

Strings

↳ `my_str1 = "python"`

~~`my_str2 = 'prog'`~~

~~`my_str3 = '''`~~

~~`Hello`~~

~~`'''`~~

↳ Accessing

~~`my_str1 = " " # Space`~~

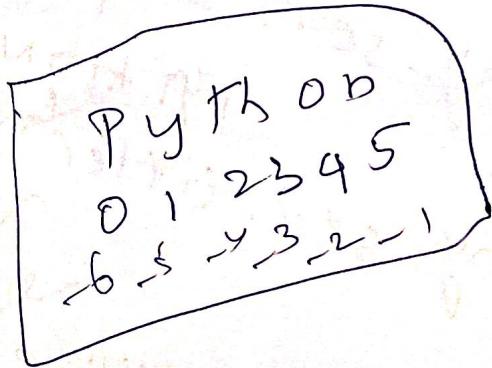
~~`my_str2 =`~~

~~`bool(my_str1) # False`~~

~~`bool(my_str2) # True`~~

~~`my_str = "python"`~~

~~`my_str[0] → P`~~



~~`my_str[-1] → n`~~

~~`my_str[:5] → Pytho`~~

~~`my_str[0:] → python`~~

~~`my_str[0:-1] → python`~~

to change

`my_str = "python"`

`del my_str`

`my_str1 = "python!"`

↳ How to delete a string.

→ Length of string

`len(my-str) # 6`

~~my-str[0] = "w" ✗~~

~~Case Conversion~~

~~my-str = "Python Scripting"~~

`my-str.lower() Python Scripting ✓`

~~my-str.swapcase()~~

`my-str.swapcase() PYTHON SCRIPTING ✓`

~~my-str.title() || Python Scripting tutorial~~

`my-str.title() || Python Scripting Tutorial ✓`

~~.Capitilize()~~

`.Capitilize() Python Scripting tutorial ✓`

~~.center()~~

`.center() Python Scripting tutorial ✓`

~~# Python Scripting tutorial~~

`# Python Scripting tutorial (lower) ✓`

- boolean
- ↳ my-str = "Hello"
 - .isalnum() → True
 - .isalpha() → True
 - .islower() # False
 - .isupper() # True
 - .startswith("P") # False
 - .startswith("H") # True
 - .startswith("llO") # True
 - .endswith("llO")
 - ↳ "Hello how are you"
 - .istitle() # False
 - .isspace() # False
 - ↳ "Hello"
 - .isspace() # True
 - .isalpha() # False
 - .isnumeric() # True
 - ↳ "123456"
 - .isnumeric() # True
 - .istitle() # True
 - ↳ "python" →

join | center | zfill

↳ my_str = "python"
 ".".join(my_str)
 → py-t-h-o-n
"t".join(my_str)
 → py

↳ my_str = "python"
 my_str.center(20)
 python
 python
 zfill(10)

↳ my_str.zfill(10)
 "0000python"

strip|split:

↳ my_str = " python ".lstrip()
 my_strip = str.strip(' - ')
 python

↳ x = " python yyyy ".rstrip('y')
 x.strip('p').rstrip('y')
 python

$x = "python is easy and it is very"$ ③

$x.split('is')$
 $\hookrightarrow ["python", "easy", "and", "it", "very popular"]$

~~Count | Index | find~~

$\hookrightarrow my_str = "Narendra Reddy"$
 $my_str.count('a') \# 2$

$\hookrightarrow \underline{index}$
 $my_str = "python is easy and it is popular"$
 $my_str.index('is') \# 7$

$.index("is", 8) \# 22$

$.index("is", 23) \# \text{error}$

$.index("is", 23) \# -1$

$.find("is", 23) \# -1$

$.find("2") \# -1$

$java_version = "java version 1.6"$

$\hookrightarrow java_version = "java version 1.6"$
 $.find("java") \# 0$

```
import os  
size = os.get_terminal_size().columns  
given_string = input(" ")  
pt(given_string, center(size).title())  
.ljust(size).title()  
.rjust(size).title()
```

python

// 80 python programming python program

python Programming

~~ABDUL~~

↳ S1 = "Hello"
S2 = "984 id123@mail.com"
S3 = "my BOOK"
S4 = "Hello"

0 1 2 3 4
-5 -4 -3 -2 -1

↳ S1 = "Hello"
S1[0] = "t"
S1[-5] = H

↳ len(S1) → 5

↳ for i in S1:
 pt(i, end="") } }

for i in range(len(S1)):
 pt(S1[i], end="")

(4)

String Literals

$s1 = "Hello"$

$s2 = "Hello"$

$s3 = "Hello"$

$s5 = "Clark"$

"Clark"

"Clark"

Index Slicing

$s1 = "Hello world"$

$s1[0] = "H"$

$s1[-7] = "o"$

$s1[0] = "w"$

$s1[1:7] = "Hello"$

$s1[1:7] = "Hello"$

$s1[1:7] = "wor"$

$s1[-5:-2] = "wor"$

$s1[-11:-2] = "wor"$

$s1[-11:-2] = "Hello"$

Step

$0:11:1$
Hello world

$0:11:2$
How world

$s1[:]$
 $s1[::-1]$ // reverse

$s2 = "Hello world"$
 $s2[-3:9:-1]$
Forward direction only
No print

$s2[8:2:-1]$
row 0

operator

ConCat

$$SI = "abc"^{11} "cde"^{11}$$

$$S_1 = \text{ACE}$$
$$S_2 = \text{ICDE}$$

$$S \vdash "abc"$$

"abcde^fg^h"

$s_1 = "absolu\text{t}e"$

$$S_1 + S_2 \stackrel{?}{=} \begin{matrix} ab \\ abc \end{matrix} \begin{matrix} ccde' \\ abc \end{matrix} \begin{matrix} abc \\ abc \end{matrix} \begin{matrix} abc \\ abc \end{matrix} \begin{matrix} abc \\ abc' \end{matrix}$$

$$S_1 \neq S_2 \neq \bar{u}^{abc} \bar{a}^{bc} \bar{a}^{bc} \bar{a}^{bc}$$

$$S_1 \neq S = \{x\}$$

$\hookrightarrow s1 = "abcdefg"^{h''}$

$\hookrightarrow S_1 = \{abc\}$
 $S_2 = \{ABC\}$
 apple < ball < cat < dog < pyth
 $a < b < c$
 $A < B < C$
 $q < g < A < B < C < 90$ $q7 < 122$

$$48^{-5^7} \stackrel{?}{=} \sin^{-5^2} \rightarrow \text{True}$$

“software”

11 Sand ware

↳ "hardware" } false

`s1 = "python"`
`s2 = "py char"`

"python" "class"

S1 = "ppy charm"

~~Class~~ ~~x~~ ~~size~~ ~~size~~ ~~size~~ ~~size~~

Find / Index

`s1 = "Hello how are you"`

`s1.find('o') # 4`

`s1.find("How") # 6`

`s1.find('K') # -1`

`s1.find("how", 0, 5)`

`# -1`

`rfind [sub, start, end]`

`s1.rfind('o') # 15`

`s1.`

`s1.index('o') # 4`

`s1.index("how") # 6`

`s1.index('x') # value error`

`s1.rindex('o') # 15`

`s1.rindex('o', 0, 15) # 7`

Count

`s1.count('o') # 3`

`s1.count("me") # 0`

Padding

(5)

↳ ljust(width, fillchar)

$s = "Hello"$

$s.ljust(10, "*")$

\rightarrow Hello *****

↳ s.rjust(10)

$s.rjust(10)$

$s = "Hello"$

$s.center(10, "*")$

\rightarrow ****Hello****

↳ zfill(width)

$s.zfill(10)$

\rightarrow 00000Hello

↳ lstrip

$s = "Hello"$

$s.lstrip()$

\rightarrow Hello

$s.lstrip("$") \#Hello$

↳ s = "Hello"

$s.lstrip()$

\rightarrow Hello

$s = "Hello$"$

$s.lstrip("$")$

↳ s = "#! Hello\$"

$s.strip("#! $")$

\rightarrow Hello

Replace

(6)

↳ $s1 = "a-b-c-d-e"$
 $s1.replace("-", "", "")$
 a,b,c,d,e
 $s1.replace("-", "", "", 3)$
 $a, b, c, d - e$
 $s1.replace("*", "m")$
 $s1.replace("", "")$
// no changes

↳ $s1 = "narendra.ny@gmail.com@yahoo"$
 $s1.replace("gmail", "yahoo")$
narendra. ~~ny~~ $@$ yahoo.com

↳ $s1 = "xy2"$
 $s2 = "abc"$
 $s1.join(s2)$
 $axy2bxg2cxy2$

y ~~join~~

↳ $s1 = "/"$
 $s2 = "abc"$
 $s1.join(s2)$
 $a/b/c$

split
 $s1 = "john smith Ajay"$
 $s1.split()$ { type(s2)
list
["john", "smith", "Ajay"] }

$s1 = "john-Smith-Ajay-Khan-james"$
 $s1.split(" ")$ some string
["john", "Smith", "Ajay", "Khan", "james"]

$s1.split("-") \Rightarrow ["john", "smith", "Ajay", "Khan", "james"]$

prefix and suffix

↳ `s1 = "python is very easy"`
`s1.startswith('python')`

`#True`
`s1.startswith('is', 7)`

`#True`
`s1.endswith('easy')`

`#True`
`s1 = "abc@gmail.com"`
`s1.endswith('.com')`

`#True`

↳ `s1 = "python is easy"`
`s1.partition("is")`
`('python', 'is', 'easy')`

`type(s2)`

`<class 'tuple'>`

↳ `s1 = "python is easy"`
`s1.partition('s')`
`('python', ' ', 'easy')`

inquiry.

(7)

↳ $s_1 = "Hello"$
 . isalpha() # True
 . islower() # False
 . isupper() # False

↳ $s_1 = "Hello how are you"$
 $s_1 = "Hello"$
 s1.title()
 # False

↳ $s = " "$
 s.isspace()
 # True

$s = " "$
 s.isspace()
 # False

len(s) # 0

"nirrif"

$s = "nirrif"$
 s.isspace()
 // whitespace

↳ $s = "n"$
 s.isprintable() → "inirifibla esc"

$s_1 = "abc"$
 s1.isidentifier() # False

"abc" s2.isidentifier()
 # True

$\hookrightarrow s1 = "7235"$

$s1.isdigit()$

#True

$s2 = "73.23"$

$s2.isdigit()$

$\hookrightarrow s2 = "37542"$

$s2.isdecimal()$

#some unicode
#True

$\rightarrow s1 = "35742"$

$s1.isnumeric()$

#True

$\rightarrow d = "abc"$

$d.casefold()$ #abc

$d.ascii()$ #True