MultiWingSpan

Home Programming Web Design Computer Science Twisting Puzzles Arduino BBC micro:bit

BBC micro:bit
Encoding Ciphers

Introduction

If you have written a Python program and flashed it to the micro:bit, you can use the REPL window to interact with it. This project turns the micro:bit into a cipher toolkit for encoding and decoding ciphers.

This page will start you off. There are algorithms for several ciphers in another section of this web site.

This page shows you that you can use the micro:bit as a platform for running Python programs for you to interact with via REPL.

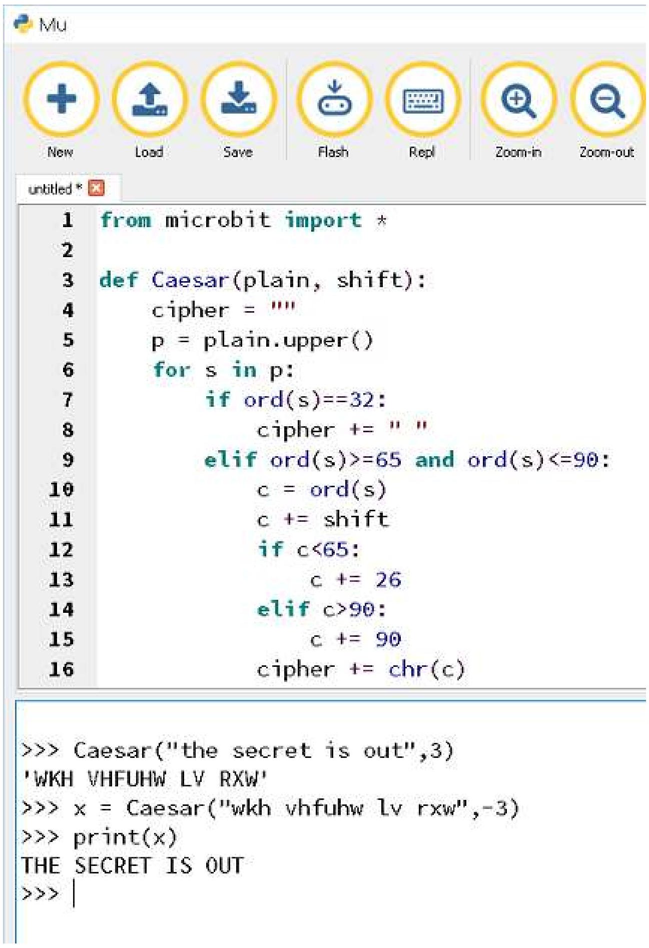
Caesar Shift

We'll start with one of the simplest ciphers, the Caesar Shift Cipher. Read more about it on **this page.**

```python
from microbit import *

def Caesar(plain, shift):
    cipher = ""
    p = plain.upper()
    for s in p:
        if ord(s)==32:
            cipher += " "
        elif ord(s)>=65 and ord(s)<=90:
            c = ord(s)
            c += shift
            if c<65:
                c += 26
            elif c>90:
                c -= 26
            cipher += chr(c)
    return cipher
```

The program is just a function. You will need to use an REPL terminal to interact with the program. Here is an example of it in action using Mu.



```python
from microbit import *

def Caesar(plain, shift):
    cipher = ""
    p = plain.upper()
    for s in p:
        if ord(s)==32:
            cipher += " "
        elif ord(s)>=65 and ord(s)<=90:
            c = ord(s)
            c += shift
            if c<65:
                c += 26
            elif c>90:
                c += 90
            cipher += chr(c)
```

```
>>> Caesar("the secret is out",3)
'WKH VHFUHW LV RXW'
>>> x = Caesar("wkh vhfuhw lv rxw",-3)
>>> print(x)
THE SECRET IS OUT
>>> |
```

Vignère

This is a slightly more complex cipher. Read about it on **this page.**

The cipher uses a keyword and more than one alphabet.

BBC Microbit

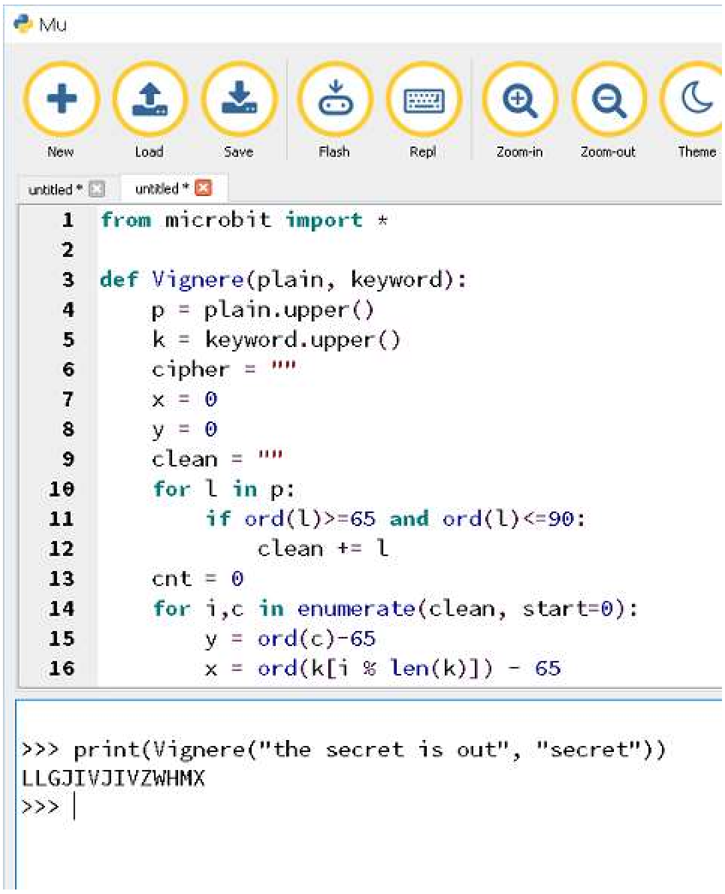
Collapse All Expand All

+ Block Editor - The Basics
+ Block Editor - Components
+ Kodu - micro:bit Worlds
+ JavaScript Blocks
+ JavaScript Blocks - Exercises
+ Blocks - Bit:Bot
+ Blocks - Bit:Commander
+ MicroPython - Starting Off
- MicroPython - Examples
- Dice Rolling
- Shut The Matrix
- Encoding Morse Code
- Encoding Ciphers
- Drawing A Maze
- Scrolling Race Track
- Vertical Scroller
- Concentration Game
- Text Entry - Accelerometer
- Charlie's Python Game
- Lights Out Game
- Sam's French Number Game
- Bryn's Concentrated Clocks
- Lattice Paths
- Knight Moves
+ MicroPython - Components
+ MicroPython - Breakout Boards
+ MicroPython - Exercises
+ MicroPython - Pi Accessories
+ MicroPython - Bit:Bot
+ MicroPython - Bit:Commander
+ MicroPython - Projects
+ MicroPython - Visual Basic
+ Other - Odds & Ends

```
from microbit import *

def Vignere(plain, keyword):
    p = plain.upper()
    k = keyword.upper()
    cipher = ""
    x = 0
    y = 0
    clean = ""
    for l in p:
        if ord(l)>=65 and ord(l)<=90:
            clean += l
    cnt = 0
    for i,c in enumerate(clean, start=0):
        y = ord(c)-65
        x = ord(k[i % len(k)]) - 65
        cipher += chr(65 + ((x+y)%26))
    return cipher
```

In action,



```
from microbit import *

def Vignere(plain, keyword):
    p = plain.upper()
    k = keyword.upper()
    cipher = ""
    x = 0
    y = 0
    clean = ""
    for l in p:
        if ord(l)>=65 and ord(l)<=90:
            clean += l
    cnt = 0
    for i,c in enumerate(clean, start=0):
        y = ord(c)-65
        x = ord(k[i % len(k)]) - 65
```

```
>>> print(Vignere("the secret is out", "secret"))
LLGJIVJIVZWHMX
>>> 
```

Challenges

Have a look at some of the algorithms in the **Ciphers** section of the site. Just add more functions to your program and call them from the REPL window.

Keep going with this idea. You can make a whole host of functions and run them directly from the REPL window.