# Data Structures - Things you can store data in

Diff data structures have different ways of
- adding data
- getting data
- initializing

```
 0  1  2
[1, 2, 3]
a[0:2]
```

| | list | dictionary |
|---|---|---|
| initialize | `[1, "a", 2]` `[]` | `fruits = {"apple": "red", "plum": "purple", "banana": "yellow"}` |
| get | `a = [1, 2, 3]` `a[0]` `a[1]` | `fruits["apple"]` |
| add | `[1, 2] + [3] → [1, 2, 3]` | `fruits["oranges"] = "orange"` `fruits["oranges"] = "green"` `fruits["apple"] = "orange"` |

`a = [1, 2]` `[a[0]] + [3] + [a[1]]`

① Make a dictionary that contains three key value pairs with the keys being the name of someone in the class (as a string) and the value being a string of something you like about them. (one-word)

② Make a function that takes in a positive integer and returns a list with elements from zero to that integer.

```
def gen_list(n):
    '''
    >>> gen_list(5)
    [0, 1, 2, 3, 4, 5]
    '''
    answer = []
```

Splicing ex:

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| E | D | L | Y | F | T |

```
e = ["E", "D", "L", "Y", "F", "T"]
e[0:2] →
e[1:4]
```

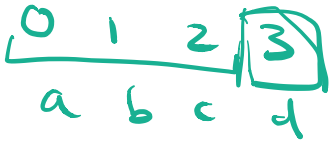
```
for i in range(n+1):
    answer += [i]
return answer
```

④ Write a function which returns a list of the first three elements of list "a" assuming list "a" contains three elements or more:

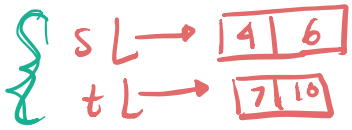
```
def get_first_three(a):
    return a[0:3]
```

a[:3]

★ Hint: use splicing

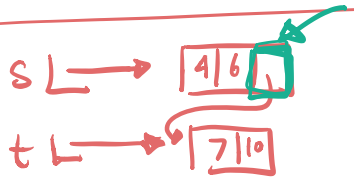



s = [4, 6]
t = [7, 10]




---

s.append(t)




s.append(5)




---

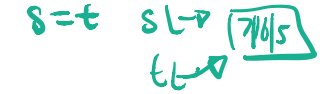
s.extend(t)


---

Append & Extend return None

```
>>> print([1].extend([2]))
None
```

s = t



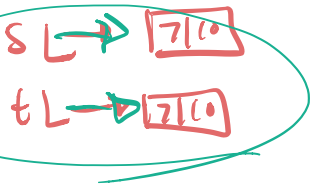

t.append(5)

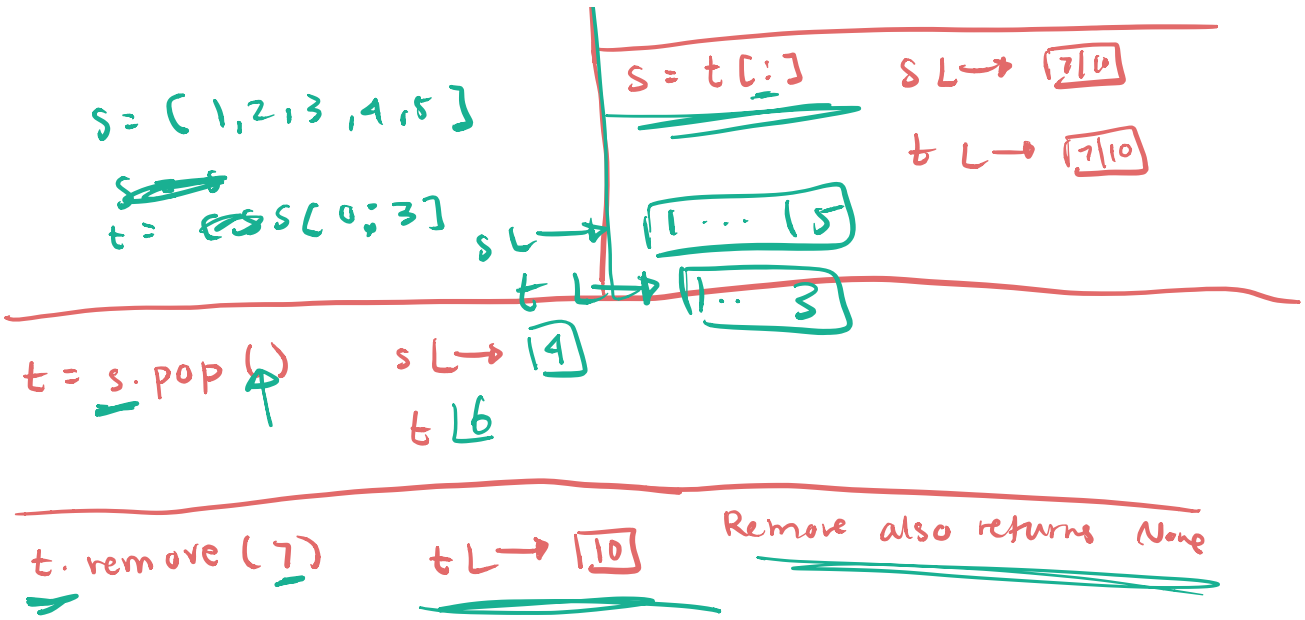
To Make a Copy

s = t


s = list(t)

r = list(s)




s = [1,2,3,4,5]
t = s[0:3]
s = t[:]
s → 7|10
t → 7|10
s →
t →
1 ... 5
1 .. 3
t = s.pop()
s → 4
t 6
t.remove(7)
t → 10
Remove also returns None