

VIDEO 12 : STRINGS

This tutorial will be very code heavy because I'm going to show you a ton of string functions. Strings are a series of characters between quotes. You can use Single quotes, double quotes or triple quotes.

CODE

```
print(type("3"))
print(type('3'))
print(type("""3"""))
```

You can see the data type for data using type

CODE

```
print(type(3))
print(type(3.14))
```

Each character is stored in a series of boxes labeled with index numbers. You can find out how many characters a string contains.

CODE

```
samp_string = "This is a very important string"
print("Length :", len(samp_string))
```

You can get characters using index numbers starting at 0.

CODE

```
samp_string = "This is a very important string"
print(samp_string[0])

# Get the last character
print(samp_string[-1])
```

SLICE

You can get a block of characters using slice. A slice is where you define what index values you want between 2 brackets.

CODE

```
samp_string = "This is a very important string"
# Get a slice by saying where to start and end
# The 4th index isn't returned
print(samp_string[0:4])

# Get everything starting at an index
print(samp_string[8:])

# More slices
```

```
print("Every Other ", samp_string[0:-1:2])
print("Reverse ", samp_string[::-1])
```

Other random string manipulations you can use

CODE

```
# Join or concatenate strings with +
print("Green " + "Eggs")
```

```
# Repeat strings with *
print("Hello " * 5)
```

```
# Convert an int into a string
num_string = str(4)
```

You can cycle through each character with for

CODE

```
samp_string = "This is a very important string"
for char in samp_string:
    print(char)
```

You can cycle through characters in pairs. Subtract 1 from the length because length is 1 more than the highest index because strings are 0 indexed. Then use range starting at index 0 through string length and increment by 2 each time through.

CODE

```
samp_string = "This is a very important string"
for i in range(0, len(samp_string)-1, 2):
    print(samp_string[i] + samp_string[i+1])
```

UNICODE

Computers assign characters with a number known as a Unicode A-Z have the numbers 65-90 and a-z 97-122. 2 functions allow you to work with unicodes.

CODE

```
# You can get the Unicode code with ord()
print("A =", ord("A"))
```

```
# You can convert from Unicode with chr
print("65 =", chr(65))
```

Shortcut Ways to Perform Math Calculations

Let's say you want to add val plus 1. You could type out val = val + 1, but there is a shortcut way val_1 += 1

This shortcut can be used for all math operations

```
val_1 -= 5
val_1 *= 3
val_1 /= 2
val_1 %= 6
```

Likewise you can also add one string to another in the same way

```
str_1 += str_2
```

There is another shortcut when you want to just increment or decrement by 1. Instead of `val_1 += 1` you can just type `val_1++` or `val_1--`.

Python Problem for you to Solve

Here is another problem you can work through. Remember it isn't important if you don't get it right. Think in new ways, search the internet and the only goal is to understand the solution.

Your code should receive a **uppercase string** and then hide it's meaning by turning it into a string of unicodes. Then it should translate the unicodes back into the original message.

SOLUTION

```
norm_string = input("Enter a string to hide in uppercase: ")

secret_string = ""

# Cycle through each character in the string
for char in norm_string:

    # Store each character code in a new string
    # += is the same as secret_string = secret_string + whatever
    secret_string += str(ord(char))

print("Secret Message :", secret_string)

norm_string = ""

# Cycle through each character code 2 at a time by incrementing by
# 2 each time through since unicodes go from 65 to 90
for i in range(0, len(secret_string)-1, 2):

    # Get the 1st and 2nd for the 2 digit number
    char_code = secret_string[i] + secret_string[i+1]

    # Convert the codes into characters and add them to the new string
    norm_string += chr(int(char_code))

print("Original Message :", norm_string)
```

2nd Python Problem for you to Solve

Now if you solved the previous problem I have another for you. Make the above work with upper and lowercase letters by changing 2 lines of code.

SOLUTION

Add these 2 changes

```
secret_string += str(ord(char) - 22)
```

```
norm_string += chr(int(char_code) + 22)
```

That's all we will cover on strings this time. In the next video we'll cover even more with strings including more problems for you to solve.