**Tarea 2**

**Mario Moreno Zamora**

**Para esta tarea se siguen casi mismos lineamientos empleados en la Tarea de Vacaciones, es decir, se debe crear un directorio que se llama "Tarea Semana 2" y dentro debe almacenarse este notebook con la nomenclatura con la salvedad de que si se trata de una pregunta de implementación se debe colocar en ese mismo ejercicio la resolución y si se aborda una pregunta abierta se debe colocar la fuente.**

**De estas últimas todas deben colocarse en Markdown (siguiendo los lineamientos de visualización) o de lo contrario no se valdrá el crédito.**[**¶**](https://render.githubusercontent.com/view/ipynb?commit=aaaf9b4ba9377d76a34db589a6ad00b34d7d845d&enc_url=68747470733a2f2f7261772e67697468756275736572636f6e74656e742e636f6d2f596f4e6f4675692f617175696e6f6861796e6164612f616161663962346261393337376437366133346462353839613661643030623334643764383435642f796f6e6f6675692f53656d616e61253230322f546172656125323053656d616e61253230322e6970796e62&nwo=YoNoFui%2Faquinohaynada&path=yonofui%2FSemana+2%2FTarea+Semana+2.ipynb&repository_id=137132890&repository_type=Repository#De-estas-%C3%BAltimas-todas-deben-colocarse-en-Markdown-(siguiendo-los-lineamientos-de-visualizaci%C3%B3n)-o-de-lo-contrario-no-se-valdr%C3%A1-el-cr%C3%A9dito.)

*#Ejercicio 1*

*#A partir de la siguiente matriz:*

matriz = [[1,2,3,4,5],

[6,7,8,9,10],

[11,12,13,14,15],

[16,17,18,19,20]]

*"""*

*Imprima los elementos de esta manera:*

*1 2 3 4 5*

*6 7 8 9 10*

*11 12 13 14 15*

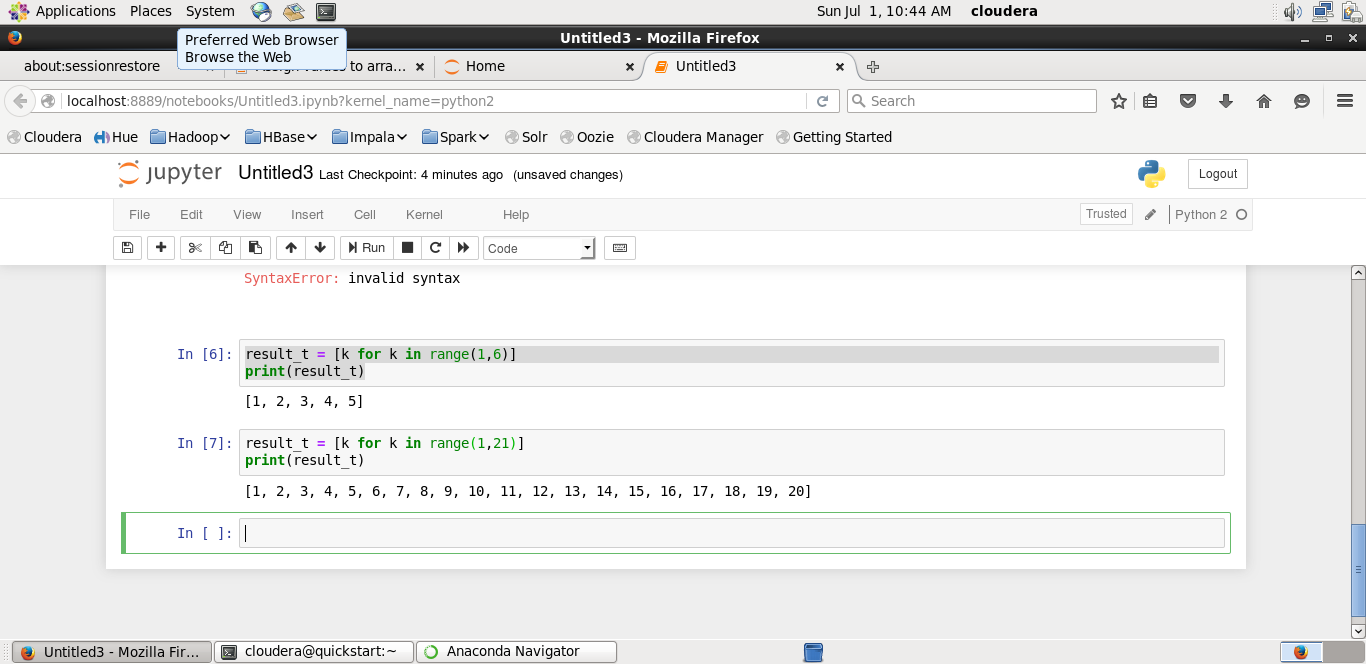
*16 17 18 19 20*

*Usando solamente un ciclo (while, for); no hay restricción de if-else, breaks, etc.*

**Result\_t = [k for k in range (1,21)]**

**Print (result\_t)**

**Result\_t = []**

****

*#Ejercicio 3*

*"""*

*En general en los lenguajes de programación una de las prácticas recomendadas para nombrar ya sea variables o funciones es*

*el "Camel Case" el cual consiste en sustituir posibles espacios con letras mayúsculas (la primera letra de la frase por lo*

*regular no se toma en cuenta), por ejemplo:*

*este es el nombre de mi variable => esteEsElNombreDeMiVariable*

*aqui esta otro nombre para la variable => aquiEstaOtroNombreParaLaVariable*

*¿Por qué Camel? porque la combinación de mayúsculas y minúsculas simulan la joroba de un camello (dato curioso debería de ser*

*un dromedario y no un camello porque el camello tiene una joroba y el dromedario dos ;))*

*Entonces, para este ejercicio teniendo en cuenta esta cadena:*

*En algun lugar de La Mancha de cuyo nombre no quiero acordarme, o ha mucho tiempo que vivia un hidalgo de los de lanza en*

*astillero, adarga antigua, rocin flaco y galgo corredor.*

*Se debe convertir bajo la nomenclatura Camel Case.*

*Para este ejercicio en particular la primera palabra de la frase se deja intacta mientras que los signos de puntuación*

*(exceptuando el último) se deben remover.*

*Sugerencias: las funciones lower() & upper() convierten una letra a minúsculas y mayúsculas respectivamente, si*

*se da el caso de que ya estaban convertidas a los tipos que se deseaban transformar no arroja error.*

*Por otra parte las funciones islower() & isupper() indican si una letra está en su modalidad minúscula o mayúscula*

*respectivamente.*

*"""*

cadena = "En algun lugar de La Mancha de cuyo nombre no quiero acordarme, o ha mucho tiempo que vivia un hidalgo de los de lanza en astillero, adarga antigua, rocin flaco y galgo corredor."

**Cadena = "En algun lugar de La Mancha de cuyo nombre no quiero acordarme, o ha mucho tiempo que vivia un hidalgo de los de lanza en astillero, adarga antigua, rocin flaco y galgo corredor."**

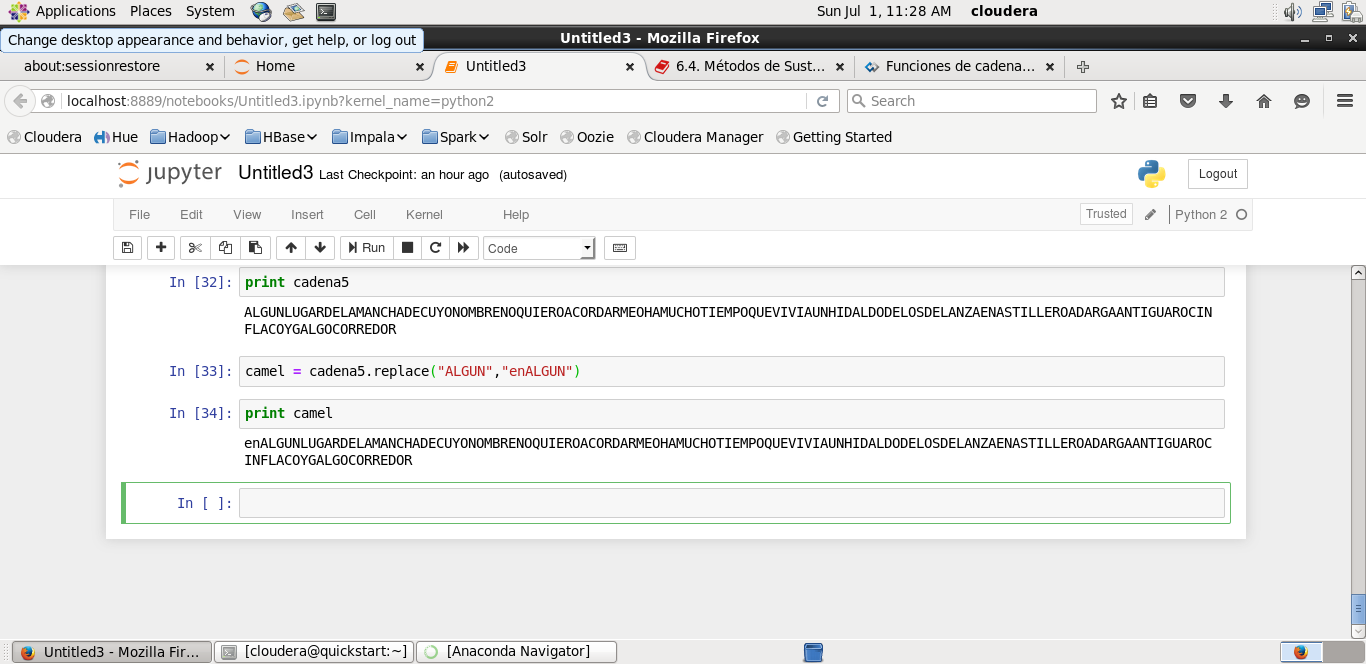
**Cadena2 = cadena.upper()**

**Cadena3=cadena2.replace (“,”,” ”)**

**Cadena4 = cadena3.replace(“ ”,””)**

**Cammel = cadena4.replace(“ENALGUN”,”enALGUN”)**

**Print cammel**

****

*#Ejercicio 4*

*"""*

*Se define "raíz digital" a la suma recursiva de todos los dígitos de un número hasta llegar a uno solo, por ejemplo*

*raiz\_digital(16)*

*=> 1 + 6*

*=> 7*

*raiz\_digital(942)*

*=> 9 + 4 + 2*

*=> 15 ...*

*=> 1 + 5*

*=> 6*

*Entonces el objetivo consiste en implementar el algoritmo de la raiz digital y probar con el número 493193, cuyo resultado*

*debe ser 2. Sugerencia: la forma recursiva es la más fácil de implementar.*

*Curso rápido para ingresar funciones en Python; basta con declarar:*

*#En def no hay tipo de retorno de dato como en otros lenuajes.*

*def nombre\_funcion(parametro\_1,...,parametro\_n):*

*#El return del tipo de dato no cuenta en Python.*

*return algo*

*"""*

**Def raíz\_digital (numero):**

**Suma = sum (int(digito) for digito in str (numero))**

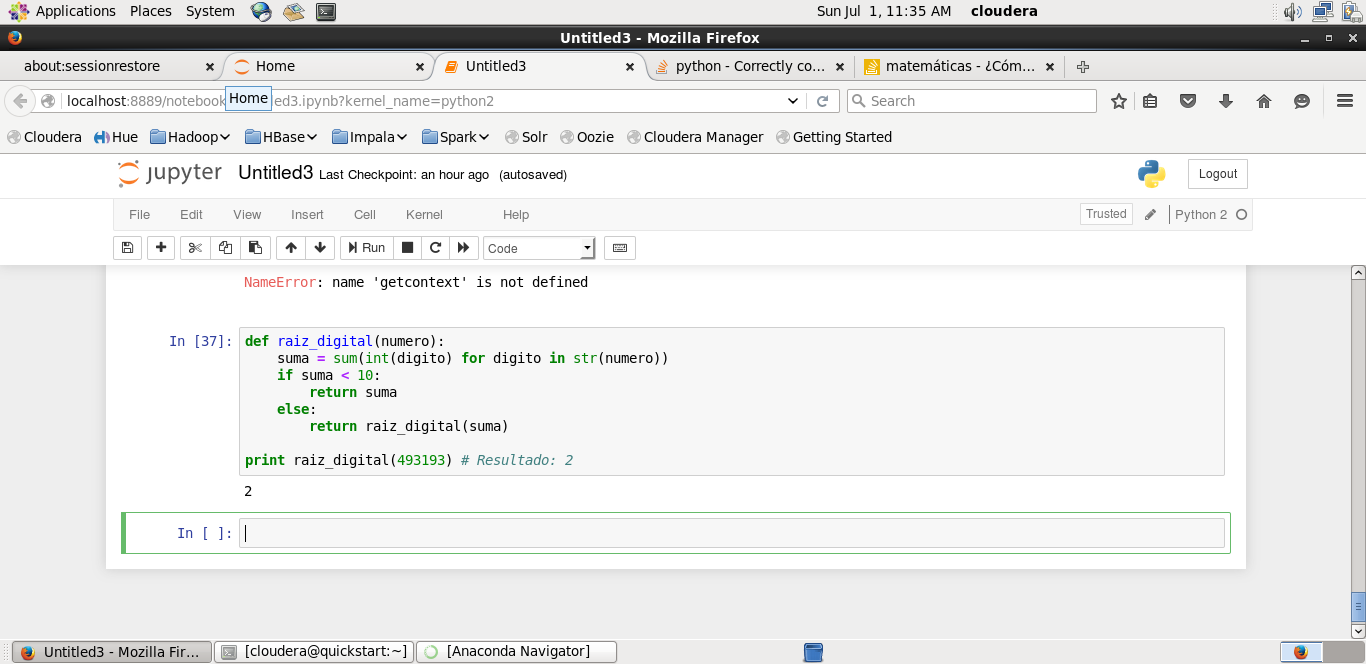
**If suma < 10:**

**Return suma**

**Else:**

**Return raiz\_digitial (suma)**

**Print raiz\_digital (493193)**

****

*#Ejercicio 7*

*"""*

*Suponga que nos encontramos en una fiesta de graduación de preparatoria, de aquéllas lindas de antaño y no*

*una conjunción de pecado como American Pie.*

*También asumamos que existen 3 tipos de personas (no nos metemos con faltas a la suposición de género): hombres (H)*

*mujeres(M) y chaperones (C).*

*Como su nombre lo dice, un chaperón debe interponerse entre un hombre y una mujer para que no existan situaciones*

*lascivas. Para nuestro caso, teniendo una reunión (lista) del tipo:*

*[H,M,H,M,H,M,H,M,H,M,H,M]*

*Deben colocarse chaperones cada vez que se encuentren una correspondencia del tipo HM o MH,*

*en nuestro caso la solución sería:*

*[H,C,M,C,H,C,M,C,H,C,M,C,H,C,M,C,H,C,M,C,H,C,M]*

*Aunque lo anterior es sólo un ejemplo pueden darse casos en el que existan parejas MM y HH.*

*Entonces, ¿en qué consiste el ejercicio?, en que dada la siguiente reunión (lista):*

*[H,M,M,H,M,M,M,H,M,H,H,M,H,M,H,M,H,M,M,H,M,H,M,M,H,M,H,M,H,H,M,M,H,M,H,M,M,H,M,H,M,H,M,H,M,H,M]*

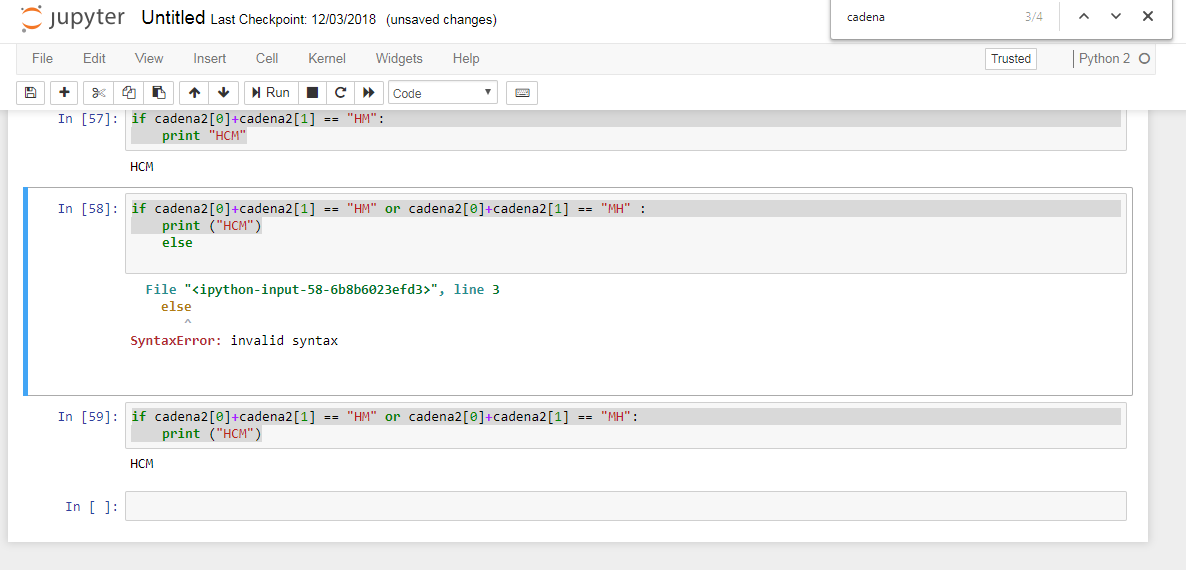
*Se debe imprimir una lista resultado con todos los chaperones puestos apropiadamente para evitar la situación lasciva*

*Mencionada.*

*"""*

**if cadena2[0]+cadena2[1] == "HM" or cadena2[0]+cadena2[1] == "MH":**

**print ("HCM")**



*#Ejercicio 9*

*"""*

*Con base en el archivo ejercicio\_5.txt de la tarea anterior cree un dataframe de Pandas considerando*

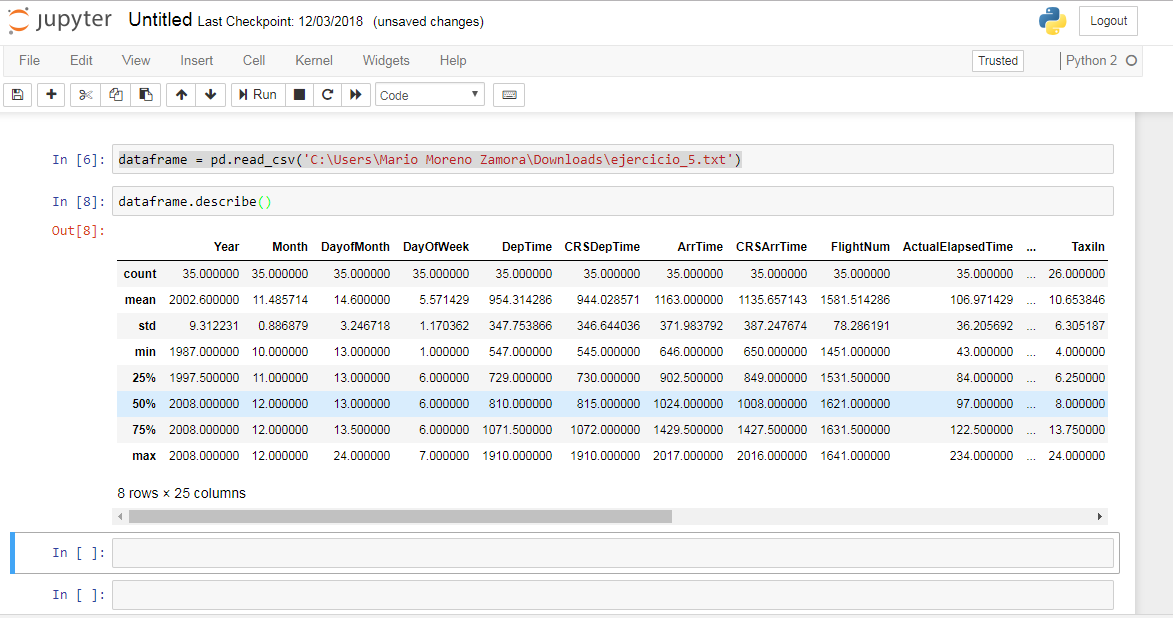
*el encabezado, por ahora no se preocupe del tipo de dato.*

*"""*

**import pandas as pd**

**dataframe = pd.read\_csv('C:\Users\Mario Moreno Zamora\Downloads\ejercicio\_5.txt')**

**dataframe.describe()**



*#Ejercicio 10*

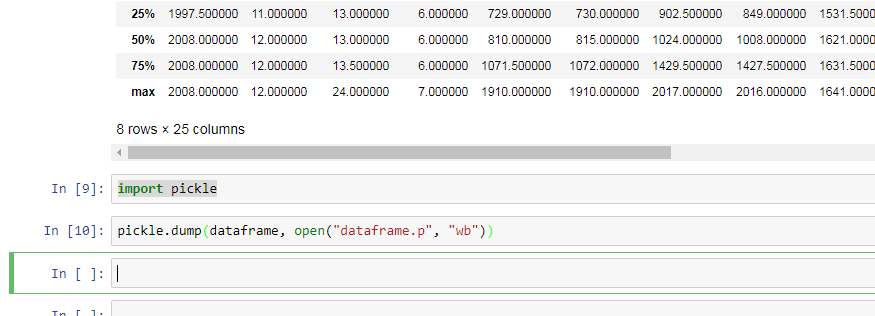
*"""Tome el dataframe de Pandas del ejercicio anterior, guárdelo como un archivo de tipo .pickle con el nombre:*

*<su\_usuario\_de\_git> y colóquelo en el subdirectorio "Tarea Semana 2"*

*"""*

**import pickle**

**pickle.dump(dataframe, open("dataframe.p", "wb"))**



*#Ejercicio 11*

"""

Para este ejercicio se debe implementar un programa que tome el lenguaje natural y lo convierta a código Morse:

https://www.electronics-notes.com/articles/ham\_radio/morse\_code/characters-table-chart.php

Usando solamente un ciclo for/while; if-else y demás no tienen limitantes.

Para las pruebas de traducción exitosa se puede ocupar este ejemplo:

Creo que aquello en lo que nos convertimos depende de lo que nuestros padres nos enseñan en pequeños momentos,

cuando no están intentando enseñarnos. Estamos hechos de pequeños fragmentos de sabiduría.

Y el siguiente sitio:

https://morsecode.scphillips.com/translator.html?utm\_source=hootsuite

NOTA: puede deshacerse de los signos de puntuación pero no de las letras acentuadas, por otro lado las mayúsculas

y minúsculas se comportan igual (upper(), lower()).

En este caso puede convertir las letras "á" ó "à" o "â" por "a" y la "ñ" por "n".

# Python program to implement Morse Code Translator

'''

VARIABLE KEY

'cipher' -> 'stores the morse translated form of the english string'

'decipher' -> 'stores the english translated form of the morse string'

'citext' -> 'stores morse code of a single character'

'i' -> 'keeps count of the spaces between morse characters'

'message' -> 'stores the string to be encoded or decoded'

'''

# Dictionary representing the morse code chart

MORSE\_CODE\_DICT = { 'A':'.-', 'B':'-...',

                    'C':'-.-.', 'D':'-..', 'E':'.',

                    'F':'..-.', 'G':'--.', 'H':'....',

                    'I':'..', 'J':'.---', 'K':'-.-',

                    'L':'.-..', 'M':'--', 'N':'-.',

                    'O':'---', 'P':'.--.', 'Q':'--.-',

                    'R':'.-.', 'S':'...', 'T':'-',

                    'U':'..-', 'V':'...-', 'W':'.--',

                    'X':'-..-', 'Y':'-.--', 'Z':'--..',

                    '1':'.----', '2':'..---', '3':'...--',

                    '4':'....-', '5':'.....', '6':'-....',

                    '7':'--...', '8':'---..', '9':'----.',

                    '0':'-----', ', ':'--..--', '.':'.-.-.-',

                    '?':'..--..', '/':'-..-.', '-':'-....-',

                    '(':'-.--.', ')':'-.--.-'}

# Function to encrypt the string

# according to the morse code chart

def encrypt(message):

    cipher = ''

    for letter in message:

        if letter != ' ':

            # Looks up the dictionary and adds the

            # correspponding morse code

            # along with a space to separate

            # morse codes for different characters

            cipher += MORSE\_CODE\_DICT[letter] + ' '

        else:

            # 1 space indicates different characters

            # and 2 indicates different words

            cipher += ' '

    return cipher

# Function to decrypt the string

# from morse to english

def decrypt(message):

    # extra space added at the end to access the

    # last morse code

    message += ' '

    decipher = ''

    citext = ''

    for letter in message:

        # checks for space

        if (letter != ' '):

            # counter to keep track of space

            i = 0

            # storing morse code of a single character

            citext += letter

        # in case of space

        else:

            # if i = 1 that indicates a new character

            i += 1

            # if i = 2 that indicates a new word

            if i == 2 :

                 # adding space to separate words

                decipher += ' '

            else:

                # accessing the keys using their values (reverse of encryption)

                decipher += list(MORSE\_CODE\_DICT.keys())[list(MORSE\_CODE\_DICT

                .values()).index(citext)]

                citext = ''

    return decipher

# Hard-coded driver function to run the program

def main():

message = "WE LOVE BIG DATA"

result = encrypt(message.upper())

print (result)

message = "-- .- .-. .. --- -- --- .-. . -. --- --.. .- -- --- .-. .- "

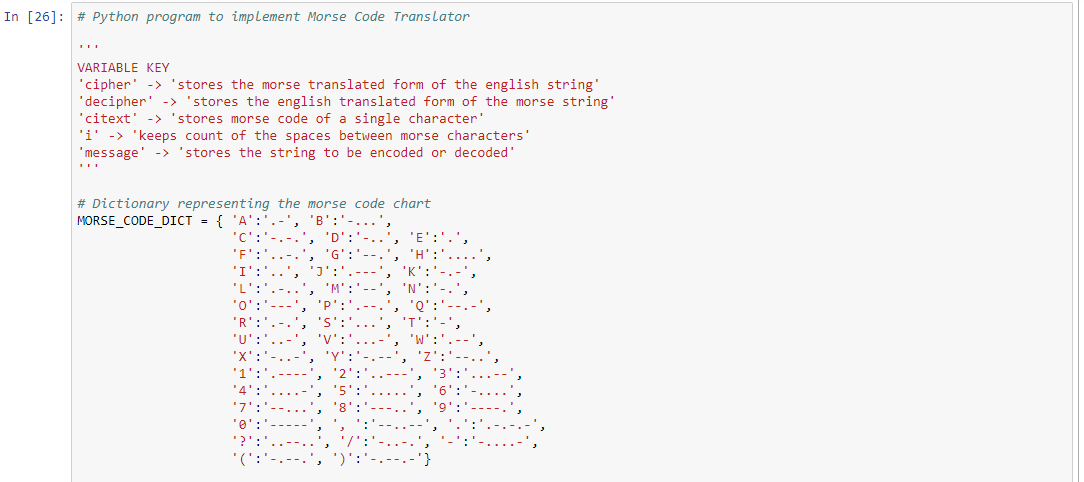
result = decrypt(message)

print (result)

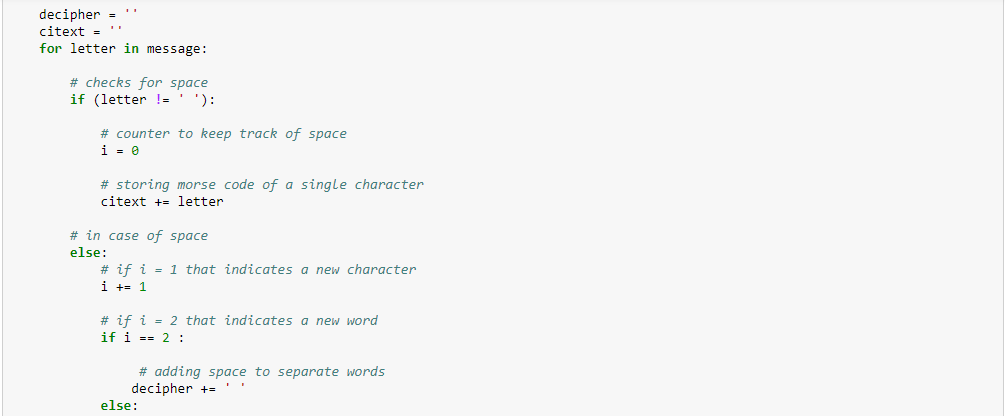
# Executes the main function

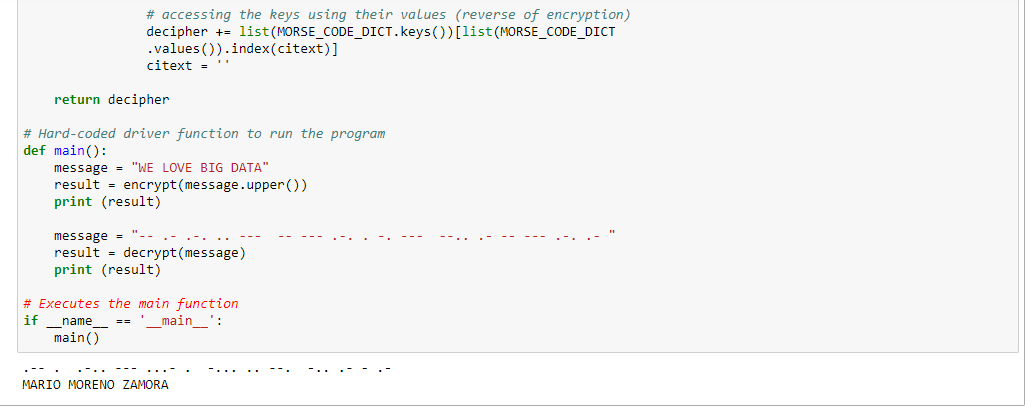
if \_\_name\_\_ == '\_\_main\_\_':

main()









Fuente: <http://localhost:8888/notebooks/Untitled.ipynb?kernel_name=python2>

*#Ejercicio 12*

*"""*

*Para repasar el uso de funciones recursivas, implemente la función de Ackerman. Muestre en pantalla el resultado de :*

*ackerman(3,4)*

*NOTA: puede crear una función (precisamente llamada ackerman) para una mejor presentación del código.*

*"""*

def ackermann(m,n):

if m == 0:

return (n + 1)

elif n == 0:

return ackermann(m - 1, 1)

else:

return ackermann(m - 1, ackermann(m, n - 1))

x=int(input("What is the value for m? "))

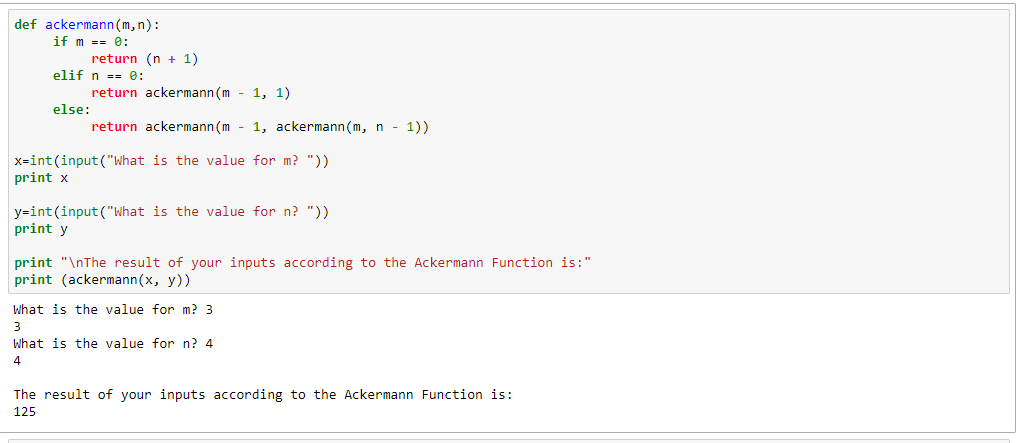
print x

y=int(input("What is the value for n? "))

print y

print "\nThe result of your inputs according to the Ackermann Function is:"

print (ackermann(3, 4))



Fuente: <http://pythonfiddle.com/ackermanns-function/>

*#Ejercicio 13.*

*"""*

*Investigue e implemente la criba de Eratóstenes; hágase valer de todas las funcionalidades vistas*

*hasta ahora (if-else, while, for, diccionarios, listas, etc.)*

*NOTA: para este ejercicio debe colocar 2 casillas, una para la explicación del tema EN MARKDOWN*

*y la otra para la implementación del mismo.*

*"""*

def prime\_eratosthenes(n):

prime\_list = []

for i in range(2, n+1):

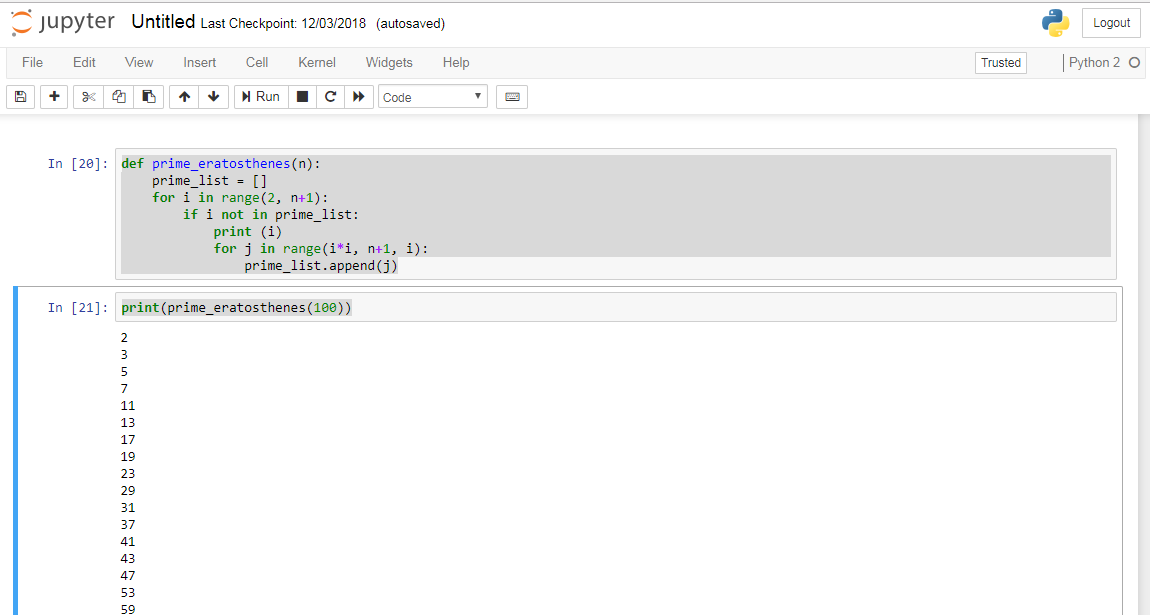
if i not in prime\_list:

print (i)

for j in range(i\*i, n+1, i):

prime\_list.append(j)

print(prime\_eratosthenes(100))



Fuente: <https://www.w3resource.com/python-exercises/list/python-data-type-list-exercise-34.php>