Strings

1. Introduction
2. Types of Characters
3. ASCII
4. Problem Solving
    1. Print all characters (Iteration)
    2. Print 1st & last char (Indexing) (Immutability)
    3. Print ASCII values [ord()]
    4. Repeat string [+, *]
    5. Count upper case [isupper()]
    6. Convert upper to lower [chr(), lower(), upper())
    7. More inbuilt functions

String

print("Hello world") | print("world Hello")

"Hello world" → string

"world Hello" → characters

print ('Hello world')

→ sequence of characters

- a-z
- A-Z
- 0-9
- special characters: $ @ ! % ( ) [ ] { } # - . —

ASCII 0-255 | Unicode

'a'-'z'

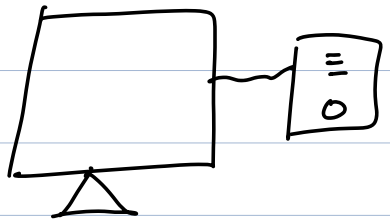
'A'-'Z'

'0'-'9'

Enter a password : ****

Q) Valid strings?

1) "abaaa" ✓
2) "ab$" ✓
3) "ab$' ✗ An opening " has to be closing "
4) 'ab223' ✓
5) "a 012" ✓
6) "a bcdef"" ✗



1/0 Binary.

Dumb

Numbers

5
10
237
1129

= $(110)_2$

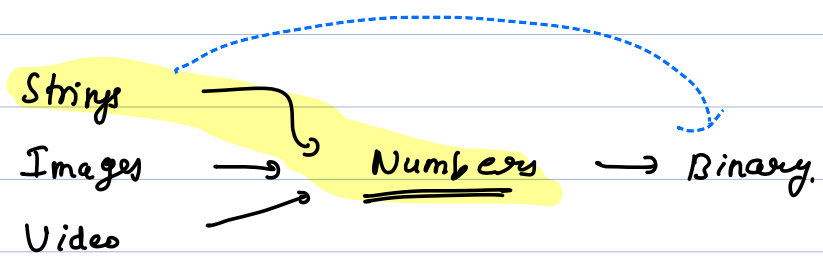
→ Binary ⇒ Base 2

Decimal

⇓

Base 10

0 - 1 - 2 - 3 - - - - - 9



"abc"

x y z

x = 23

x = 97

x = 65

x = 48

x = 12

55 X

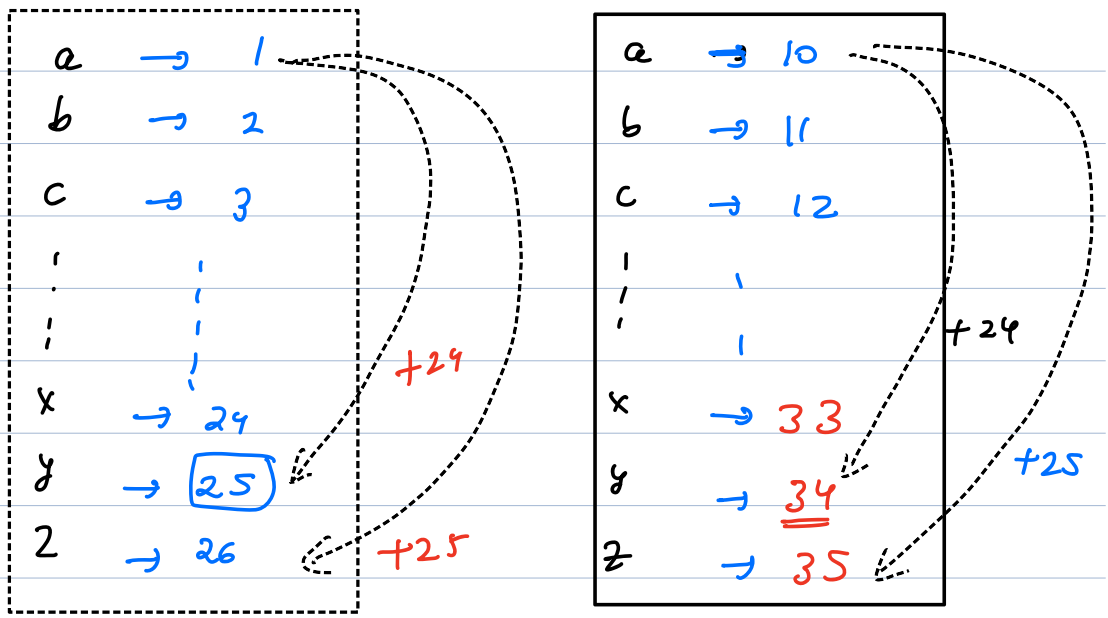
Text msg

A

WA

B

WA



26

A ⟶ B

Xyz 24, 25, 26 opq

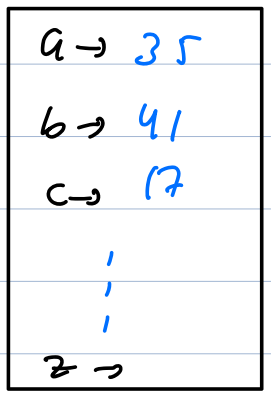
a → 10
b → 11
c → 12
d → 13
e → 14
f → 15
g → 16
h → 17

i → 18
j → 19
k → 20
l → 21
m → 22
n → 23
o → 24
p → 25

q → 26
r
s
t
u
v
w
x

y
z

Donald Trump → Poland Trump



Random.

ASCII

↓

American Standard Code For Information Interchange.

Remember

Lowercase		Uppercase		Numbers
'a'	→ 97	'A'	→ 65	'0' → 48
'b'	→ 98	'B'	→ 66	'1' → 49
'c'	→ 99	'C'	→ 67	'2' → 50
⋮	⋮	⋮	⋮	⋮
'y'	→ 121	'Y'	→ 89	⋮
'z'	→ 122	'Z'	→ 90	'9' → 57

+24 (Lowercase → Uppercase)

+25 (Uppercase → Numbers)

[97 − 122] = 122 − 97 + 1 = 25 + 1 = 26

[a, b] = b − a + 1

[1, 10] = 10 − 1 + 1

included

[2, 10] = 10 − 2 + 1 = 9

2, 3, 4, 5, 6, 7, 8, 9, 10

E J O T Y

↑ ↑ ↑ ↑ ↑

5 10 15 20 25

P ⇒ 16 +64 = 80

K ⇒ 11

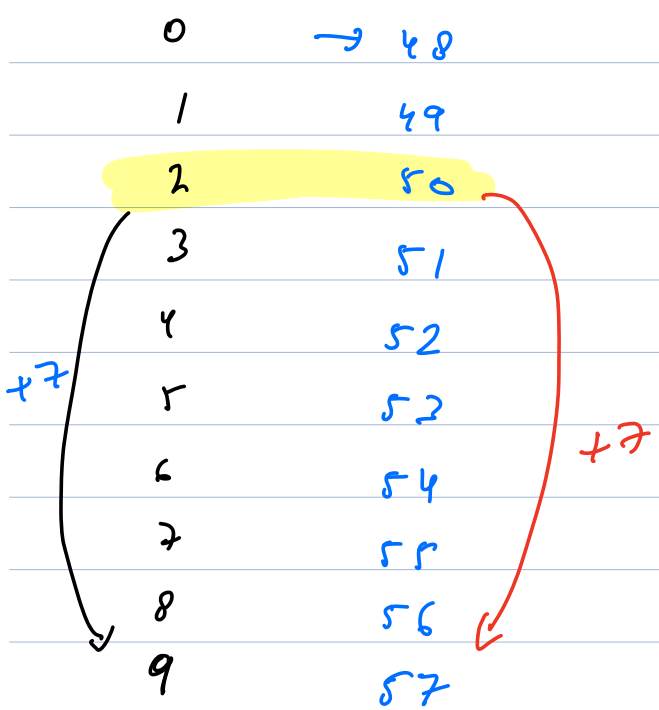
A: 65 1+64

B: 66 2+64

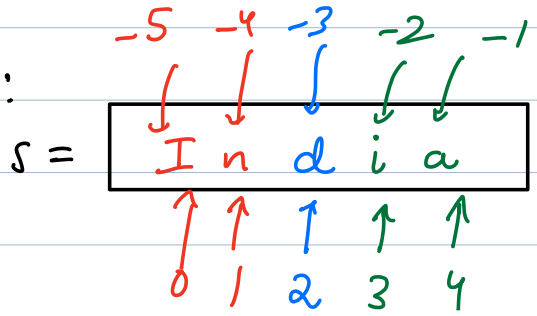
C: 67 3+64

D: 68 +64

E: 69 = 5



Indexing:



s[0] # I

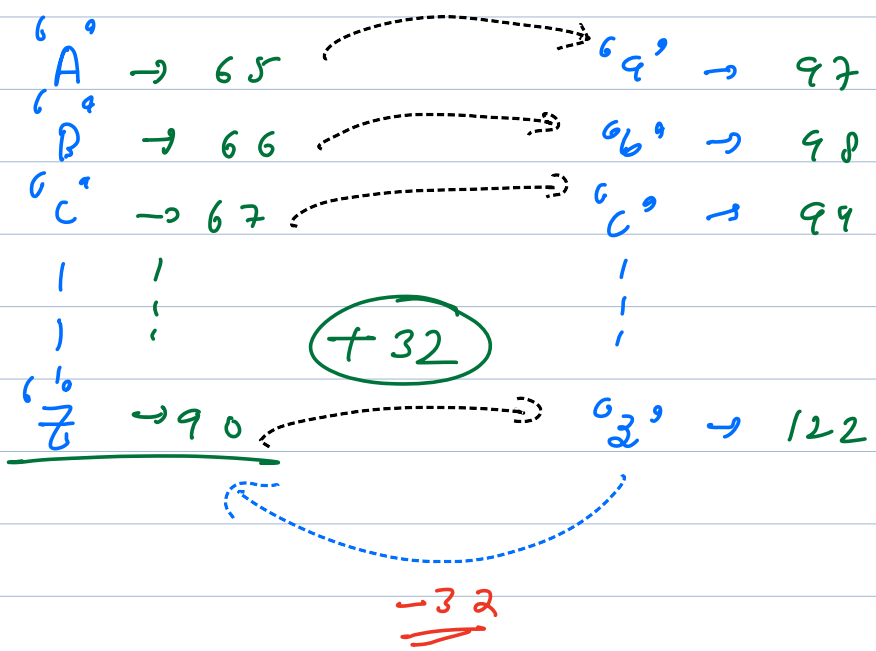
s[-1] # a

String Slicing

s[:2] # 'In'

s[-2:] # 'ia'

Conversion Upper to Lower



★ $2^5 = 32$