

Compound data types

- tuple
- list
- dictionary
- set

List

```
x = ["a", 1, 3]
```

```
x.append(4)
```

Tuple

```
my_tuple = (-1, -2, -3)
```

```
my_tuple = my_tuple + (-4,)
```

$(-1, -2, -3) + (-4,)$

$(-1, -2, -3, -4)$

Set

```
my_set = {1, 2, 3}
```

```
my_set.add("a")
```

```
my_set.add(1)
```

Dictionary

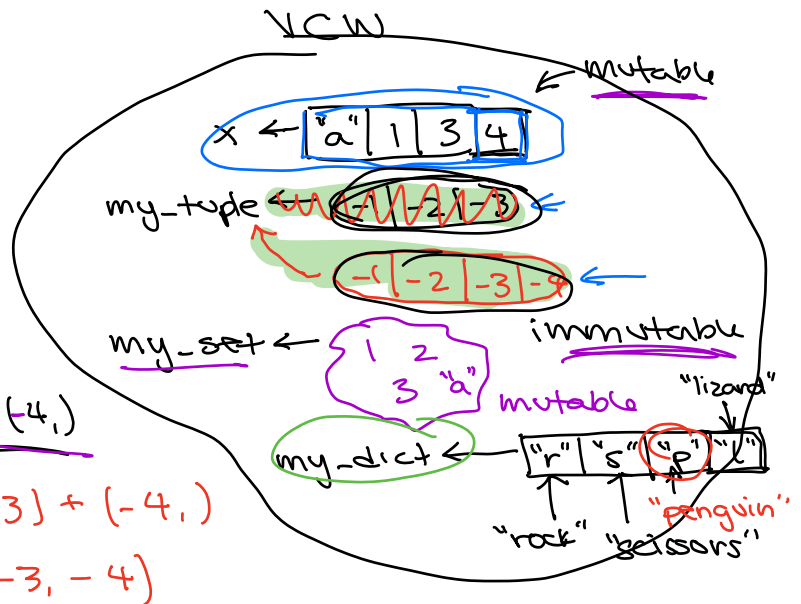
```
my_dict = {"r": "rock", "s": "scissors",  
           "p": "paper"}
```

```
my_dict["p"] → "paper"
```

```
my_dict["rock"] → error!
```

```
my_dict["l"] = "lizard"
```

```
my_dict["p"] = "penguin"
```



Loops

- for loops

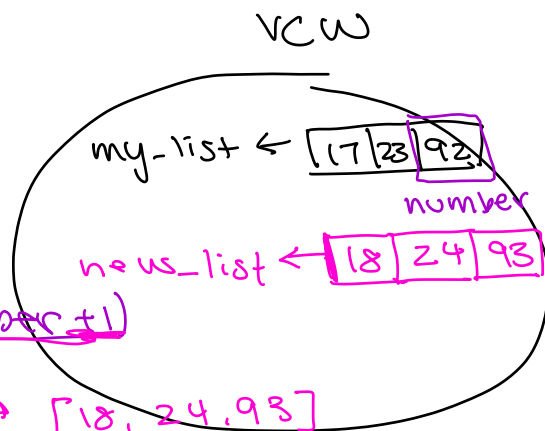
```
my_list = [17, 23, 92]
```

```
new_list = []
```

```
for number in my_list:
```

```
new_list.append(number + 1)
```

```
print(new_list) → [18, 24, 93]
```

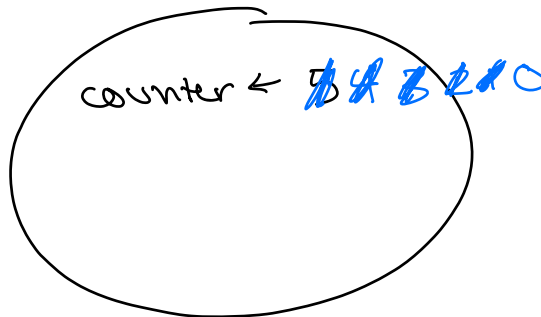


- while loops

```
counter = 5
while counter > 0:
    print(counter)
    counter = counter - 1
print("BLAST OFF")
```

5
4
3
2
1

VCW



5
4
3
2
1
BLAST OFF

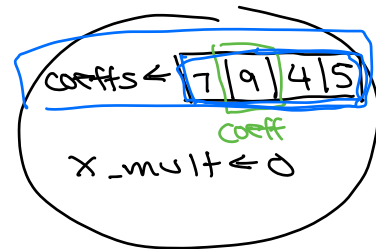
Functions

$$f(x) = 7 + 9x + 4x^2 + 5x^3$$

coeffs = [7, 9, 4, 5]

x = 4.2

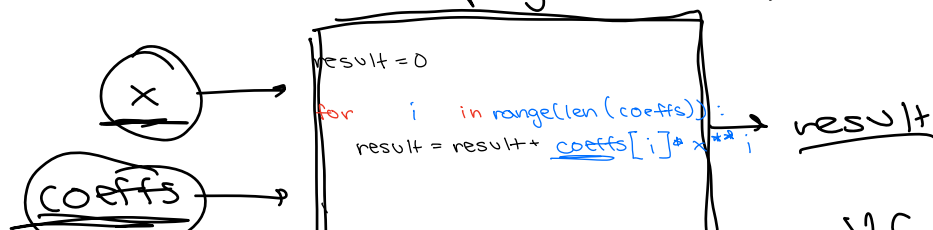
result = 0



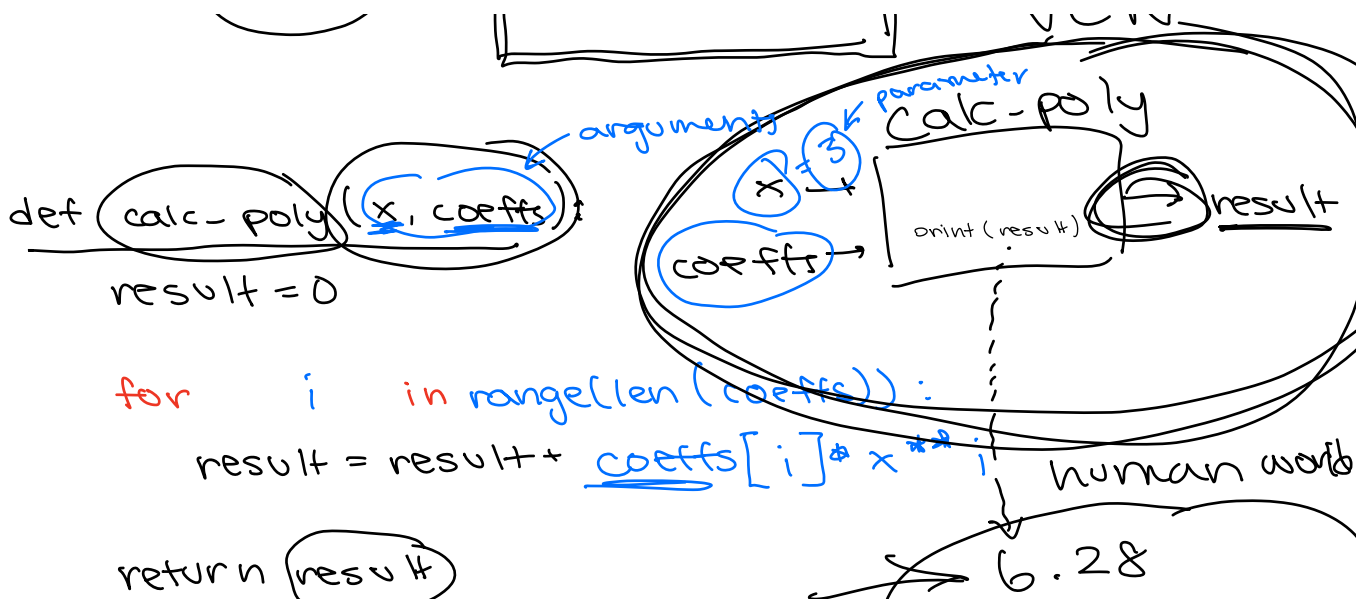
```
for i in range(len(coeffs)):
    result = result + coeffs[i] * x ** i
```

print(result)

def calc_poly(x, coeffs):



1 1 1 1 1



range(x) → 0, 1, ..., x-1
 print(something) → something gets printed
 len("dog") → 3

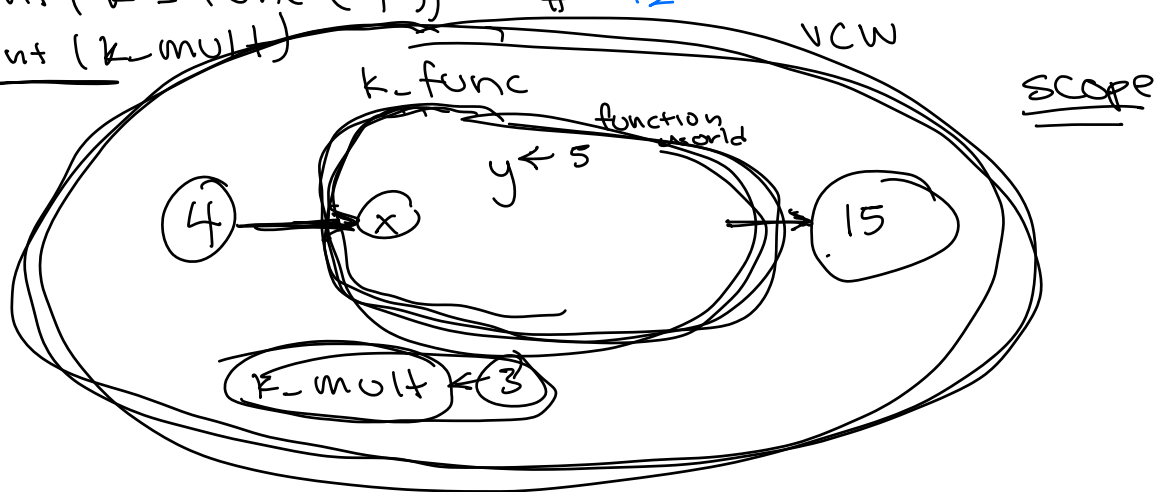
k_mult = 3
 def k_func(x):

y = x + 1
 return k_mult * y

x = 5
 y = x + 1

x ← 5
 y ← 6

print(k_func(4)) # 12
 print(k_mult)



```

pi = 3.14
def perimeter(r):
    return pi * 2 * r
  
```

```

p = perimeter(5)
print(p) # 31.4
  
```

$$P = (2\pi r)^2$$

$$A = \pi r^2$$

```

def area(r):
    p = perimeter(r)
    return (p / 2) ** 2
  
```

