

Strings

→ Can be enclosed `<` and `>`

Eg:- `str1 = 'Hi'`
`str2 = "Hello"`

→ `" "` is used to include
's in strings.

Eg:- `"Student's."`

→ Length of string :-

len(str).

→ Escape sequence characters:-

`\n` → newline

Eg:- `str = "Hi \n Hello"`

It \rightarrow take space.

BASIC OPERATIONS ON STRINGS:-

i). Concatenation :-

str1 = "Hi"

str2 = "Hello"

print(str1 + str2) \rightarrow HiHello.

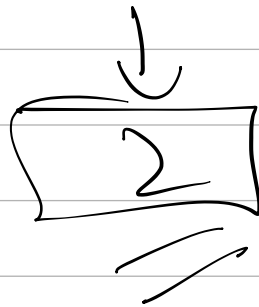
ii). Length of string :-

Eg:-

len()

str1 = "Hi"

print(len(str1))



★). Spaces are also counted in len().

INDEXING IN STRING :-

str = Hi Hello
0 1 2 3 4 5 6 7

str[0] = 'H'

str[4] = 'e'

★). Only can be used to
access characters in string.
Eg:- a = str[0]
a → stores 'H'.

★). Cannot modify strings
using indexing.

gstr = 'Hi Hello'
0 1 2 3 4 5 6 7

str[2] = g // not possible.

Slicing

→ Accessing parts of a string.

str1 = "Hi Hello"
0 1 2 3 4 5 6 7

str1[4:7] = "ell"
↓ ↗ not included.
included

// To get "Hello"

str1[3:8] = "Hello"
↓
len(str1)

Alternate,

str1[3:len(str1)]

*]- str1[:L]

Default starts from 0.

*). str1[5:]

Default goes till end.

Negative Index:-

M a n g o
-5 -4 -3 -2 -1

str1[-3:-1] \rightarrow "ng"

String Functions

str1 = "Hi Hello"

*). str.endswith("lo")

\downarrow
returns true if str1
ends with 'lo'.

★). str.capitalize():

↓
This makes first letter
of string Capital and
prints.

Note:- Only while using
this function it works,
original string is not changed.

// To change the original string.

```
str1 = "hi"
```

```
str1 = str1.capitalize()  
print(str1)
```

↓
"Hi"

✓

★). str.replace (old, new)

Eg:-

str1 = "Hi Hello Man"

str1.replace('e', 'a')



makes

str1 = "Hi Hallo Man"



★). str1.find(word)



returns

1st index

of 1st occurrence

Eg:-

str1 = "Hi Hello Man"
0 1 2 3 4 5 6 7 8 9 10 11

str1.find("a")



prints 4



Egⁿ str1 = "Gg Man from Man."
0 1 2 3 4 5 6 7 8 9 10 11 12

str1.find("Man")

↓
returns index of first "Man"

i.e 3
//

*). str1.count("Man")

↓
returns 2 as

there are 2 occurrences
//

Conditional Statements

if - elif - else (SYNTAX).

Order of checking.

if \rightarrow elif \rightarrow else.

★). If we use if two or more times, All 'if' are executed.

★). Elif \rightarrow only executes if is false.

★). Else \rightarrow executes when both if and elif are false.

Nested If :-

Eg:- Check can vote or cannot. But if age is less than or equal to 20, print also "Hi teen"

```
if (age >= 18):
```

```
    print("can vote")
```

```
    if (age <= 20):
```

```
        print("Hi teen")
```

```
else:
```

```
    print("cannot vote")
```
