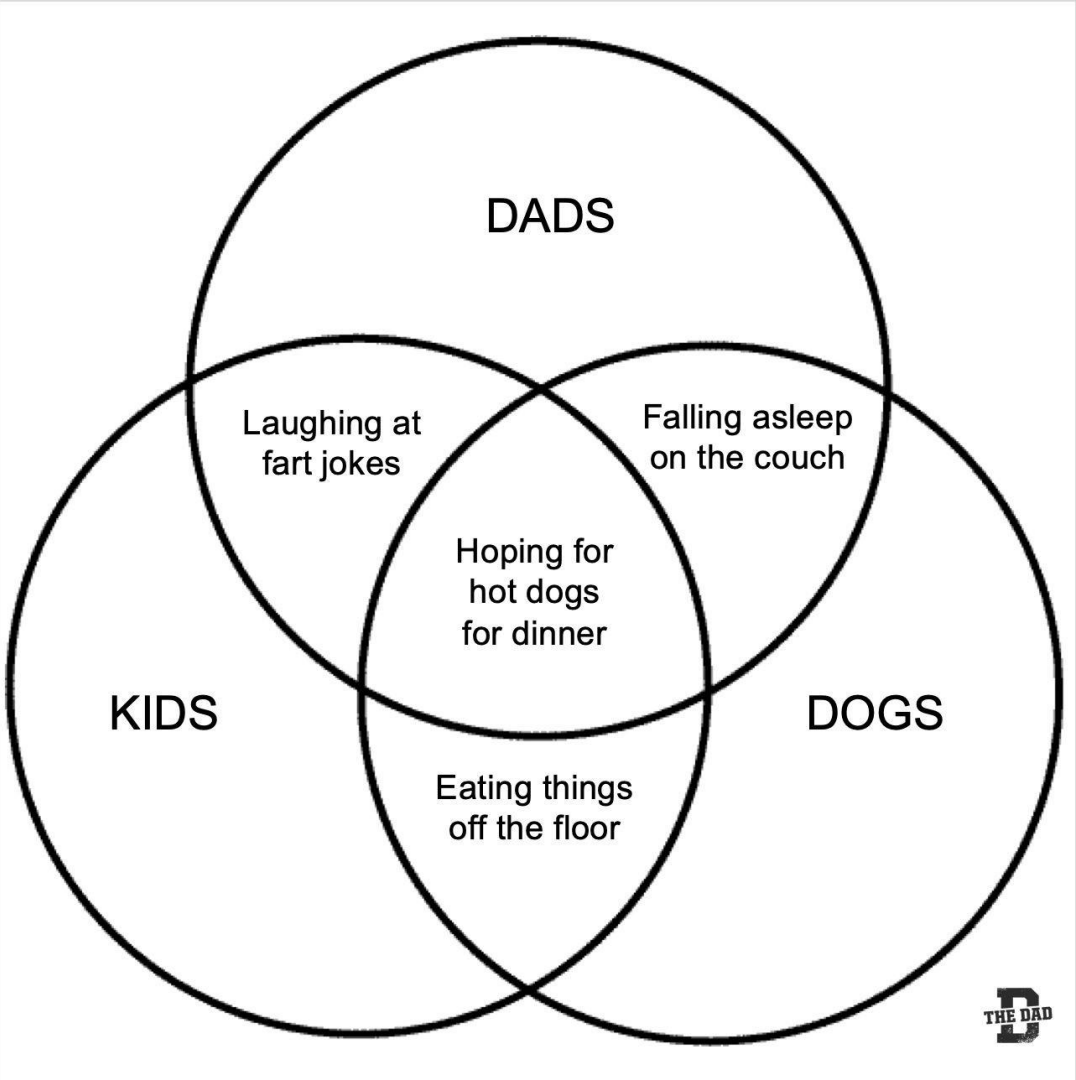
CSc 110 Sets

Benjamin Dicken



Set

- A **set** is (another) data structure
- Helpful ways of thinking about it
  - A dictionary without the values
  - A “bag" of elements

Similarities

```
# Dictionary creation
ds = {'a':8, 'b':7, 'c':4}

# Dictionary loop
for key in ds:
    print(key)
```

```
# Set creation
ds = {'a', 'b', 'c'}

# Set loop
for element in ds:
    print(element)
```

Differences

```
ds = {'a':8, 'b':7, 'c':4}

# Remove from dictionary
del ds['c']

# Add to dictionary
ds['e'] = 20

# Create empty
ds_2 = {}
```

```
ds = {'a', 'b', 'c'}

# Set removal
ds.remove('c')

# Adding to set
ds.add('e')

# ???
ds_2 = {}
```

What will print?

```
numbers = {1, 2, 3, 4, 'word'}
numbers.add(5)
numbers.remove(5)
numbers.add(1)
numbers.remove(7)
numbers.add(5)
print(numbers)
```

What will print?

```
numbers = {1, 2, 3, 4, 'word'}
numbers.add(5)
numbers.remove(5)
numbers.discard(5)
numbers.add(1)
numbers.discard('words')
numbers.add(2)
print(numbers)
```

Looping through a set

- What would print?

```
names = {"Jones", "James", "Zac"}
for i in range(0, len(names)):
    print(names[i])
```

Looping through a set

- Why What would print?

```
names = {"Jones", "James", "Zac"}
for i in range(0, len(names)):
    print(names[i])
```

- Elements cannot be “looked up” by index (position) in the data structure
- You would end up with an error:

```
TypeError: 'set' object does not support indexing
```

Looping through a set

- Use this instead:

```
names = {"Ben", "James", "Zac"}
for name in names:
    print(name)
```

- Iterates through the *elements* of the set, not indexes

Differences from a Dictionary

```
ds = {'a':8, 'b':7, 'c':4}

# Get value from dictionary
value = ds['c']

# Change value in dictionary
ds['c'] = 23
```

```
ds = {'a', 'b', 'c'}

# ?

# ?
```

What would be in grades?

```
grades = set()
letters = ['C', 'B', 'E', 'C', 'A', 'B', 'B', 'A']
for l in letters:
    if l in grades:
        grades.remove(l)
    else:
        grades.add(l)
print(grades)
```

What will happen?

```
grades = {'A+', 'A', 'B', 'E', 'D', 'E', 'E-'}
grade_counts = {'A':5, 'B':10, 'C':7, 'D':4, 'E':2}
for element in grades:
    if element not in grade_counts:
        grades.discard(element)
    else:
        del grade_counts[element]
print(grades)
```

What will happen?

```
grades = {'A+', 'A', 'B', 'E', 'D', 'E', 'E-'}
grade_counts = {'A':5, 'B':10, 'C':7, 'D':4, 'E':2}
for element in grades:
    if element in grade_counts:
        del grade_counts[element]
print(grade_counts)
```

Activity

Exercise: Counting names

- Implement a program that . . .
  - Reads in a text file formatted like the example to the right named **names.txt**
  - Notice that some names repeat
  - The program should count how many unique names there are!
  - **Don't use a list or dictionary**

```
Lebron James
James Harden
Chris Paul
Chris Tucker
Kevin Durant
James Harden
Steve Tucker
Steve Smith
Eric Bledsoe
Steve Caroll
Chris Paul
Sally Smith
Kevin Durant
James Jones
Chris Paul
```

Exercise: Counting names

```
names = set()
names_file = open('names.txt', 'r')
for line in names_file:
    name = line.strip('\n')
    names.add(name)
print(len(names))
```

```
Lebron James
James Harden
Chris Paul
Chris Tucker
Kevin Durant
James Harden
Steve Tucker
Steve Smith
Eric Bledsoe
Steve Caroll
Chris Paul
Sally Smith
Kevin Durant
James Jones
Chris Paul
```

Exercise: Counting names

- Implement a program that . . .
  - Reads in a text file formatted like the example to the right named **names.txt**
  - Notice that some names repeat
  - The program should count how many unique names there are!
  - **Don't use a set or dictionary**

```
Lebron James
James Harden
Chris Paul
Chris Tucker
Kevin Durant
James Harden
Steve Tucker
Steve Smith
Eric Bledsoe
Steve Caroll
Chris Paul
Sally Smith
Kevin Durant
James Jones
Chris Paul
```

Exercise: Counting names

```
names = []
names_file = open('names.txt', 'r')
for line in names_file:
    name = line.strip('\n')
    if name not in names:
        names.append(name)
print(len(names))
```

```
Lebron James
James Harden
Chris Paul
Chris Tucker
Kevin Durant
James Harden
Steve Tucker
Steve Smith
Eric Bledsoe
Steve Caroll
Chris Paul
Sally Smith
Kevin Durant
James Jones
Chris Paul
```

Exercise: Counting names

- Implement a program that . . .
  - Reads in a text file formatted like the example to the right named **names.txt**
  - Notice that some names repeat
  - The program should count how many unique names there are!
  - **Don't use a set or list**

```
Lebron James
James Harden
Chris Paul
Chris Tucker
Kevin Durant
James Harden
Steve Tucker
Steve Smith
Eric Bledsoe
Steve Caroll
Chris Paul
Sally Smith
Kevin Durant
James Jones
Chris Paul
```

Exercise: Counting names

```
names = {}
names_file = open('names.txt', 'r')
for line in names_file:
    name = line.strip('\n')
    names[name] = ''
print(len(names))
```

```
Lebron James
James Harden
Chris Paul
Chris Tucker
Kevin Durant
James Harden
Steve Tucker
Steve Smith
Eric Bledsoe
Steve Caroll
Chris Paul
Sally Smith
Kevin Durant
James Jones
Chris Paul
```