VIDEO 8 : MORE STRING FUNCTIONS

This time we'll cover most of the string methods that you'll ever need, that have not been covered previously. I'll display some examples with a brief explanation which should be self explanatory.

CODE

```
# Strings have many methods we can use beyond what I covered last time
rand_string = "    this is an important string    "

# Delete whitespace on left
rand_string = rand_string.lstrip()

# Delete whitespace on right
rand_string = rand_string.rstrip()

# Delete whitespace on right and left
rand_string = rand_string.strip()

# Capitalize the 1st letter
print(rand_string.capitalize())

# Capitalize every letter
print(rand_string.upper())

# lowercase all letters
print(rand_string.lower())
```

Lists will be covered in detail later, but I want to show them briefly here. This is how you turn a list into a string and then separate each item with a space.

CODE

```
a_list = ["Bunch", "of", "random", "words"]
print(" ".join(a_list))
```

A space was defined as the thing that would separate the items in the last example. Something that is used to separate data is called a delimiter. So if we had "pig, cow, turtle" the comma and space would be the delimiter because it comes between each piece of meaningful data.

This is how we turn a string into a list and print it out.

CODE

```
rand_string = "this is an important string"
a_list_2 = rand_string.split()

for i in a_list_2:
    print(i)
```

We can find how many times a string occurs in a string

CODE

```
rand_string = “this is an important string”
print("How many is :", rand_string.count(“is"))
```

Get an index for a matching string

CODE

```
rand_string = “this is an important string”
print("Where is string :", rand_string.find(“string"))
```

Replace a string

CODE

```
rand_string = “this is an important string”
print(rand_string.replace("an ", "a kind of "))
```

Python Problem for you to Solve

It’s time to test what you have learned. You will create an acronym generator. The user will enter a string and then convert it to an acronym with uppercase letters like this

```
Convert to Acronym : Random Access Memory
RAM
```

Give it a go. I’m sure you can do it.

SOLUTION

```
# Ask for a string
orig_string = input("Convert to Acronym : ")

# Convert the string to all uppercase
orig_string = orig_string.upper()

# Convert the string into a list
list_of_words = orig_string.split()

# Cycle through the list
for word in list_of_words:

    # Get the 1st letter of the word and eliminate the newline
    print(word[0], end="")

print()
```

More String Methods

For our next problem some additional string methods are going to be very useful.

CODE

```
# Returns True if characters are letters or numbers
# Whitespace is false
print("Is z a letter or number :", letter_z.isalnum())

# Returns True if characters are letters
print("Is z a letter :", letter_z.isalpha())

# Returns True if characters are numbers (Floats are False)
print("Is 3 a number :", num_3.isdigit())

# Returns True if all are lowercase
print("Is z a lowercase :", letter_z.islower())

# Returns True if all are uppercase
print("Is z a uppercase :", letter_z.isupper())

# Returns True if all are spaces
print("Is space a space :", a_space.isspace())
```

2nd Python Problem for you to Solve

This problem will really wrack your brain, so don't worry if you can't solve it. Feel free to use all the resources of the internet to try.

We are going to make a Caesar's Cipher. Encryption is super popular so let's take a look at one of the first. Here is what you'll have to program.

Receive a message and then encrypt it by shifting the characters by a requested amount to the right. A becomes D, B becomes E for example. Also decrypt the message back again.

You should check if a character is a letter and if not leave it as its default.

HINTS

1. A-Z have the numbers 65-90 in unicode
2. a-z have the numbers 97-122
3. You get the unicode of a character with ord(yourLetter)
4. You convert from unicode to character with chr(yourNumber)
5. Use isupper() to decided which unicodes to work with
6. Add the key (number of characters to shift) and if bigger or smaller then the unicode for A, Z, a, or z increase or decrease by 26

SOLUTION

```
# Receive the message to encrypt and the number of characters to shift
message = input("Enter your message : ")
key = int(input("How many characters should we shift (1 - 26)"))

# Prepare your secret message
secret_message = ""
```

```
# Cycle through each character in the message
for char in message:

    # If it isn't a letter then keep it as it is in the else below
    if char.isalpha():

        # Get the character code and add the shift amount
        char_code = ord(char)
        char_code += key

        # If uppercase then compare to uppercase unicodes
        if char.isupper():

            # If bigger than Z subtract 26
            if char_code > ord('Z'):
                char_code -= 26

            # If smaller than A add 26
            elif char_code < ord('A'):
                char_code += 26

        # Do the same for lowercase characters
        else:
            if char_code > ord('z'):
                char_code -= 26
            elif char_code < ord('a'):
                char_code += 26

        # Convert from code to letter and add to message
        secret_message += chr(char_code)

    # If not a letter leave the character as is
    else:
        secret_message += char

print("Encrypted :", secret_message)

# To decrypt the only thing that changes is the sign of the key
key = -key

orig_message = ""

for char in secret_message:
    if char.isalpha():
        char_code = ord(char)
        char_code += key

        if char.isupper():
            if char_code > ord('Z'):
                char_code -= 26
            elif char_code < ord('A'):
                char_code += 26
        else:
            if char_code > ord('z'):
```

```
            char_code -= 26
        elif char_code < ord('a'):
            char_code += 26

    orig_message += chr(char_code)

else:
    orig_message += char

print("Decrypted :", orig_message)
```

Awesome Job if you solved that! Really awesome super job if you understood all of the code! The goal is only to make you think like a programmer and to understand the finished code.

I'll give you a break this time and not force you to do a quiz since the problems were so difficult. I won't be easy on you next time though. In the next video we'll finally start talking about functions.