# CSX3001/ITX3001
# FUNDAMENTALS OF COMPUTER PROGRAMMING

## CLASS 06 UNDERSTANDING STRING

CHARACTER, STRING OPERATORS, STRING METHODS, TYPE CONVERSION, AND ESCAPE SEQUENCES

PYTHON

## STRING RECAP

In programming terms, we usually call text a *string*. When you think of a string as a collection of letters, the term makes sense. All the letters, numbers, and symbols in this worksheet could be a string. We create a string by putting quotes around text, like this:

```
message = "What is your name?"
```

Then, to see what's inside message, we could enter print(message), like this:

```
print(message)
```

```
What is your name?
```

You can also use single quotes to create a string, like this:

```
message = 'What is your name?'
```

## CHARACTER

Aforementioned, string is a collection of letters or in programming term "string is a sequence of characters". Computer do not deal with characters, they deal with numbers (binary). This conversion of character to a number is called encoding, and the reverse process is decoding. ASCII and Unicode are some of the popular encoding used. Let's see how ASCII value used for representative of a character.

```
asciiOfA = ord('A')
print(asciiOfA)
```

The build-in ***ord*** method accepts a single character and gives you an associated ascii value. Also, if you want to convert from ascii number to a character, you could do like this:

```
charForA = chr(65)
print(charForA)
```

The build-in ***chr*** method accepts a number which is an ascii value and the method gives you an associated character.

## STRING OPERATIONS

The basic string operations you should know by now:

| | | |
|---|---|---|
| + | "Hello" + " " + "World" | String concatenation |
| * | "Awesome" * 100 | Creates multiple copies of a string |

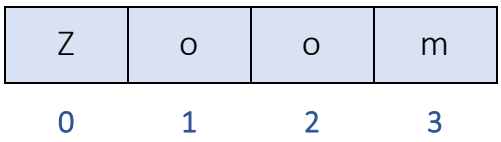
If you would like to check whether this string contains another string, you can use the operation ***in*** or ***not in***. See examples below.

```
str1 = "This is test"
str2 = "test"

if str2 in str1:
   print("Yes, some str2 is in str1")
else:
   print("Oh no")
```

```
name = "Trudy"

if name not in "Alice,Bob":
   print("Access denied")
else:
   print("Access granted")
```

You can extract a character from a string by specify the character's index. The index always starts from zero. The index notation is square brackets [index]. See the examples below.

```
name = "Zoom"
firstChar = name[0]
print(firstChar)
```




What happen if you specify the character's position #4 for the previous code?

Ans:

---

You can also extract part of characters from a string by specify two numbers in the index notation. Try the code below.

```
x = "Hello World"
print(x[0:5])
print(x[6:11])
```

If we want just some part of **"llo"**, what would be the index?

Ans: x[ ..... : ..... ]

If we want just some part of **"Wor"**, what would be the index?

Ans: x[ ..... : ..... ]

If we want just some part of **"o W"**, what would be the index?

Ans: x[ ..... : ..... ]

## STRING WITH FOR-LOOP

You can traverse each character in the string with for-loop like this:

```
message = "abcdefghijklmnopqrstuvwxyz"

for c in message:
    if c not in "aeiou":
        print(c)
    else:
        print("*")
```

The variable c in the for-loop represent each character in each loop round. The number of loop repetition is equal to the length of the string (in this case 26 times). The example logic inside the loop replace all the occurrences of a, e, i, o, u with * (asterisk).

## STRING METHODS

Try code in the second column and note the result in the third column. More explanations will be given in the class.

| | | |
|---|---|---|
| len() | len("Alice") | |
| capitalize() | "Alice".capitalize() | |
| upper() | "Alice".upper() | |
| lower() | "Alice".lower() | |
| isupper() | "Alice".isupper() | |
| islower() | "Alice".islower() | |
| find() | "Alice".find("ice") | |
| replace() | "Alice".replace("ice", "ex") | |
| isdigit() | "1232".isdigit() | |
| isalpha() | "3".isalpha() | |

| | | |
|---|---|---|
| split() | "Alice,Bob".split(",") | |
| join() | ", ".join(["Alice", "Bob"]) | |

more https://www.programiz.com/python-programming/methods/string

## TYPE CONVERSION

Convert from other type to string, you can use ***str()*** syntax.

```
i = 5
x = str(x)

f = 232.453543
y = str(f)

b = False
z = str(b)
```

## ESCAPE CHARACTER

| Escape Sequence | Description |
|---|---|
| \newline | Backslash and newline ignored |
| \\ | Backslash |
| \' | Single quote |
| \" | Double quote |
| \t | ASCII Horizontal Tab |

## ◆ STRING EXERCISES

Complete the following exercises in Python IDLE or Jupyter notebook.

1) Write a Python code to take a sentence, then the code prints the sentence using uppercase letters. Expected input/output:

```
Write a sentence: My friend and I want to go shopping and dining.
MY FRIEND AND I WANT TO GO SHOPPING AND DINING.
```

2) Write a Python code to take a sentence and a character that you want to count. Expected input/output:

```
Enter a sentence: Lady on the moon.
Enter a character:o
Total no. of character o in this sentence = 3
```

3) Based on the exercise3, write a Python code to keep asking the character to be counted until the user enter n or N to end.

4) Write a Python code to take a sentence, then the code prints the sentence with only alphabet with whitespaces. Expected input/output:

```
Enter a sentence: I'm 9 years old.
Im  years old
```

5) Based on the exercise5, write a python code without using build-in string methods. You can use ord() and chr() to solve this exercise.

6) Write a python code to take a sentence, then the code prints the total number of alphabets, uppercase letters, lowercase letters, numeric and other letters. Expected input/output:

```
Enter string: Mr. Tom Fox is 56 years old.
 Total alphabet = 18
 Uppercase = 3
 Lowercase = 15
 Numeric = 2
 Other letters (including space) = 8
```

7) Write a Python code to take a sentence and slice characters into two half and print each half accordingly. If a total number of characters is an odd number, the first slice will have (n+1)/2 characters. Expected inputs/outputs:

Sample 1:

```
Write a sentence: 12345678
First half:  1234
Second half: 5678
```

Sample 2:

```
Write a sentence: 123456789
First half:  12345
Second half: 6789
```

Sample 3:

```
Write a sentence: I am a good boy
First half:  I am a g
Second half: ood boy
```

8) Reversing a string can be done in Python by using string slicing feature. For instant, try the following Python code to observe the slicing results.

```
myStr = “Hello world!”
revMyStr = myStr[::-1]
print(revMystr)
```

Write a program to reverse an input string without using string slicing.

Sample Run:
Input String: Hello World!
Reversed String: !dlroW olleH

9) Replacing substring(s) in a given string, str, in Python can be done by using replace() function.

Syntax: str.replace(old, new [, count])
str – the given string
old – the replaced substring
new – the new substring which replace old
count (optional) – the number of times old substring will be replaced with new substring, if it is not specified, replace() will replace all occurrence of old.

An example of using replace():
orgStr = “You are my sunshine, you are my everything.”
repStr = org.replace(“are”,”were”)
print(repStr)

Write a program to replace all old substring(s) with new substring(s) without using replace() function.
Sample Run:

Input String: You are my sunshine, you are my everything.
Replaced string: are
Replaced with: were
The output string: You were my sunshine, you were my everything.

10) Write a program to count how many digit(s) appear in the input string.
Sample Run:
Input string: 123$
There are 3 digits in the string.

Input string: Hello again!
There is no digit at all!

Input string: ho ho ho 12 haha 345 hehe 5555
There are 9 digits in the string.

## ASSIGNMENTS

Complete the following exercises in Python IDLE. You must name the python file as, {your-id}_class0{number}_{course-code}_{section-number}_assignment{number}.py for example, for assignment 1 will be named,

6120001_class06_csx3001_541_assignment1.py

1) Ask the user to enter a string and count number of uppercase letters, lowercase letters and spaces in a string and toggle case the given string (convert lowercase to uppercase and vice versa). Expected input/output:

```
Enter string: I am Harry Potter

3 Uppercase
11 Lowercase
3 Spaces
i AM hARRY pOTTER
```

2) A substitution cipher is a technique of cryptography where you replace each character of the plaintext by another character, according to some mapping scheme. Write a program that performs a simple substitution cipher, by replacing each character from A-Z by the next 3 values after it, like this:

A → D, B → E, C → F ... X → A, Y → B, Z → C

Expected input/output:

```
Enter a message: I LOVE YOU

L ORYH BRX
```

Note that: Apply the mapping for uppercase alphabet only.