



range() and
Dictionary Practice

Reminders

Want help with a concept we've covered since the last quiz? Go to Tutoring!

- **Tonight:** 5-7pm in Sitterson Hall (SN) 008
- **Tomorrow:** 5-7pm in SN 011

Submit a ticket to be seen via the CSXL website! (see details on the website's Support page)

Match the Data Structure to its Application

Set

List

Dictionary

Store a bunch of
tasks in a specific
order

Keeping track of
inventory in a store
(names of items and
the number in stock)

Store the jersey
numbers of UNC's
basketball team

Match the Data Structure to its Application

List

Store a bunch of tasks in a specific order



Dictionary

Keeping track of inventory in a store (names of items and the number in stock)



Set

Store the jersey numbers of UNC's basketball team



```
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)
```

2.2. What will be printed?

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

2.3. What will be printed?

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

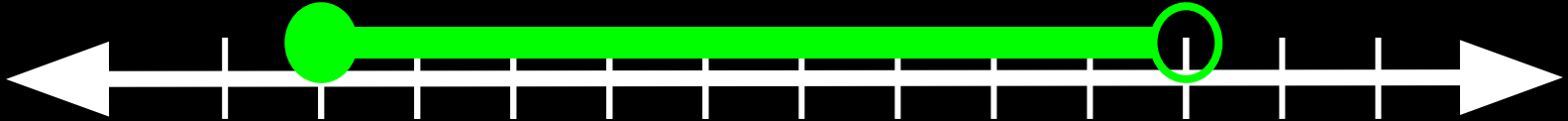
2.4. What will be printed?

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

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"]
```

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

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:

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

Output:

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

Output:

```
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

```
1 def group_names(names: list[str]) -> dict[str, int]:
2     groups: dict[str, int] = {}
3     first_letter: str
4     for n in names:
5         first_letter = n[0]
6         if first_letter in groups:
7             groups[first_letter] += 1
8         else:
9             groups[first_letter] = 1
10    return groups
11
12 ppl: list[str] = ["Karen", "Emily", "Kris"]
13 result: dict[str, int] = group_names(names=ppl)
14 print(result)
15 result["I"] = 1
16 result.pop("E")
17 print(result)
```

Extra Practice

```
1 def artist_counts(playlist: dict[str, str]) -> dict[str, int]:
2     artists: dict[str, int] = dict()
3     for track in playlist:
4         art: str = playlist[track]
5         if playlist[track] not in artists:
6             artists[art] = 1
7         else:
8             artists[art] += 1
9     return artists
10
11
12 songs: dict[str, str] = {
13     "Showgirl": "Taylor",
14     "Hello": "Erykah",
15     "Fiat": "Butcher",
16     "Woo": "Erykah",
17 }
18
19 print(artist_counts(songs))
```