**Game 24 using Python**

24 pulls on a range of mathematical thinking and skills, including addition, subtraction, multiplication, division, order of operations, and number sense. Using Python to play 24 is a creative and fun way to practice math skills whole class or individually while also introducing students to the python language and how to run programs using IDLE.

For this program I used IDLE, a python environment that can be found on the Python website under downloads, <https://www.python.org/downloads/>.

1. Open a new file

 

1. Copy and paste this code into the new file

'''

 The 24 Game

 Given any four digits in the range 1 to 9, which may have repetitions,

 Using just the +, -, \*, and / operators; and the possible use of

 brackets, (), show how to make an answer of 24.

 An answer of "!" will generate a new set of four digits (if your stuck).

 Otherwise you are repeatedly asked for an expression until it evaluates to 24

 Note: you cannot form multiple digit numbers from the supplied digits,

 so an answer of 12+12 when given 1, 2, 2, and 1 would not be allowed.

'''

from \_\_future\_\_ import division, print\_function

import random, ast, re

import sys

if sys.version\_info[0] < 3: input = raw\_input

def choose4():

 'four random digits >0 as characters'

 return [str(random.randint(1,9)) for i in range(4)]

def welcome(digits):

 print (\_\_doc\_\_)

 print ("Your four digits: " + ' '.join(digits))

def check(answer, digits):

 allowed = set('() +-\*/\t'+''.join(digits))

 ok = all(ch in allowed for ch in answer) and \

 all(digits.count(dig) == answer.count(dig) for dig in set(digits)) \

 and not re.search('\d\d', answer)

 if ok:

 try:

 ast.parse(answer)

 except:

 ok = False

 return ok

def main():

 digits = choose4()

 welcome(digits)

 trial = 0

 answer = ''

 chk = ans = False

 while not (chk and ans == 24):

 trial +=1

 answer = input("Expression %i: " % trial)

 chk = check(answer, digits)

 if answer.lower() == 'q':

 break

 if answer == '!':

 digits = choose4()

 print ("New digits:", ' '.join(digits))

 continue

 if not chk:

 print ("The input '%s' was wonky!" % answer)

 else:

 ans = eval(answer)

 print (" = ", ans)

 if ans == 24:

 print ("Thats right!")

main()

It should look like this:



1. Save the program as “game\_24.py”; you can either do this by going to the file menu or shortcut keys, ie. Command + s on a Mac; Conrol + s on a PC
2. Run the Module. When you click “Run Module” or F5, the Python Shell will appear and look like the image in Step 5
3. Lastly! Play the game