Python for Beginner
Hack The Caesar Cipher

a Python practical guide by Tan Pham

Download Best Udemy Dev Course....2
Intro....3
Python Essental 4
Installation....4
Python Install....4
Pycharm IDE Installation....5
If Statement....6
Example check a number is even or odd....6
Hint....6
Solution....7
For Statement....7
Exmple : Count the Number of Even and Odd....7
Hint....7
Solution....7
String....8
Function....8
Encrypt Message with Caesar Cipher....9
Manually Encrypt....9
Automate Encrpyt....10
Decrypt Caesar Cipher Message....11
Manually Decrypt....11
Automate Decrypt....12
Hack Caesar Cipher Message....14
The Brute-Force Attack....14

Download Best Udemy Dev Course

http://devcourses.co

What is Good Course Look Like ?

- Start from Beginner

- Step by Step Explained Method
- Alot of Example to Make Thing Clear
- Alot of Exercise, Quize
- Real Life Hand on Projects Solve Real Problems
- Boot up to Launch New Carrier

Intro

The Caesar cipher is named after Julius Caesar, who, according to Suetonius, used it with a shift of three to protect messages of military significance. While Caesar's was the first recorded use of this scheme, other substitution ciphers are known to have been used earlier.[4][5]

"If he had anything confidential to say, he wrote it in cipher, that is, by so changing the order of the letters of the alphabet, that not a word could be made out. If anyone wishes to decipher these, and get at their meaning, he must substitute the fourth letter of the alphabet, namely D, for A, and so with the others."

— Suetonius, *Life of Julius Caesar* 56

His nephew, Augustus, also used the cipher, but with a right shift of one, and it did not wrap around to the beginning of the alphabet:

"Whenever he wrote in cipher, he wrote B for A, C for B, and the rest of the letters on the same principle, using AA for Z."

— Suetonius, *Life of Augustus* 88

Evidence exists that Julius Caesar also used more complicated systems,[6] and one writer, Aulus Gellius, refers to a (now lost) treatise on his ciphers:

"There is even a rather ingeniously written treatise by the grammarian Probus concerning the secret meaning of letters in the composition of Caesar's epistles."

— Aulus Gellius, *Attic Nights 17.9.1–5*

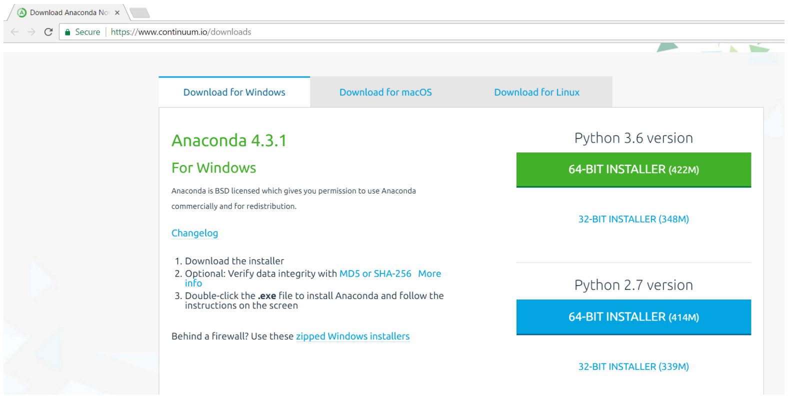
It is unknown how effective the Caesar cipher was at the time, but it is likely to have been reasonably secure, not least because most of Caesar's enemies would have been illiterate and others would have assumed that the messages were written in an unknown foreign language.[7] There is no record at that time of any techniques for the solution of simple substitution ciphers. The earliest surviving records date to the 9th century works of Al-Kindi in the Arab world with the discovery of frequency analysis.[8]

Python Essental

Installation

Python Install

We will install python with anaconda distribution, I recommand this because all the package already tested carefully. Go to https://anaconda.org/ and click to download



Anaconda 4.3.1

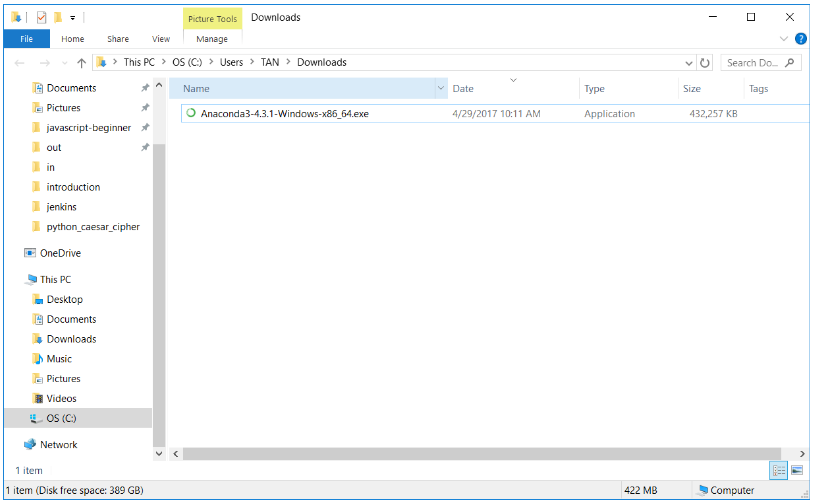
For Windows

Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.

1. Download the installer
2. Optional: Verify data integrity with MD5 or SHA-256 More info
3. Double-click the **.exe** file to install Anaconda and follow the instructions on the screen

Behind a firewall? Use these zipped Windows installers

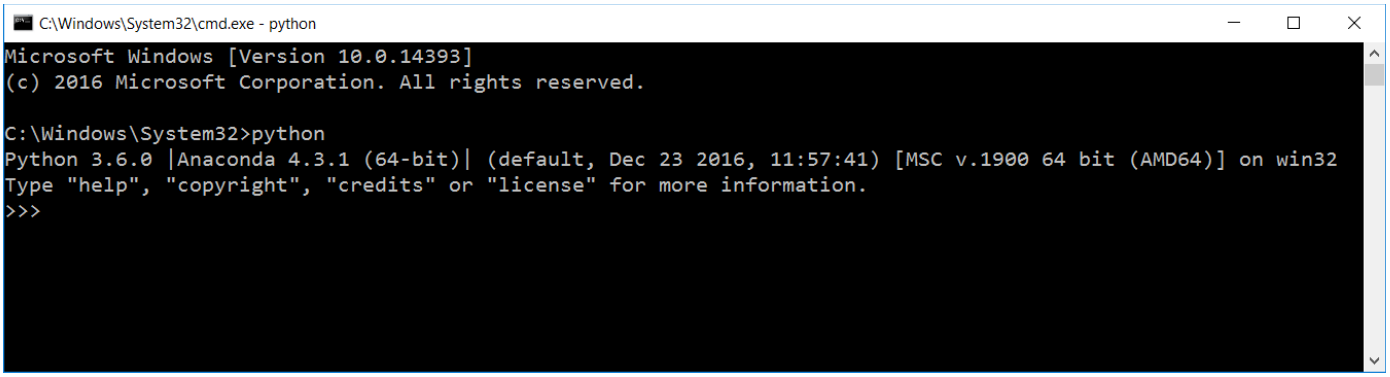
Click to 64 bit and Python 3.6 version and download installer to local



Name	Date	Type	Size	Tags
Anaconda3-4.3.1-Windows-x86_64.exe	4/29/2017 10:11 AM	Application	432,257 KB	

Doublick to installer and install with default option.

After install open command prompt and type python to check if install is successfully



C:\Windows\System32\cmd.exe - python

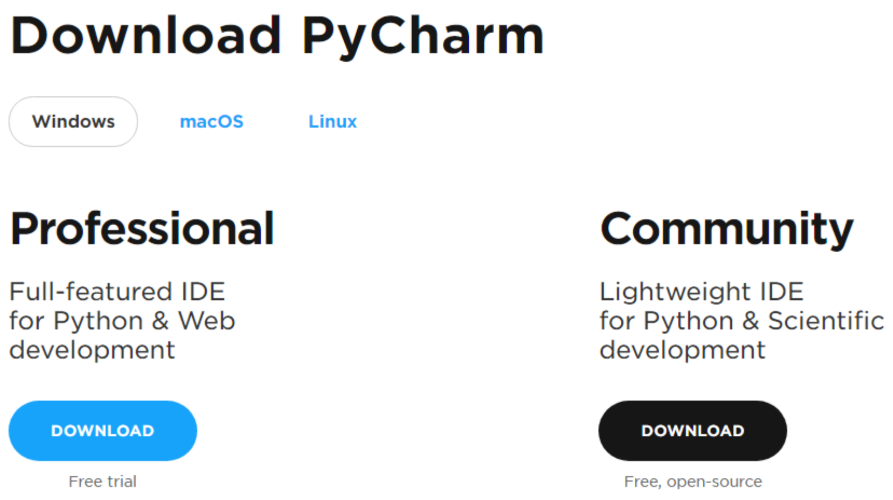
```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\System32>python
Python 3.6.0 |Anaconda 4.3.1 (64-bit)| (default, Dec 23 2016, 11:57:41) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Pycharm IDE Installation

Pycharm is really good IDE to working with python programming language. In this section I will show you how to install Pycharm IDE.

Go to https://www.jetbrains.com/pycharm/download/#section=windows

and click to download community version



Download PyCharm

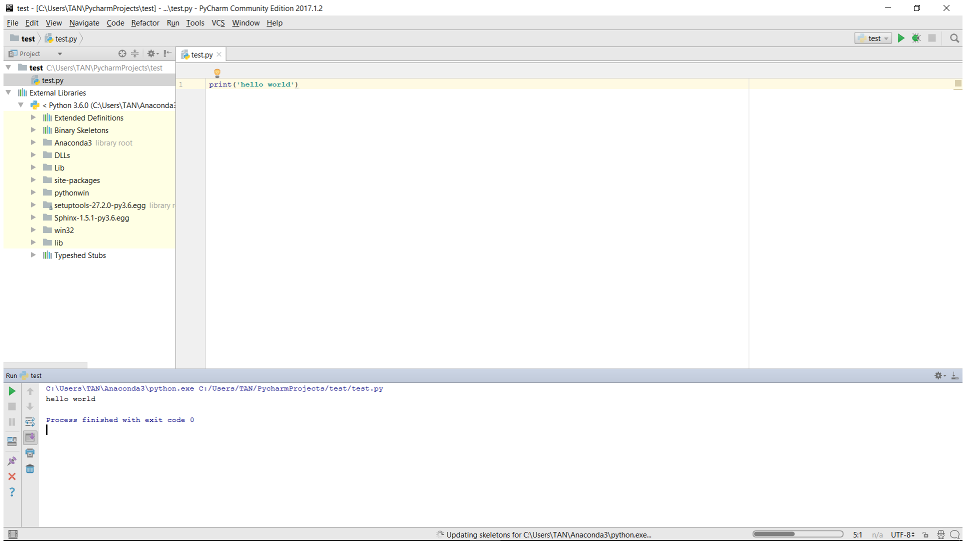
Professional

Full-featured IDE for Python & Web development

Community

Lightweight IDE for Python & Scientific development

After install with default option, try to call Pycharm



test - [C:\Users\TAN\PycharmProjects\test] - ...\test.py - PyCharm Community Edition 2017.1.2
File Edit View Navigate Code Refactor Run Tools VCS Window Help
test
test.py
Project
test C:\Users\TAN\PycharmProjects\test
test.py
External Libraries
< Python 3.6.0 (C:\Users\TAN\Anaconda3
Extended Definitions
Binary Skeletons
Anaconda3 library root
DLLs
Lib
site-packages
pythonwin
setuptools-27.2.0-py3.6.egg library r
Sphinx-1.5.1-py3.6.egg
win32
lib
Typeshed Stubs
test.py
Run test
5:1 n/a UTF-8

```
print('hello world')
```

```
C:\Users\TAN\Anaconda3\python.exe C:/Users/TAN/PycharmProjects/test/test.py
hello world

Process finished with exit code 0
```

Updating skeletons for C:\Users\TAN\Anaconda3\python.exe...

Now typing in Sublimetext : ***print('hello world')***

Then try to run the code and you will see 'hello world' printout.

If Statement

In order to write useful programs, we almost always need the ability to check conditions and change the behavior of the program accordingly. Conditional statements give us this ability. The simplest form is the **if** statement, which has the genaral form:

```
if BOOLEAN_EXPRESSION:
    STATEMENTS
```

A few important things to note about if statements:

1. The colon (:) is significant and required. It separates the **header** of the **compound statement** from the **body**.
2. The line after the colon must be indented. It is standard in Python to use four spaces for indenting.
3. All lines indented the same amount after the colon will be executed whenever the BOOLEAN_EXPRESSION is true.

Example check a number is even or odd

Find whether a given number is even or odd, print out an appropriate message

Hint

- Use **%** operator to get remaining of division between a number and 2

- Use ***if*** statement

Solution

```
#number = 4
number = 5
mod = num % 2
if mod > 0:
    print("This is an odd number.")
else:
    print("This is an even number.")
```

For Statement

The for loop processes each item in a sequence, so it is used with Python’s sequence data types - strings, lists, and tuples.

Each item in turn is (re-)assigned to the loop variable, and the body of the loop is executed.

The general form of a for loop is:

```
for LOOP_VARIABLE in SEQUENCE:
    STATEMENTS
```

This is another example of a compound statement in Python, and like the branching statements, it has a header terminated by a colon (:) and a body consisting of a sequence of one or more statements indented the same amount from the header.

Exmple : Count the Number of Even and Odd

Write a Python program to count the number of even and odd numbers from a series of numbers. Go to the editor
Sample numbers : numbers = (1, 2, 3, 4, 5, 6, 7, 8, 9)
Expected Output :

```
Number of even numbers : 5
Number of odd numbers : 4
```

Hint

- Use **%** to check if a number divible for 2
- Use **in** to check every item inside tuple

Solution

```
numbers = (1, 2, 3, 4, 5, 6, 7, 8, 9) # Declaring the tuple
```

```
count_odd = 0
count_even = 0
for x in numbers:
        if not x % 2:
             count_even+=1
        else:
             count_odd+=1
print("Number of even numbers :",count_even)
print("Number of odd numbers :",count_odd)
```

String

Following are some essential string function will be used in our Caesar Cipher program

Title	Code	Return
Create a new string	s = “hello world” print(s)	hello world
Calculate length of string	len(“hello world”)	Return the length of string “hello world” **11**
Check if string is inside a string	“hello” in “hello world”	Because “hello” is inside “hello world”, so this check return **True**
Find character position inside string	s = “hello world” s.find(‘e’)	‘e’ position inside string “hello world” **1.** String is index from 0.
Access character inside string by index	s = “hello world” s[1]	String index from 0 so return will be ‘e’

String Concatenation	s = “hello” + “ ” + “world”
	Print(s)
		“hello world”

Function

In Python, a **function** is a named sequence of statements that belong together. Their primary purpose is to help us organize programs into chunks that match how we think about the problem.

The syntax for a **function definition** is:

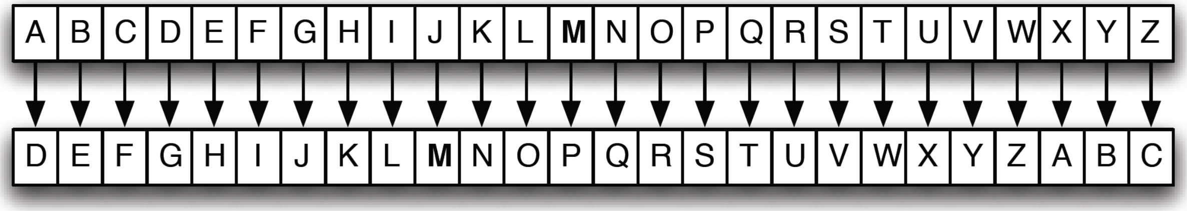
```
def NAME( PARAMETERS ):
   STATEMENTS
```

For example we create a funtion that return number of even inside a number list

```
# Input : a list of number
# Output : number of even

def number_of_even(list_number):
   even_number = 0

   for x in list_number:
       if x % 2 == 0:
           even_number = even_number + 1

   return even_number

numbers = [1,2,3,4,5,6,7,8,9]
print('number of even : ', number_of_even(numbers))
```

Encrypt Message with Caesar Cipher

Manually Encrypt



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

D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Caesar Cipher with Shift Key = 3

- To encrypt a message use Caesar Cipher, you will encrypt each alphabe character in message.
- To encrypt a character, you will replace the original character by a new character.
- The new character is found from original one by just shift original a number of time on alphabe table. The number of shift is called KEY. For example, if KEY = 3, “A” character will become “D” as show up above table.
- Using above rules, message “***I love programming***” will become “***L oryh surjudpplqj***”

Automate Encrpyt

Using python programming language, we will transfer sequent of encrypt to a program with following notes:

- Using ***for in*** loop to access every character inside a string
- Using ***upper()*** function to make all character inside string become upper case
- Using ***find()*** function to findout character position inside string
- Using ***len()*** global function to get length of string
- Access character inside string with index

We get following code

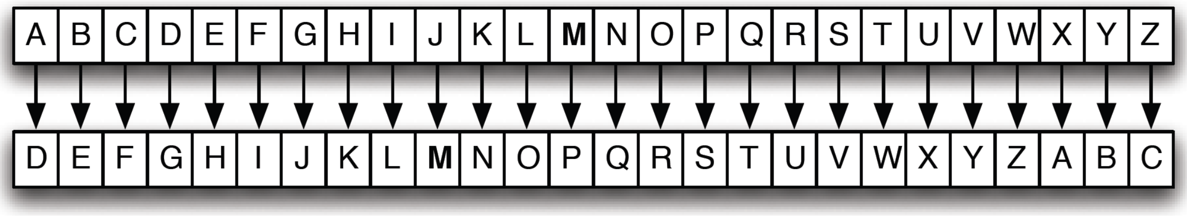
```
org_message = 'I love programming'

# org_message = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

alphabe = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
key = 3

# make all character inside original message upper case
org_message = org_message.upper()

encrypt_message = ''

# print all character
for org_character in org_message:

      if org_character in alphabe:

            org_position = alphabe.find(org_character)
            new_position = org_position + key

            if new_position > (len(alphabe) - 1):
                  new_position = new_position - len(alphabe)

            new_character = alphabe[new_position]

      else:
            new_character = org_character

      encrypt_message = encrypt_message + new_character

print(encrypt_message)
```

After put encrypt code into function, We have following code

```
def  encrypt_caesar_cipher(org_message, key):

      alphabe = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

      # make all character inside original message upper case
      org_message = org_message.upper()

      encrypt_message = ''

      # print all character
      for org_character in org_message:

            if org_character in alphabe:

                  org_position = alphabe.find(org_character)
                  new_position = org_position + key

                  if new_position > (len(alphabe) - 1):
                        new_position = new_position - len(alphabe)

                  new_character = alphabe[new_position]

            else:
                  new_character = org_character

            encrypt_message = encrypt_message + new_character

      print(encrypt_message)

      return encrypt_message

#######################################
org_message = 'I love programming'
key = 3
encrypt_caesar_cipher(org_message,key)
```

Decrypt Caesar Cipher Message

Manually Decrypt



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

D E F G H I J K L **M** N O P Q R S T U V W X Y Z A B C

- Decrypt a message is process of decrypt every character inside message.
- To decrypt message, we should know the KEY (shilft number which used to encrypt)
- A character is decrypted by simply shift back KEY times. For example with KEY = 3, “D” character will become “A”. And message “***L oryh surjudpplqj***” will be decrypt as ***“I love programming”***

Automate Decrypt

Folowing python code will doing decrypt

```
encrypt_message = 'L ORYH SURJUDPPLQJ'

alphabe = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

key = 3

# make all character inside original message upper case
encrypt_message = encrypt_message.upper()

org_message = ''

# print all character
for encrypt_character in encrypt_message:

      if encrypt_character in alphabe:

            encrypt_position = alphabe.find(encrypt_character)
            org_position = encrypt_position - key

            if org_position < 0:
                  org_position = encrypt_position - key + len(alphabe)

            org_character = alphabe[org_position]

      else:
            org_character = encrypt_character
      org_message = org_message + org_character

print(org_message)
```

Putting above code into function, We will have decrypt function as below

```
def decrypt_caesar_cipher(encrypt_message, key):

      alphabe = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

      # make all character inside original message upper case
      encrypt_message = encrypt_message.upper()

      org_message = ''

      # print all character
      for encrypt_character in encrypt_message:

            if encrypt_character in alphabe:

                  encrypt_position = alphabe.find(encrypt_character)
                  org_position = encrypt_position - key

                  if org_position < 0:
                        org_position = encrypt_position - key +
len(alphabe)

                  org_character = alphabe[org_position]

            else:
                  org_character = encrypt_character

            org_message = org_message + org_character

      print(org_message)

      return org_message

################################################
encrypt_message = 'L ORYH SURJUDPPLQJ'
key = 3
decrypt_caesar_cipher(encrypt_message, key)
```

Hack Caesar Cipher Message

The Brute-Force Attack

Have only 26 characters in alphabe, it mean we only need to try out 25 keys in order to findout the original message. The way of trying all possibility like this called brute-force attack.

Below is code to try out all 25 differerence possible keys.

- This code use decrypt function which already created from decrypt part.
- To try out 25 possibility, we use ***rang()*** function combine with ***for*** statement

```
def decrypt_caesar_cipher(encrypt_message, key):

      alphabe = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

      # make all character inside original message upper case
      encrypt_message = encrypt_message.upper()

      org_message = ''

      # print all character
      for encrypt_character in encrypt_message:

            if encrypt_character in alphabe:

                  encrypt_position = alphabe.find(encrypt_character)
                  org_position = encrypt_position - key

                  if org_position < 0:
                        org_position = encrypt_position - key +
len(alphabe)

                  org_character = alphabe[org_position]

            else:
                  org_character = encrypt_character

            org_message = org_message + org_character

      print(org_message)

      return org_message
```

```
def brute_force_caesar_cipher(encrypt_message):

      for key in range(1,26):
            decrypt_caesar_cipher(encrypt_message, key)


################################################


encrypt_message = 'L ORYH SURJUDPPLQJ'
brute_force_caesar_cipher(encrypt_message)
```

Run this program and we can see what out put make sense.

```
K NQXG RTQITCOOKPI
J MPWF QSPHSBNNJOH
I LOVE PROGRAMMING
H KNUD OQNFQZLLHMF
G JMTC NPMEPYKKGLE
F ILSB MOLDOXJJFKD
E HKRA LNKCNWIIEJC
D GJQZ KMJBMVHHDIB
C FIPY JLIALUGGCHA
B EHOX IKHZKTFFBGZ
A DGNW HJGYJSEEAFY
Z CFMV GIFXIRDDZEX
Y BELU FHEWHQCCYDW
X ADKT EGDVGPBBXCV
W ZCJS DFCUFOAAWBU
V YBIR CEBTENZZVAT
U XAHQ BDASDMYYUZS
T WZGP ACZRCLXXTYR
S VYFO ZBYQBKWWSXQ
R UXEN YAXPAJVVRWP
Q TWDM XZWOZIUUQVO
P SVCL WYVNYHTTPUN
O RUBK VXUMXGSSOTM
N QTAJ UWTLWFRRNSL
M PSZI TVSKVEQQMRK
```