20.10.2022

Curs 3

Șiruri de caractere
(stringuri)

Șir de caractere = o secvență de caractere indexat de la 0

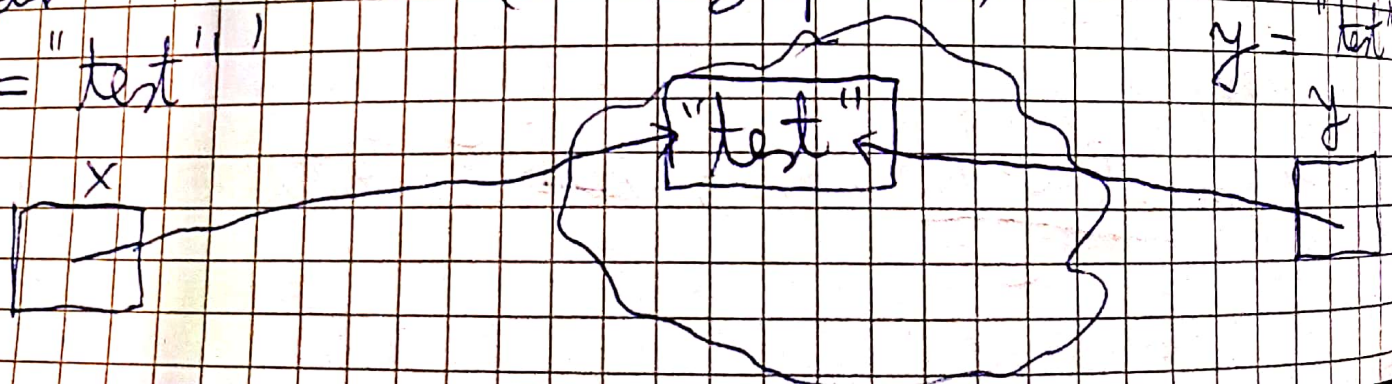
IMUTABILĂ

conținutul său nu mai poate fi modificat

```
s = "test"
s[4] = 'a'
s = s[:1] + 'a' + s[2:]
```

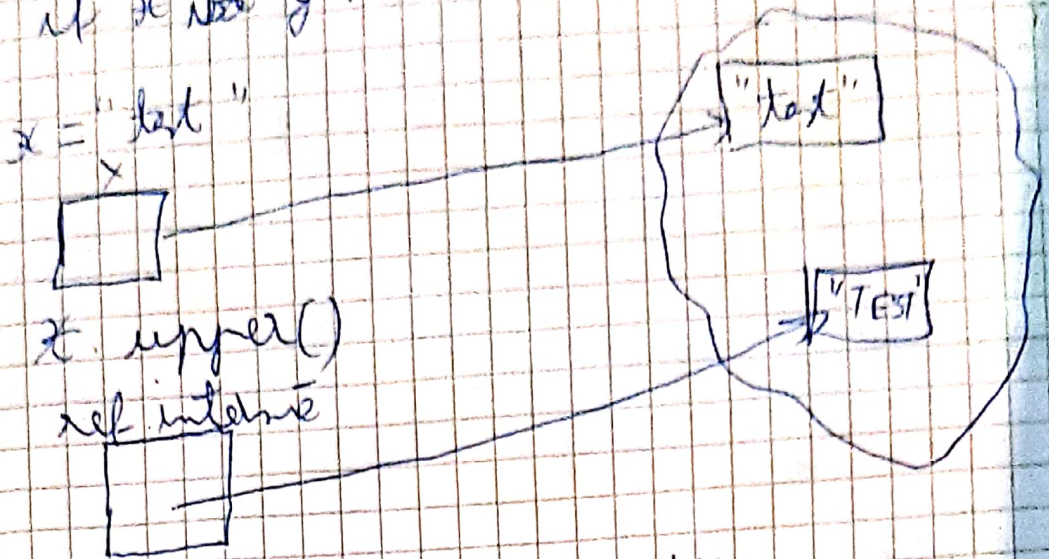
Bazin de șiruri (string pool)

```
x = "test"
y = "test"
```



if x == y : → O(n)
if x is y : → O(1)

x = "test"



x.upper()
ref interne

print(x) => "test" !!!
=> x = x.upper()

s = "te"
t = "st"
v = s + t ?
v = v + "st"

s = 'test'
s = "test"
s = "test" → s = "Ana
s = 'Ana
are
mere'
print(s) → "Anaaremere"

v = "te" + "st"
intern(sir) → modulul sys
import sys
s = sys.intern(s)

s = "Ana
are
mere"
print(s) → "Ana
are
mere"

x = 10
print(f"{{{x}}}") → "{10}"

Accesarea elem unei vol de tip

indexare directă →	0	1	2	3	4	5	6
▷ =	"n"	"a"	"o"	"a"	"y"	"a"	"o"
	-14	-10	-9	-8	-4	-6	-5

▷[3] = ▷[-8] = "a"

AM "a"

Secv. de indici (slice)

▷[2:6] = "ogal"

incluziv excluziv

▷[-8:-4] = ▷[-8:7] = [3:-4]

▷[:-3] = "pe"

▷[::-1]

Operatori pt. șiruri

+, *, op. relaționali

"doc" * 3 = "dodocdoc"

"A" < "a"

"ce" in "merge" = True

Funcții pt. șiruri

a) len(șir)

len("text") = 4 len([1, 2, 3]) =

b) str(expresie)

str(1 + 2 == 3) = "True"

c) min(șir) / max(șir)

d) ord(caracter)
ord("A") = 65

e) chr(x)
chr(65) = "A"

funcție(x, ...)

Metode pt. șiruri (clase de.)
șir.metodă()
sau
alt șir

1) metode pt. formatare
a) strip([șir])
parametru opțional []

s = " test "
s.strip() = "test"
s = "programare"
s.strip("prac") = "ogram"

b) center(lățime, [caracter])
s = "test"
s.center(10) = " test "

c) format
s = "Ana are {mere} !"
s.format(7, "verzi")
s = "Ana are {1} mere {0} !"
s.format("verzi", 7)
s = "Ana are {x} mere {ddd} !"

1. format(...)

2) Metode pt. tipul caracterelor

lower()
upper()
swapcase()
title()

3) Metode de clasificare

isascii()
isalpha() ← la nivel de șir!
isdigit()

4) Metode de căutare

a) count(șir, [start], [stop]) ← numără doar aparițiile disjuncte
"babcbabcc".count("abc") = 2
"aaaa".count("aa") = 2

b) find (—//—, —//—, —//—)
c) rfind (—//—, —//—, —//—)
d) startswith (—//—, —//—, —//—)
e) endswith (...)
f) replace (șir_1, șir_2, [nr])

5) Metode de construire/împărțire

a) .join([listă șiruri])

s = ", ".join(["Ana", "Ion", "Mihai"])

s = "Ana, Ion, Mihai"

b) .split([caracter])

"Ana are mere, pere și prune"

t = s.split() => t = ["Ana", "are", "mere", ...]