range() and Dictionary Practice

Announcements

- **Reminder: EX04: Dictionary Utils** due Tuesday, 10/7

```
1 vend: dict[str,str] = {"A1":"Oreos", "A2":"Lays", "B1":"Coke", "B2":"7up"}
2 flavors: set[str] = {"Orange", "Cherry", "Lime"}
```

2.1. What will be printed?

```
1 for prod in vend:
2   print(prod)
```

A1, A2, B1, B2

2.2. What will be printed?

```
1 for prod in vend:
2   print(vend[prod])
```

"B2" "7up"

Oreos, Lays, Coke, 7up

2.3. What will be printed?

```
1 for flav in flavors:
2   print(flav)
```

Orange, Cherry, Lime (order varies, but all values in set are printed)

2.4. What will be printed?

```
1 if "Berry" in flavors:
2   print("Available!")
3 else:
4   print("Out...")
```

Out...

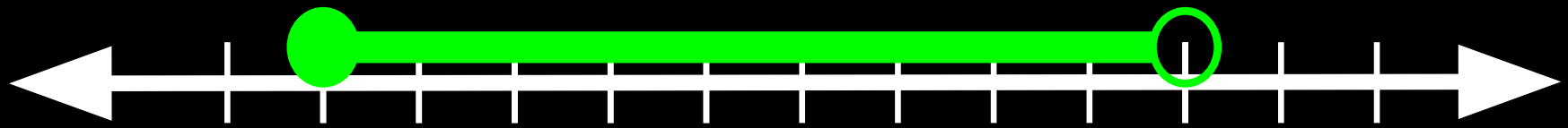
2.5. What will be printed?

```
1 def buy(vm: dict[str,str])->str:
2   for thing in vm:
3     return thing
4   return "Other"
5
6 print(buy(vm=vend))
```

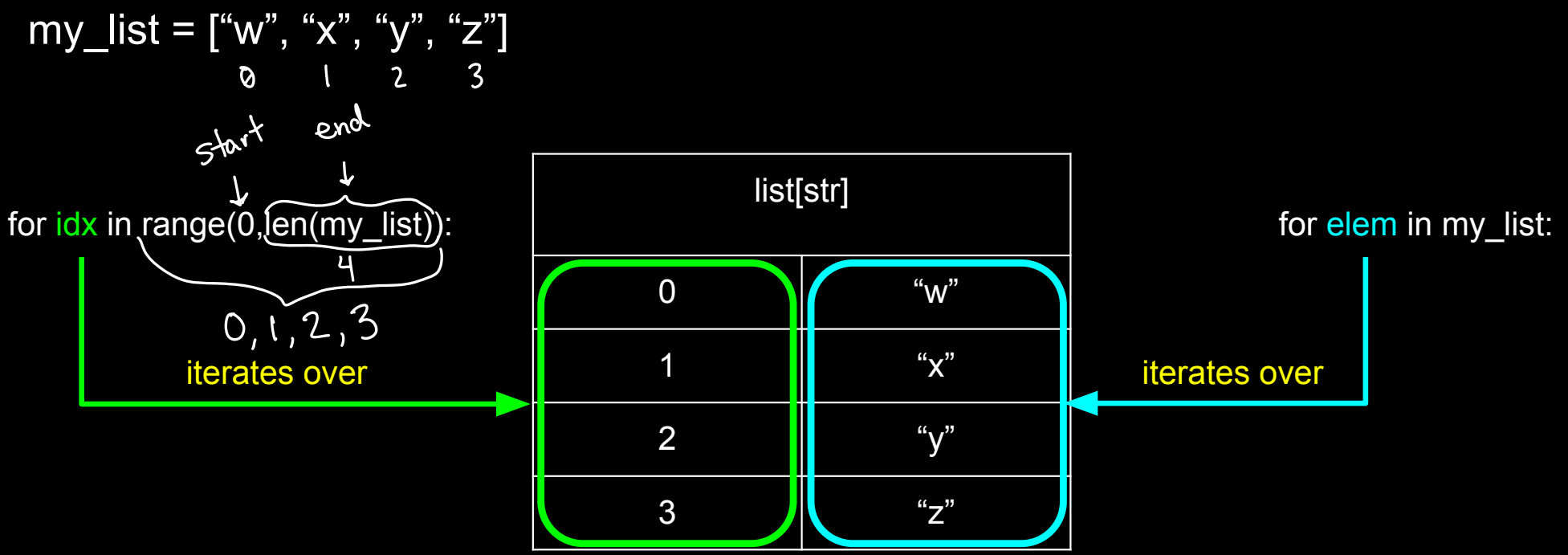
"A1"

A1

Range



- A type of sequence you can loop over.
- Includes start point, does **not** include end point, and *steps* through every point in between
- Constructor: range(start, end, [step = 1])
- Examples:
  - range(1, 5) stops at numbers 1, 2, 3, 4
  - range(1, 6, 2) stops at numbers 1, 3, 5


my_list = ["w", "x", "y", "z"]
0 1 2 3
start
end
for idx in range(0,len(my_list)):
4
0,1,2,3
iterates over
list[str]
0
"w"
1
"x"
2
"y"
3
"z"
for elem in my_list:
iterates over

my_list = [“w”, “x”, “y”, “z”]

for idx in range(0,len(my_list)):

iterates over

list[str]	
indexes	elements
0	“w”
1	“x”
2	“y”
3	“z”

for elem in my_list:

iterates over

```
for idx in range(0,len(my_list)):
    print(idx)
```

Output:

0
1
2
3

```
for idx in range(0,len(my_list)):
    print(my_list[idx])
```

Output:

w
x
y
z

```
for elem in my_list:
    print(elem)
```

Output:

w
x
y
z

Memory Diagram

```python
def group_names(names: list[str]) -> dict[str, int]:
    groups: dict[str, int] = {}
    first_letter: str
    for n in names:
        first_letter = n[0]
        if first_letter in groups:
            groups[first_letter] += 1
        else:
            groups[first_letter] = 1
    return groups

ppl: list[str] = ["Karen", "Emily", "Kris"]
output: dict[str, int] = group_names(names=ppl)
print(output)
output["I"] = 1
print(output)
```

dict at id:2

Stack

Globals

group_names	id:0
ppl	id:1
output	id:2

group_names

RA	13
RV	id:2
first_letter	~~"K"~~ ~~"E"~~ "K"
names	id:1
groups	id:2

Heap

id:0 fn lines 1-10

id:1 list[str]

0	"Karen"
1	"Emily"
2	"Kris"

id:2 dict[str, int]

"K"	~~1~~ 2
"E"	1
"I"	1

Output

{"K": 2, "E": 1}

{"K": 2, "E": 1, "I": 1}

Extra Practice

```python
def artist_counts(playlist: dict[str, str]) -> dict[str, int]:
    artists: dict[str, int] = dict()
    for track in playlist:
        art: str = playlist[track]
        if playlist[track] not in artists:
            artists[art] = 1
        else:
            artists[art] += 1
    return artists


songs: dict[str, str] = {
    "Showgirl": "Taylor",
    "Hello": "Erykah",
    "Fiat": "Butcher",
    "Woo": "Erykah",
}

print(artist_counts(songs))
```