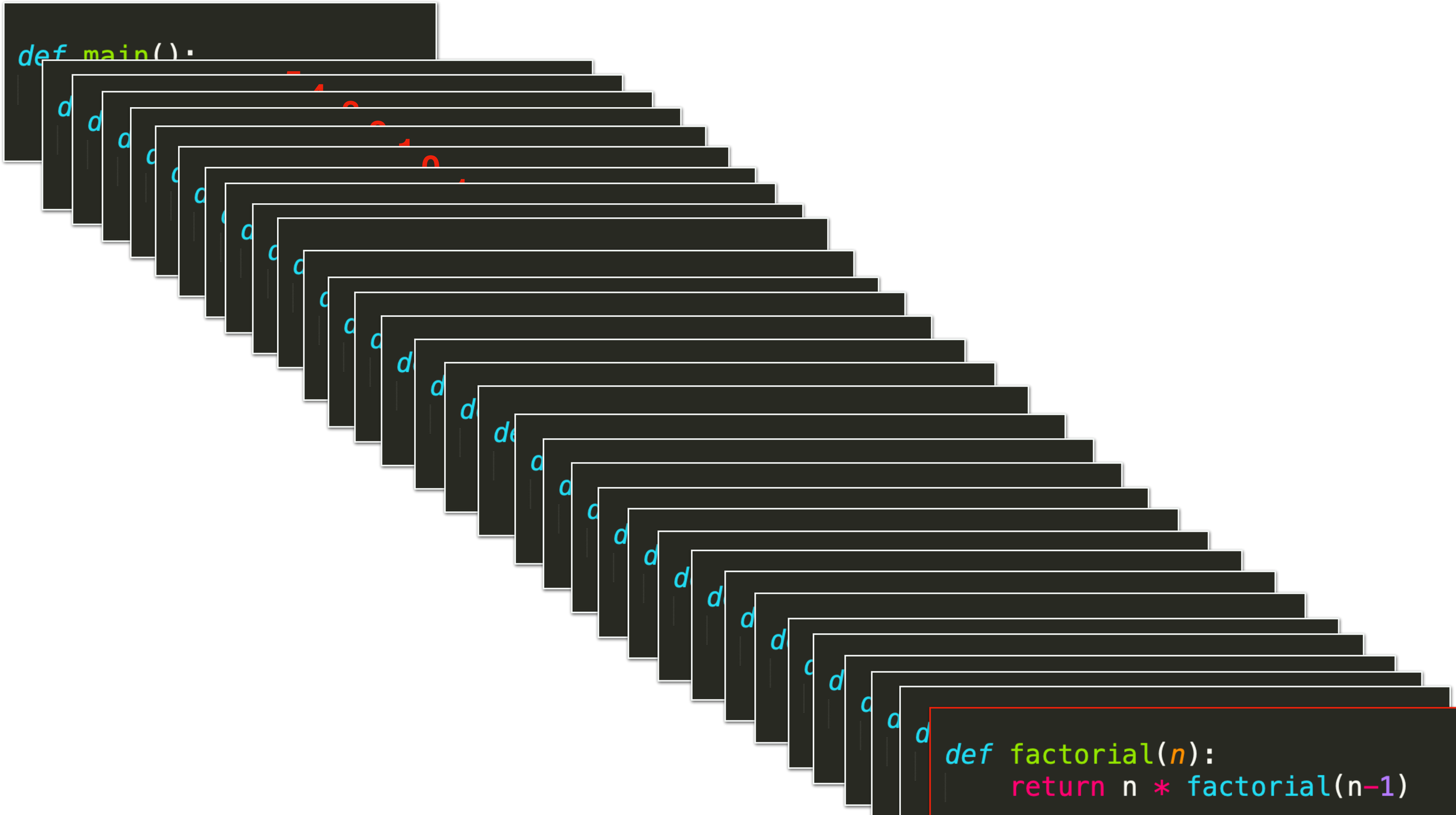
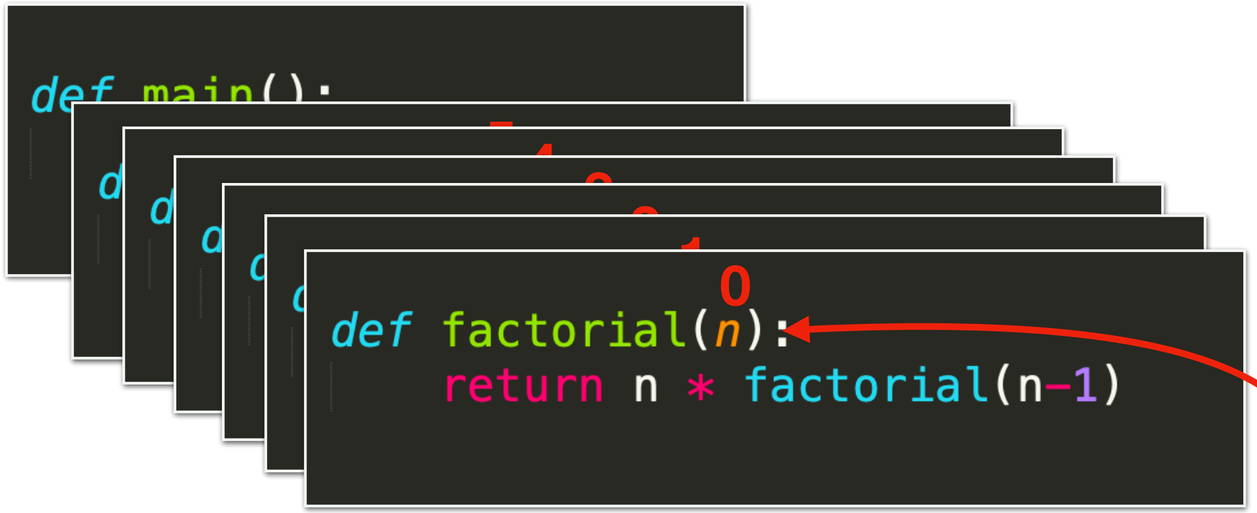
SC101

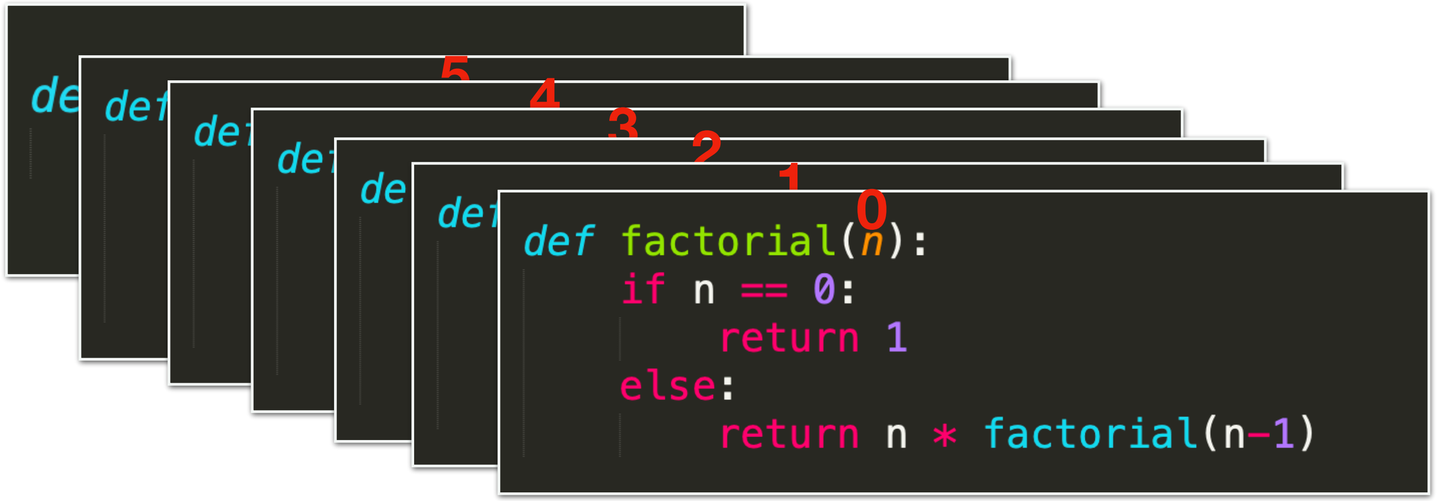
Week5

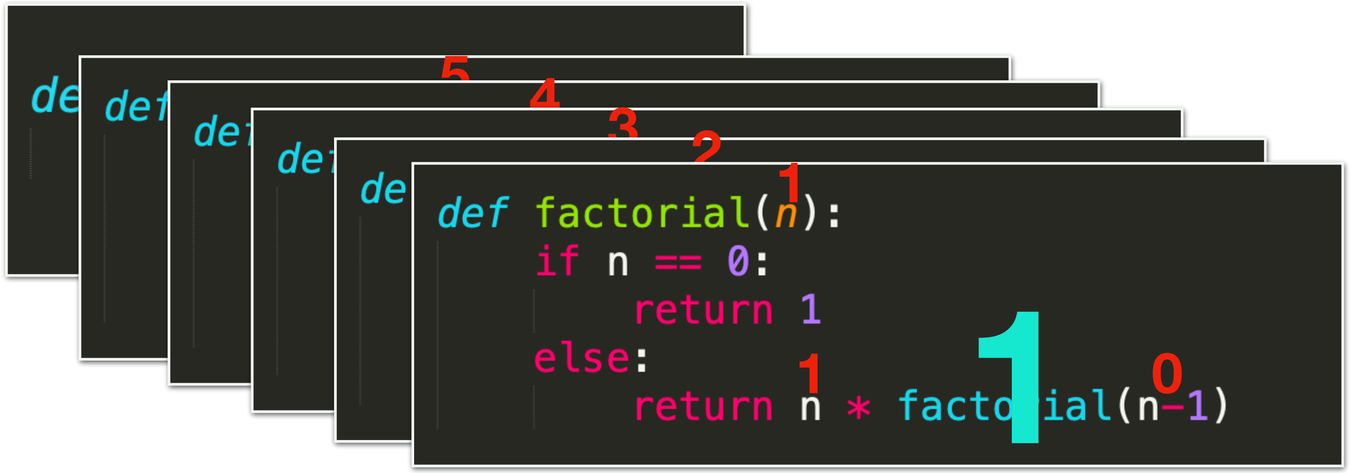

def factorial(n):
return n * factorial(n-1)

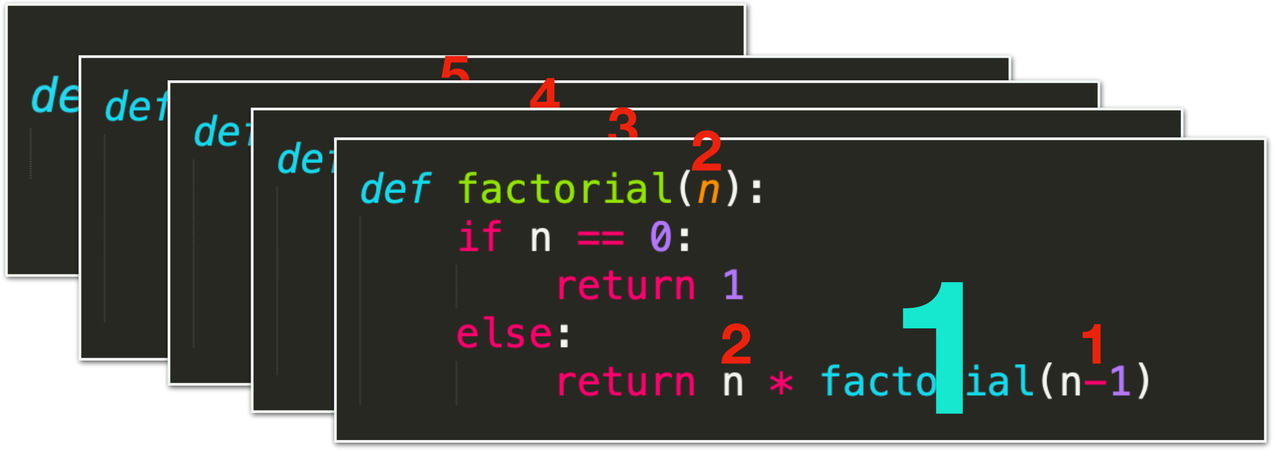
Stack Overflow

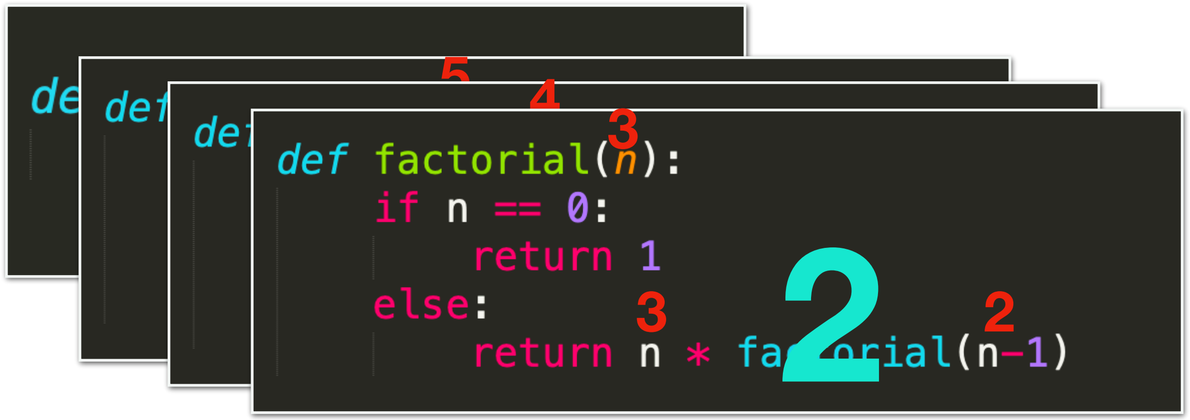


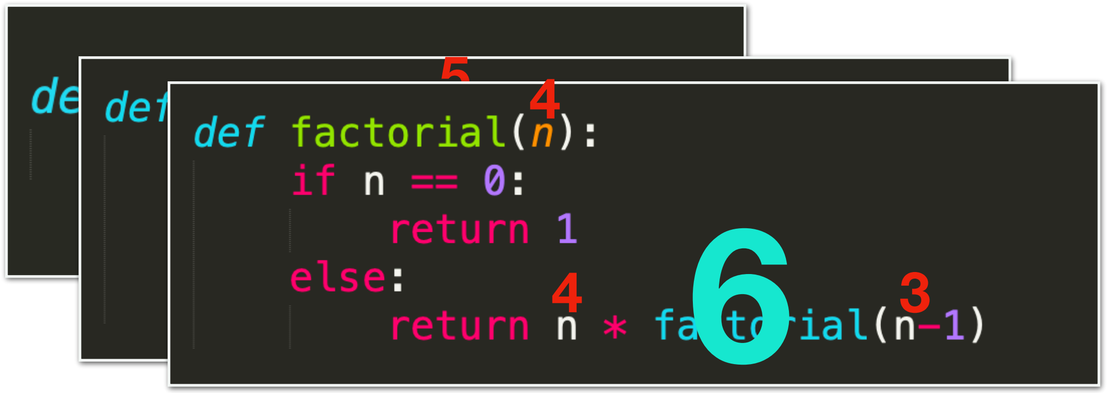
Base Case

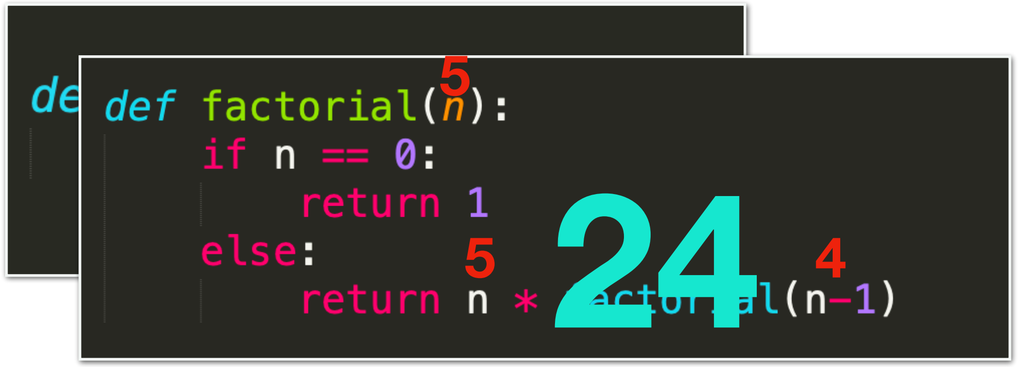

5
4
3
2
1
0
def factorial(n):
if n == 0:
return 1
else:
return n * factorial(n-1)


5
4
3
2
1
def factorial(n):
if n == 0:
return 1
else:
return n * factorial(n-1)
1
1
0


5
4
3
2
def factorial(n):
if n == 0:
return 1
else:
return n * factorial(n-1)
2
1
1


5
4
3
def factorial(n):
if n == 0:
return 1
else:
return n * factorial(n-1)
3
2
2


5
4
def factorial(n):
if n == 0:
return 1
else:
4
6
3
return n * factorial(n-1)

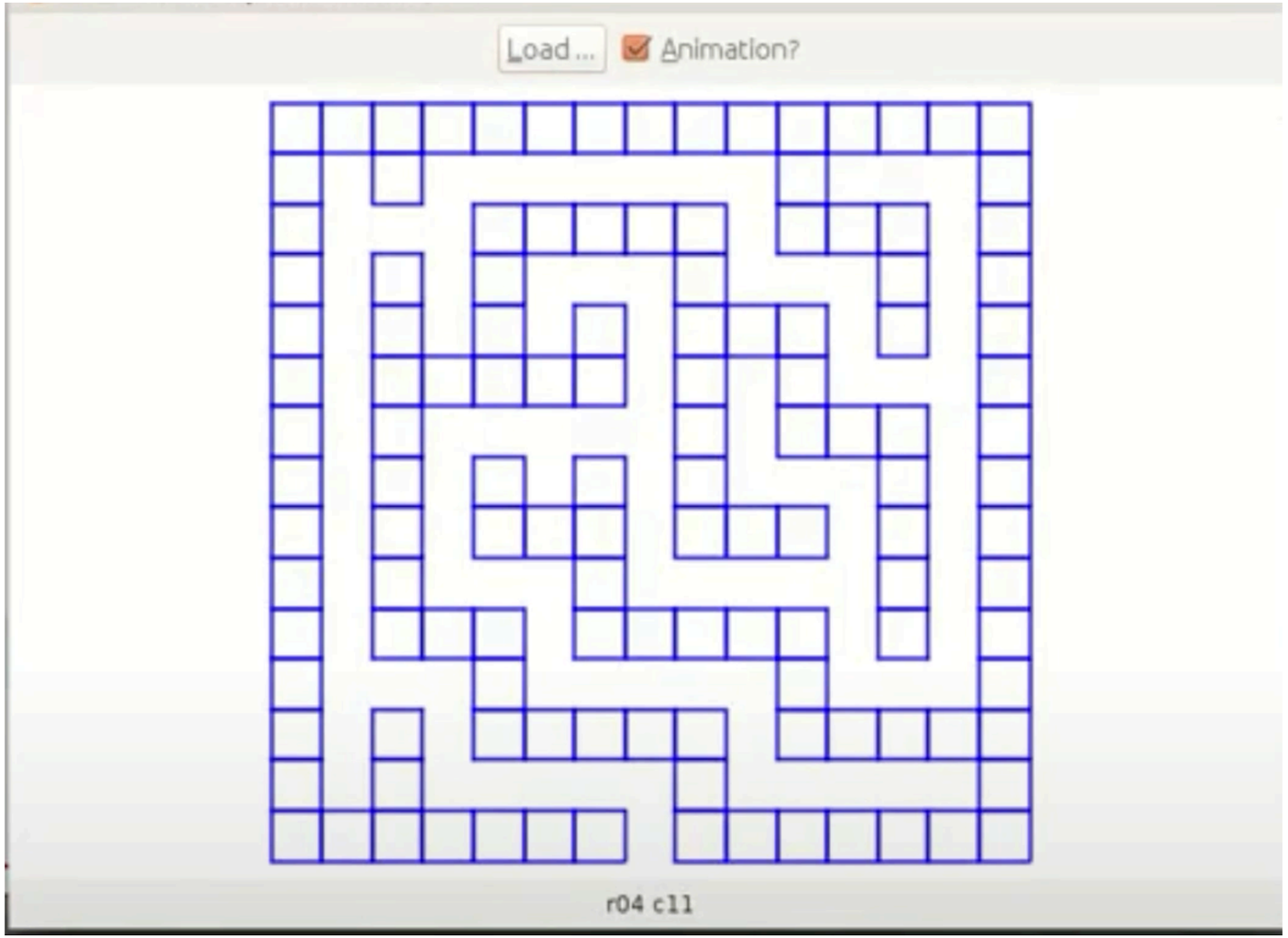

5
def factorial(n):
if n == 0:
return 1
else:
5
24
4
return n * factorial(n-1)

```
def main():
    print(factorial(5))
```

120

Self Similarity

Backtracking


Load ...
Animation?
r04 c11

```
def main():
    permutation([1, 2, 3])
```

```
def main():
    permutation([1, 2, 3])
```

```
def main():
    def permutation(lst):
        permutation_helper(lst, [], len(lst))
```

```
def main():
    def permutation(lst):
        permutation_helper(lst, [], len(lst))
```

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [ ]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [ ]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [ ]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 1

lst = [1, 2, 3]

current_lst = [ ]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 1

lst = [1, 2, 3]

current_lst = [ ]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 1

lst = [1, 2, 3]

current_lst = [ ]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 1

lst = [1, 2, 3]

current_lst = [ ]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 1

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 1

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 1

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 1

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 1

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [1, 2]

ans_len = 3

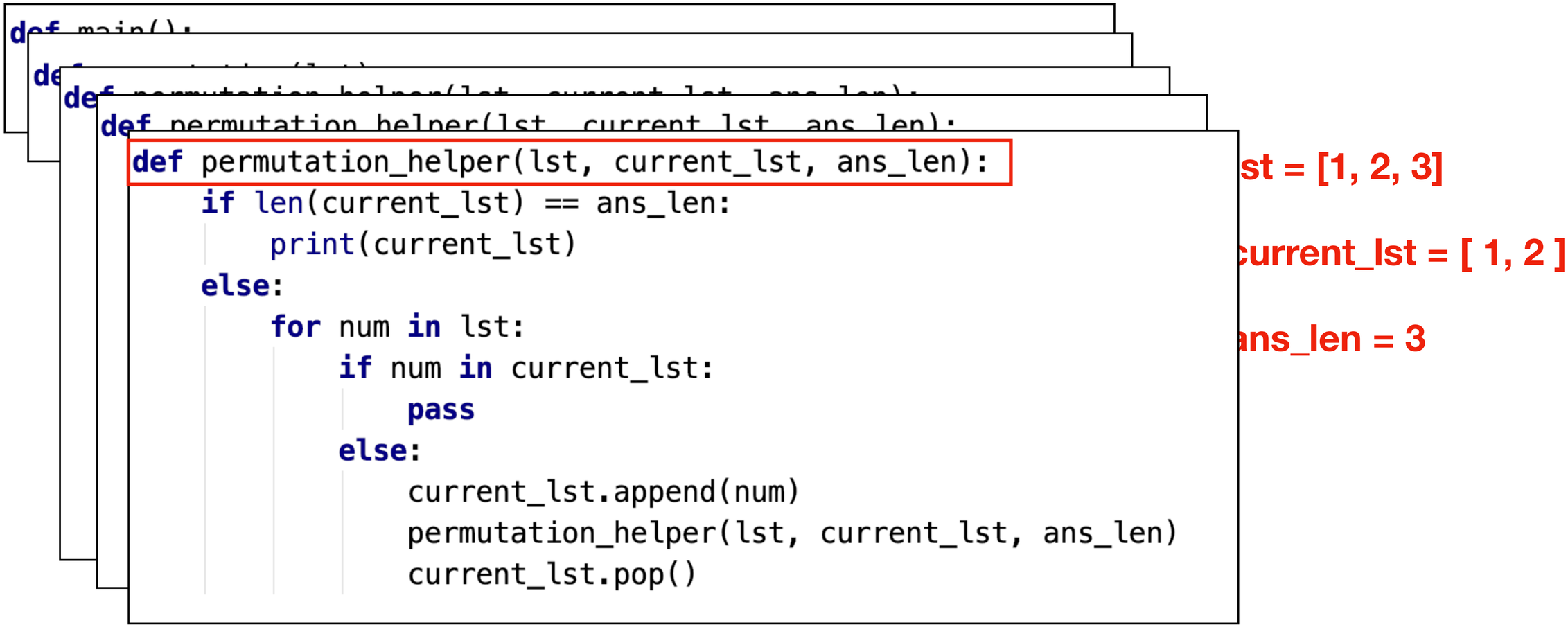
```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [1, 2]

ans_len = 3


def permutation_helper(lst, current_lst, ans_len):
if len(current_lst) == ans_len:
print(current_lst)
else:
for num in lst:
if num in current_lst:
pass
else:
current_lst.append(num)
permutation_helper(lst, current_lst, ans_len)
current_lst.pop()
st = [1, 2, 3]
current_lst = [1, 2]
ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

st = [1, 2, 3]

current_lst = [1, 2]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

st = [1, 2, 3]

current_lst = [1, 2]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 1

st = [1, 2, 3]

current_lst = [1, 2]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 1

st = [1, 2, 3]

current_lst = [1, 2]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

st = [1, 2, 3]

current_lst = [1, 2]

num = 1

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

st = [1, 2, 3]

current_lst = [1, 2]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

st = [1, 2, 3]

current_lst = [1, 2]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

st = [1, 2, 3]

current_lst = [1, 2]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

st = [1, 2, 3]

current_lst = [1, 2]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

st = [1, 2, 3]

current_lst = [1, 2]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

st = [1, 2, 3]

current_lst = [1, 2]

ans_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

st = [1, 2, 3]

current_lst = [1, 2, 3]

ans_len = 3

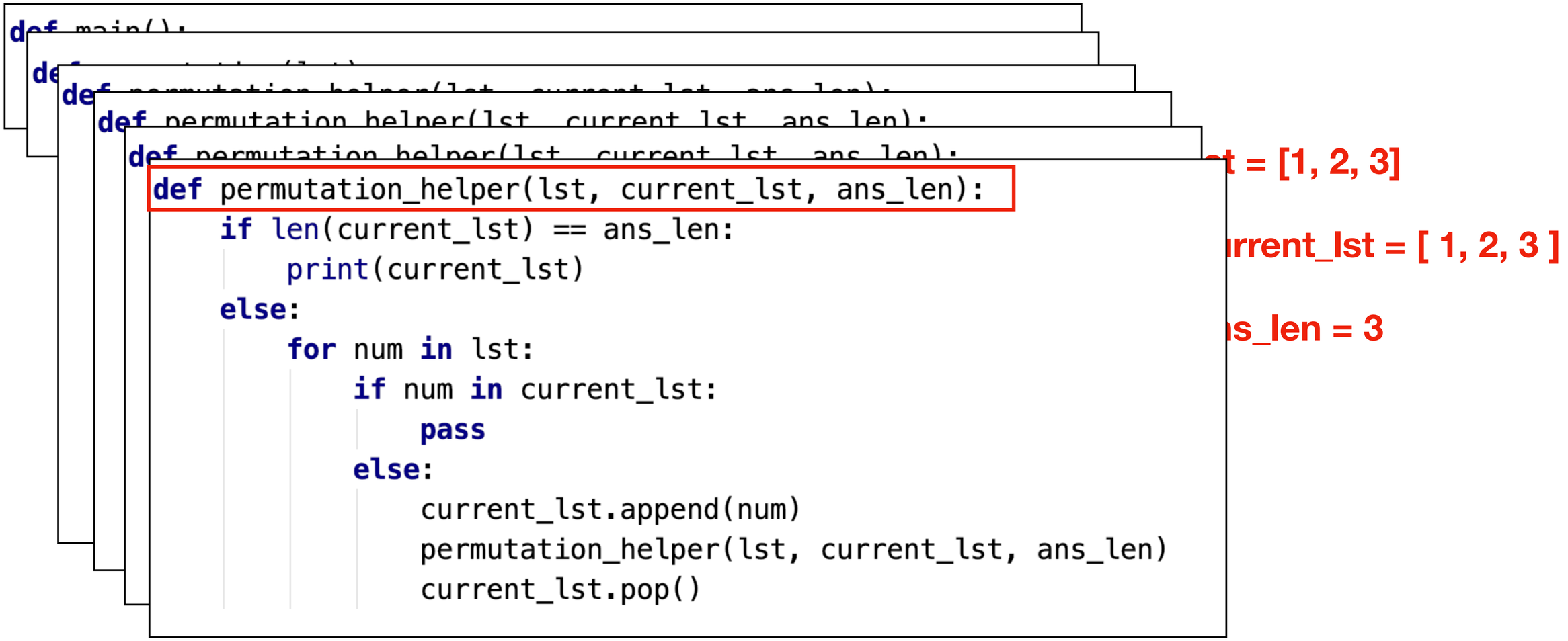
```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

st = [1, 2, 3]

current_lst = [1, 2, 3]

ans_len = 3




```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

t = [1, 2, 3]

rrent_lst = [1, 2, 3]

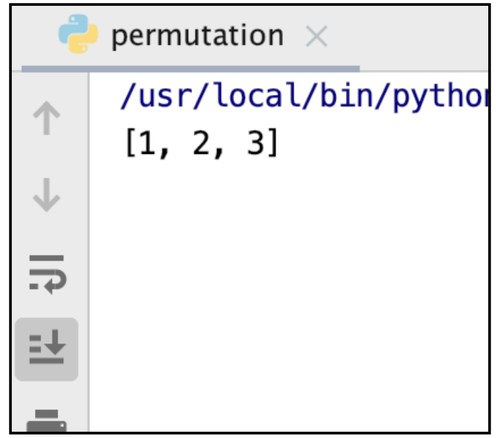
s_len = 3

```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

t = [1, 2, 3]

rrent_lst = [1, 2, 3]

s_len = 3



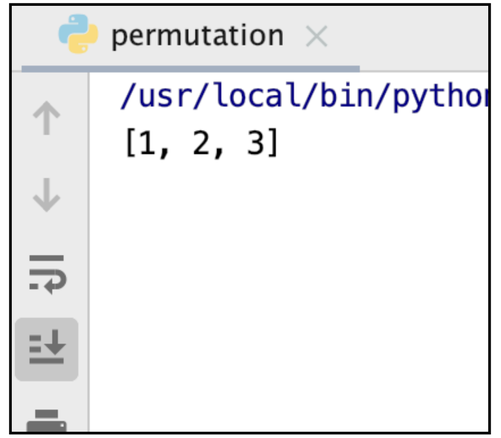
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [1, 2, 3]

num = 3

ans_len = 3



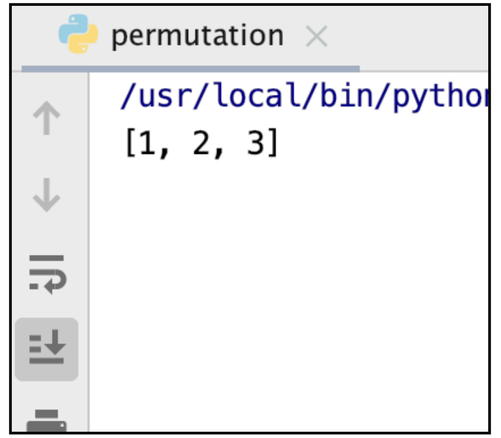
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [1, 2, 3]

num = 3

ans_len = 3



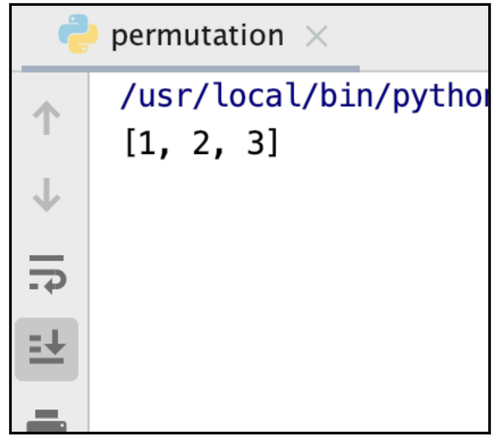
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [1, 2]

num = 3

ans_len = 3



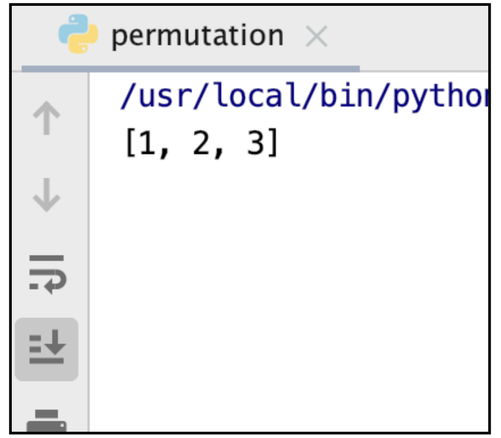
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [1, 2]

ans_len = 3



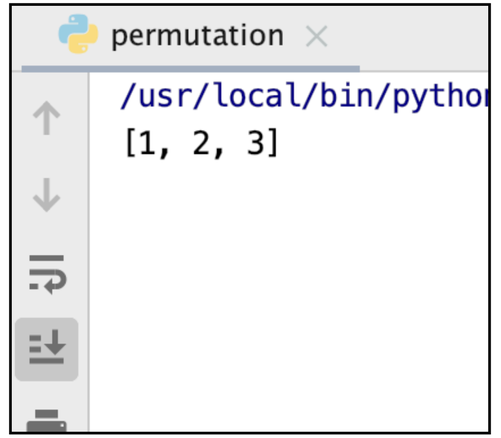
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [1, 2]

ans_len = 3



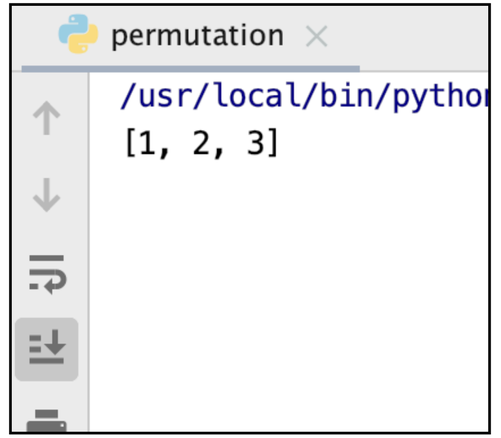
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3



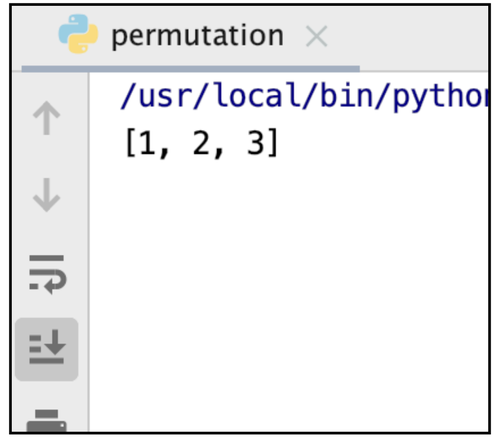
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3



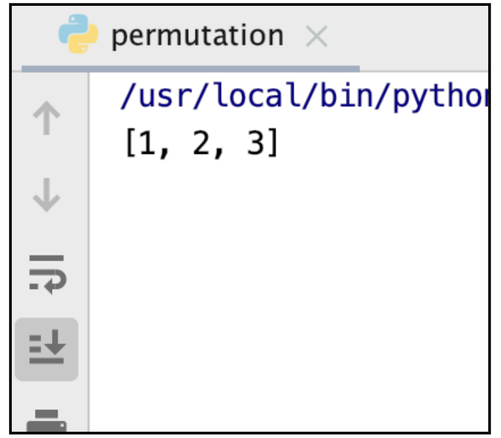
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3



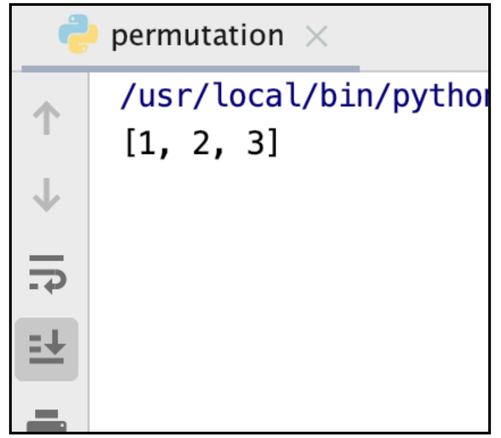
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [1, 3]

ans_len = 3



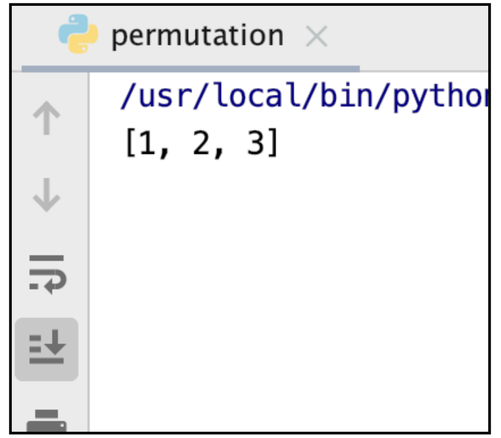
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [1, 3]

ans_len = 3

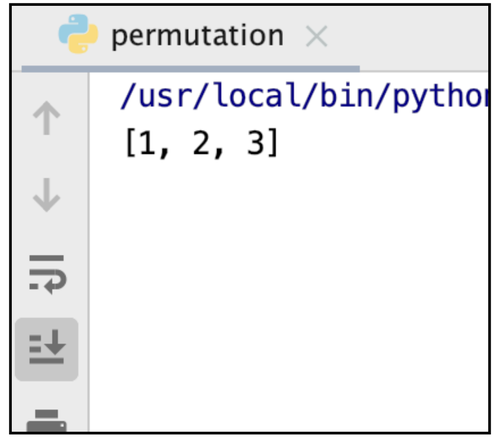


```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [1, 3]

ans_len = 3

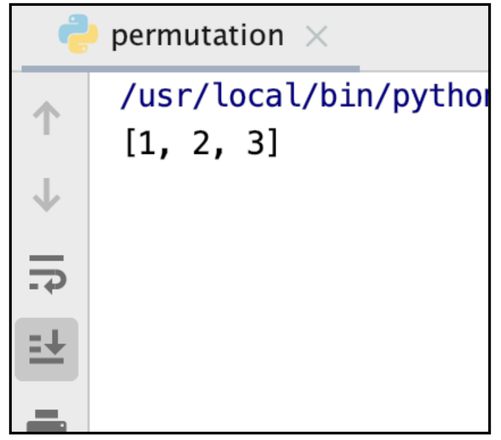


```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [1, 3]

ans_len = 3

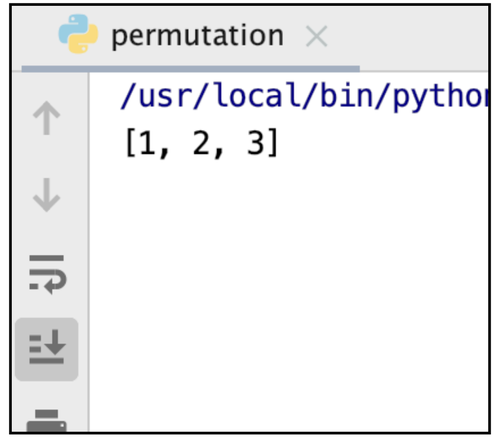


```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [1, 3]

ans_len = 3



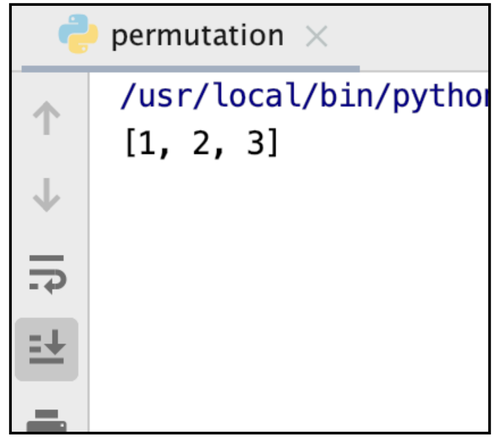
```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 1

lst = [1, 2, 3]

current_lst = [1, 3]

ans_len = 3



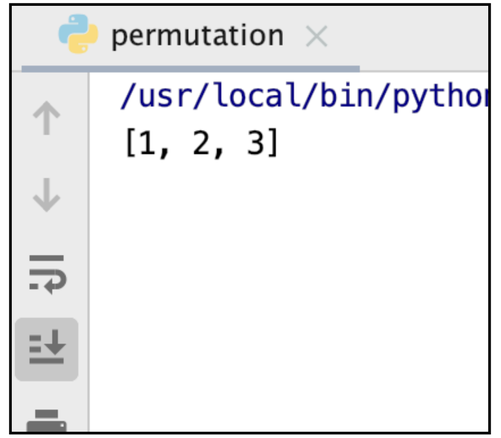
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [1, 3]

num = 1

ans_len = 3



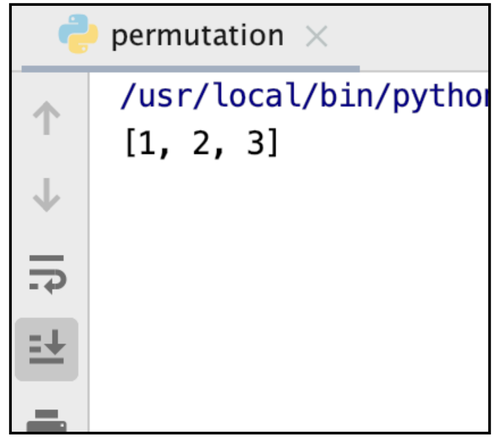
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 1

lst = [1, 2, 3]

current_lst = [1, 3]

ans_len = 3



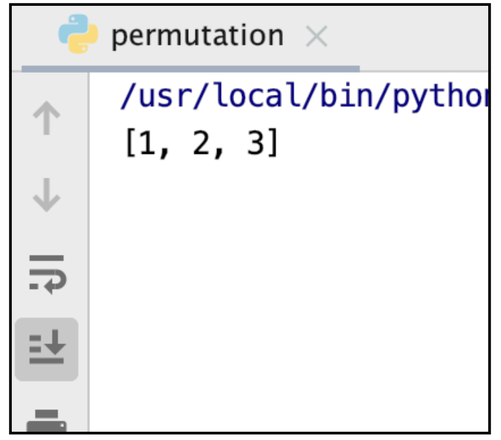
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [1, 3]

ans_len = 3



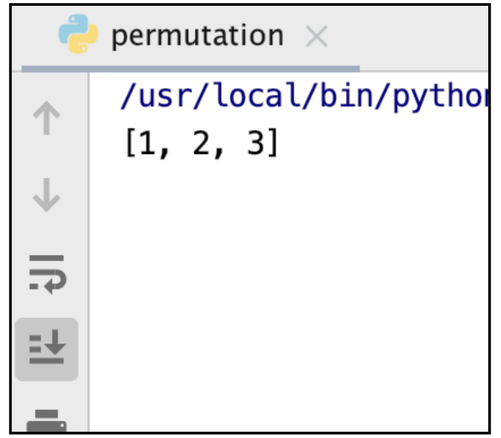
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [1, 3]

ans_len = 3



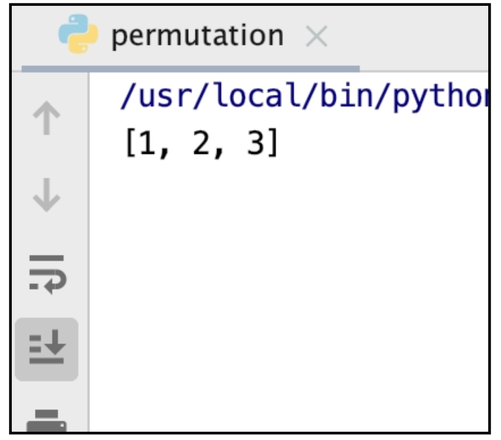
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [1, 3, 2]

ans_len = 3



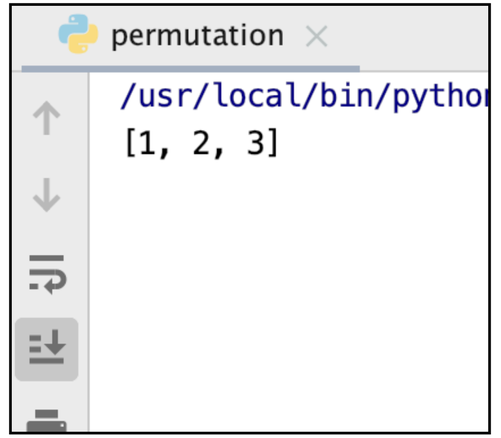
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [1, 3, 2]

ans_len = 3

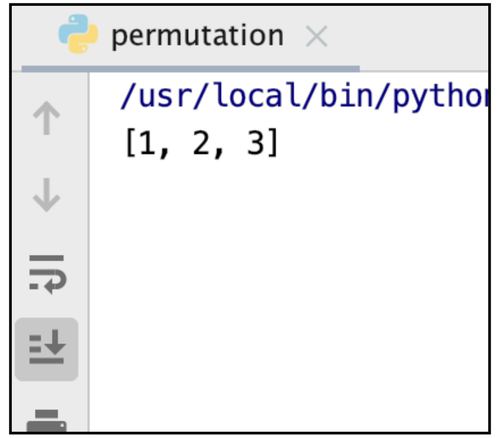


```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [1, 3, 2]

ans_len = 3

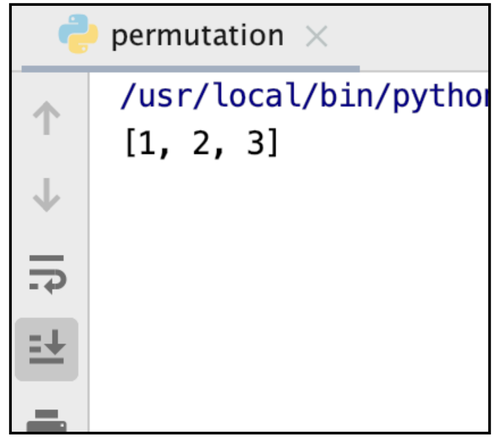


```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [1, 3, 2]

ans_len = 3

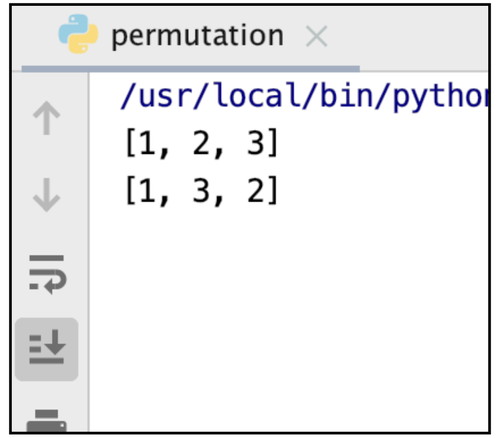


```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

lst = [1, 2, 3]

current_lst = [1, 3, 2]

ans_len = 3



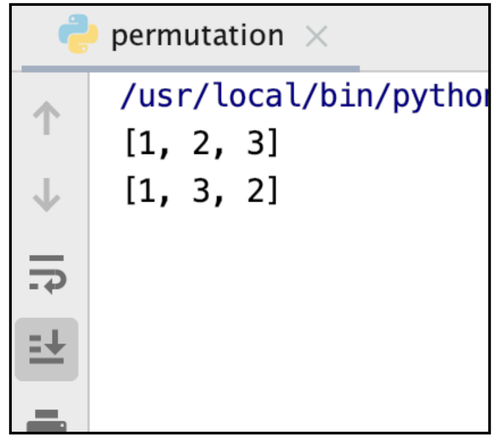
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [1, 3, 2]

ans_len = 3



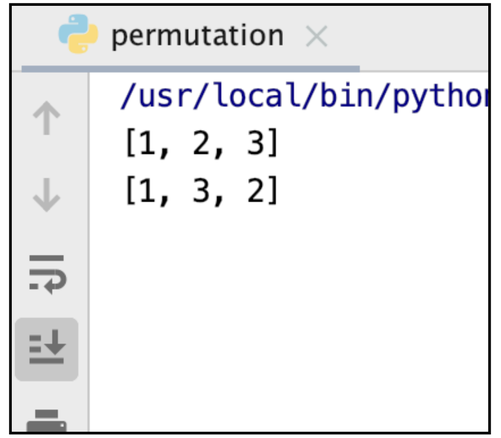
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [1, 3]

ans_len = 3



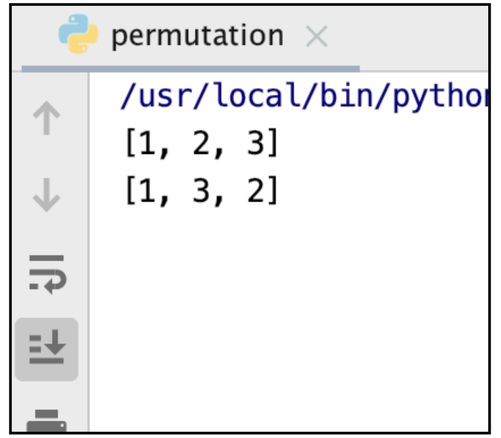
```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [1, 3]

ans_len = 3



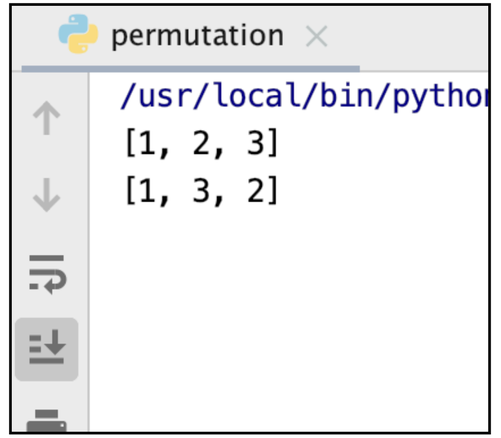
```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [1, 3]

ans_len = 3



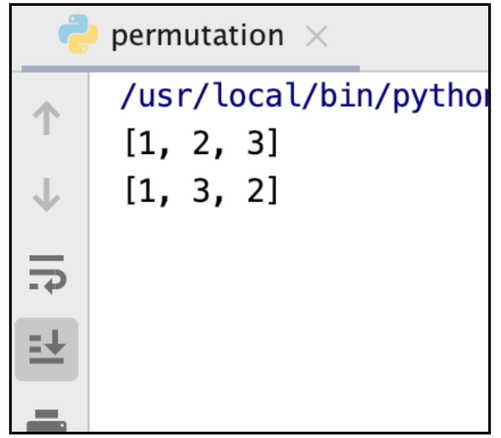
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [1, 3]

ans_len = 3



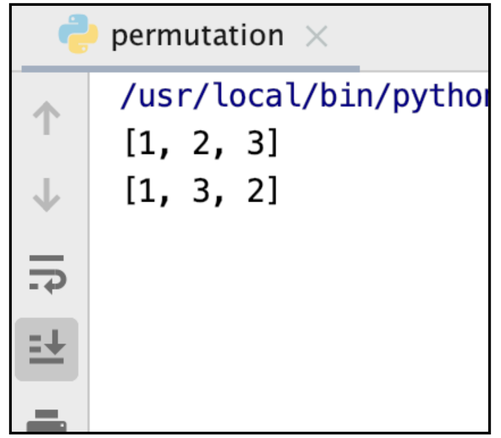
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [1, 3]

ans_len = 3



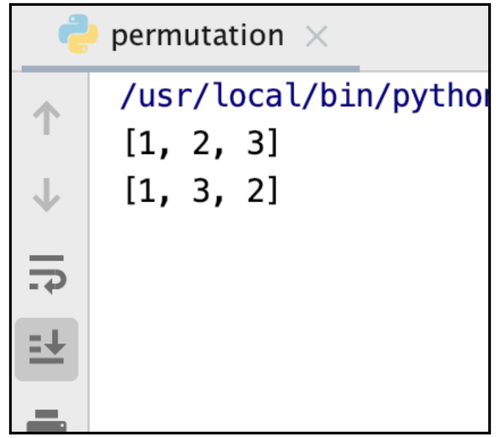
```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3



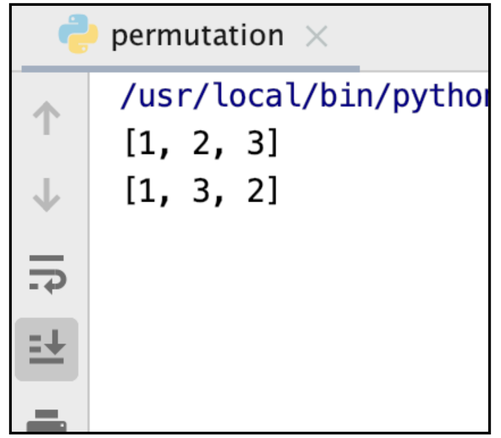
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 1

lst = [1, 2, 3]

current_lst = [1]

ans_len = 3



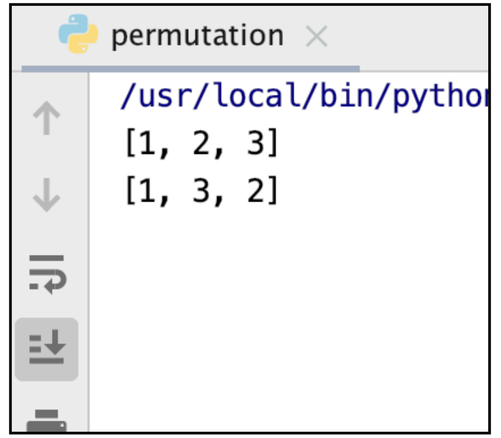
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 1

lst = [1, 2, 3]

current_lst = [ ]

ans_len = 3



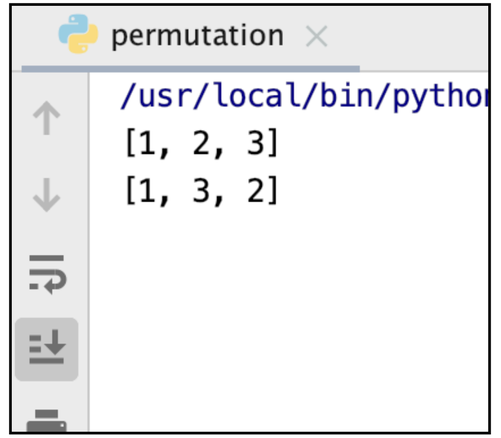
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [ ]

ans_len = 3



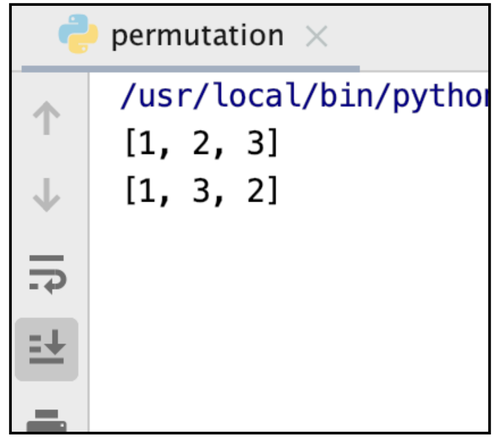
```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [ ]

ans_len = 3



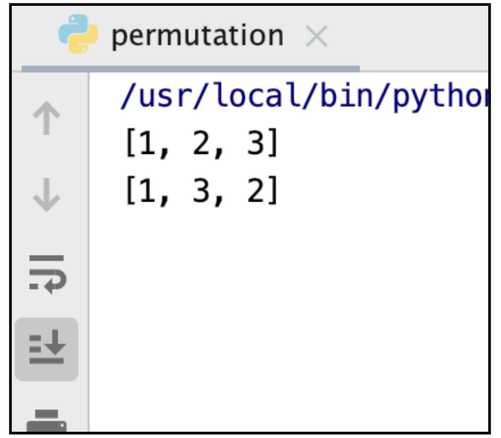
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [ ]

ans_len = 3



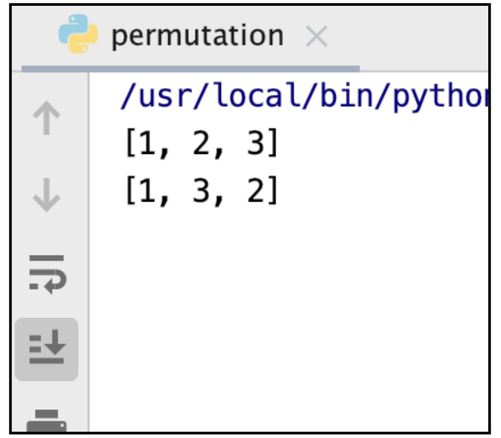
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [2]

ans_len = 3



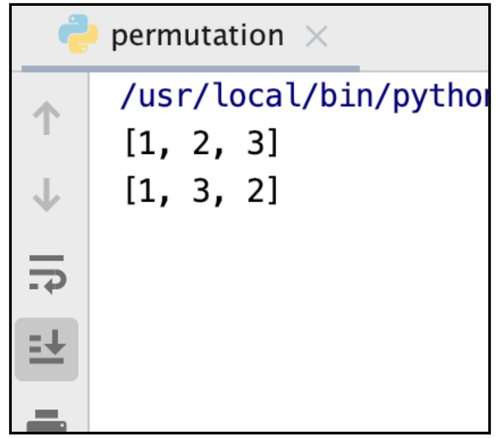
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [2]

ans_len = 3



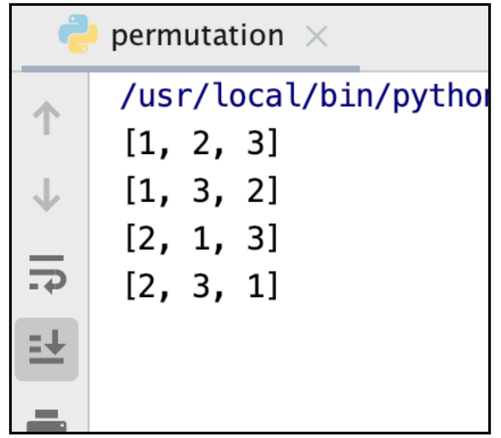
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [2]

ans_len = 3



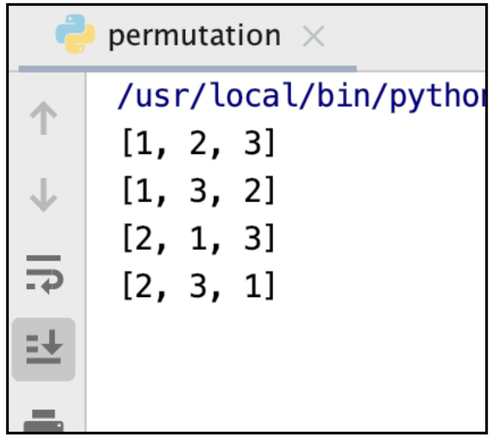
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 2

lst = [1, 2, 3]

current_lst = [ ]

ans_len = 3



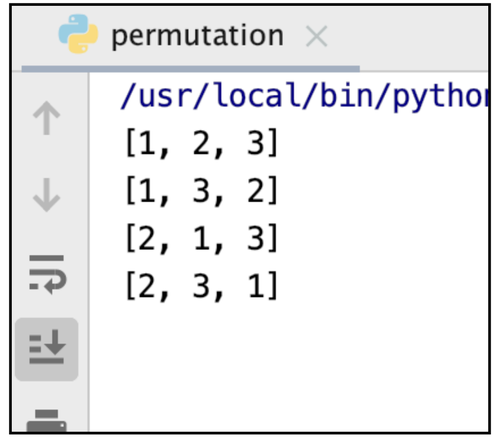
```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [ ]

ans_len = 3



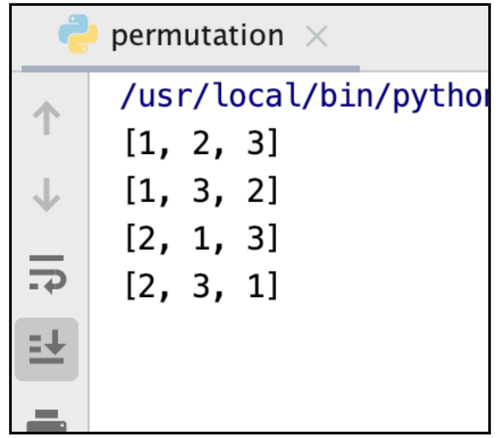
```
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [ ]

ans_len = 3



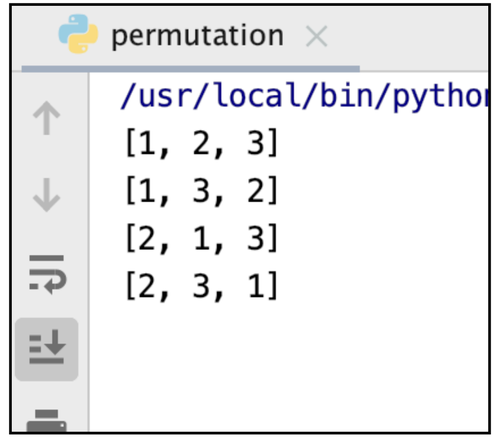
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [ ]

ans_len = 3



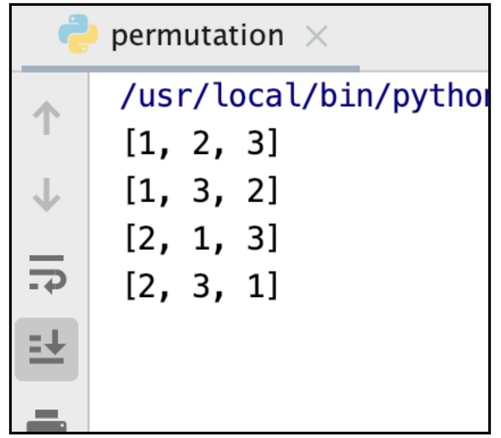
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [3]

ans_len = 3



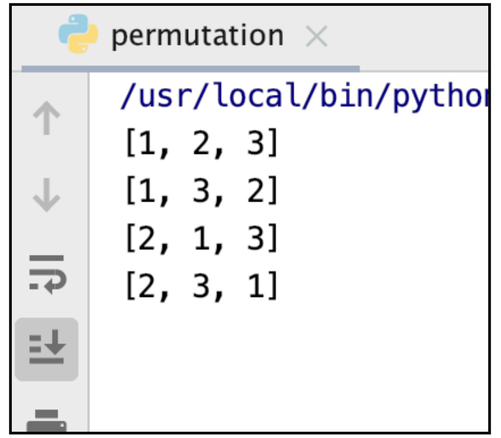
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [3]

ans_len = 3



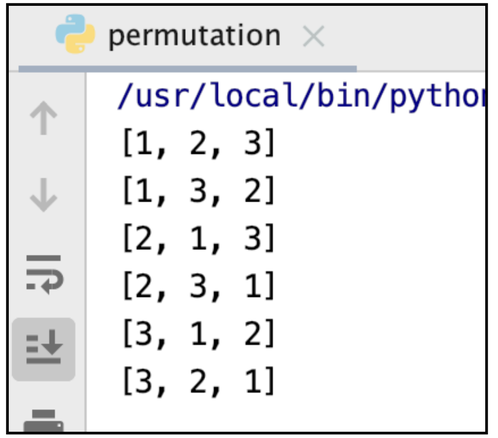
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [3]

ans_len = 3



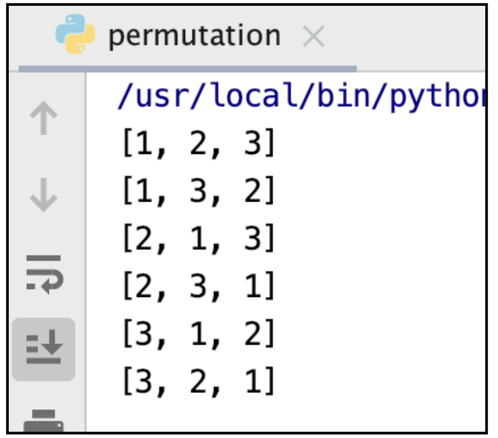
```python
def permutation_helper(lst, current_lst, ans_len):
    if len(current_lst) == ans_len:
        print(current_lst)
    else:
        for num in lst:
            if num in current_lst:
                pass
            else:
                current_lst.append(num)
                permutation_helper(lst, current_lst, ans_len)
                current_lst.pop()
```

num = 3

lst = [1, 2, 3]

current_lst = [ ]

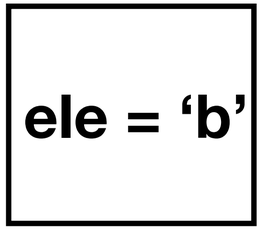
ans_len = 3



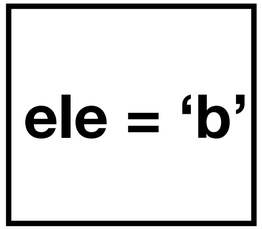
```python

def main():
    subsets(['a', 'b', 'c', 'd'])


def subsets(lst):
    subsets_helper(lst, [])


def subsets_helper(lst, chosen):
    if len(lst) == 0:
        print(chosen)
    else:
        # Choose
        element = lst.pop()

        # Explore without
        subsets_helper(lst, chosen)

        # Explore with
        chosen.append(element)
        subsets_helper(lst, chosen)

        # Unchoose
        chosen.pop()
        lst.append(element)

```

lst = [‘a’, ‘b’] / chosen = [ ]

lst = [‘a’, ‘b’] / chosen = [ ]

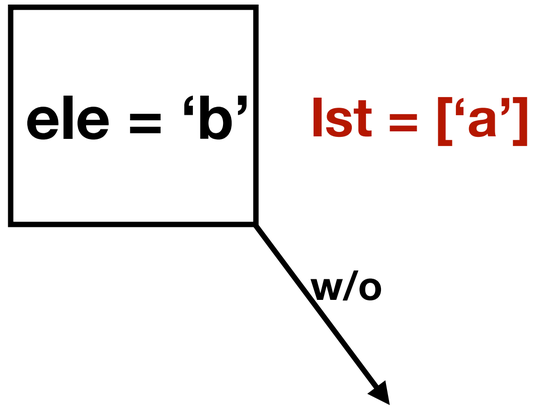


lst = [‘a’, ‘b’] / chosen = [ ]

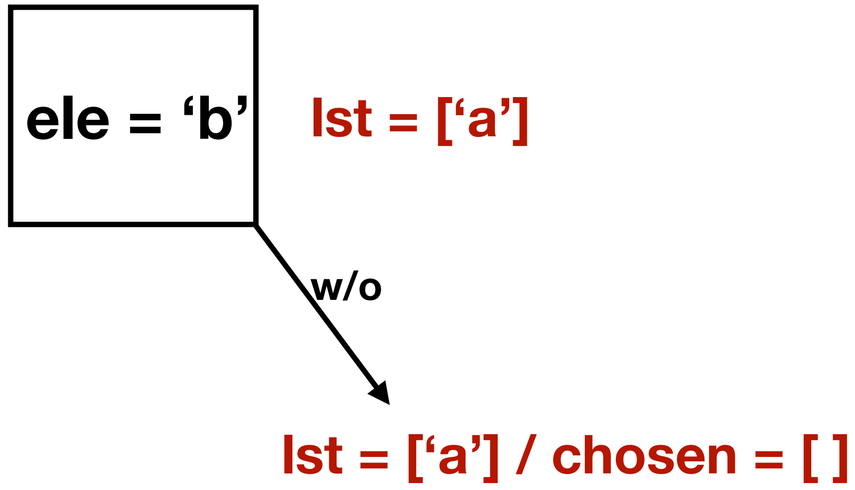


lst = [‘a’]

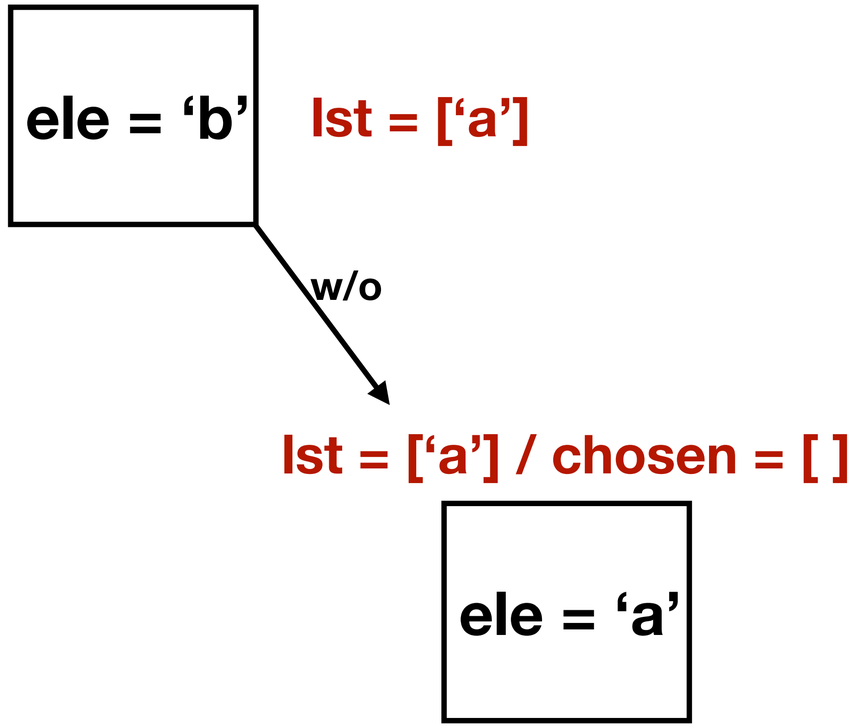
lst = [‘a’, ‘b’] / chosen = [ ]



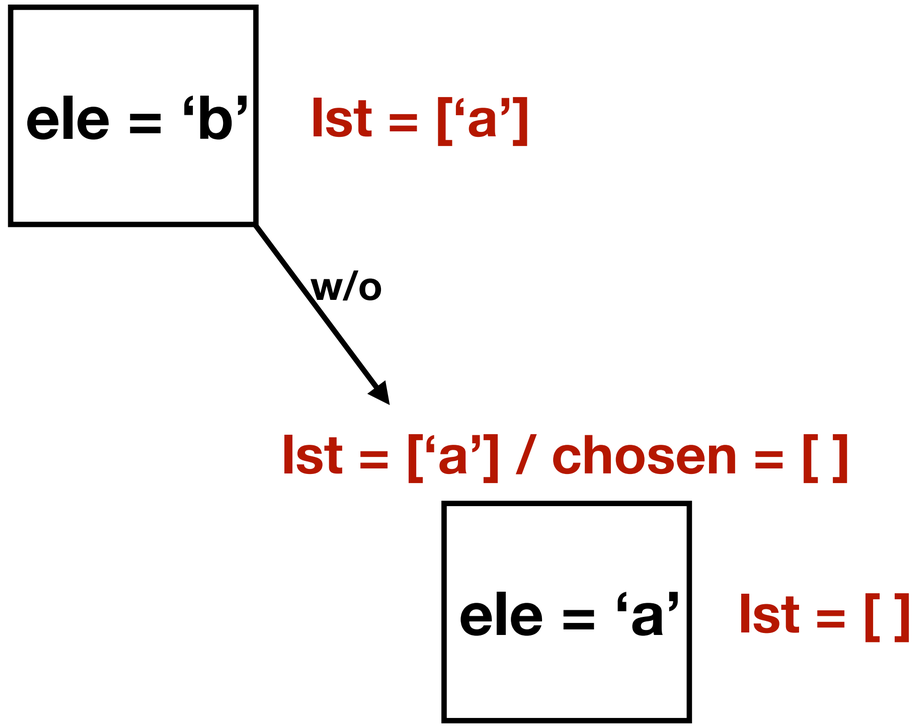
lst = [‘a’, ‘b’] / chosen = [ ]



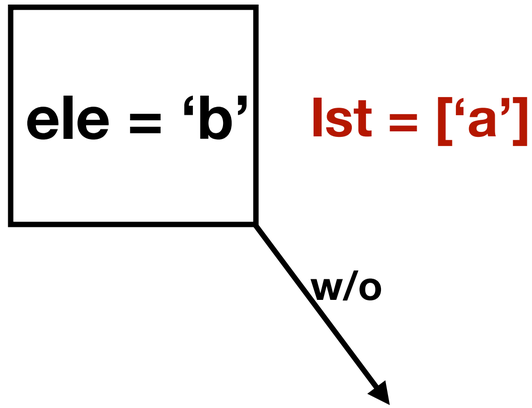
lst = [‘a’, ‘b’] / chosen = [ ]



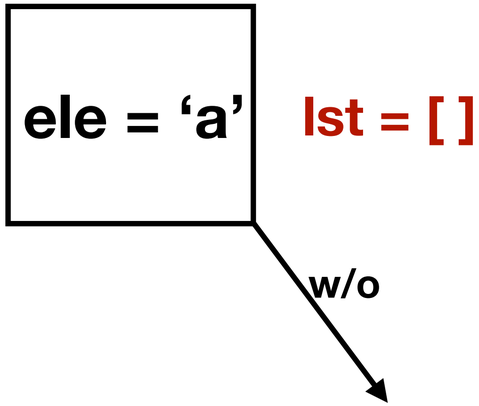
lst = [‘a’, ‘b’] / chosen = [ ]



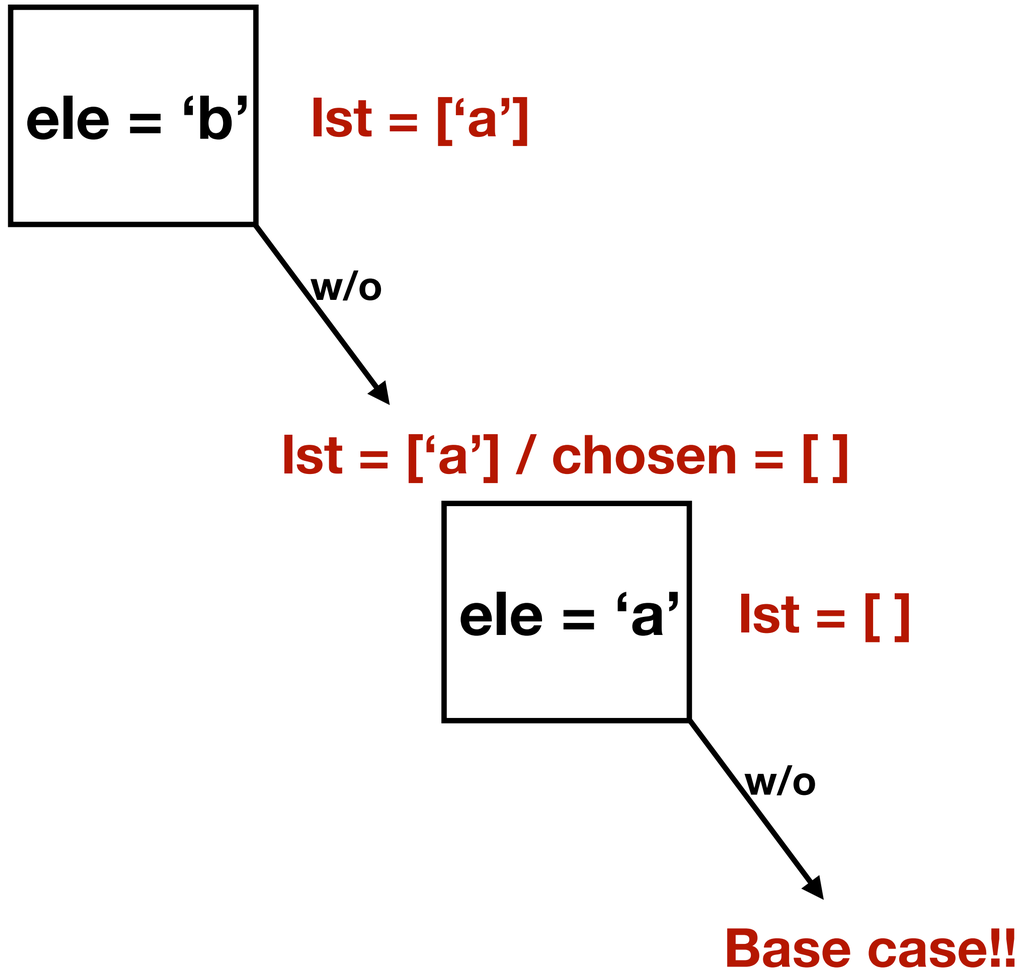
lst = [‘a’, ‘b’] / chosen = [ ]



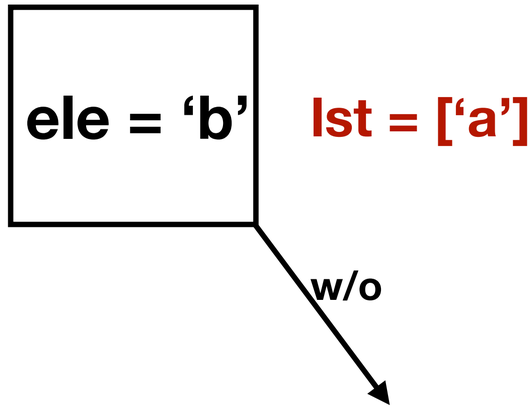
lst = [‘a’] / chosen = [ ]



lst = [‘a’, ‘b’] / chosen = [ ]

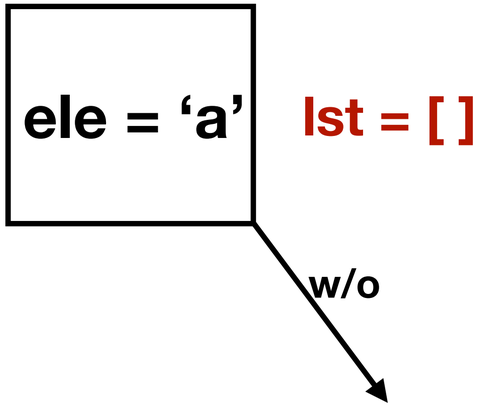


lst = [‘a’, ‘b’] / chosen = [ ]



lst = [‘a’]

lst = [‘a’] / chosen = [ ]

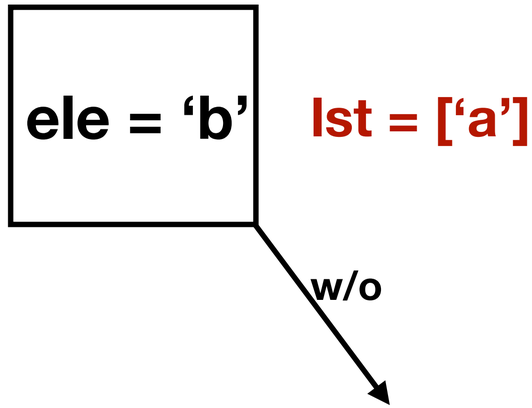


lst = [ ]

Base case!!

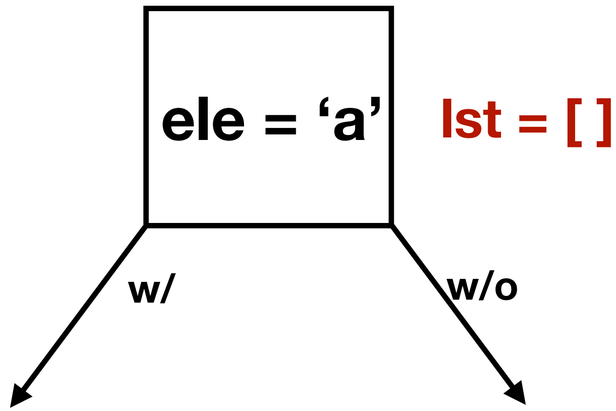
chosen = [ ]

lst = [‘a’, ‘b’] / chosen = [ ]



lst = [‘a’]

lst = [‘a’] / chosen = [ ]

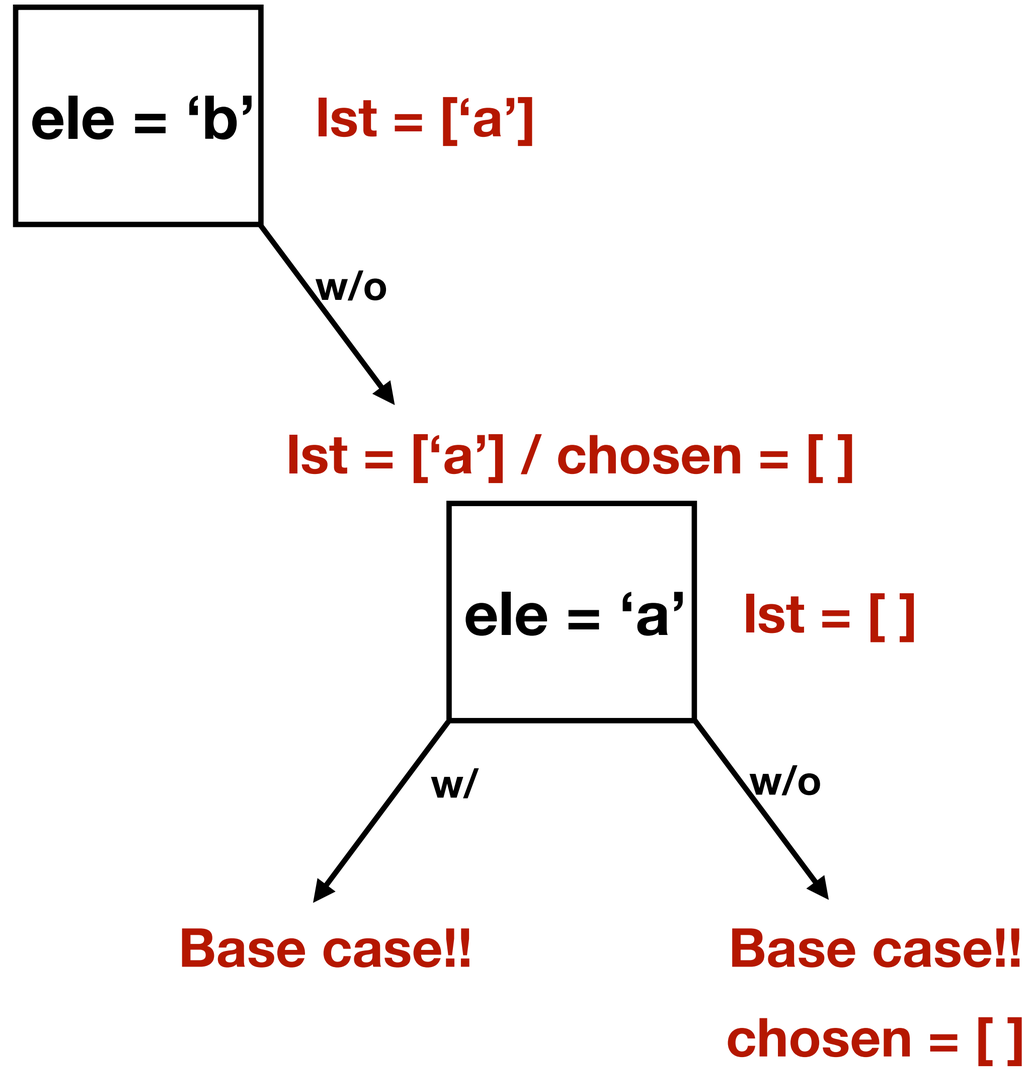


lst = [ ]

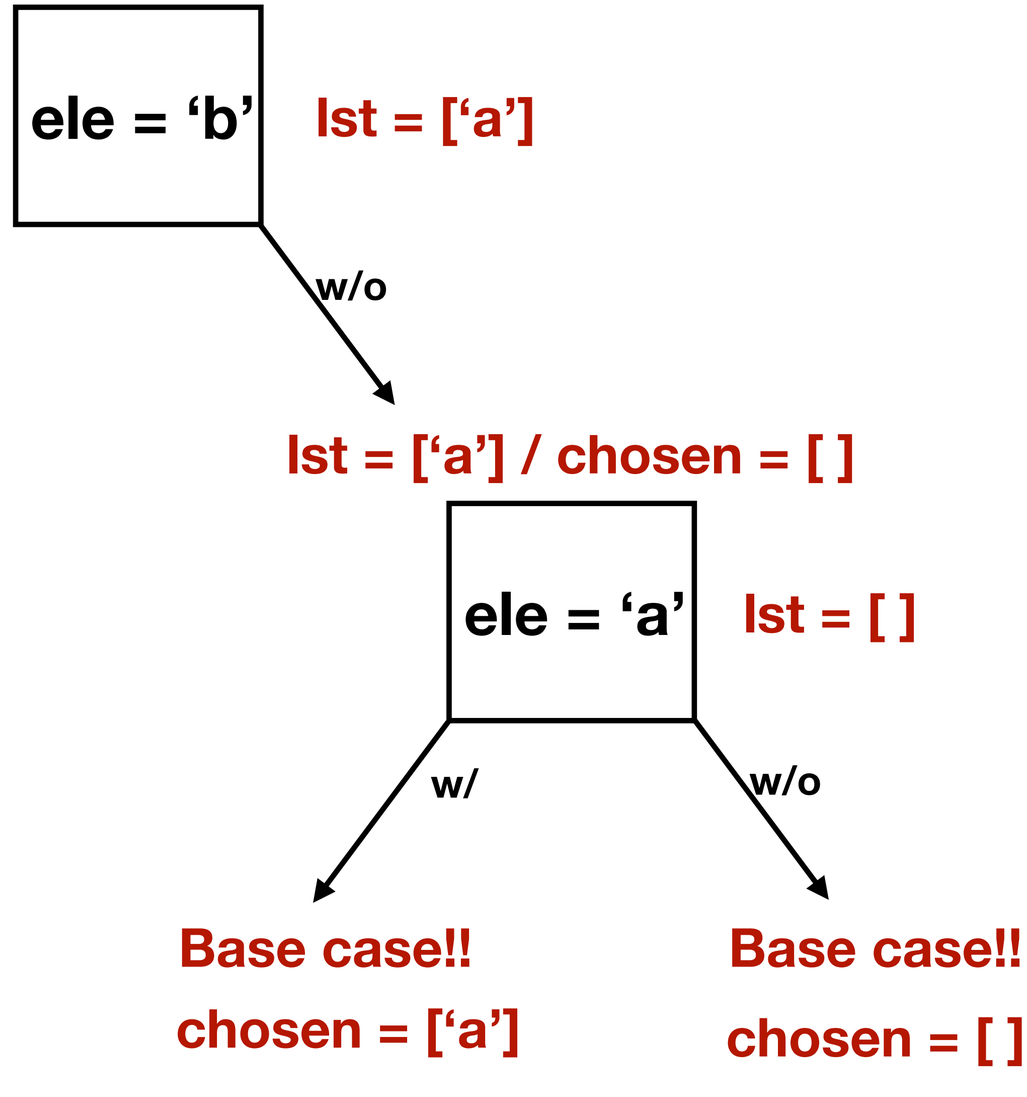
Base case!!

chosen = [ ]

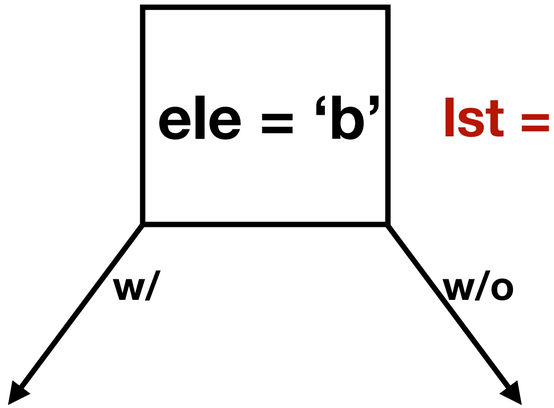
lst = [‘a’, ‘b’] / chosen = [ ]



lst = [‘a’, ‘b’] / chosen = [ ]

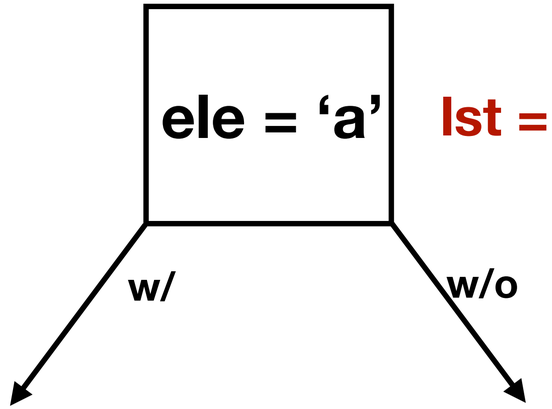


lst = [‘a’, ‘b’] / chosen = [ ]



lst = [‘a’]

lst = [‘a’] / chosen = [ ]



lst = [ ]

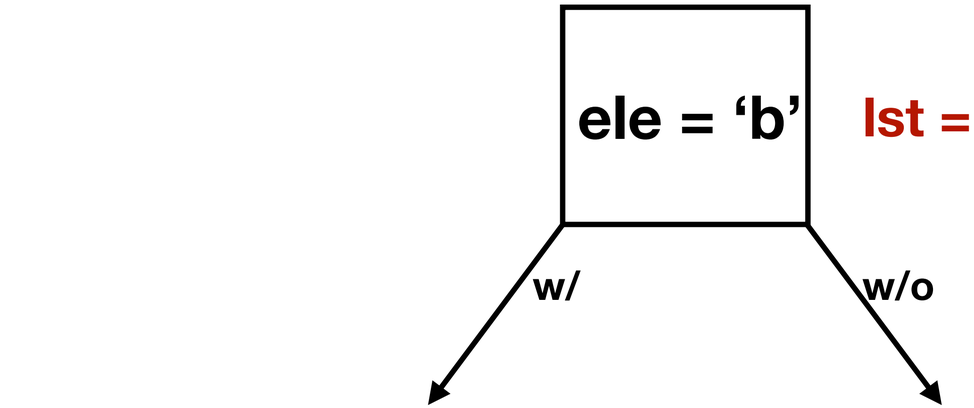
Base case!!

chosen = [‘a’]

Base case!!

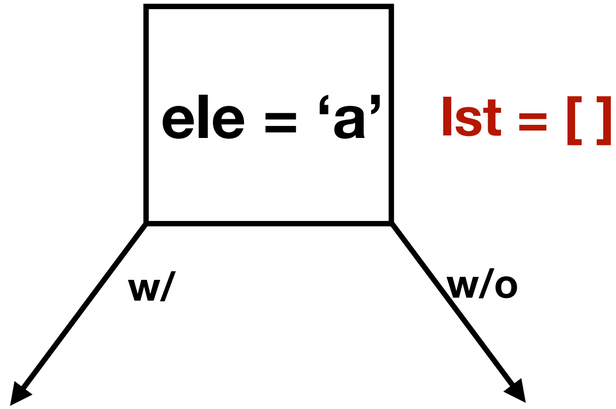
chosen = [ ]

lst = [‘a’, ‘b’] / chosen = [ ]



lst = [‘a’] / chosen = [‘b’]

lst = [‘a’] / chosen = [ ]



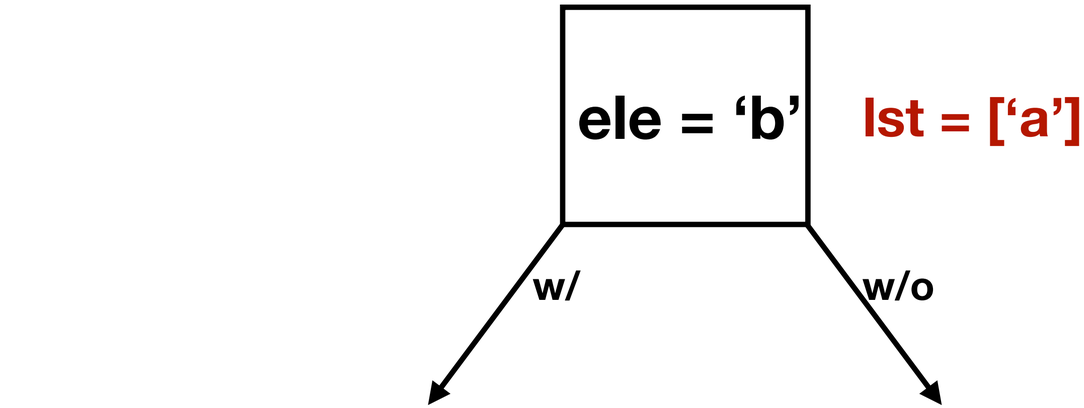
Base case!!

chosen = [‘a’]

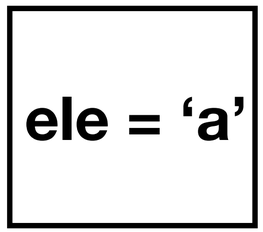
Base case!!

chosen = [ ]

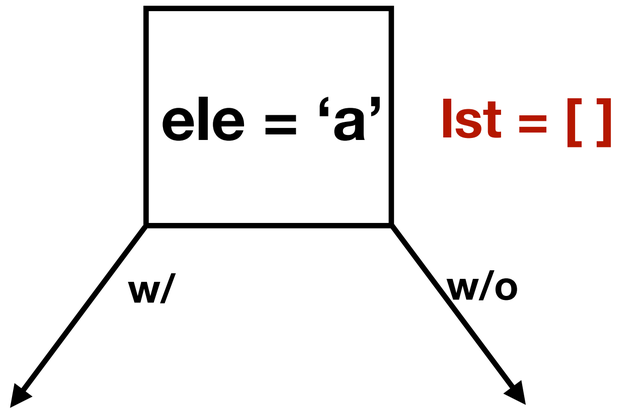
lst = [‘a’, ‘b’] / chosen = [ ]



lst = [‘a’] / chosen = [‘b’]



lst = [‘a’] / chosen = [ ]



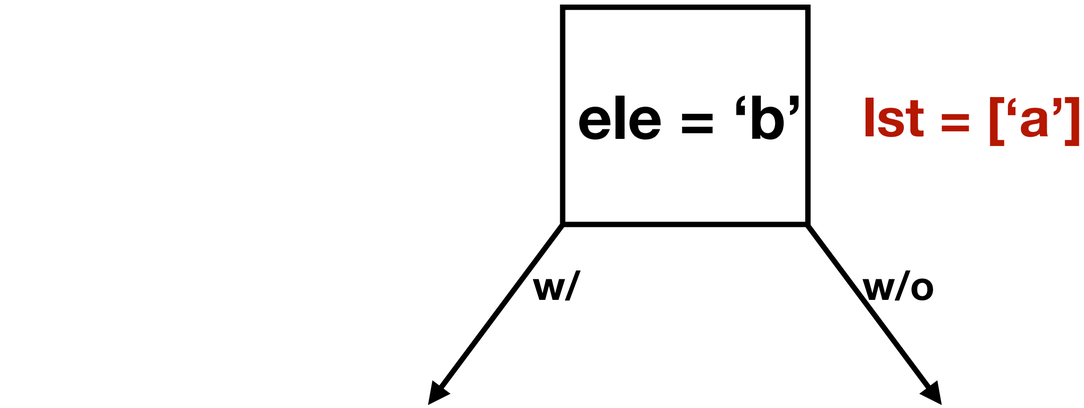
Base case!!

chosen = [‘a’]

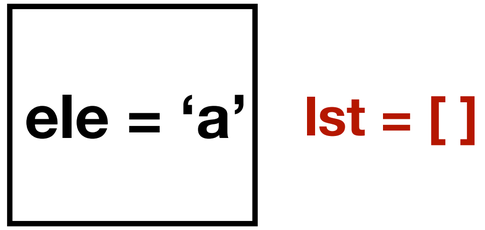
Base case!!

chosen = [ ]

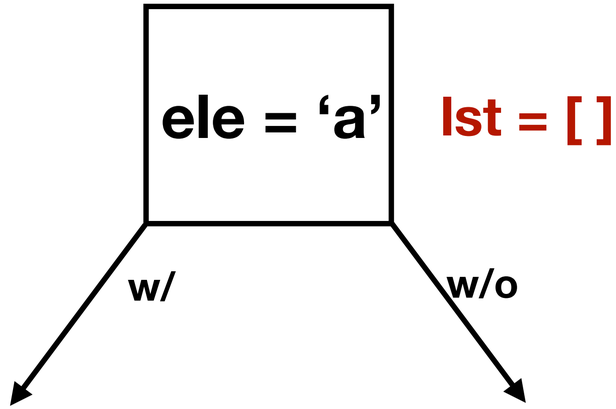
lst = [‘a’, ‘b’] / chosen = [ ]



lst = [‘a’] / chosen = [‘b’]



lst = [‘a’] / chosen = [ ]



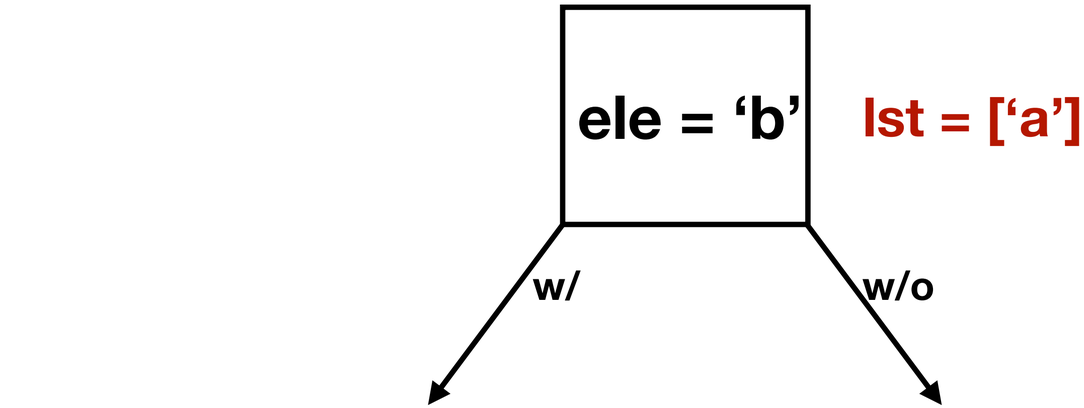
Base case!!

chosen = [‘a’]

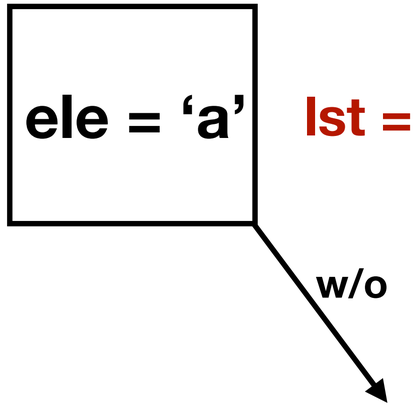
Base case!!

chosen = [ ]

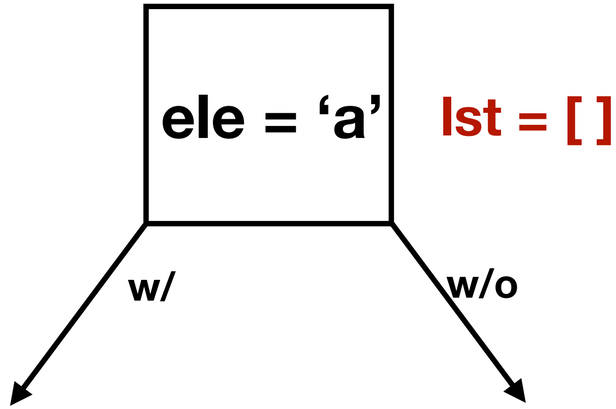
lst = [‘a’, ‘b’] / chosen = [ ]



lst = [‘a’] / chosen = [‘b’]



lst = [‘a’] / chosen = [ ]



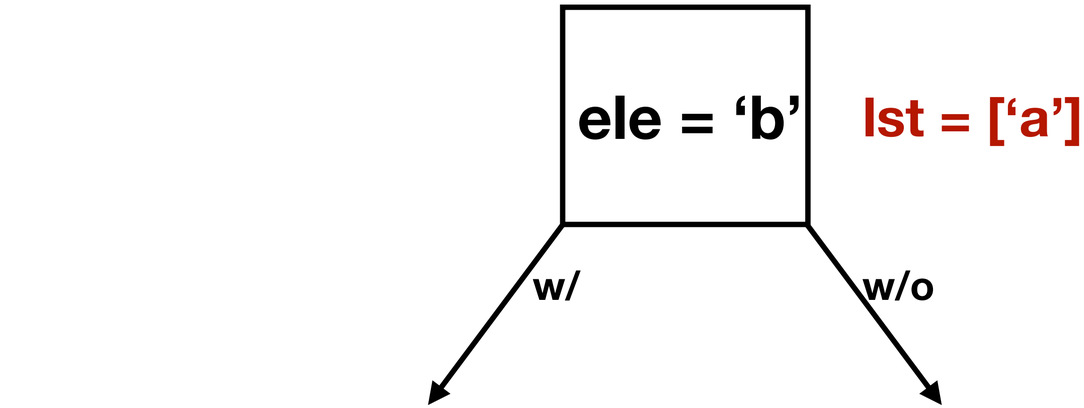
Base case!!

chosen = [‘a’]

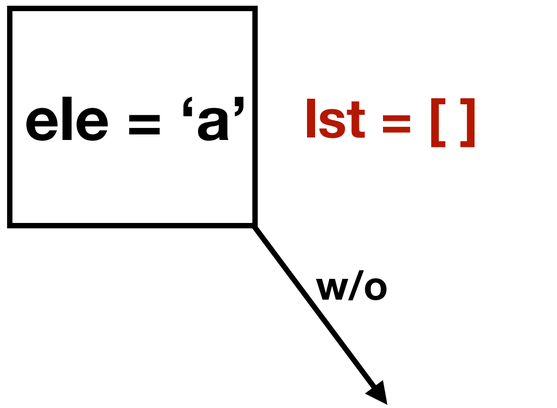
Base case!!

chosen = [ ]

lst = [‘a’, ‘b’] / chosen = [ ]

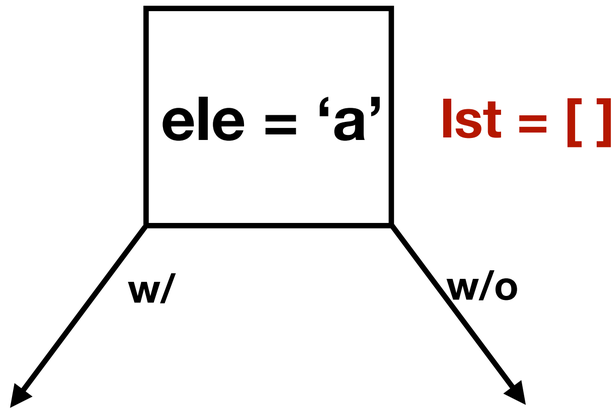


lst = [‘a’] / chosen = [‘b’]



Base case!!

lst = [‘a’] / chosen = [ ]



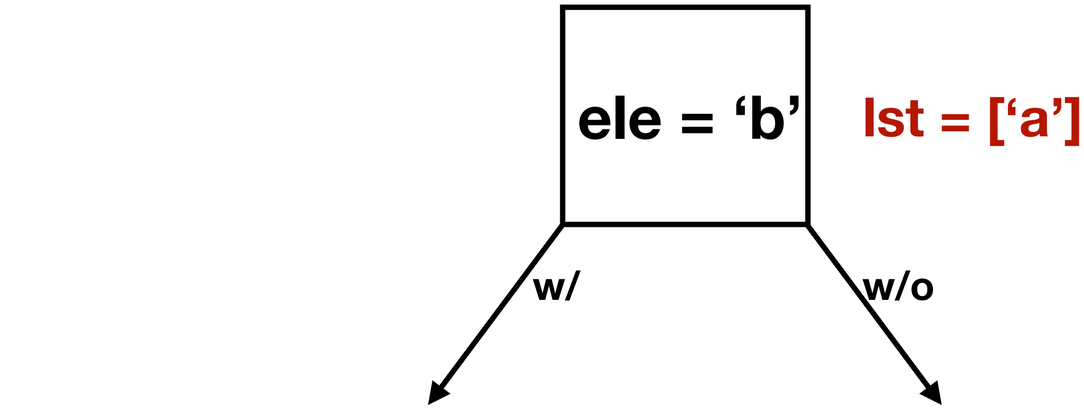
Base case!!

chosen = [‘a’]

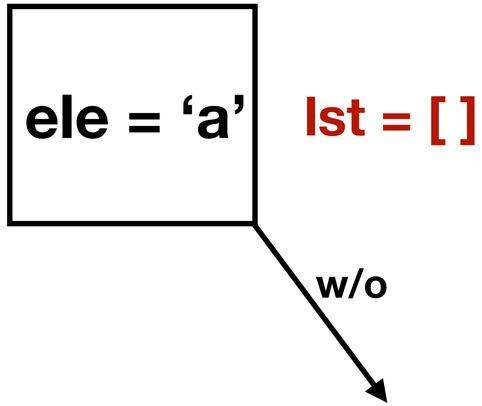
Base case!!

chosen = [ ]

lst = [‘a’, ‘b’] / chosen = [ ]



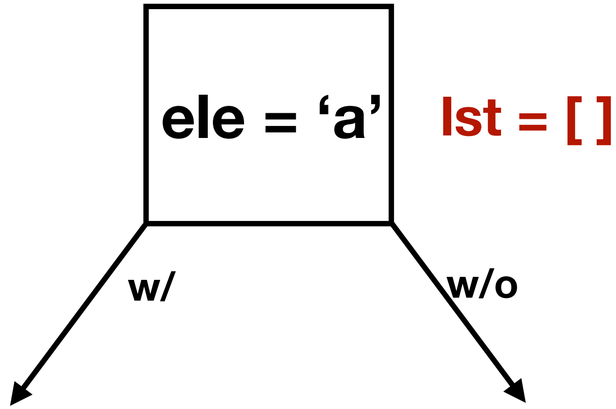
lst = [‘a’] / chosen = [‘b’]



Base case!!

chosen = [‘b’]

lst = [‘a’] / chosen = [ ]



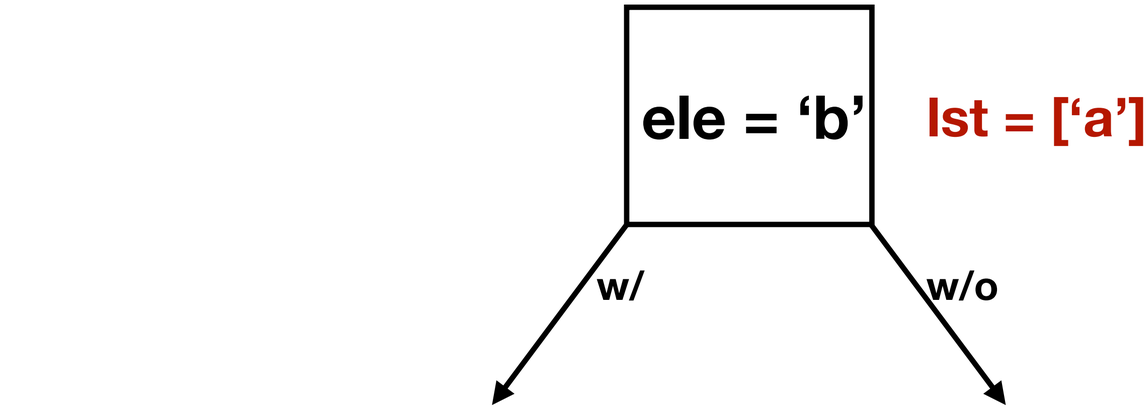
Base case!!

chosen = [‘a’]

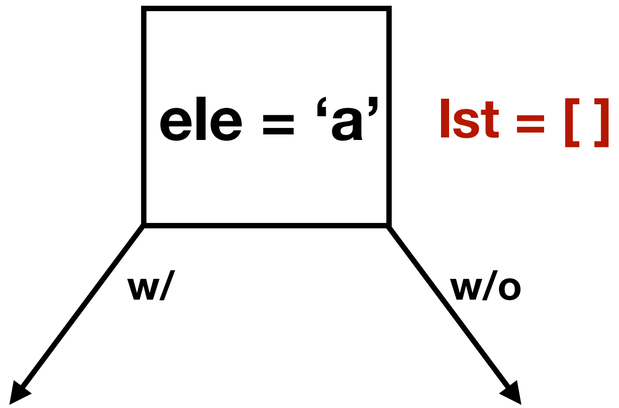
Base case!!

chosen = [ ]

lst = [‘a’, ‘b’] / chosen = [ ]



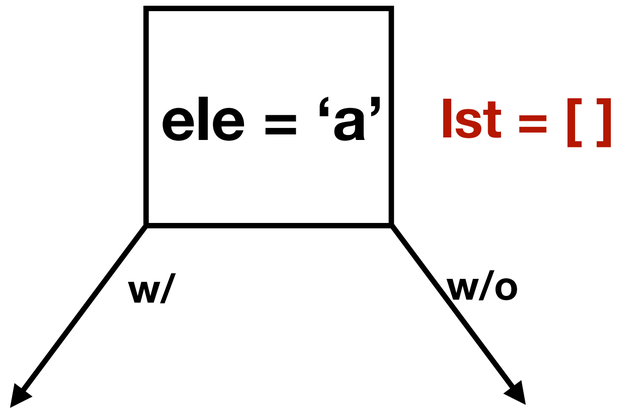
lst = [‘a’] / chosen = [‘b’]



Base case!!

chosen = [‘b’]

lst = [‘a’] / chosen = [ ]



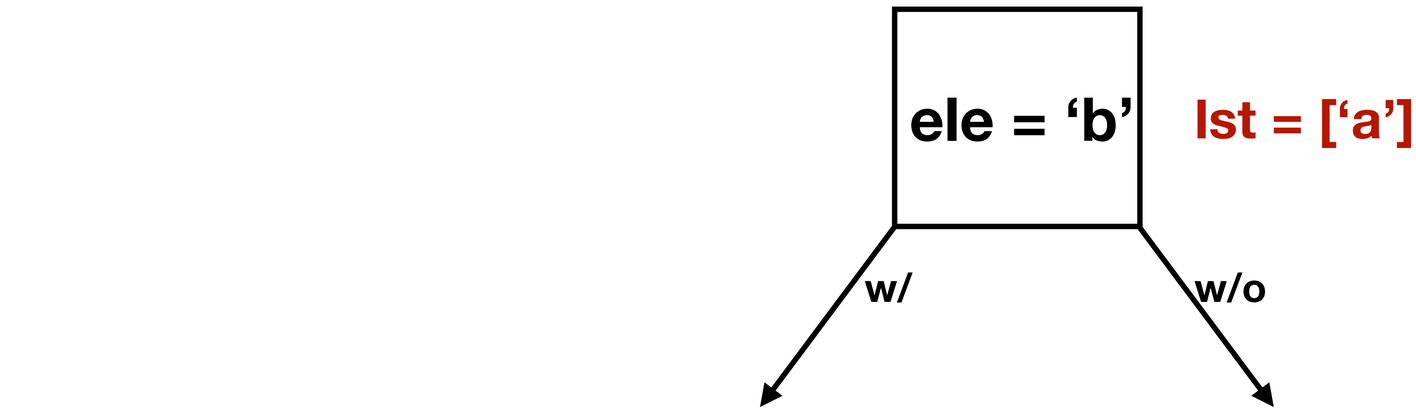
Base case!!

chosen = [‘a’]

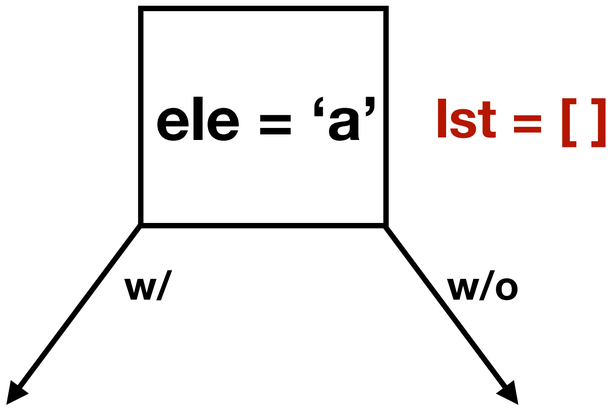
Base case!!

chosen = [ ]

lst = [‘a’, ‘b’] / chosen = [ ]



lst = [‘a’] / chosen = [‘b’]



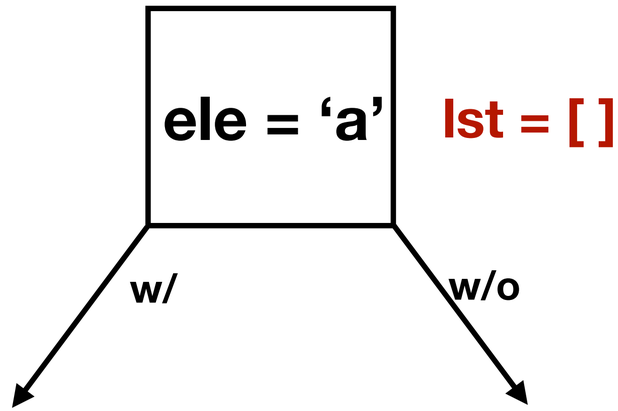
Base case!!

chosen = [‘a’, ‘b’]

Base case!!

chosen = [‘b’]

lst = [‘a’] / chosen = [ ]

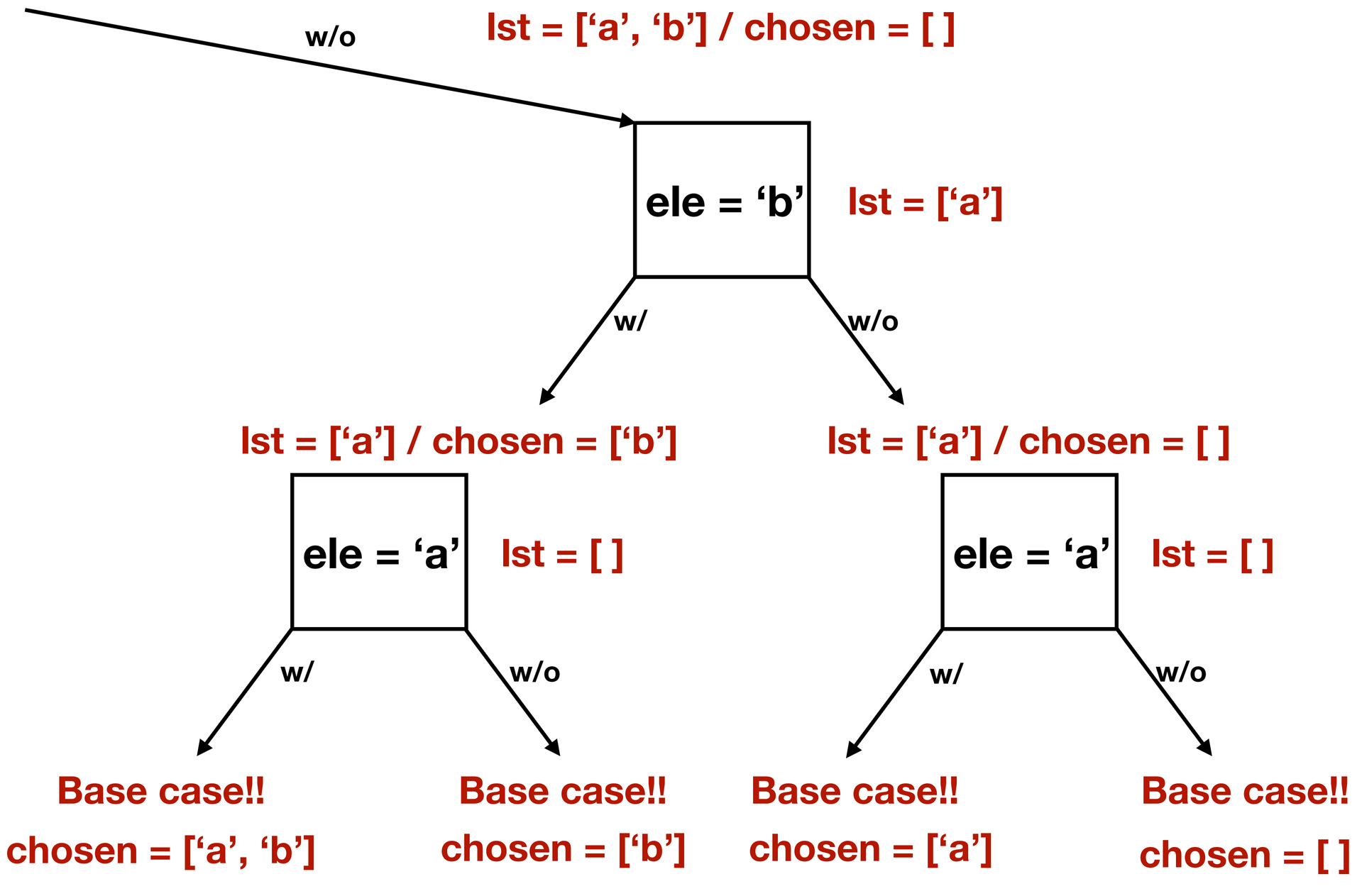


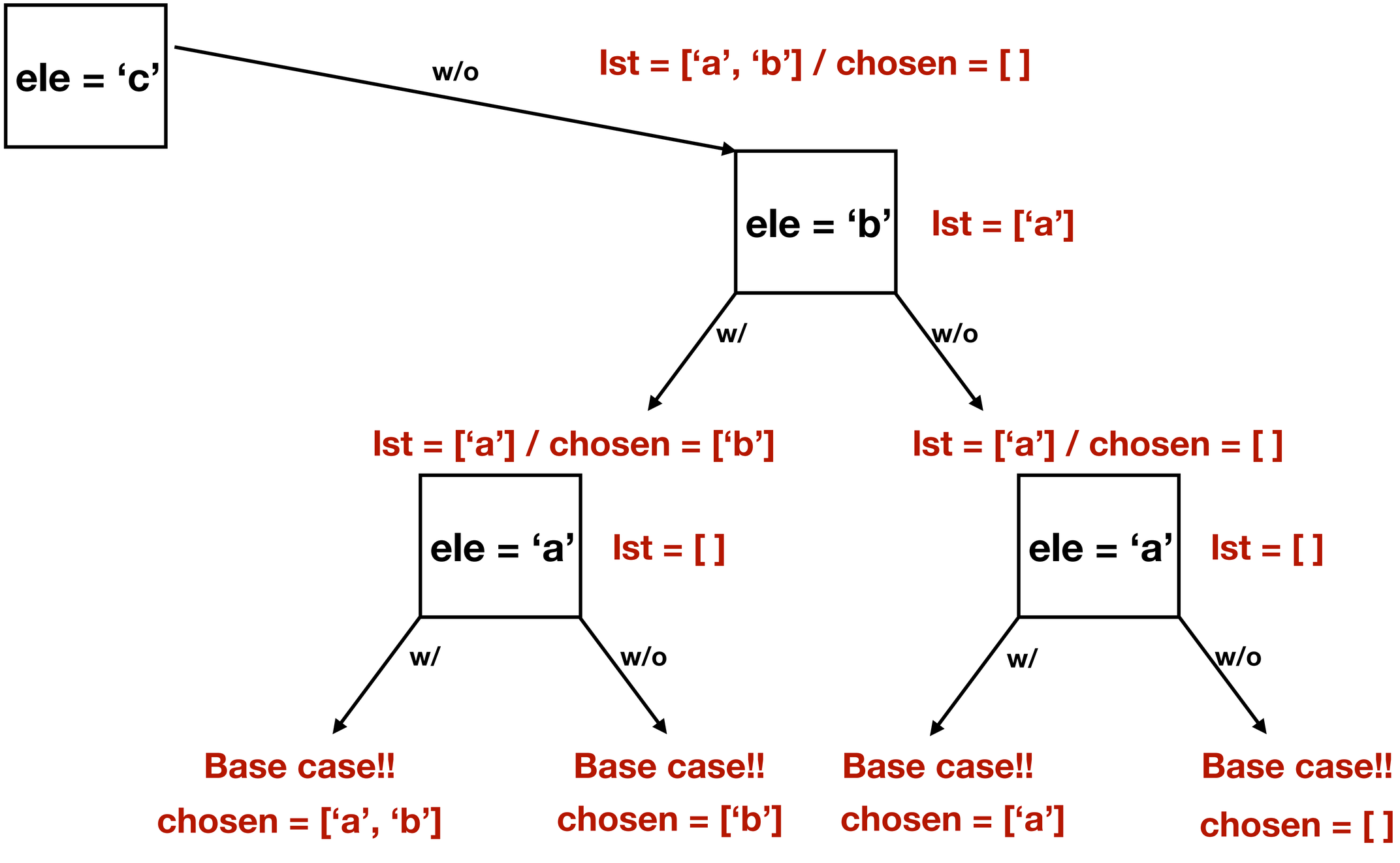
Base case!!

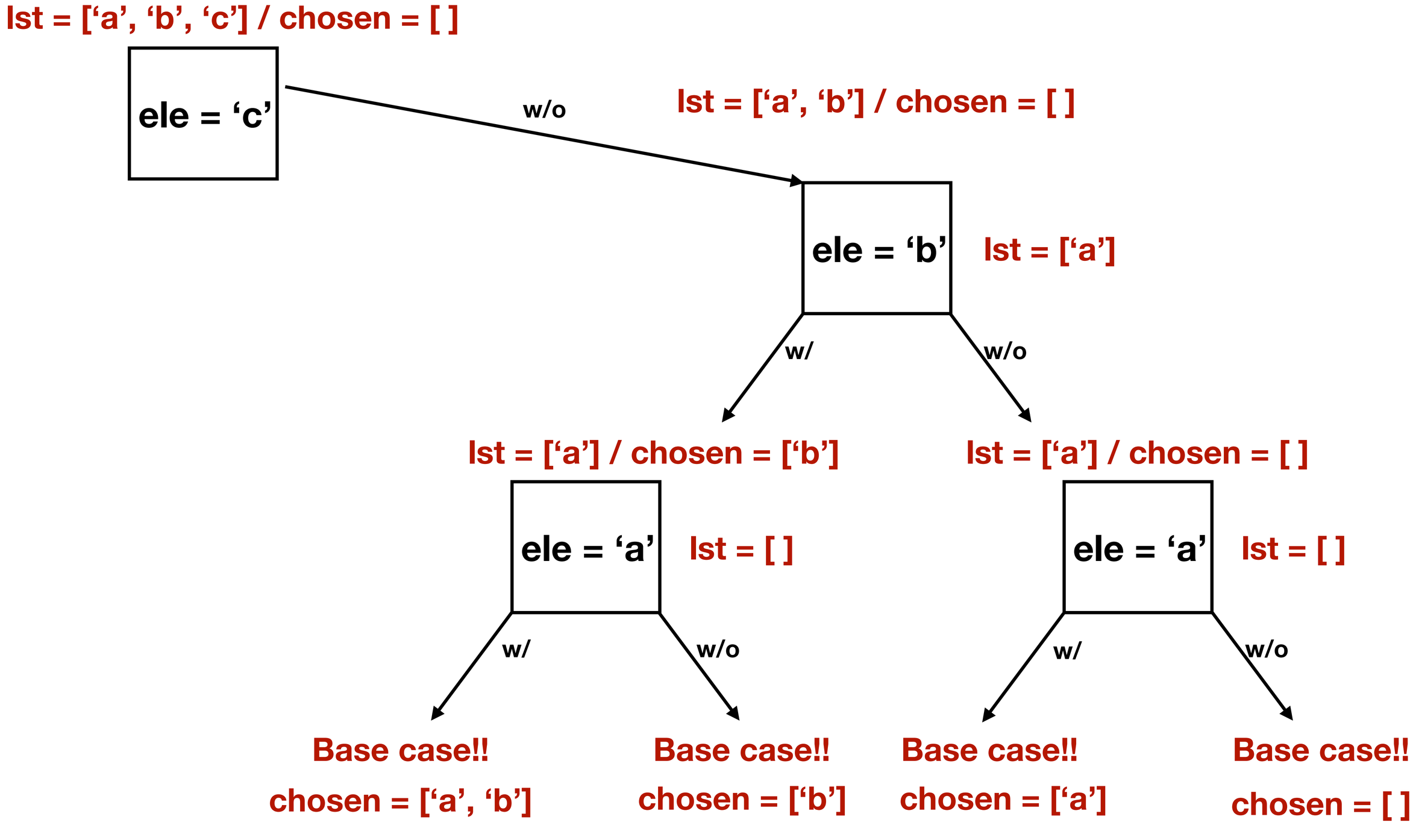
chosen = [‘a’]

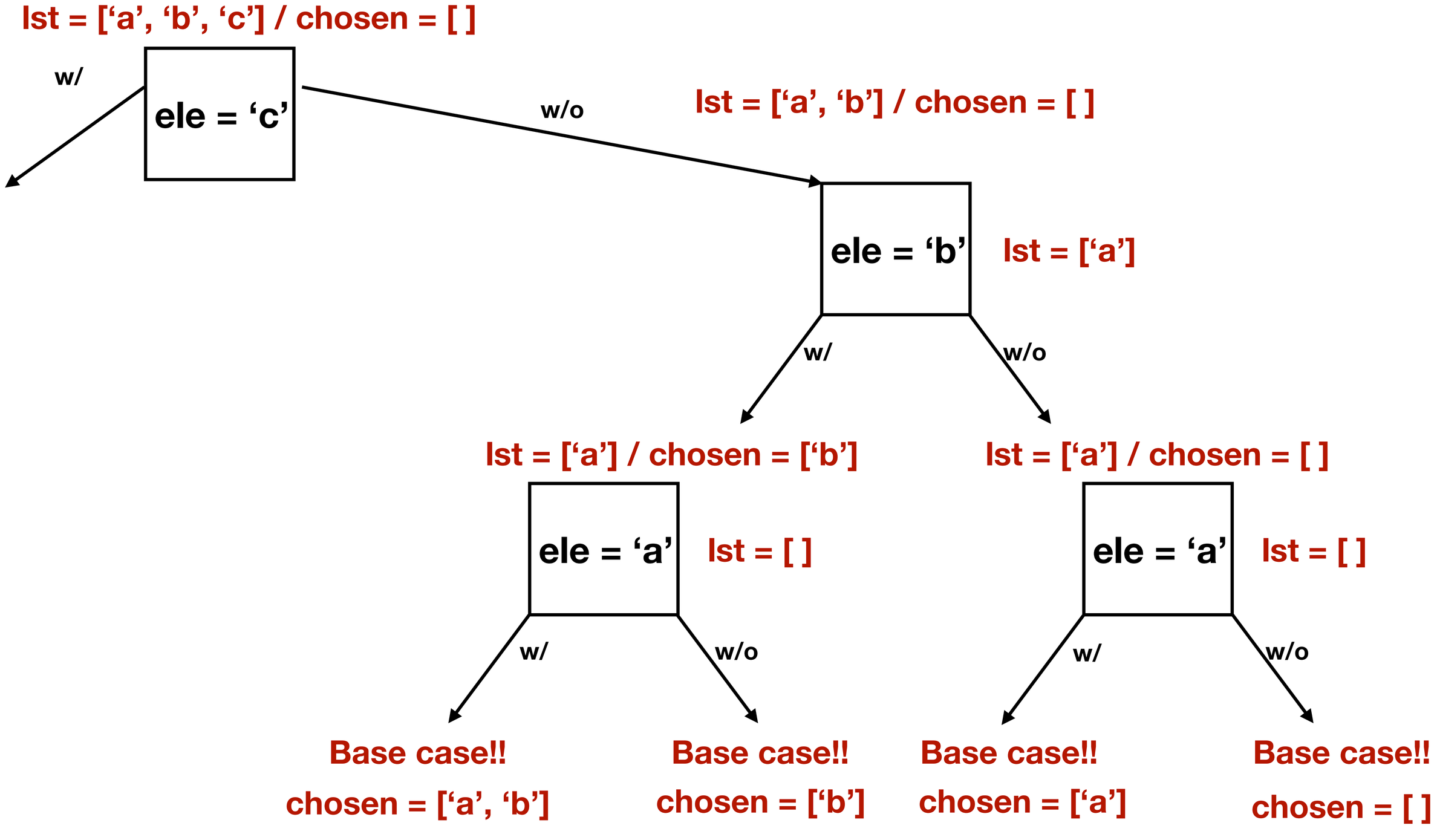
Base case!!

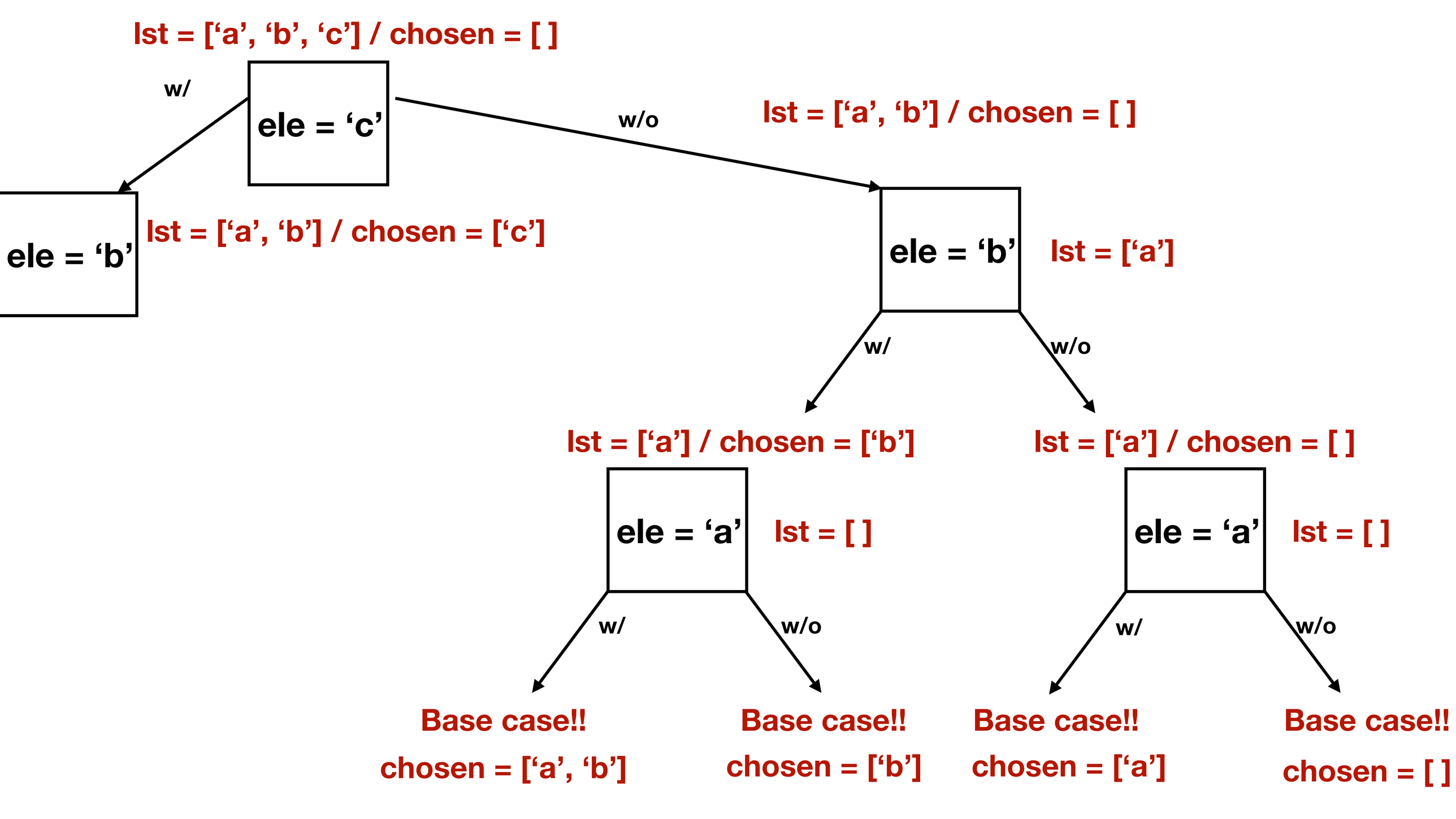
chosen = [ ]


w/o
lst = [‘a’, ‘b’] / chosen = [ ]
ele = ‘b’
lst = [‘a’]
w/
w/o
lst = [‘a’] / chosen = [‘b’]
lst = [‘a’] / chosen = [ ]
ele = ‘a’
lst = [ ]
w/
w/o
ele = ‘a’
lst = [ ]
w/
w/o
Base case!!
chosen = [‘a’, ‘b’]
Base case!!
chosen = [‘b’]
Base case!!
chosen = [‘a’]
Base case!!
chosen = [ ]


ele = ‘c’
w/o
lst = [‘a’, ‘b’] / chosen = [ ]
ele = ‘b’
lst = [‘a’]
w/
w/o
lst = [‘a’] / chosen = [‘b’]
lst = [‘a’] / chosen = [ ]
ele = ‘a’
lst = [ ]
w/
w/o
ele = ‘a’
lst = [ ]
w/
w/o
Base case!!
chosen = [‘a’, ‘b’]
Base case!!
chosen = [‘b’]
Base case!!
chosen = [‘a’]
Base case!!
chosen = [ ]


lst = [‘a’, ‘b’, ‘c’] / chosen = [ ]
ele = ‘c’
w/o
lst = [‘a’, ‘b’] / chosen = [ ]
ele = ‘b’
lst = [‘a’]
w/
w/o
lst = [‘a’] / chosen = [‘b’]
lst = [‘a’] / chosen = [ ]
ele = ‘a’
lst = [ ]
w/
w/o
ele = ‘a’
lst = [ ]
w/
w/o
Base case!!
chosen = [‘a’, ‘b’]
Base case!!
chosen = [‘b’]
Base case!!
chosen = [‘a’]
Base case!!
chosen = [ ]


lst = [‘a’, ‘b’, ‘c’] / chosen = [ ]
w/
ele = ‘c’
w/o
lst = [‘a’, ‘b’] / chosen = [ ]
ele = ‘b’
lst = [‘a’]
w/
w/o
lst = [‘a’] / chosen = [‘b’]
lst = [‘a’] / chosen = [ ]
ele = ‘a’
lst = [ ]
w/
w/o
ele = ‘a’
lst = [ ]
w/
w/o
Base case!!
chosen = [‘a’, ‘b’]
Base case!!
chosen = [‘b’]
Base case!!
chosen = [‘a’]
Base case!!
chosen = [ ]


lst = [‘a’, ‘b’, ‘c’] / chosen = [ ]
w/
ele = ‘c’
w/o
lst = [‘a’, ‘b’] / chosen = [ ]
ele = ‘b’
lst = [‘a’, ‘b’] / chosen = [‘c’]
ele = ‘b’
lst = [‘a’]
w/
w/o
lst = [‘a’] / chosen = [‘b’]
lst = [‘a’] / chosen = [ ]
ele = ‘a’
lst = [ ]
ele = ‘a’
lst = [ ]
w/
w/o
w/
w/o
Base case!!
chosen = [‘a’, ‘b’]
Base case!!
chosen = [‘b’]
Base case!!
chosen = [‘a’]
Base case!!
chosen = [ ]