Strings

🎯 Session Objectives

- 🔤 Understand string indexing and slicing
- 🔧 Explore common string methods and operations

97 > 65

'a' > 'A'

dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char
0	0	000	NULL	32	20	040	space	64	40	100	@	96	60	140	`
1	1	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	2	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	3	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	4	004	EOT	36	24	044	$	68	44	104	D	100	64	144	d
5	5	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	6	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	7	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
8	8	010	BS	40	28	050	(	72	48	110	H	104	68	150	h
9	9	011	TAB	41	29	051	)	73	49	111	I	105	69	151	i
10	a	012	LF	42	2a	052	*	74	4a	112	J	106	6a	152	j
11	b	013	VT	43	2b	053	+	75	4b	113	K	107	6b	153	k
12	c	014	FF	44	2c	054	,	76	4c	114	L	108	6c	154	l
13	d	015	CR	45	2d	055	-	77	4d	115	M	109	6d	155	m
14	e	016	SO	46	2e	056	.	78	4e	116	N	110	6e	156	n
15	f	017	SI	47	2f	057	/	79	4f	117	O	111	6f	157	o
16	10	020	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM	57	39	071	9	89	59	131	Y	121	79	171	y
26	1a	032	SUB	58	3a	072	:	90	5a	132	Z	122	7a	172	z
27	1b	033	ESC	59	3b	073	;	91	5b	133	[	123	7b	173	{
28	1c	034	FS	60	3c	074	<	92	5c	134	\	124	7c	174	|
29	1d	035	GS	61	3d	075	=	93	5d	135	]	125	7d	175	}
30	1e	036	RS	62	3e	076	>	94	5e	136	^	126	7e	176	~
31	1f	037	US	63	3f	077	?	95	5f	137	_	127	7f	177	DEL

www.alpharithms.com

codIng → 99

CodinG → 67

```
# String Case Methods
_str = 'Python is Awesome 🔥'
print(len(_str))
```

```
19
```

```
print(_str.upper())
```

```
PYTHON IS AWESOME 🔥
```

```
print(_str.lower())
```

```
python is awesome 🔥
```

```
print(_str.title()) # Convert the first letter of each word to uppercase and the rest to lowercase
```

```
Python Is Awesome 🔥
```

```
print(_str.capitalize()) # Convert the first letter of whole sentence to uppercase and rest to lower
```

```
Python is awesome 🔥
```

```python
# String Operations
str1 = 'Coding'
str2 = 'Ninjas'
new_string = str1 + ' ' + str2 # Concatenation
print(new_string)
```

```
Coding Ninjas
```

```python
name = 'Aditya'
age = 29
print(f'Hi, {name}, You are {age} year old.')
```

```
Hi, Aditya, You are 29 year old.
```

```python
# TypeError: can only concatenate str (not "int") to str
name = 'Aditya'
age = 29
print('Hi' + name + ', You are ' + str(age) + ' year old.')
```

```
HiAditya, You are 29 year old.
```

```python
print(type(age))
```

```
<class 'int'>
```

```python
# Repeat (*)
_echo = 'Ninjas\n'
print(_echo * 3)
```

```
Ninjas
Ninjas
Ninjas
```

```python
# String Comparison
_str1 = 'Python'
_str2 = 'Java'
print(_str1 == _str2) # False
print(_str1 != _str2) # True
print(_str1 >= _str2) # True
print(_str1 <= _str2) # False
```

```
False
True
True
False
```

```
_str1 = 'Swim'
_str2 = 'Swimming'
print(_str1 == _str2) # False
print(_str1 != _str2) # True
print(_str1 >= _str2) # False
print(_str1 <= _str2) # True
```

```
False
True
False
True
```

```
_str1 = 'coding'
_str2 = 'Coding'
print(_str1 == _str2) # False
print(_str1 != _str2) # True
print(_str1 >= _str2) # True
print(_str1 <= _str2) # False
```

```
False
True
True
False
```

```
_str1 = 'coding'
_str2 = 'Zodiac'
print(_str1 == _str2) # False
print(_str1 != _str2) # True
print(_str1 >= _str2) # True
print(_str1 <= _str2) # False
```

```
False
True
True
False
```

```
_str1 = 'cODING'
_str2 = 'Coding'
print(_str1 == _str2) # False
print(_str1 != _str2) # True
print(_str1 >= _str2) # True
print(_str1 <= _str2) # False
```

```
False
True
True
False
```

```
_str1 = 'swim'
_str2 = 'Swimming'
print(_str1 == _str2) # False
print(_str1 != _str2) # True
print(_str1 >= _str2) # True
print(_str1 <= _str2) # False
```

```
False
True
True
False
```

```
's' > 'S' [small character having greater ASCII Values > 'S' having low ASCII Value]
length also plays an important role if the previous characters going to tie.
```

```
# Common String Methods
# replace
_str = 'Python is Awesome!'
new_str = _str.replace('Awesome', 'Fantastic')
print(new_str)
```

```
Python is Fantastic!
```

```
print(_str)
```

```
Python is Awesome!
```

```
_str = 'Python is Awesome!'
new_str = _str.replace('o','O')
print(new_str)
```

```
PythOn is AwesOme!
```

```
# split()
_text = 'Indore@MadhyaPradesh@India'
print(_text.split('@'))
```

```
['Indore', 'MadhyaPradesh', 'India']
```

```
topics = 'SQL*Python*Excel*PowerBI'
topic_list = topics.split('*')
print(topic_list)
```

```
['SQL', 'Python', 'Excel', 'PowerBI']
```

```
topic_list[1]
```

```
'Python'
```

```
split_str = "ViratKohli MSD RohitSharma Sachin"
print(split_str.split())
```

```
['ViratKohli', 'MSD', 'RohitSharma', 'Sachin']
```

```
file_path = "http://localhost:8889/notebooks/anaconda_projects/CN-Python-TTS/PythonSession.ipynb?"
path_list = file_path.split('/')
print(path_list)
```

```
['http:', '', 'localhost:8889', 'notebooks', 'anaconda_projects', 'CN-Python-TTS', 'PythonSession.ipynb?']
```

```
# String Format -> .format() / f-string
name = 'Jasaswini'
age = 27
gender = 'Female'
print('Hi, {}.... You are {} years old. Congratulation! You are {}!'.format(name,age,gender))
```

```
Hi, Jasaswini.... You are 27 years old. Congratulation! You are Female!
```

```
# String Format -> .format() / f-string
name = 'Jasaswini'
age = 27
gender = 'Female'
print(f'Hi, {name}.... You are {age} years old. Congratulation! You are {gender}!')
```

```
Hi, Jasaswini.... You are 27 years old. Congratulation! You are Female!
```

```
# Strip() -> same like trim
print('    Python Programming      '.strip())
```

```
Python Programming
```

```
# Strip() -> same like trim
print('******Python Programming*******'.strip('*'))
```

```
Python Programming
```

```
# Strip() -> same like trim
print('******Python Programming'.strip('*'))
```

```
Python Programming
```

```
# Strip() -> same like trim
print('Python Programming********'.strip('*'))
```

```
Python Programming
```

```
# Strip() -> same like trim
print('$$$$$$$Python Programming********'.strip('*').strip('$'))
```

```
Python Programming
```

```
# Strip() -> same like trim
print('$$$$$$$Python Programming********'.strip('*$'))
```

```
Python Programming
```

```
# index() -> position
_str = "This is a Python Course!"
print(_str.index('is'))
```

```
2
```

```python
# index() -> position
_str = "This is a Python Course!"
# print(_str.index('python')) # ValueError: substring not found
print(_str.index('Python'))
```

```
10
```

```python
# index() -> position
_str = "This is a Python Course!"
print(_str.index('is',3,10)) # index(substr,start,stop)
print(_str.index('is',3)) # index(substr,start,stop)
```

```
5
5
```

```
index(substr, start = 0, stop = 'len(text)')
```

```python
# String Checks() -> Returns Boolean Result
print("CodingNinjas".isalpha()) # True
print("Ninja99".isalpha()) # False
print("Ninja99".isalnum()) # True
print("CodingNinja".isupper()) # False
print("CODING".isupper()) # True
print("coding".islower()) # True
print("123456789".isnumeric()) # True
print("coding123".isnumeric()) # False
print("coding123".isnumeric()) # False
print('Mentorship Program!77'.swapcase()) # lower -> upper & upper -> lower
```

```
True
False
True
False
True
True
True
False
False
mENTORSHIP pROGRAM!77
```

```python
# startswith (Boolean Return)
_str = 'Hey, Welcome to the World of Programming!'
print(_str.startswith('hey')) # False
print(_str.startswith('Hey')) # True
print(_str.startswith('Hello')) # False
print(_str.startswith('Hi')) # False
print(_str.startswith('H')) # True
print(_str.startswith('Welcome',5)) # True
```

```
False
True
False
False
True
True
```

```python
# endswith (Boolean Return)
_str = 'Hey, Welcome to the World of Programming!'
print(_str.endswith('hey')) # False
print(_str.endswith('Program')) # False
print(_str.endswith('Programming!')) # True
print(_str.endswith('!')) # True
print(_str.endswith('g!')) # True
print(_str.endswith('ing!',-4)) # True
print(_str.endswith('ing!')) # True
print(_str.endswith('ing!',-3)) # False
```

```
False
False
True
True
True
True
True
False
```

```python
# .count() -> 'substring' -> count the substring
_str = "This is a Python Course!"
print(_str.count('is'))
```

```
2
```

```python
print('is' in _str)
```

```
True
```

What is an Array: