

```

# LCYS_2022 / L.M. Saxton

# https://pypi.org/project/bcrypt/
# !pip install bcrypt
import bcrypt

class CreateUsername():
    '''Create a username within a set of parameters'''
    def __init__(self, username):
        '''Initialize username'''
        self.username = username

    # Username functions:

    def userLength(self):
        '''Check the length of the username'''
        if len(self.username)>5 and len(self.username)<=16:
            pass
        else:
            print("\t- Username needs to be between 5 and 16 characters long")

    def userSpaces(self):
        '''Check if username has unauthorized spaces'''
        text = self.username
        count = 0
        for char in text:
            if char == ' ':
                count = count+1
        if count == 0:
            pass
        else:
            print("\t- Username should not have any spaces")

    def userSpecSym(self):
        '''Check if username has unauthorized special symbols'''
        specialsymb = (
            '!', '@', '#', '$', '&', '*', '?', '"', "'", "/", "\\", '|', '-', '{', '}',
            '[', ']', '+', '=', '^', '%', '.', ':', ';', '~', '`', '<', '>'
        )
        if not any(char in specialsymb for char in self.username):
            pass
        else:
            print("\t- Username should not have any special characters")

    def successUserLength(self):
        '''Return value if username length is within range'''
        if len(self.username)>5 and len(self.username)<=16:
            return(int(1))
        else:
            return(int(0))

    def successUserSpaces(self):
        '''Return value if username has no spaces'''
        text = self.username
        count = 0
        for char in text:
            if char == ' ':
                count = count+1
        if count == 0:
            return 1
        else:
            return 0

    def successUserSpecSym(self):
        '''Return value if username has no special symbols'''
        specialsymb = (
            '!', '@', '#', '$', '&', '*', '?', '"', "'", "/", "\\", '|', '-', '{', '}',
            '[', ']', '+', '=', '^', '%', '.', ':', ';', '~', '`', '<', '>'
        )
        if not any(char in specialsymb for char in self.username):
            return 1
        else:
            return 0

```

```

class CreatePassword():
    '''Create a password within a set of parameters'''
    def __init__(self, password):
        '''Initialize password'''
        self.password = password

    def passLength(self):
        '''Check length of password'''
        if len(self.password)>8 and len(self.password)<=16:
            pass
        else:
            print("\t- Password needs to be between 8 and 16 characters long")

    def passUpper(self):
        '''Check if password has uppercase character(s)'''
        if not any(char.isupper() for char in self.password):
            print('\t- Password should have at least one uppercase character')
        else:
            pass

    def passDigit(self):
        '''Check if password has digit(s)'''
        if not any(char.isdigit() for char in self.password):
            print('\t- Password should have at least one number')
        else:
            pass

    def passSpaces(self):
        '''Check if password has no spaces'''
        text = self.password
        count = 0
        for char in text:
            if char == ' ':
                count = count+1
        if count == 0:
            pass
        else:
            print("\t- Password should not have any spaces")

    def passSpecSym(self):
        '''Check if password has special symbol(s)'''
        specialsymb = ('!', '@', '#', '$', '&', '*')
        if not any(char in specialsymb for char in self.password):
            print(
                '\t- Password should have at least one of the following symbols:'
                '\n\t ! @ # $ % & *'
            )
        else:
            pass

    def successPassLength(self):
        '''Return value if password length accepted'''
        if len(self.password)>8 and len(self.password)<=16:
            return 1
        else:
            return 0

    def successPassUpper(self):
        '''Return value if password has uppercase character(s)'''
        if not any(char.isupper() for char in self.password):
            return 0
        else:
            return 1

    def successPassDigit(self):
        '''Return value if password has digit(s)'''
        if not any(char.isdigit() for char in self.password):
            return 0
        else:
            return 1

    def successPassSpaces(self):
        '''Return value if password has no spaces'''
        text = self.password
        count = 0

```

```
for char in text:
    if char == ' ':
        count = count+1
if count == 0:
    return 1
else:
    return 0

def successPassSpecSym(self):
    '''Return value if password has special symbol(s)'''
    specialsymb = ('!', '@', '#', '$', '&', '*')
    if not any(char in specialsymb for char in self.password):
        return 0
    else:
        return 1

class EncryptPass():
    '''Encrypt the accepted password'''
    def __init__(self, password):
        '''Initialize password to hash'''
        self.password = password

    def encryptPass(self):
        '''Hash password with bcrypt'''
        b_password = self.password.encode('ASCII')
        # Generates a unique value with every hash
        salt = bcrypt.gensalt()
        # Generates the hash value
        hashed = bcrypt.hashpw(b_password, salt)
        return(hashed)
```