

ȘIRURI DE CARACTERE

→ indexate de la 0 și IMUTABILE, nu se term cu '0', lung. este memorată dinamic, se află în O(1)

! $s = \text{"test"}$ # test $s = \text{"test"}$ # TEST
 $s.upper()$ $s = s.upper()$

Bazăm de șiruri; $s = \text{"test"}$ și $t = \text{"test"}$ vor avea ca. referință
 pt. a verifica dacă două șiruri sunt identice putem scrie $s == t$ ceea ce are O(1)

$s = \text{"test"}$
 $x = \text{"st"}$ → $s == x$ = False
 $t = \text{"te"} + x$
 $s = \text{"test"}$
 $x = \text{"st"}$
 $t = s.lower() + x$ → $s == t$ = True
 ↳ este o metodă lentă

indici poz. și neg.

	0	1	2	3
s	t	e	s	t
	-4	-3	-2	-1

! oglinditul unui str
 str[::-1]

înlocuirea car. de pe poz p

$s = s[:p] + \text{"car"} + s[p+1:]$
 ↳ slice ↳ exclusiv ↳ inclusiv

	0	1	...	p	...	m-1
s						

operatori pt. prelucrare

$+$ = concatenare
 $*$ = concatenare repetată
 $<, <=, >, >=, ==, !=$ pt. comparare lexicografică

funcții predefinite

len(s) → întoarce nr. de caractere
str(s) → transformă val. exp. într-un șir de caractere
min(s) → cel "mai mic" car.
ord(c) → furnizarea codul ASCII
chr(n) → furnizarea car. cu acel cod ASCII

metode pt. prelucrare

1. strip([car])

s = "www.exemplu.org" → s = "exemplu"

s = s.strip("...misgroun?!")

! dacă nu specificăm nimic, metoda va elimina spațiile de la sf. și începu

2. center([latime, car])

s = "test" → implicit spațiu

s = s.center(18, ".") → s = .. test..

3. format(ma. var.)

• accesare secvențială

s = "Ana are {} mere {}!"

s = s.format(5, "roșii") → Ana are 5 mere roșii!

• accesare pozițională

s = "Ana are {1} mere {0}!"

s = s.format("roșii", 3+4)

• accesare prin nume

s = "Ana are {numar} mere {culoare}!"

s = s.format(culoare = "roșii", numar = 7)

4. lower()

= litere mari → litere mici

5. upper()

= litere mici → litere mari

6. swapcase()

= mici → mari & mari → mici

7. title()

= fiecare cuvânt va fi de forma: Litere mare mică mică...

8. capitalize()

= șir → propoziție : Ana are mere

9. isascii()

verifică dacă toate caracterele sunt ASCII

10. isalpha()

— " — sunt litere

11. isdigit()

— " — cifre

12. isnumeric()

— " — caractere numerice

13. isalnum()

— " — litere / cifre

14. isspace()

— " — spații albe (tab, space...)

15. islower()

— " — litere mici

16. isupper()

— " — litere mari

17. istitle()

— " — prima literă din cuvânt ↑, restul ↓

True
sau
False

18. count (subsir, start, stop)

↳ optional

S = "hahahaha"
print (S.count ("aha")) → 2

! va număra și din interiorul cuvintelor

19. find (subsir, start, stop) → prima poz. unde apare subsirul / -1
↳

20. rfind (" ") → cea mai mare poz. / -1

21. startswith (pref., start, stop) → dacă are acel prefix sau nu

22. endswith (" ")

23. replace (subsir, nou, max)
↳

24. split (separator) → creează o listă cu cuv. delimitate de ac. sep.
↳ poate fi doar unul idacă vrem ca strtok, trebuie să dăm
↳ implicit spațiu replace întâi

25. join (listă - subsir) → refac propoz. cu un separator ales de noi

" " .join (lista) → va pune spații