

Lesson 2: Binary Search

Programming Fundamentals in Python

Lesson 1 Recap

- FizzBuzz
- Prime numbers below one million

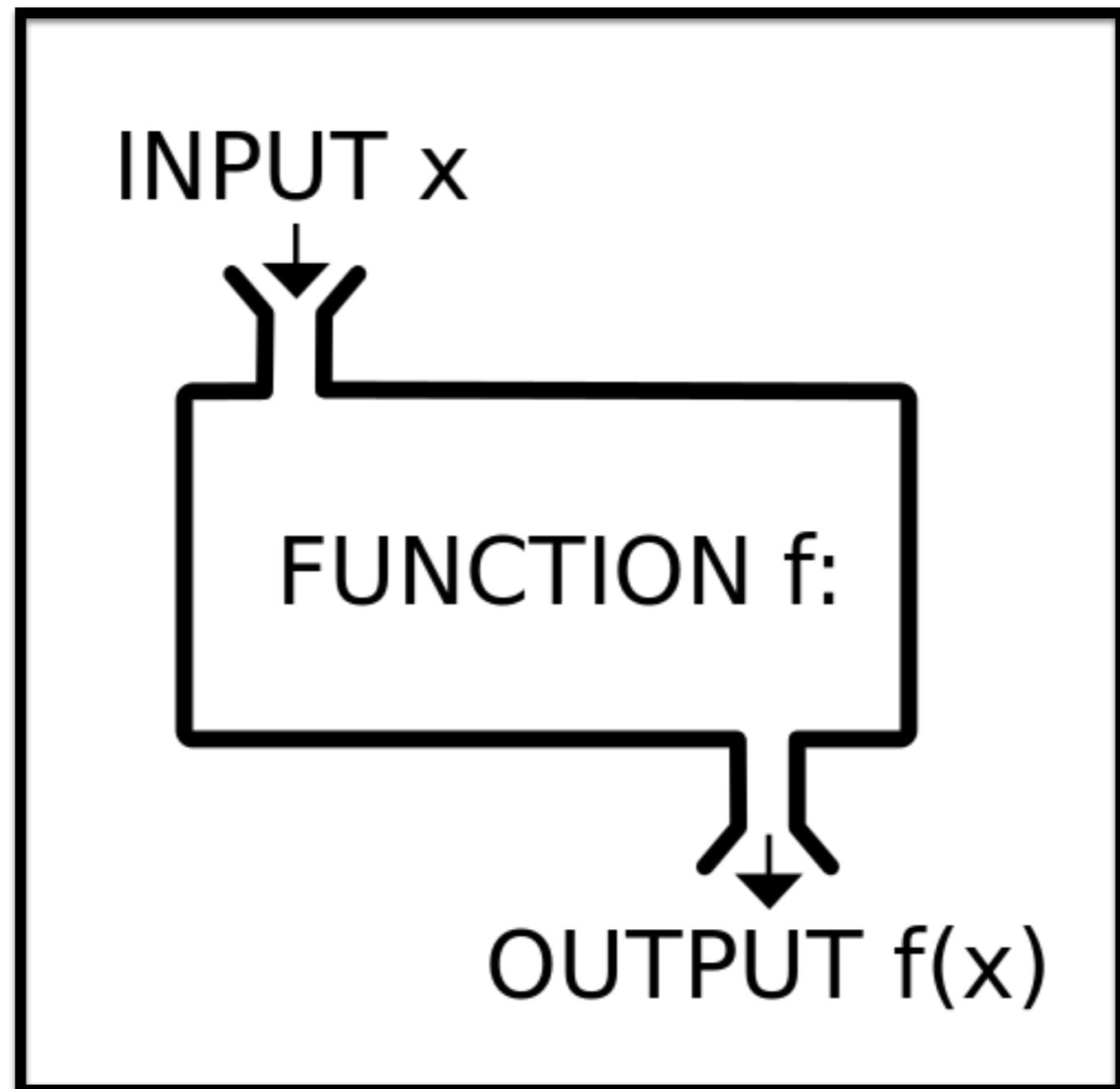
Today's Goal

- Implement “What Number am I thinking of?”
 - Using binary search

What Number Am I Thinking Of Demo

Breakdown

- Functions
- Binary Search
- Lists
- Dictionaries
- Sets



Functions

f(x)

Functions

- Functions are recipes!
- Re-use your code

Functions

```
def square(x):  
    return x * x  
  
square(5) # value is 25
```

Functions

```
def multiply(a, b):  
  
    total = 0  
  
    for i in range(0, a):  
  
        total = total + b  
  
    return total  
  
multiply(2,3) # 2 * 3  
  
3 + 3
```



```
def make_lasagna():

    noodles = cook_noodles()

    sauce = make_sauce()

    combined = combine_in_tray(get_tray(),
                                noodles,
                                sauce)

    lasagna = bake(combined)

    return lasagna
```

```
def fire_the_missiles():

    connect_to_norad()

    authenticate("obama", "topsecretpass")

    send_missile_code(1234)
```



Lists

Lists

- For ordered data
- Most common data structure

```
numbers = [1,2,3]
```

```
numbers[0] # value is 1
```

```
numbers[1] = -1
```

```
# numbers is now [1,-1,3]
```

```
numbers.append(4)
```

```
# numbers is now [1,-1,3,4]
```



Dictionaries

Dictionaries

- Unordered data
- “Keys” and “Values”

Dictionaries

```
favorite_foods = {"David": "Burrito"}
```

```
print(favorite_foods["David"])
```

```
favorite_foods["Jackie"] = "Sushi"
```

```
favorite_foods["David"] = "Pizza"
```

```
print(favorite_foods["David"])
```

Dictionaries

```
favorite_foods = {"David": "Burrito"}
```

```
# These two lines are equivalent to the  
above line
```

```
favorite_foods = {}
```

```
favorite_foods["David"] = "Burrito"
```

```
bank_accounts = {  
    "David": 10,  
    "Jackie": 100,  
    "BillGates": 5000000,
```

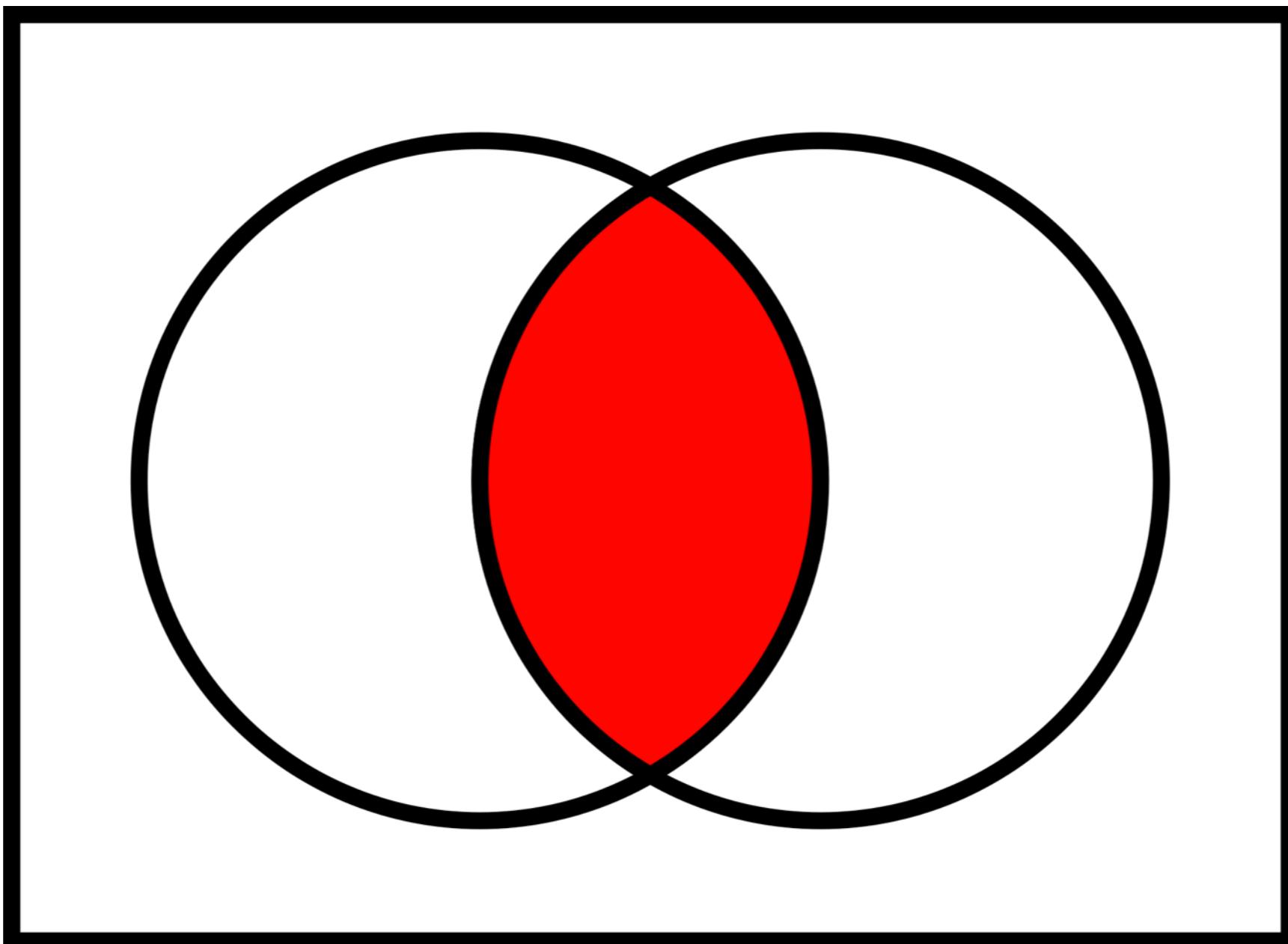
```
}
```

```
amount = 500
```

```
bank_accounts["David"] += amount
```

```
bank_accounts["BillGates"] -= amount
```

```
print("Thanks, Bill!")
```



Sets

Sets

- Unordered data
- Only holds UNIQUE elements
 - No duplicates

Sets

```
primes = {2, 3, 5, 7} # literal

for i in range(1,10):

    if i in primes:

        print("I found a prime:", i)

    else:

        print("%d was not a prime." % i)
```

Sets

```
primes = set()
```

```
primes.add(2)
```

```
primes.add(3)
```

```
primes.add(5)
```

```
primes.add(7)
```

Sets

```
primes = set([2,3,5,7]) # create from  
list
```

Sets

```
primes = {2, 3, 5, 7}
```

```
primes.add(4)
```

```
# A: What is the value of primes?
```

```
# {2, 3, 5, 4, 7}
```

```
primes.add(5)
```

```
# B: What is the value of primes?
```

Binary Search

Binary search

steps: 0



Sequential search

steps: 0



Binary Search Whiteboard Demonstration

```
target = 42
```

```
values = range(1,101)
```

```
lower_bounds = 0
```

```
upper_bounds = 100
```

```
found = False
```

```
while !found:  
  
    position = int((lower_bounds +  
                    upper_bounds) / 2)  
  
    if values[position] < target:  
  
        lower_bounds = position  
  
    elif values[position] > target:  
  
        upper_bounds = position  
  
    else:  
  
        found = True  
  
        print("found target %d at position %d" % (target,  
position))
```

```
target = 42
values = range(1,100)
lower_bounds = 0
upper_bounds = 100

found = False

while !found:
    position = int((lower_bounds +
                     upper_bounds) / 2)
    if values[position] < target:
        lower_bounds = position
    elif values[position] > target:
        upper_bounds = position
    else:
        found = True
        print("found target %d at position %d" %
              (target, position))
```

Recap

- Functions
- Lists
- Dictionaries
- Sets
- Binary Search

Homework Assignment

- “What Number Am I Thinking Of?”

What Number Am I Thinking Of?

- Ask the user to think of a number from 1-100
- Guess a number
- Ask the user if their number is higher, lower, or equal to the number that you just guessed
 - If not the number, use binary search to loop and converge on the number
 - If the number, tell the user what their number is and how many guesses it took to find it

What Number Am I Thinking Of?

```
response = input("Was this number  
(G)reater, (L)esser, or (E)qual to your  
number?\n")  
  
if response == "G":  
    # handle greater  
  
elif response == "L":  
    #handle lesser
```

Bonus Assignment

Using the zipcodes.txt dataset:

- `find_zipcode(zip)`: Find the position of a given zipcode using binary search
- `find_unique_zipcodes()`: Eliminate duplicates and print the list
- `find_zipcode_count()`: How many times does each zip code appear?
- `find_zipcode_range(zip)`: Building on your `find_zipcode()` solution, find the first and last instance of a given zipcode

Next Week

Adventure game!

Using:

CSV & Higher-order Functions