

fruits[2] ↗ returns item in index 2

easy if thing I'm looking for nicely maps to indexes 0 to len-1.

Imagine phone numbers

"Amp" → "321-999-0123"

"Erick" → "407-123-4567"

```
nums = [ ["Amp", "321-999-0123"],  
          ["Erick", "407-123-4567"],  
          ...  
          ]
```

100,000
names

num = -1

for i in range(len(nums)):

if nums[i][0] == "Caroline":
 num = nums[i][1]

if num != -1

print("Caroline's # is", num)

Key issue: I don't know (which index)
Caroline is in! So I have to look at
all of them.

nums["Caroline"]

Instead of finding which integer index Caroline is in, what if "Caroline" was the index itself?

This exists and it's a dictionary.

Dictionary associates words with their definitions.

"cat" → "animal..."
"orange" → "fruit..."
↑ ↑
KEYS VALUES

① To add an entry

nums["Caroline"] = "407-429-3617".

② Check if entry exists

if "Caroline" in nums:
 print(nums["Caroline"])

③ Change entry (same as adding)

nums["Caroline"] = "321-929-0061"

Example #1

Censor

replace [badword] = less offensive word

message display substitute all
badwords with their approved replacement

File

hello bob

how are you?

str = "hello bob how are you? \n"

msg = str.split()

msg → ["hello", "bob", "how", "are", "you?"]

Example #2

voting example

"Celine"

"Davis"

"Celine"

Candidate NumVotes

Key Value

Celine → 3

Davis → 1

Ali → 1

↓ "Ali" → defect wasn't in,
"Celine" add entry

Sets

fruit = set()

fruit.add("orange")

fruit.add("pineapple")

fruit.add("orange") → NO CHANGE

Set operators

Union

| ~~∪~~

(everyone belongs
at least 1 class)

Intersection

& ~~∩~~

(everyone who belongs
to both)

Difference

—

$A - B$ → everyone in A who is
not in B

Symmetric

difference

↗

$(A - B) \cup (B - A)$

everyone in exactly
one set