1. String’s indexed character cannot to be assigned a new value, as Strings are **immutable.**  
Example: name = "Reinforcement"  
print(id(name)) #73472  
name[0] = "V" # Raises TypeError

2. **+=** operator is used to concatenate strings, it does not violate Python’s string immutability Property. Because doing so new creates a new association with data and variable. E.g. str\_1="a" and str\_1+="b. Effect of this statements to create string ab and reassign it to variable str\_1, any string data is not actually modified.

3. A Character in string can be indexed using string name followed by index number of character in square bracket. **Positive Indexing** i.e. first index is 0 and so on, or **Negative Indexing** i.e. last letter is -1 and so on can be used to index a character.

Example:

in\_string **=** "iNeuron Full Stack Data Science"

print(in\_string[9],in\_string[10],in\_string[2]) *# Positive Indexing*

print(in\_string[**-**1],in\_string[**-**5],in\_string[**-**2]) *# Negative Indexing*

Output: u l e

e i c

4. We can access elements of sequence datatypes by using slicing and indexing. Indexing is used to obtaining individual element while slicing for sequence of elements.

Example:

in\_string **=** "iNeuron Full Stack Data Science"

print(in\_string[1],in\_string[3],in\_string[5]) *# Indexing*

print(in\_string[1:15]) *# Slicing*

Output: N u o

Neuron Full St

5. Indexed characters and sliced substrings have datatype **String**.

Example:

in\_string **=** "iNeuron Full Stack Data Science"

print(type(in\_string[3])) *# Indexing -> str*

print(type(in\_string[1:10])) *# Slicing -> str*

Output: <class 'str'>

<class 'str'>

6. Object that contains sequence of character datatypes are called String.

7. **+**, **+=** and **\*** allow to combine one or more smaller strings to create a larger string. **<string>.join(<sep>)** method joins element of iterable type like list and tuple to get a combined string.

Example:

in\_string **=** 'iNeuron '

in\_string **+=** 'Full Stack Data Science'

print(in\_string **+** ' FSDS')

print('FSDS '**\***3)

print(" "**.**join(['I','N','E','U','R','O','N'])) *# List Iterable*

print(" "**.**join(('I','N','E','U','R','O','N'))**.**lower()) *# Tuple Iterable*

Output: iNeuron Full Stack Data Science FSDS

FSDS FSDS FSDS

I N E U R O N

i n e u r o n

8. Checking the target string with **in** or **not** Operators before using the index method to find a substring just helps confirming availability of substring and thus avoid raising of **ValueError.**  
**Example:**  
in\_string = "Kunal"  
in\_string.index('x') # Raises ValueError  
in\_string.index('u') # 1

9. The String Operators and built-in methods to Produce Simple Boolean (True/False) Results are:

* **in**
* **not**
* **<string>.isalpha()**
* **<string>.isalnum()**
* **<string>.isdecimal()**
* **<string>.isdigit()**
* **<string>.islower()**
* **<string>.isnumeric()**
* **<string>.isprintable()**
* **<string>.isspace()**
* **<string>.istitle()**