

# Joseph's Beginner Python Projects

## Joseph's Goal Projects

1. A version of [Mandarin Mojo](#) in which I can input a list of characters
  - a. Variation 1: A version of [Mandarin Mojo](#) in which I can input any two linked pieces of information that I would put on a flashcard (such as language vocabulary, capitals of countries, etc.).
  - b. Variation 2: A version of a [Mandarin Mojo](#)-like game in which the user can import a frequency list (probably a spreadsheet with two columns: *word* in column A and *frequency* in column B), and then in the game the words will be presented proportionally according to that frequency.
2. Make an image black and white, so that our writing sample scans never have a bluish tint
3. Get all articles from New York Times about China
  - a. I would like to have a script which will allow me to change which country it searches for, and produces an excel spreadsheet with the following data in different columns: date of publication, author, title of article, text of article.
  - b. I'd like to get this information for every article published in the New York times which is focused on China, but I'd also like to be able to edit this in order to find articles about any country. I want to include articles that are about [COUNTRY], not merely mention the name of the country. This script should function regardless of what country name is used, whether China, Angola, Russia, Scotland, or any other country.
4. Proportionally assign PIVP tasks
  - a. Make a simple GUI for website in which the user clicks one of four buttons:
    - i. "who should I assign QR to?"
    - ii. "who should I assign NTC to?"
    - iii. "who should I assign M&U to?"
    - iv. "who should I assign FC to?"
  - b. When clicked, the buttons will give you the name of a person. It can be selected from a random list of strings, and I can set the proportionality as I like. For example, if Joseph should do it 50% of the time, Lara 40% of the time, and Rachel 10% of the time, I can just have it randomly select an entry from the following list: [Joseph, Joseph, Joseph, Joseph, Joseph, Lara, Lara, Lara, Lara, Rachel]
5. Use goodreads spreadsheet to email return current lowest prices for multiple books
6. Daily HSK question via email

## From Reddit

1. ✓ [Magic 8 Ball](#)
2. ✓ [Pythagorean Triples Checker](#)
3. ✎ [Coin Estimator By Weight](#)
4. ✓ [Mad Libs Story Maker](#)

# Joseph's Beginner Python Projects

5. [≈ Change Calculator](#)
6. [Mean, Median, and Mode](#)
7. [Higher Lower Guessing Game](#)
8. [Multiplication Table](#)
9. [Fibonacci Sequence](#)
10. [Hangman Game](#)
11. [Menu Calculator](#)
12. [Dice Rolling Simulator or Creating a Dice Simulator](#)
13. [Count and Fix Green Eggs and Ham](#)
14. [What's My Number?](#)
15. [Factors of a Number](#)
16. [Countdown Clock](#)
17. [Turn Based Pokemon Style Game](#)
18. [A Variation of 21](#)
19. [Compare Recent Karma](#)
20. [Watch for new TIL facts](#)
21. [Random Wikipedia Article](#)
22. [What's the Weather?](#)

## [Michigan State University CS1 Python Programming Projects Archive](#)

- **Beginnings: a set of simple projects when you have little to work with**
  - First Steps -- little to work with.
    - [Richter Scale](#): Input some numbers, do some simple arithmetic, output results.
    - [Windchill](#): Input some numbers, do some simple arithmetic, output results.
    - [Gasoline](#): Input some numbers, do some simple arithmetic on gas and oil quantities, output results.
    - [Measurement](#): Input some numbers, do some simple arithmetic to do silly conversions such as furlongs/fortnight, output results.
    - [Population](#): Input some numbers, do some simple arithmetic to estimate today's U.S. population, output results.

# Joseph's Beginner Python Projects

- **≈ Time Travel:** Input some numbers, do some simple arithmetic to calculate travel time between stars, output results.
- **≈ Debt :** Calculate the height of a stack of bills based on the national debt. .
- **Simple Programs.**
  - **✓ Number Puzzle: Slayer:** Check solutions for a number-word puzzle.
  - **≈ Garden:** Calculate plantings and mulch for a garden.
  - **Einstein:** Solve a simple number puzzle attributed to Einstein.
  - **MacArthur:** Solve a simple number puzzle attributed to MacArthur.
  - **Tones:** Calculate frequencies to generate tones on a computer speaker.
  - **Turtle Graphics Angles:** Use Turtle Graphics to draw two lines and the calculate the angle between them.
  - **NinetyNine :** Simple number game.
- **Control: With selection and repetition we can begin to do something**
  - **Draw :** Draw regular patterns using TurtleGraphics.
  - **Making Change :** Making change.
  - **Number Puzzle: SplitNumber:** Split numbers into increasing sequences, e.g. 154152 -> 15, 41, 52.
  - **WaterBilling:** Calculate water bills based on category and usage.
  - **CarRental:** Calculate rates for a car rental agency based on category and mileage.
  - **Trig:** Implement basic trigonometric functions using power series.
  - **LatinSquares:** Input two numbers: order and start. Build a Latin square of that order with that starting number.
  - **Number Theory: Persistence:** Numbers can have interesting properties. Here we look at additive persistence where you sum the digits of a number and then sum the digits of the sum—continue until you get a single digit.
  - **Number Theory: Pologinal:** The ancient Greeks found number properties to be interesting. A pologinal number is a particular geometric arrangement of a number of balls. Here we find two triangular numbers that form a given square number.
  - **Ancient Egyptian Multiplication:** Weird Multiplication: To multiply A by B repeatedly multiply A by 2 while dividing B by 2 until the division has a remainder. If B becomes odd, add A to it. Sum the resulting A and B.
  - **Calculator:** Make a simple, basic arithmetic calculator. Accept an expression in the form "number operator number", and calculate the result.
  - **Bank File Processing:** Read a file of banking transactions and keep a ledger.
- **Working With Strings**
  - **Gibberish :** Generate gibberish from English according to a particular pattern.
  - **WHO\_Measles :** Given a file of World Health Organization raw data, answer queries such as "what is the vaccination level for Measles in Africa?"
  - **Matching DNA :** Simple DNA sequence matching.
  - **Palindromic Integers:** Use the 196-algorithm to construct palindromic integers starting from any integer.

# Joseph's Beginner Python Projects

- **Number-Guessing Game:** Create a number-guessing game that provides hints on the number and position of correct digits in the guess (similar to Mastermind).
- **Hangman:** Create a hangman game (without the graphics of the hangman itself).
- **Mastermind:** Create a Mastermind game.
- **Caesar Cypher:** Encode and decode strings using a Caesar cypher.
- **Shannon's Throbac:** For fun Claude Shannon built a machine to do Roman Numeral arithmetic. We simulate it.
- **Rock Paper Scissors:** Create a game of Rock-Paper-Scissors.
- **Functions (simple)**
  - **Flag:** Draw the American flag using TurtleGraphics.
  - **Tesselations:** Tile a plane using TurtleGraphics.
  - **StringLibrary:** Implement a library of string functions (not using the built-in string functions).
  - **WeFeelFine:** Use functions from a module to gather data for analysis from the "We Feel Fine" page. The result is a collection of people's feelings scraped from blogs.
  - **GDP vs. Employment:** Using GDP and employment data from the web, prompt for a year and provide the data for that year.
  - **Scrambled Words:** Research shows that scrambled words can be read if the first and last letter are not changed, e.g. "Elingsh uinervtisy." Read in a file and scramble the words except first letters, last letters, and punctuation.
  - **Cracking a Caesar Cipher:** Decrypt Caesar Ciphers using letter frequencies.
- **Lists and Tuples**
  - **Risk:** Analyze Center for Disease Control data on health risks.
  - **Bestsellers:** Answer queries about bestsellers such as "What were the bestsellers in 2008?".
  - **Venus:** Determine suitable craters for mining on Venus using NASA data.
  - **Baseball:** Answer queries about a baseball season such as "list the players with the top 8 batting averages."
  - **Apple Stock :** Given a log file of Apple stock prices, calculate some simple statistics.
  - **Scrambled English:** Scramble English words into a scrambled, but readable, form.
  - **Basketball:** Who is best basketball player in the NBA? Get data from the web and analyze it.
  - **Latin:** Search for Latin roots, suffixes, and prefixes in a file.
  - **Anagrams:** Given a word find all the anagrams of that word in a word list.
  - **Data Mining:** Data Mining of online Google stock data.
  - **Build Query to Search Files:** Build a simple query to search files ignoring stop words.
  - **Equity:** Calculate Return on Equity (ROE) for stocks in a data file.
  - **Build HTML:** Build HTML files from a database of information.
  - **Sunspots:** Using NOAA data from the web analyze sunspot data to predict the next sunspot peak.
  - **Pascal's Triangle:** Construct Pascal's Triangle to a specified depth.
  - **Auto Mileage:** Using online data on automobiles, gather engine size and mileage data.

# Joseph's Beginner Python Projects

Use least squares to find a linear regression and correlation. Graph it.

- **Packaging:** Minimize the amount of cardboard used to box a set of glasses with specified dimensions.

## • Dictionaries and Sets

- **Document Retrieval:** Build a simple document retrieval system.
- **Poverty:** Use US Census county data for MI to answer queries such as what is the poverty rate for children in a particular county.
- **Wikipedia:** Parse a Wikipedia log file and respond to queries such as "who edited the most articles?"
- **TagCloud2:** Build a tag cloud of a 2012 presidential debate.
- **WordCompletion:** Simple word completion.
- **TagCloud:** Build a tag cloud of the 2008 vice-presidential debates.
- **Movies & Actors:** Using the online Internet Movies DataBase build a dictionary of movies and actors. Given two movies find all the actors in the movies (AND), find the common actors in the movies (OR), find the actors in either movie but not both (EXCLUSIVE\_OR). Given an actor, find all the co-actors that actor has acted with.
- **BMI: Body Mass Index:** Using real data from the Internet on a set of people, calculate their BMI, and then use least squares to find a linear regression and correlation. Extra credit: graph it.
- **Breast Cancer Classifier:** Build a cancer classifier from breast cancer study data to predict if a given cancer is malignant or not.
- **Income Classifier:** Build an income classifier from individuals' characteristics (over 30K records from the Internet) to predict if the individual's income will be greater than \$50K or not.
- **Vocabulary Quiz:** Build a vocabulary quiz program.
- **Concordance:** Build a simple concordance.
- **Sunspots:** Determine cycles in sunspot data (data goes back to 1749).
- **SpellCheck:** Build a file spell checker that includes suggestions.
- **Natural Language Processing:** Calculate unigrams and bigrams in a text file.
- **Natural Language Processing II:** Create Markov chains from a given text and then use the chains to create new text.
- **Jobs:** Grab data from data.gov and analyze private and public employment under different presidents.

## • Classes

- using instructor-designed classes to prepare for designing classes. We have found solitaire to be an excellent medium: we provide card and deck **classes**; students build a game. There are numerous variations on solitaire: e.g. **World Of Solitaire**.
- **FreeCell:** Solitaire.
- **CanField:** Solitaire.
- **SeaHaven:** Solitaire.
- **TexasHoldM:** Poker.
- **BlackJackSquare:** Solitaire with blackjack-style scoring.

# Joseph's Beginner Python Projects

- [EastHaven](#): A solitaire variation where cards from the stock go to all tableau columns.
  - [Golf Relaxed](#): Another solitaire variation.
  - [Aces Up](#): A solitaire variation that is a kids game.
  - [Spider](#): Yet, another solitaire variation.
- **Class Design**
    - [Currency](#) : Currency calculator based on queries to Google for the latest rates.
    - [GoogleMap](#) : Query Google Maps to get a route between cities.
    - [TurtleSnowPeople](#) : Create a set of classes of graphical components for snow people.
    - [Flags](#) : Create a set of classes of graphical components for Flags.
    - [Quadrilaterals](#) : Given an outline of a class, fill in the details.
    - [YouTube](#) : Design classes to support using the Google API to make queries on YouTube.
    - [Triangles](#) : Given an outline of a class, fill in the details.
    - [Elevator Simulator](#): Create three classes (Building, Elevator, and Customer) to use in simulating an elevator.
    - [Face Drawing Program](#): Create five classes of facial features (using Turtle Graphics) and build a face drawing tool around them.
    - [Minesweeper](#): Create a Minesweeper class and design a Minesweeper game around it.
    - [Protein Transcription](#): Create a transcriber class so you can take in a sequence of DNA bases, use a mapping of amino acids to codons, and yield a sequence of amino acids.
    - [Queue Simulation](#): There are four ATM machines in the student union. Should students form one queue or four? Create an event class and build an event-driven simulation around that class. Use the simulator to answer the question.
    - [Game of Life](#): Create a class for the Game of Life and then build the game around it.
    - [Traveler's Dilemma](#): Write a program that plays the Travler's Dilemma using at least one class and one exception. Test your game with ten different pairs of people. Analyze the results.
    - [Dilbert's Carpet Fishing](#): A Dilbert cartoon has him playing an office game he calls "Carpet Fishing." Using at least one class, write the game.
    - [Turtle Classes](#): Create a set of classes of graphical components of your own design for use in creating a TurtleGraphics picture of your own design.
    - [Dating Service](#) : Create a dating service with your own matching algorithm supported by classes of your own design.
    - [Dominos](#): Create a Dominos game supported by classes of your own design.
    - [TurtleCar](#): Create a library of classes to support drawing a car using Turtle graphics.

## Python Programming Examples

1. ✓ [Python Program to Print Hello world!](#)
2. ✓ [Python Program to Add Two Numbers](#)

# Joseph's Beginner Python Projects

3. ✓ [Python Program to Find the Square Root](#)
4. [Python Program to Calculate the Area of a Triangle](#)
5. [Python Program to Solve Quadratic Equation](#)
6. [Python Program to Swap Two Variables](#)
7. [Python Program to Generate a Random Number](#)
8. [Python Program to Convert Kilometers to Miles](#)
9. [Python Program to Convert Celsius To Fahrenheit](#)
10. [Python program to check if a number is positive, negative or zero](#)
11. [Python Program to Check if a Number is Odd or Even](#)
12. [Python Program to Check Leap Year](#)
13. [Python Program to Find the Largest Among Three Numbers](#)
14. [Python Program to Check Prime Number](#)
15. [Python Program to Print all Prime Numbers in an Interval](#)
16. [Python Program to Find the Factorial of a Number](#)
17. [Python Program to Display the multiplication Table](#)
18. [Python Program to Print the Fibonacci sequence](#)
19. [Python Program to Check Armstrong Number](#)
20. [Python Program to Find Armstrong Number in an Interval](#)
21. [Python Program to Find the Sum of Natural Numbers](#)
22. [Python Program To Display Powers of 2 Using Anonymous Function](#)

# Joseph's Beginner Python Projects

23. [Python Program to Find Numbers Divisible by Another Number](#)
24. [Python Program to Convert Decimal to Binary, Octal and Hexadecimal](#)
25. [Python Program to Find ASCII Value of Character](#)
26. [Python Program to Find HCF or GCD](#)
27. [Python Program to Find LCM](#)
28. [Python Program to Find Factors of Number](#)
29. [Python Program to Make a Simple Calculator](#)
30. [Python Program to Shuffle Deck of Cards](#)
31. [Python Program to Display Calendar](#)
32. [Python Program to Display Fibonacci Sequence Using Recursion](#)
33. [Python Program to Find Sum of Natural Numbers Using Recursion](#)
34. [Python Program to Find Factorial of Number Using Recursion](#)
35. [Python Program to Convert Decimal to Binary Using Recursion](#)
36. [Python Program to Add Two Matrices](#)
37. [Python Program to Transpose a Matrix](#)
38. [Python Program to Multiply Two Matrices](#)
39. [Python Program to Check Whether a String is Palindrome or Not](#)
40. [Python Program to Remove Punctuations From a String](#)
41. [Python Program to Sort Words in Alphabetic Order](#)

# Joseph's Beginner Python Projects

42. [Python Program to Illustrate Different Set Operations](#)
43. [Python Program to Count the Number of Each Vowel](#)
44. [Python Program to Merge Mails](#)
45. [Python Program to Find the Size \(Resolution\) of a Image](#)
46. [Python Program to Find Hash of File](#)

## Weekly beginner Python exercises

- a. [Pick Word](#) 
- b. [Tic Tac Toe Game](#) 
- c. [Max Of Three](#) 
- d. [Tic Tac Toe Draw](#) 
- e. [Check Tic Tac Toe](#) 
- f. [Guessing Game Two](#) 
- g. [Draw A Game Board](#) 
- h. [File Overlap](#) 
- i. [Read From File](#) 
- j. [Write To A File](#) 
- k. [Element Search](#) 
- l. [Decode A Web Page Two](#) 

# Joseph's Beginner Python Projects

- m. [Cows And Bulls](#) 
- n. [Decode A Web Page](#) 
- o. [Password Generator](#) 
- p. [Reverse Word Order](#) 
- q. [List Remove Duplicates](#) 
- r. [Fibonacci](#) 
- s. [List Ends](#) 
- t. [Check Primality Functions](#) 
- u. [List Overlap Comprehensions](#) 
- v. [Guessing Game One](#) 
- w. [Rock Paper Scissors](#) 
- x. [List Comprehensions](#) 
- y. [String Lists](#) 
- z. [List Overlap](#) 
- aa. [Divisors](#) 
- bb. [List Less Than Ten](#) 
- cc. [Odd Or Even](#) 
- dd. [Character Input](#) 

# Joseph's Beginner Python Projects

## Project List

### Text Based Adventure Game:

*Overview:* Remember *Adventure*? Well, we're going to build a more basic version of that. A complete text game, the program will let users move through rooms based on user input and get descriptions of each room. To create this, you'll need to establish the directions in which the user can move, a way to track how far the user has moved (and therefore which room he/she is in), and to print out a description. You'll also need to set limits for how far the user can move. In other words, create "walls" around the rooms that tell the user, "You can't move further in this direction." The tricky parts here will involve setting up the directions and keeping track of just how far the user has "walked" in the game. I suggest sticking to just a few basic descriptions or rooms, perhaps 6 at most. This project also continues to build on using user-inputted data. It can be a relatively basic game, but if you want to build this into a vast, complex world, the coding will get substantially harder, especially if you want your user to start interacting with actual objects within the game.

*Concepts to keep in mind:* Strings, Variables, Input/Output, If/Else Statements, Print, List, Integers

### Sentence Generator

*Overview:* A series of different parts of sentences will be randomly put together to come up with new interesting sentences.

*What you will be Using:* Random, Integers, Print, Strings, Breaks, Functions, For, Range

*My Thoughts on Project:* A very fun beginning project, have fun with this... Come up with some wacky sentences! Also, this project really gets you to experiment with strings, piecing them together, randoming from a set of strings, and more.

*Download my Source Code:* <http://www.mediafire.com/?2dhtt0ynjdy>

### Area Calculator (in progress)

*Overview:* The user will be prompted with a menu where he/she will select a shape. Then the user will give the appropriate information needed to solve for the area, and the computer will give the area! Hope you all have taken geometry!

*What you will be Using:* Input/Output, Integers, Variables, Strings, Print, If/Elif/Else

*My Thoughts on Project:* Great if you understand geometry and want to write a program that will do a little homework for you! This program is great for learning variables and creation of math related projects.

*Download my Source Code:* <http://www.mediafire.com/?m2mm2mjxiy9>

### Address Book

*Overview:* The user wants to create an address book and downloads your program. How would you make it? Create a program that prompts the user for the information in most address

# Joseph's Beginner Python Projects

books and then stores it in a .txt file!

*What you will be Using:* Input/Output, Print, Python File Commands, If/Elif/Else

*My Thoughts on Project:* This is a great intermediate Python project once you are really going with Python. This will teach you how to access files, edit them, save them, delete them, and more. If you get stuck you can look at my source code but I strongly suggest that you use Google and then try to apply what you find into your program.

*Download my Source Code:* <http://www.mediafire.com/?fy9mx3w02dz>

## ✓ Guess My Number

*Overview:* The computer randomly generates a number. The user inputs a number, and the computer will tell you if you are too high, or too low. Then you will get to keep guessing until you guess the number.

*What you will be Using:* Random, Integers, Input/Output, Print, While (Loop), If/Elif/Else

*Download my Source Code:* <http://www.mediafire.com/?iahny9tgk0m>