Test Scores
		NA	**17 Operations With Python Lists** Additional list operations: extend, append, insert, remove, split, join, index, count **Assignment:** Practice video topics

			18 Additional List Operations Additional list operations: max, min, sum, copy, clone, tuples vs lists **Assignment 08:** Madlib (a starter code file is provided in the code zip file)
		NA	**19 Sorting and Reversing Lists** Case insensitive sorts with lambdas, reversing lists **Assignment:** Practice video topics
			20 Nested For Loops Nested for loops and nested lists. **Assignment 09:** Player Score (a starter code file is provided in the code zip file)
			21 How to Generate and Use Random Numbers Using random numbers to create variety and unpredictability in programs, randint() and choice() functions **Assignment 10:** Random (Silly) Sentences
			22 Manipulating Strings in Python Additional string functions and processing letter by letter, in, reversing a string, immutability, slices, formatting in print statements **Assignment 11:** Pig Latin
		NA	**23 How to Create While Loops** Usage and construction of a while loop, avoiding infinite loops **Assignment:** Practice video topics
		NA	**24 Using a Boolean Variable to Control a While Loop** Using boolean variables to control loop **Assignment:** Practice video topics
			25 More on While Loops Using boolean variables to control loop **Assignment 12:** Race/Contest Simulation
		NA	**26 Introduction to Python Functions** Introduction to functions, how to define and call **Assignment:** Practice video topics

			27 Passing Arguments to a Function Passing arguments to a function **Assignment 13:** Tip Calculator with a Function
		NA	**28 Returning Values from a Function** Returning values from functions **Assignment:** Practice video topics
		NA	**29 Using Functions With Named Arguments And Default Values** Named arguments, default argument values **Assignment:** Practice video topics
		NA	**30 Modularizing Code With Functions** Jumbled word program example using functions, modularizing code **Assignment:** Practice video topics
			31 Dog-Human Age Calculator Assignment Given Using Functions Functions assignment given **Assignment 14:** Dog-Human Age Calculator
		NA	**32 Variable Scope, Global And Local Variables** Variable scope, global and local variables **Assignment:** Practice video topics
			33 Functions – Rock, Paper, Scissors Assignment Given Using Functions Functions assignment given **Assignment 15:** Rock, Paper, Scissors
		NA	**34 Tic Tac Toe: Program Introduction and Overview** Overview of how to approach a longer, more complex program in terms of logic and data representation **Code File:** tic-tac-toe-01.py (all 10 tic tact toe code files are provided in the code zip file)
		NA	**35 Tic Tac Toe: Board Setup** Display instructions, create the board list, display the game board **Code File:** tic-tac-toe-02.py

		NA	**36 Tic Tac Toe: Player's Move** Get the player's single move, validating their input as numeric 0 to 8 **Code File:** tic-tac-toe-03.py
		NA	**37 Tic Tac Toe: Player Loop** Player selects x or o as their marker, validating whether the desired square is empty/available, looping to fill the board with player moves **Code File:** tic-tac-toe-04.py
		NA	**38 Tic Tac Toe: Computer's Move (Random Selection)** Randomizing who goes first, adding a preliminary computer_turn which selects a move at random, alternate turns between player and computer until board is full **Code File:** tic-tac-toe-05.py
		NA	**39 Tic Tac Toe: Check For a Win or a Tie** Check to see whether there is a win or a tie, one version uses many if statements, a second version uses nested lists **Code File:** tic-tac-toe-06.py, tic-tac-toe-07.py
		NA	**40 Tic Tac Toe: Improve The Computer's Playing Logic (If Statements)** Improve the computer's playing logic, simpler version using if statements **Code File:** tic-tac-toe-08.py
		NA	**41 Tic Tac Toe: Improve The Computer's Playing Logic by Testing For A Win** Improve the computer's playing logic, more advanced version which loops through all board positions testing for a win **Code File:** tic-tac-toe-09.py
		NA	**42 Tic Tac Toe: Final Wrapup** Add an outer loop to play multiple rounds, keep track of player and computer wins and ties **Code File:** tic-tac-toe-10.py
		NA	**43-57 Assignment Solutions** View my solutions after you code each assignment
