| Unit 1 Introduction to Python Problems [Learning Plan Index - Python](https://docs.google.com/document/d/1B5yWb6wCSRhqD42iWxCi7bmLPY2EqvU6pbiEQT0zs20/edit?usp=sharing)    *Unit 01 of Python Programming - Introduction to Python Problems* | |
| --- | --- |
| Learning Targets  This unit we will…  Look at installing Python and the PyCharm IDE on your computer as well as review some of the basics of Python so you can write your first program.  I can…   * Install python and the pycharm IDE and create a “Hello World” program * Understand some of the history of python. * Generate basic output with a print statement. * Generate mathematical output with a print statement. * Use a python Turtle to draw shapes and move around the turtle canvas using x & y coordinates.   Vocabulary: print, turtle, cartesian coordinate system, mathematical operations, IDE, python, pycharm | |
| Learn About It!  *You can explore some, or all of these resources. If you want to see a resource again, go for it!*  [Learning Plan Index - Python](https://docs.google.com/document/d/1B5yWb6wCSRhqD42iWxCi7bmLPY2EqvU6pbiEQT0zs20/edit?usp=sharing) *The Colab documents review the concepts of each unit with code you can run and modify.*  **Overview and Background**  The Python programming language was released in 1991 and if you are curious about the [history check out this webpage](https://www.geeksforgeeks.org/history-of-python/). To install and run Python on your computer you are going to need a computer, Windows or Mac, and not a chromebook. If all you have is a chromebook, you can start out using [Repl.it](https://repl.it/), but will need to use a computer once we get to using GUI’s later in the course. Please notify the teacher if you need access to a computer. You hopefully at this point have an understanding of computer systems, their components and an idea of how a program stores things in memory. If not, check out this [learning plan](https://docs.google.com/document/d/1BAf_NA2nJ9vNsnTWz9QINuo2VlQuPKjpAvZTLfol0Lc/edit?usp=sharing) from my Intro class that covers the different components of a computer system.  Being a programmer is about [critical thinking and problem solving](https://docs.google.com/presentation/d/1d6XRC4aZCTsDjJgwXsjX2nv8G1h10VODCLuszzfiP18/edit?usp=sharing). Breaking a problem down to smaller and smaller solvable problems so that you can create a working algorithm.  **Installing Python & PyCharm IDE**  To write Python code you need to [install the Python language](https://www.python.org/downloads/) on your computer. Once you have installed Python you need to install the IDE (Integrated Development Environment) [PyCharm](https://www.jetbrains.com/pycharm/download/). On the downloads page are instructions for configuring PyCharm so that it sees your Python installation. Here is a [quick video (5:20)](https://drive.google.com/file/d/17WoWCQGr9g2rCglFEwyNJOIo9ybTYPX7/view) that shows the first steps of how to fire up PyCharm, create a new Python file and run it.  **Colab Documents**  Colaboratory Documents are a version of Jupyter Notebooks. These documents allow for the combination of text and executable python code. In essence it is like having a digital textbook with code that works and you can play with to see how it works. I have created a series of Colab Docs for this class, that in essence summarize the concepts of each unit. To view them you will need to [install the documents on your Google drive](https://docs.google.com/document/d/1aiTy4a8bQAJkBPJwtXctleVcT5OJi00Hm8rUupZrss4/edit?usp=sharing). Once you have them installed you can open my documents and interact with them. If you want you can copy my documents so you can edit them. I have an [index document](https://docs.google.com/document/d/1B5yWb6wCSRhqD42iWxCi7bmLPY2EqvU6pbiEQT0zs20/edit?usp=sharing) for all the Colab Documents used in this class and each document has a link back to the index. | |
| Evidence of Learning  *Choose one activity to show what you learned.*  [Grading Rubric](https://docs.google.com/document/d/1shjqolaw_5tSX9T5OJ2FZuBeon7K3hDrYEJ5m1ltSEw/edit?usp=sharing) | **Unit Programs**  Review [Colab - Unit 1 Basic Output & Program Requirements](https://colab.research.google.com/drive/1ZgN_CmscR1gjSLjXV6NZaXSpAURil4eE?usp=sharing) and then do the program problems listed below. There are tips, sample code, and links to sample code that you will use within the Colab document. There will be two sets of problems to do, one with python and one with python turtle. The first set of programs can be done in one file, there is a picture of what your output should look like below. Name the file **Unit01\_YourLastName.py**, if you do this set of problems in [repl.it](https://repl.it/) name the repl.it Unit01\_YourLastName and turn the share link into the classroom. This [short video (2:37)](https://drive.google.com/file/d/11tBWkMDkirAdTC_AaATPvJbUSWT6WttO/view) shows how to get a share link, it is from my Intro class and talks about HTML, but you can ignore that.  **Unit01\_YourLastName**   1. Problem 1 (10 points) - Print out “PHS” using print statements:      1. Problem 2 (10 points) - Compute the following expression      1. Problem 3 (10 points) - Write a program that displays the sum of the series 1+2+3+4+5+6+7+8+9 2. Problem 4 (10 points) - If the area of a circle can be found with radius2 \* pi then output the area of a circle with a radius of 13.86. If you put “import math” at the top of your program you can use math.pi for the value of pi. Your solution must raise the radius to a power of two, no 13.86 \* 13.86. 3. Problem 5 (10 points) - output the area and perimeter of a rectangle with side lengths of 4.5 and 7.5 4. Problem 6 (15 points) - (Population projection) The US Census Bureau projects population based on the following assumptions:    1. One birth every 7 seconds    2. One death every 13 seconds    3. One new immigrant every 45 seconds   Problem 6 - Write a program to display the population for each of the next five years. Assume the current population is 312,032,486 and one year has 365 days. Hint: in Python, you can use the integer division operator // to perform division. The result is an integer. For example, 5 // 4 is 1 (not 1.25) and 10 // 4 is 2 (not 2.5). ALL MATH MUST HAPPEN IN THE CODE!! You may only use the numbers given, here is 60 & 24 & 365 for calculating seconds. The computer is the calculator and going forwards you are going to get user input so you won’t be able to hard code things. Hard coding things is hard to maintain.  Program output should be similar to the sample output below: |
|  | |
| **Unit01\_Turtle\_YourLastName**  The following programs can be done in one file. You can comment out each individual section with ‘’’ and ‘’’while you are working. Name your program Unit01\_Turtle\_YourLastName.py, again this can be done with a Turtle repl.it. Use the [Turtle Resource at python.org](https://docs.python.org/3/library/turtle.html) and the resources at the bottom of the [first colab document](https://colab.research.google.com/drive/1ZgN_CmscR1gjSLjXV6NZaXSpAURil4eE?usp=sharing).   1. Use a turtle to draw a five pointed star: (10 points)      1. Use a turtle to draw a rectanguloid: (10 points)      1. Use five turtles to draw the x y lines of the cartesian coordinate system and then (15 points)    1. Place a red triangle in the upper left quadrant with the word triangle above it, in red.    2. Place a green square in the upper right with the word square above it, in green.    3. Place a blue pentagon in the lower left with the word pentagon above it, in blue.    4. Place a pink circle in the lower right with the word circle above it, in pink..     Submit both of your programs to the Google Classroom assignment and hit Turn in. | |

