Assignment_1

June 3, 2019

1 Assignment 1

Realfagslektormaster, programing course

Author: Paul Knutson

The assignment here is the make an encryption algorithm, based on a Caesar cipher. This implementation uses and LSO function (Left Shift Operation) for encryption, and an RSO (Right Shift Operation, a reversed LSO) for decryption. The shift size can be decided by the user, but the default (which is used below) is 1. LSO turns a into b, b into c, etc. If the shift is greater, a may turn into c, b to d, etc. At the end, the decryption function uses .title() in a simple attempt to make the decrypted text prettier. Had the symbol list included both uppercase and lowercase, this would %not be necessary.

```
In [1]: # List of all acceptable symbols.
        symbols = list("ABCDEFGHIJKLMNOPQRSTUVWXYZÆØÅ0123456789 -,.!?".lower())


        def LSO(msg, shift=1):
            """Left Shift Operation."""

            # Make the string a list of lower case chars.
            msg = list(msg.lower())

            # For each letter in message.
            for i in range(len(msg)):

                # Get index of msg-letter in symbols.
                index = symbols.index(msg[i])

                # If not found, return.
                if index == -1:
                    return

                # LSO swap. Modulate to stay within bounds.
                msg[i] = symbols[(index + shift) % len(symbols)]
            return "".join(msg)
```

```
def RSO(msg, shift=1):
    """Right Shift Operation (opposite of LSO)."""
    return LSO(msg, len(symbols) - shift)


text = "Hello World!"
print("Kryptert:\t", LSO(text))
print("Dekryptert:\t", RSO(LSO(text)).title())
```

```
Kryptert:         ifmmp-xpsme?
Dekryptert:         Hello World!
```