

String Concept

* Justifying text with `rjust()`, `ljust()` and `center()`

① `'Hello'.center(10)`

O/P: `'Hello'`

② `'Hello'.rjust(10)`

\Rightarrow `'Hello'`

③ `'Hello'.ljust(10)`

O/P: `'Hello'`

An optional second argument `rjust` and `ljust` will specify a fill character apart from the space character -

Eg: `'Hello'.center(10, '*')`

O/P: `*****Hello*****`

`'Hello'.ljust(20, '-')`

O/P: `Hello-----`

Date : / /

'Hello' * 8 Just ('200g' = ')

OP ! == == == == == == == Hello '

{ Python String Cheat sheet }

s. Capitalized () → Capitalize first word of the string in the list

s. lower () Convert all string to lower case

s. swapcase () If Capital Converts to all small and vice versa

s. title () Capital each first character present in list

s. upper () Convert it into upper case.

{ Sequence operation }

s2 in s → Return true If s contain s2

s + s2 → Concat s and s2

len(s) → length of s

min(s) → Smallest character of s

max(s) → Largest character of s

Sequence operation IF

`s2 not in s` - Return true if `s` does not contain `s2`

`s * integer` → Concat # 'Hello' ⇒ HelloHelloHello

`s[index]` - Character at index of `s`

`s[i:j:k]` - Slice of `s` from `i` to `j` with step `k`

`s.count(s2)` - Count `s2` in `s`

White Space

`s.center(width)` ⇒ Centre `s` with black padding of width # 'b'
⇒ ' b '

`sisspace()` ⇒ Return true if space -

`s.ljust(width)` ⇒ Left Justify `s` with total size of width # 'Hello'

→ 'Hello'

`s.rjust(width)` ⇒ Right Justify `s` with total size of width # 'Hello'

`strip()` → Remove whitespace ⇒ 'Hello'

Find / Replace

`s.index(s2, i, j)` = Index of first occurrence of s_2 in s after i and before j index

`s.find(s2)` = Return lowest index of s_2 in s

`s.index(s2)` = Return lowest index of s_2 in s If not found give value error

`s.replace(s2, s3)` = Replace s_2 with s_3 in s

`s.replace(s2, s3, count)` \Rightarrow Replace s_2 with s_3 at most count times.

`s.rfind(s2)` \Rightarrow Return highest index of s_2 in s

`s.rindex(s2)` \Rightarrow Return highest index of s_2 in s If not found give value error

Case 11

`s.casefold` \Rightarrow Casefold s (aggressive lowercasing for caseless matching) $\#$ 'Borai' \Rightarrow 'SSorat'

`s.islower()` = Return true if s is lower case

`s.istitle()` = Return true if follow Rule of title

`s.isupper()` = Return true if all is upper case

Inspection !

`s.endswith(s2)` \Rightarrow Return true if end with s2

`s.isalnum()` \Rightarrow Return true if all alphanumeric (0, 9, a, z, exclude special char)

`s.isalpha()` \Rightarrow Return true if s is alphabetic (a-z)

`s.isdecimal()` \Rightarrow Return true if s is decimal (0-9)

`s.isnumeric()` \Rightarrow Return true if s is numeric
`'12345'` \rightarrow True, `"12.35"` \rightarrow False

`s.startswith(s2)` Return true if s started with s2

Splitting !

Suppose \Rightarrow `s = 'Hello'`

`s.join(['123'])` \Rightarrow Return s join by iterable

\Rightarrow '1Hello 2Hello 3Hello'

`s.partition(sep)` \Rightarrow Partition string at sep and

Return 3 tuple with part before the sep itself and part after ~~'Hello'~~ 'Hello'

\Rightarrow 'he', 'l', llo

s.partition → # Hello → 'He', 'l', 0

s.rsplit(sep, maxsplit) → Return list of split by sep with right most maxsplit performed

s.splitlines() → Return a list of line in
'Hello\nworld' →
['Hello', 'World']

In spection 11

s[i:j] → Slicing S from i to j

s.endswith((s1, s2, s3)) → Return true If s ends with any of string tuple s1, s2, s3

s.isdigit() → Return true If s is digit

s.isidentifier() — Return true If s is valid identifier

s.isprintable() — Return true If s is printable

White Space !!

s. center (width, pad) \Rightarrow # s. centre (10, '*')
 \Rightarrow [**Hello***] output

s. expandtabs (integer) \Rightarrow Replace all tabs with space of
 tabsize integer —
 \Rightarrow 4 # Hello\tWorld \Rightarrow Hello...World

~~s. lstrip~~
 s. lstrip \Rightarrow Remove leading whitespace from s # 'Hello'
 \Rightarrow 'Hello'

s. rstrip \Rightarrow Remove trailing whitespace from s # 'Hello'
 \Rightarrow 'Hello'

s. zfill (width) = left fill 's' with ASCII '0' digit with
 to total length width # '42' \Rightarrow '0042'

text = '42'

print (text.zfill(4)) #

O/P '0042' Ans.