

Welcome to CS103!

Are there “laws of physics”  
in computer science?

# Key Questions in CS103

- What problems can you solve with a computer?
  - ***Computability Theory***
- Why are some problems harder to solve than others?
  - ***Complexity Theory***
- How can we be certain in our answers to these questions?
  - ***Discrete Mathematics***

# The Teaching Team



***Amy Liu***  
(Instructor)



***Benson Kung***  
(TA)



***Annika Mauro***  
(TA)

***Amy's Email: [liuamyj@cs.stanford.edu](mailto:liuamyj@cs.stanford.edu)***

***Staff Email List: [cs103-sum2223-staff@lists.stanford.edu](mailto:cs103-sum2223-staff@lists.stanford.edu)***

# Course Website

**<https://cs103.stanford.edu>**

All course  
content will be  
hosted here,  
except for  
lecture videos.

# Prerequisite / Corequisite

# CS106B

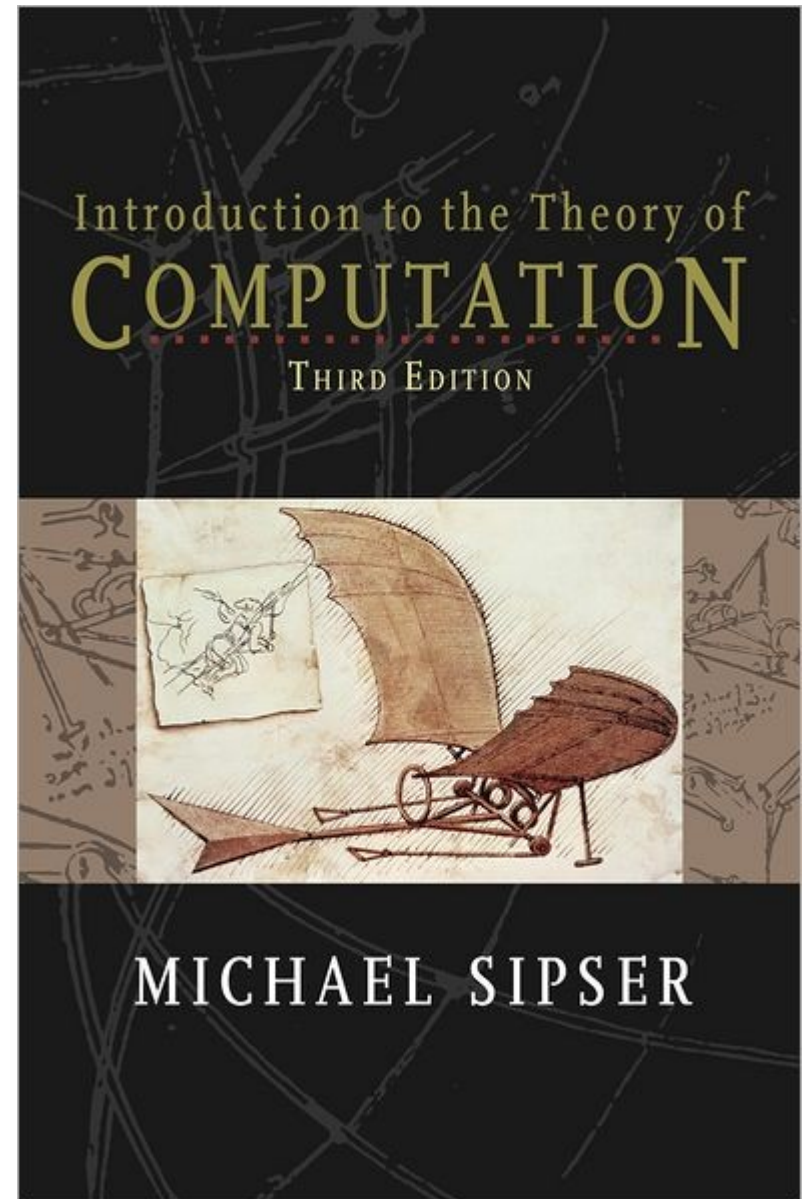
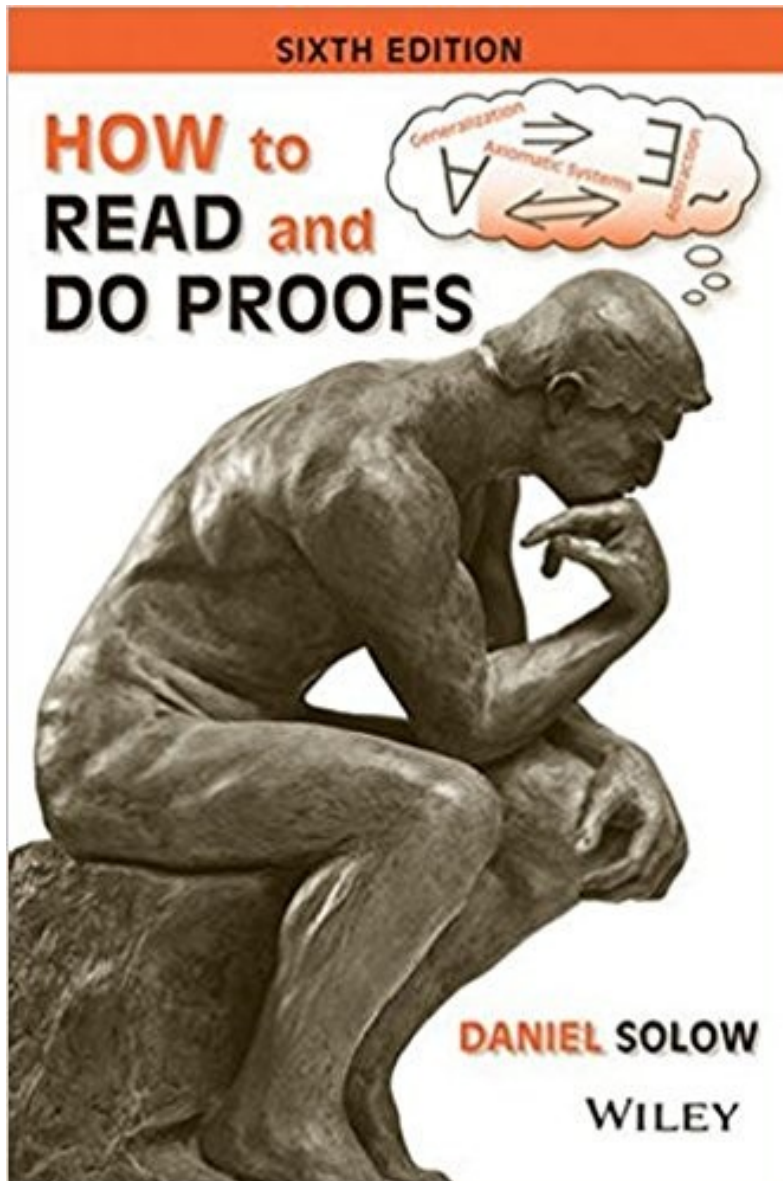
Some problem sets will have small coding components. We'll also reference some concepts from CS106B, particularly recursion, throughout the quarter.

There aren't any math prerequisites for this course - high-school algebra should be enough!

# Problem Set 0

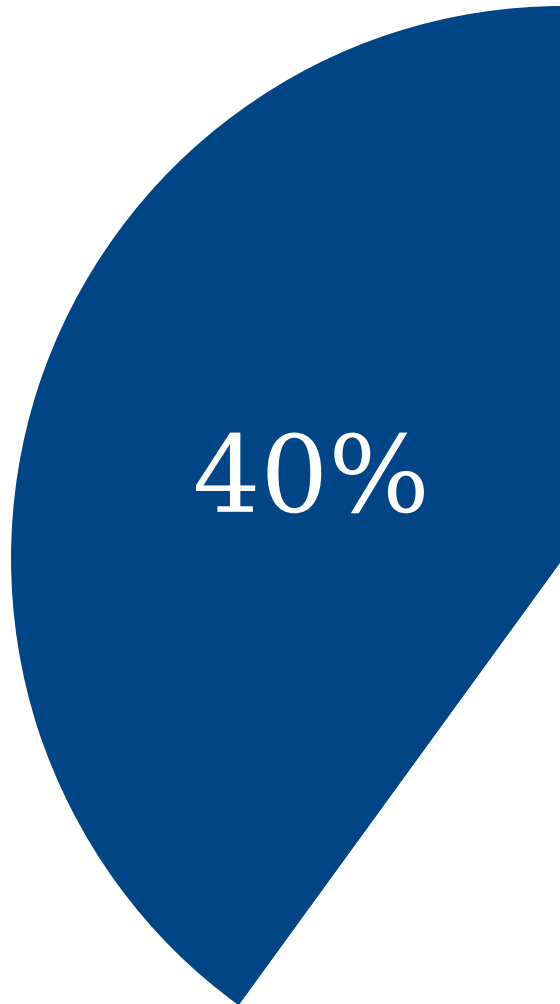
- Your first assignment, Problem Set 0, goes out today. It's due Friday at 4:00PM Pacific.
- This assignment requires you to set up your development environment and to get set up on the various tools and platforms we will use in the course.
- There's no coding involved, but it's good to start early anyway in case you encounter any technical issues setting up.

# *Recommended* Reading



# Grading

# Grading

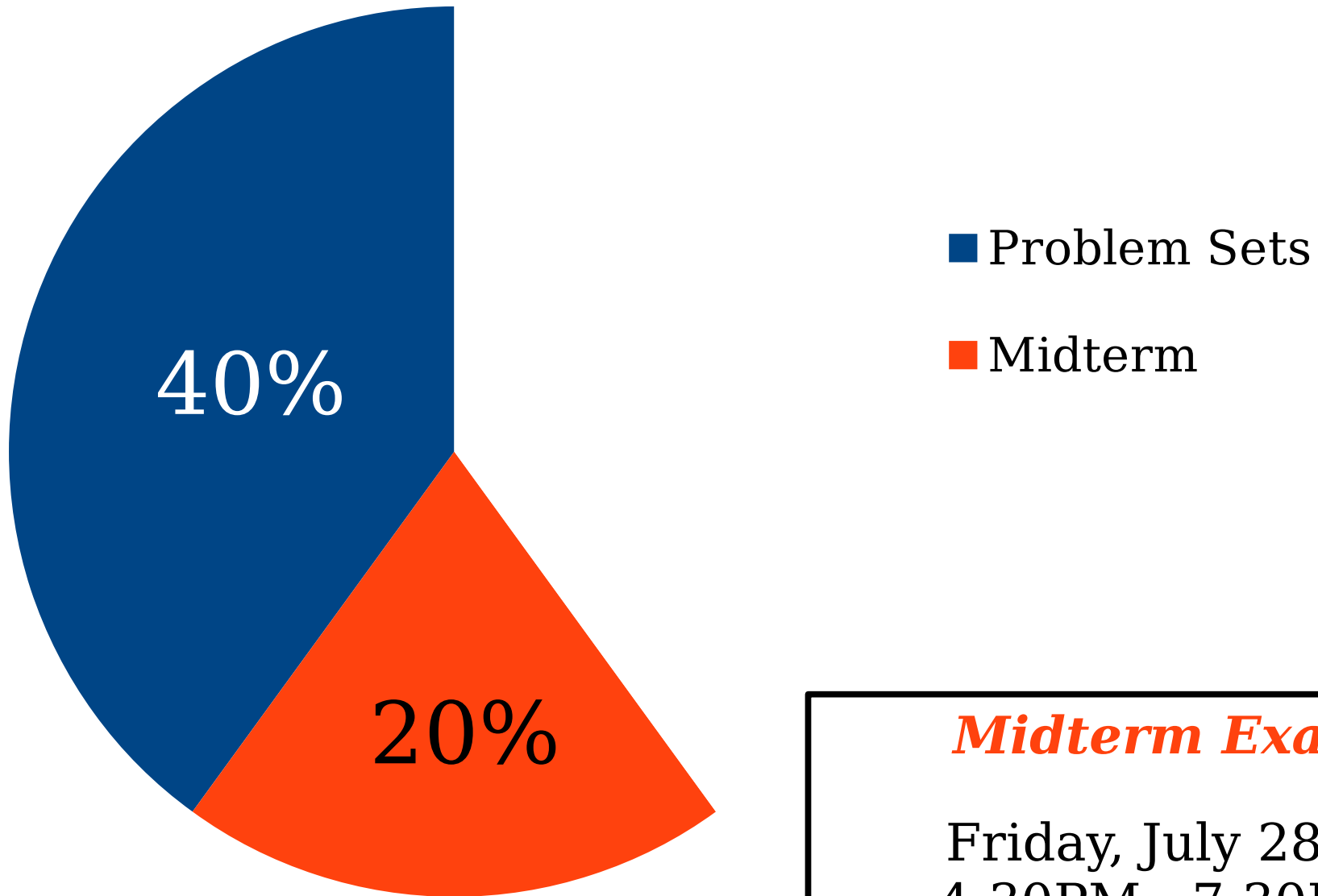


■ Problem Sets

## ***Eight Problem Sets***

Problem sets may be completed individually or in pairs.

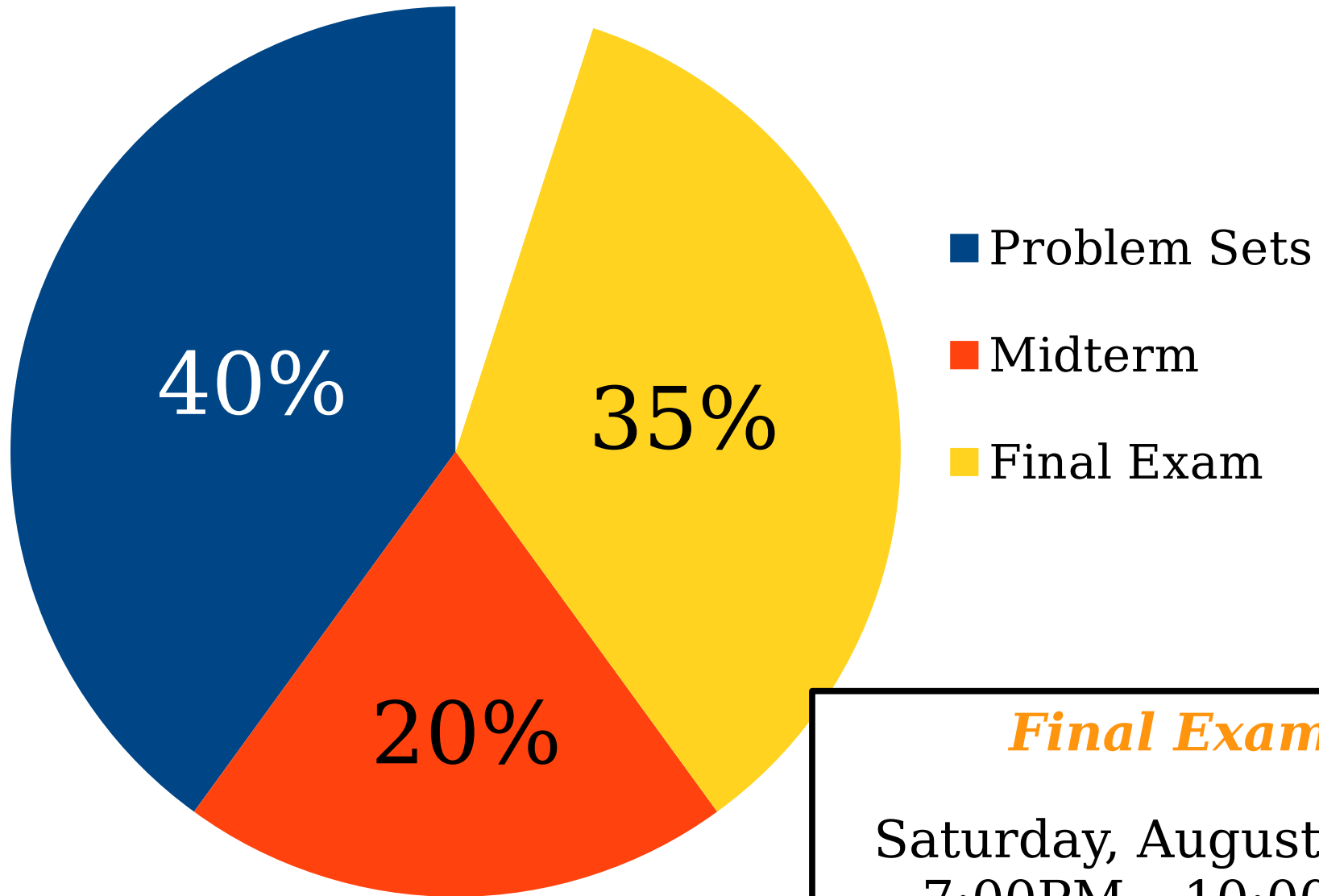
# Grading



## ***Midterm Exam***

Friday, July 28<sup>th</sup>,  
4:30PM - 7:30PM

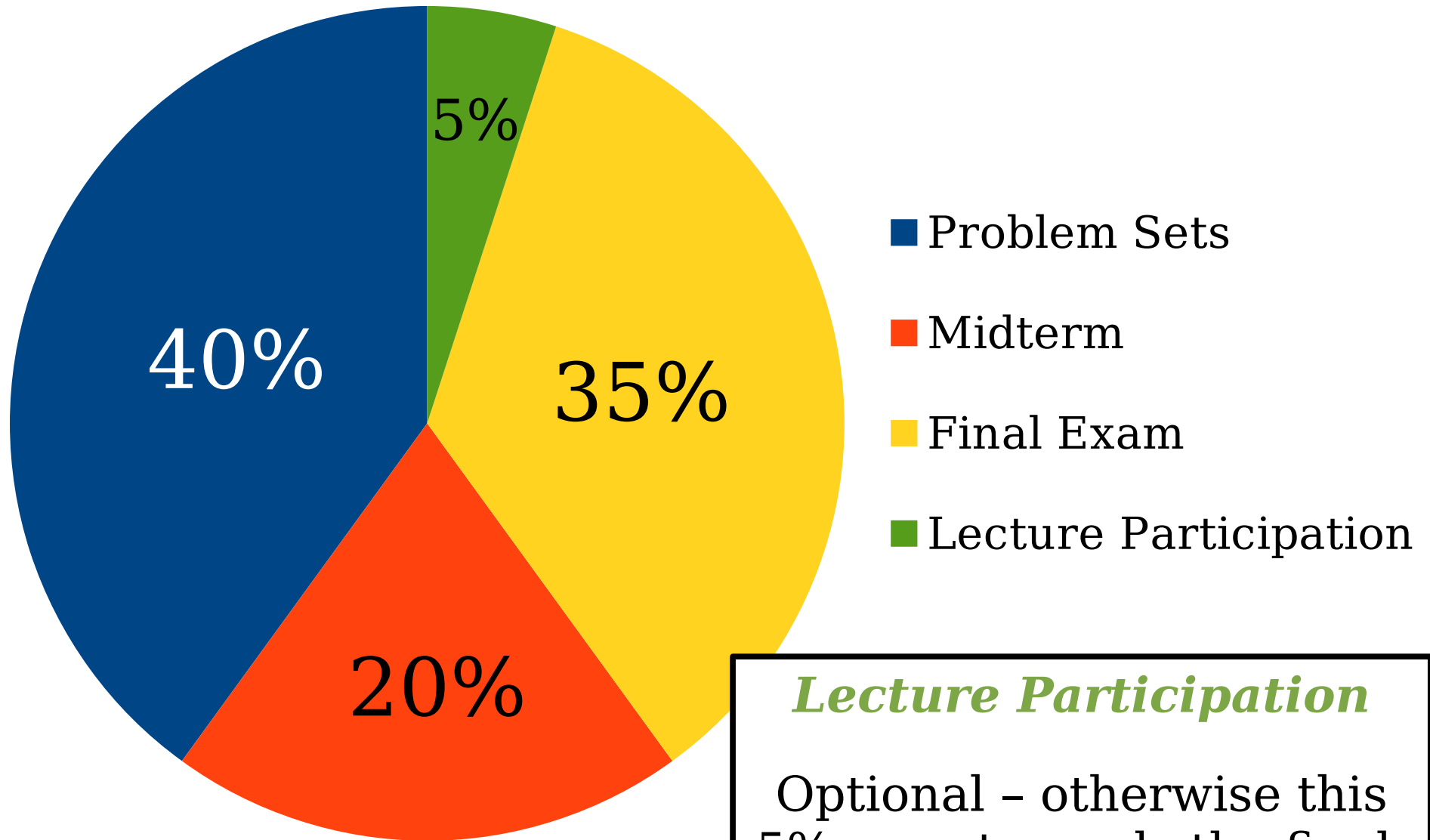
# Grading



***Final Exam***

Saturday, August 19<sup>th</sup>,  
7:00PM - 10:00PM

# Grading



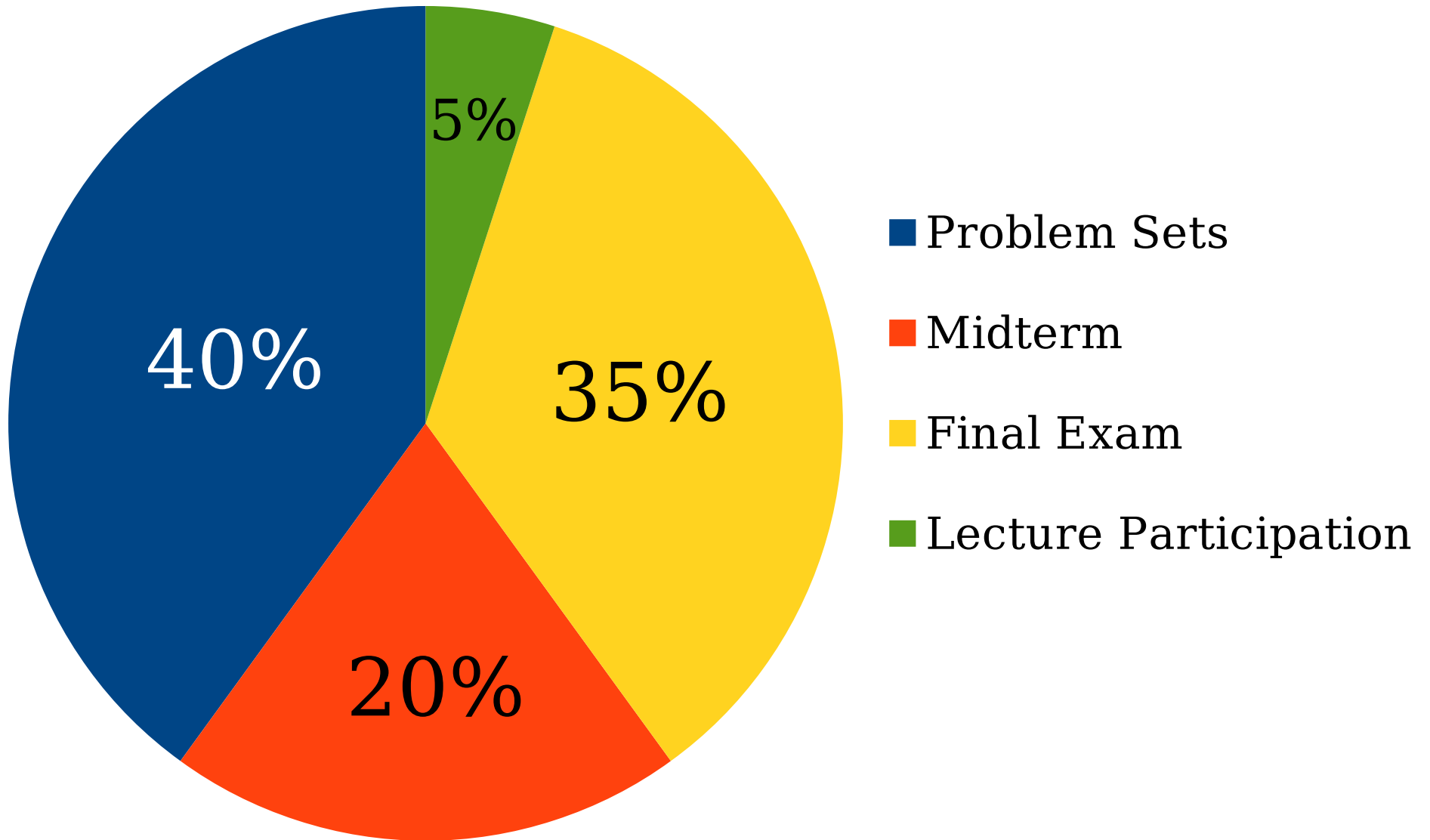
## *Lecture Participation*

Optional - otherwise this 5% goes towards the final.

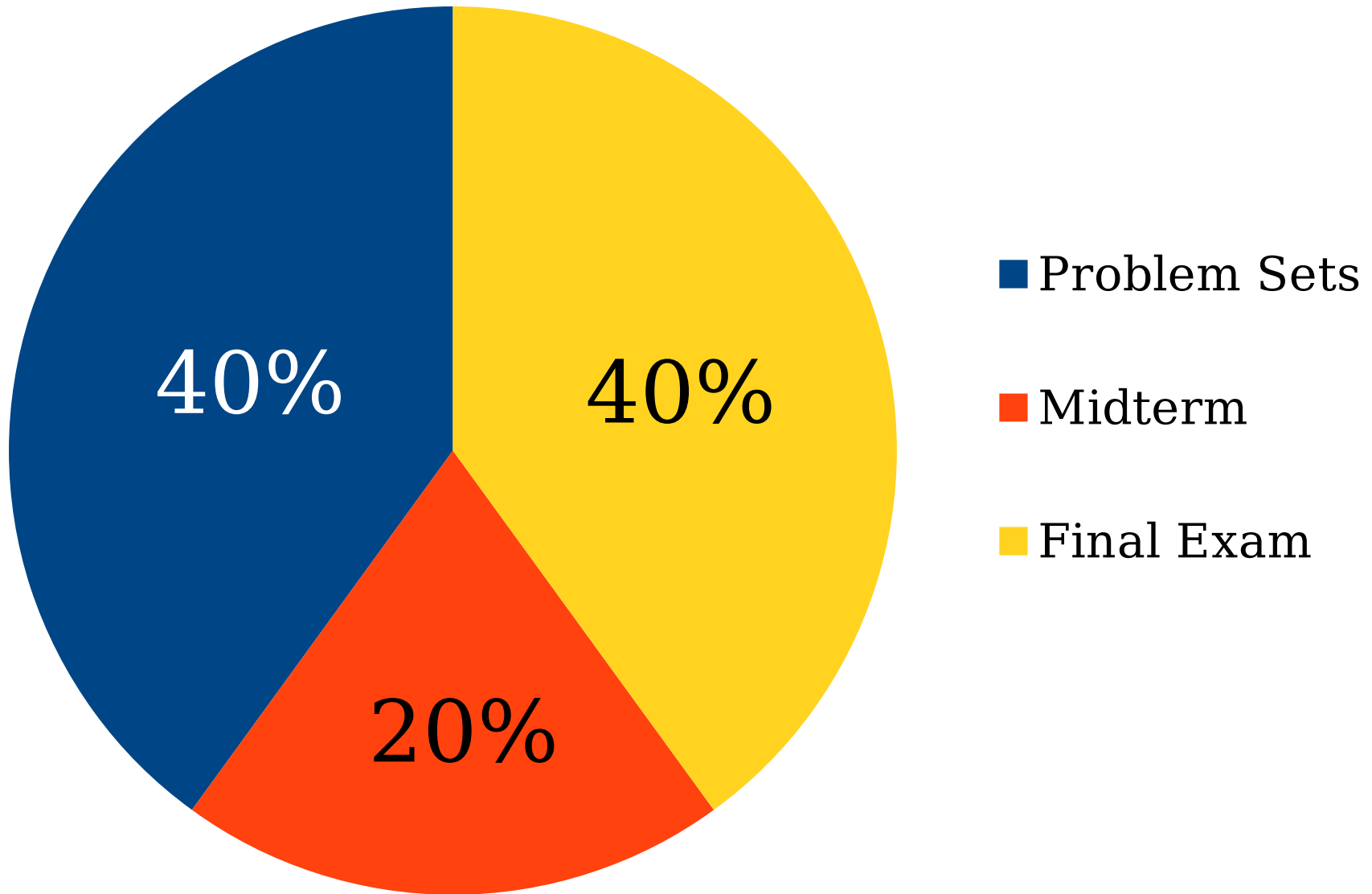
# Lecture Participation

- Lecture attendance/participation is recorded via Poll Everywhere.
- Submit answers to all questions during a lecture (regardless of response correctness) to receive credit.
- 4 excused absences across the quarter.
- For SCPD students taking the course remotely, each lecture will have participation questions that are due by the next lecture.
- Non-SCPD students looking to participate remotely – email [cs103-sum2223-staff@lists.stanford.edu](mailto:cs103-sum2223-staff@lists.stanford.edu) to discuss accommodations/exceptions.

# Grading



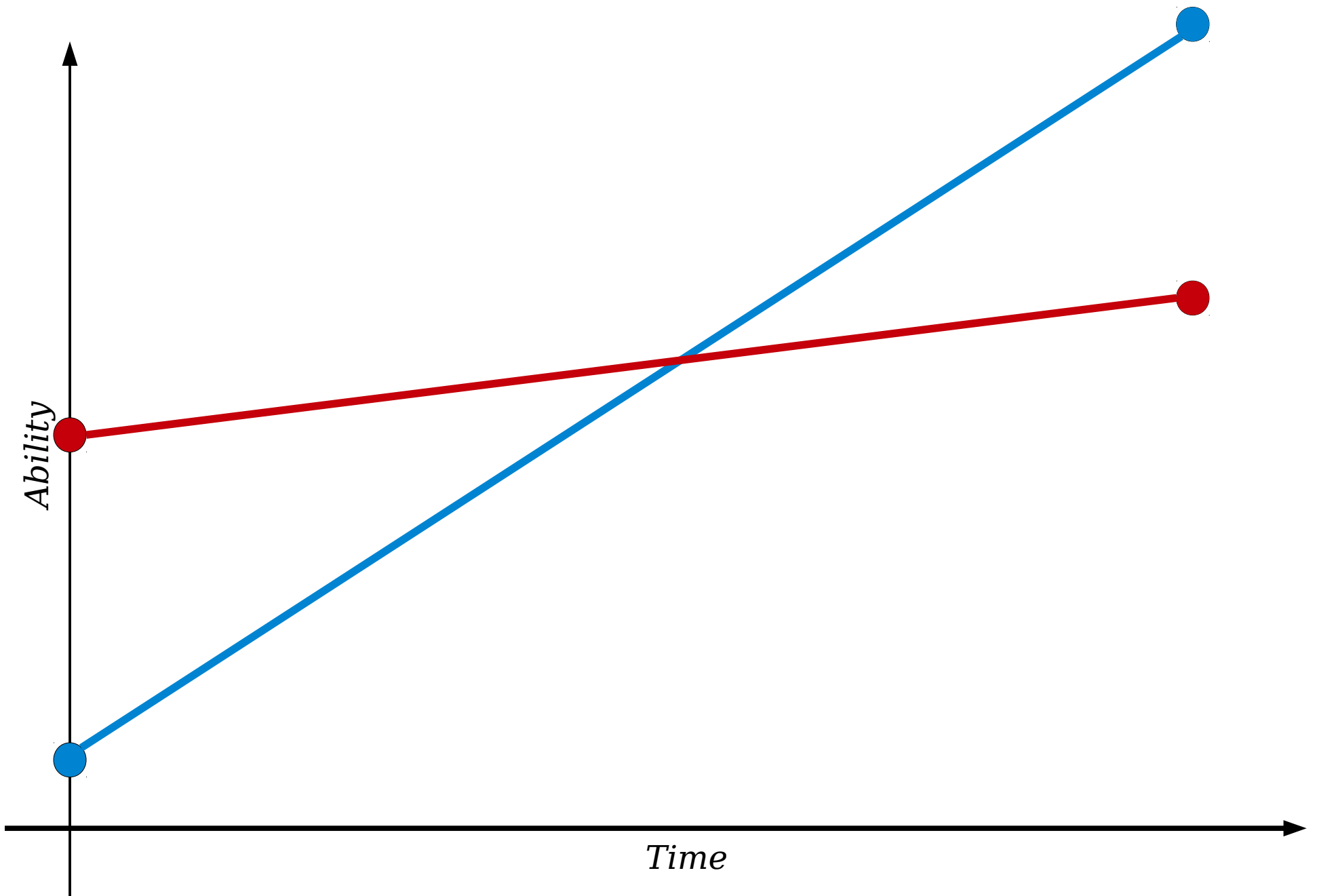
# Alternate Grading Breakdown



# How to Succeed in CS103

# Proof-Based Mathematics

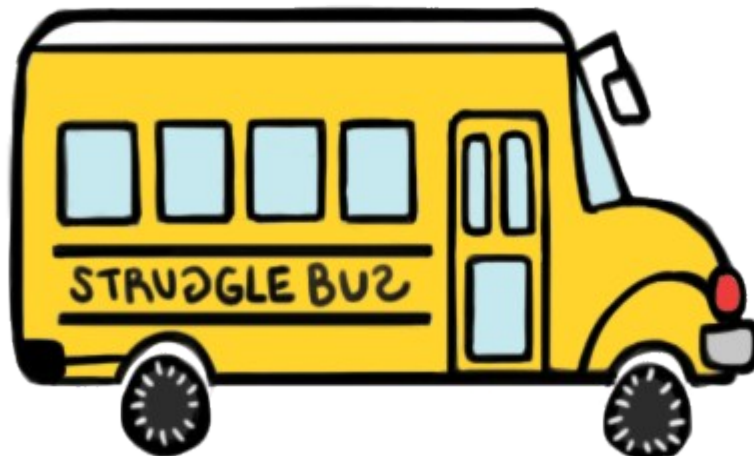
- Most high-school math classes – with the exception of geometry – focus on *calculation*.
- CS103 focuses on *argumentation*.
- Your goal is to *see why things are true*, not *check that they work in a few cases*.
- Be curious! Ask questions. Try things out on your own. You'll learn this material best if you engage with it and refuse to settle for a “good enough” understanding.



“A little slope makes up for a lot of y-intercept.”  
- John Ousterhout

# Don't Psych Yourself Out

- It is ***perfectly normal*** to get stuck or be confused when learning math.
- We've all been on the Struggle Bus. Don't be afraid to ask for help!



# Getting Good at Math

- ***Engage with the concepts.*** Work through lots of practice problems. Play around with new terms and definitions on your own time to see how they work.
- ***Ask for help when you need it.*** We're here to help you. We want you to succeed, so let us know what we can do to help!
- ***Work in groups.*** Get help from the TAs, your problem set partner, and other students.

We've got a big journey ahead of us.

***Let's get started!***

# Introduction to Set Theory

“CS103 students”

“Cool people”

“The chemical elements”

“Cute animals”

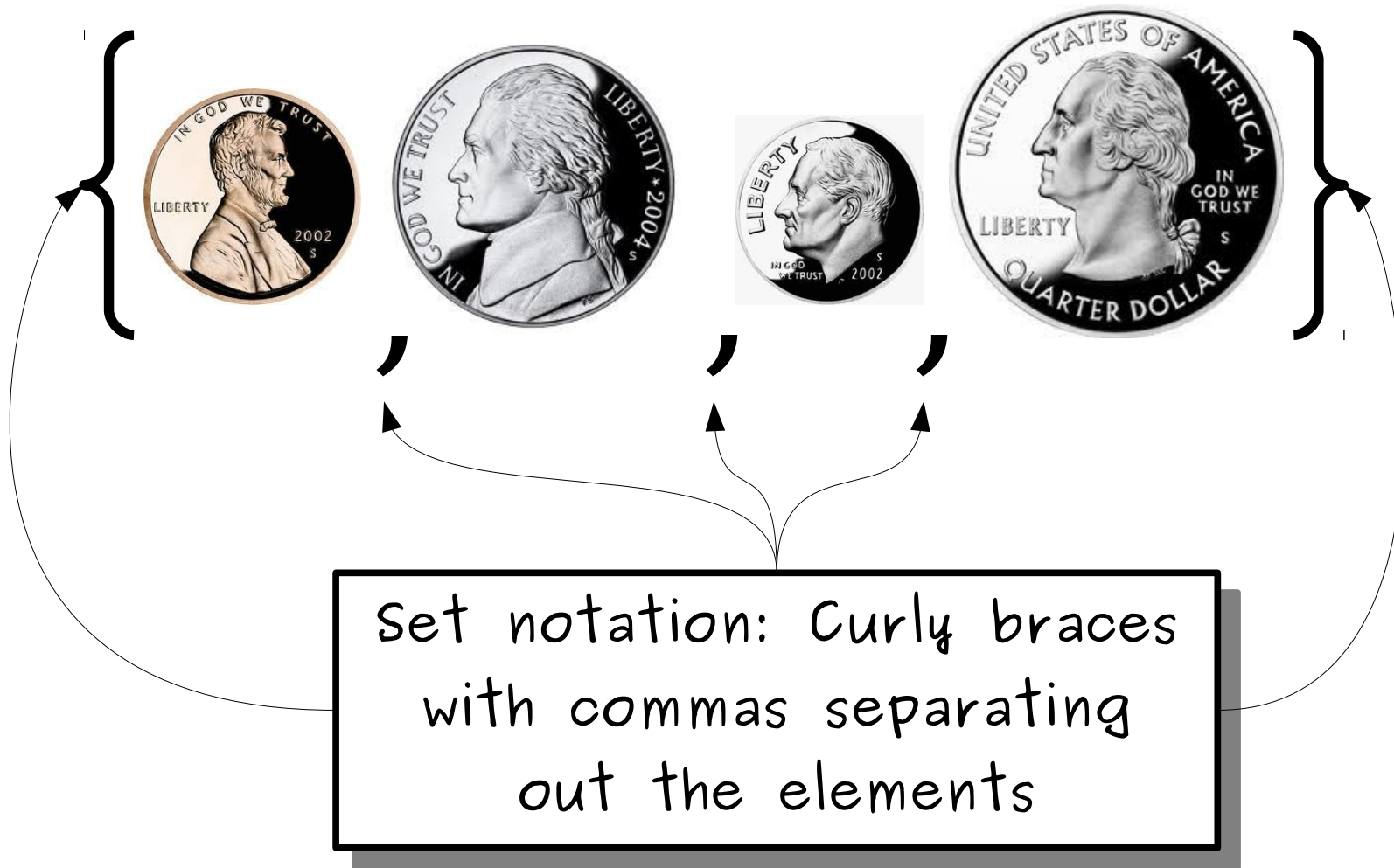
“US coins”

A ***set*** is an unordered collection of distinct objects, which may be anything, including other sets.

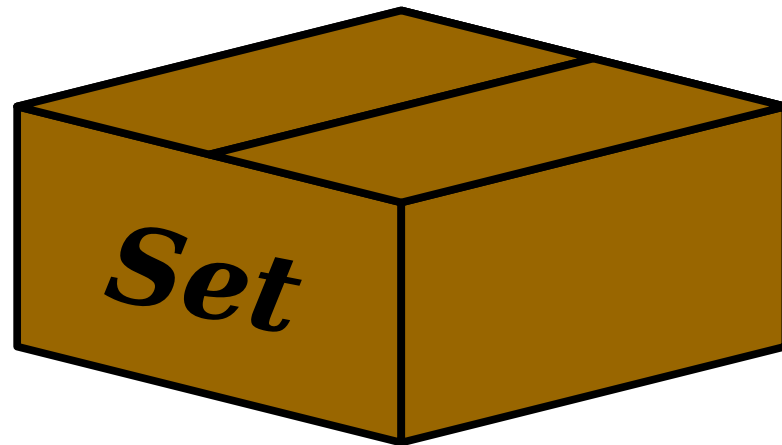


---

A **set** is an unordered collection of distinct objects, which may be anything, including other sets.

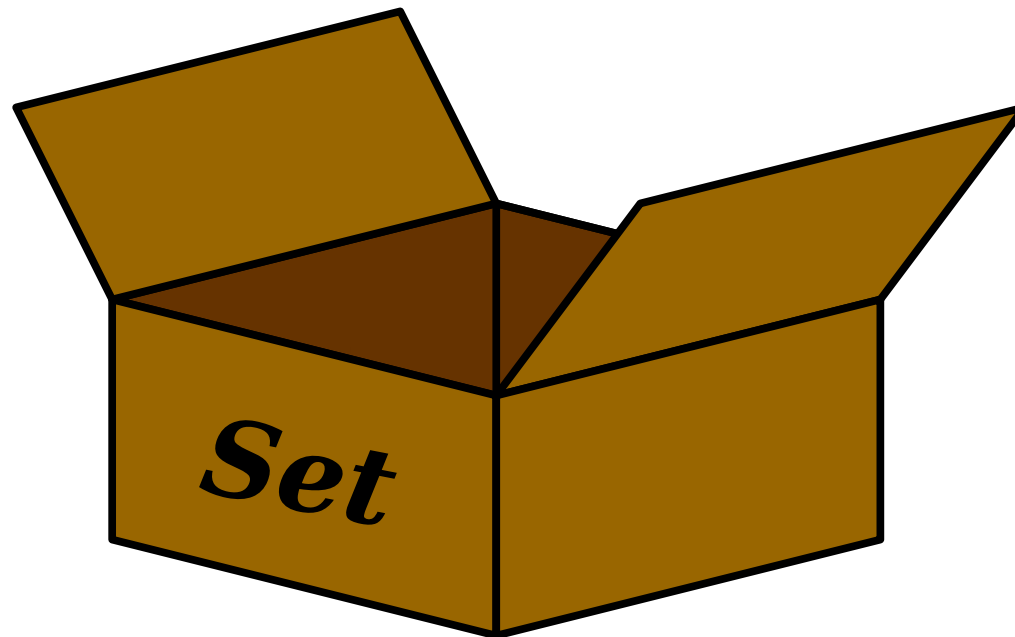


A **set** is an unordered collection of distinct objects, which may be anything, including other sets.



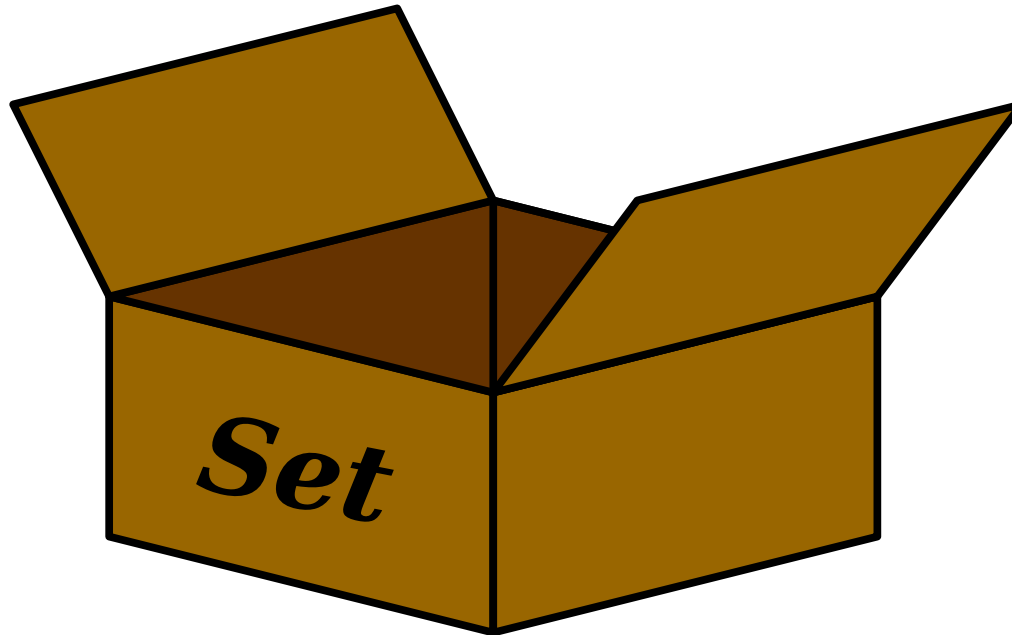
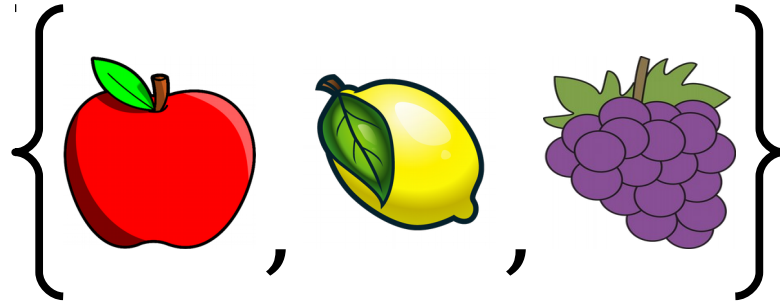
---

Two sets are equal when they have the same contents, ignoring order.



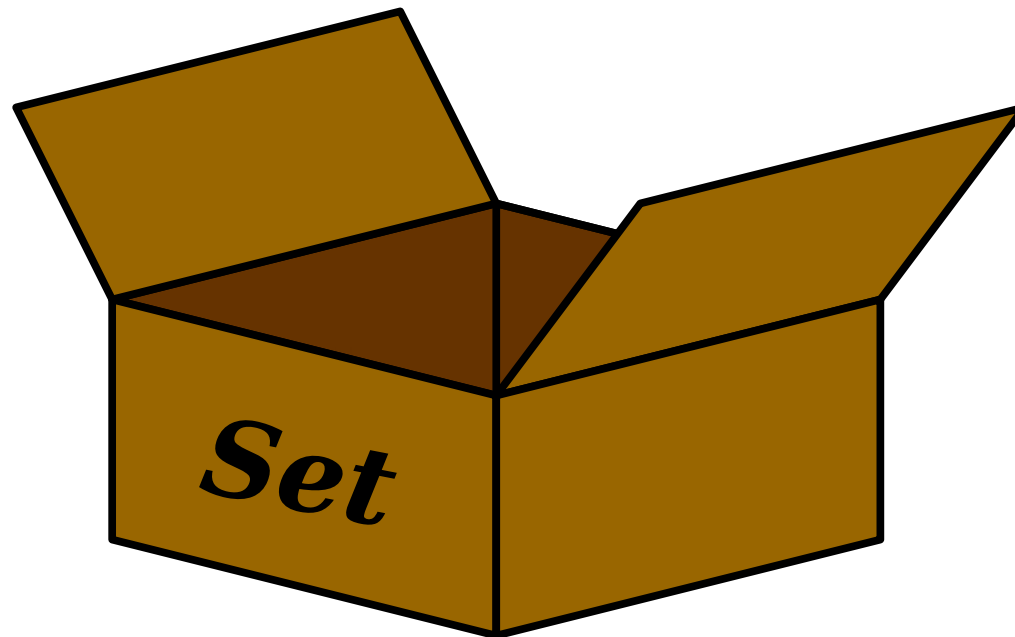
---

Two sets are equal when they have the same contents, ignoring order.



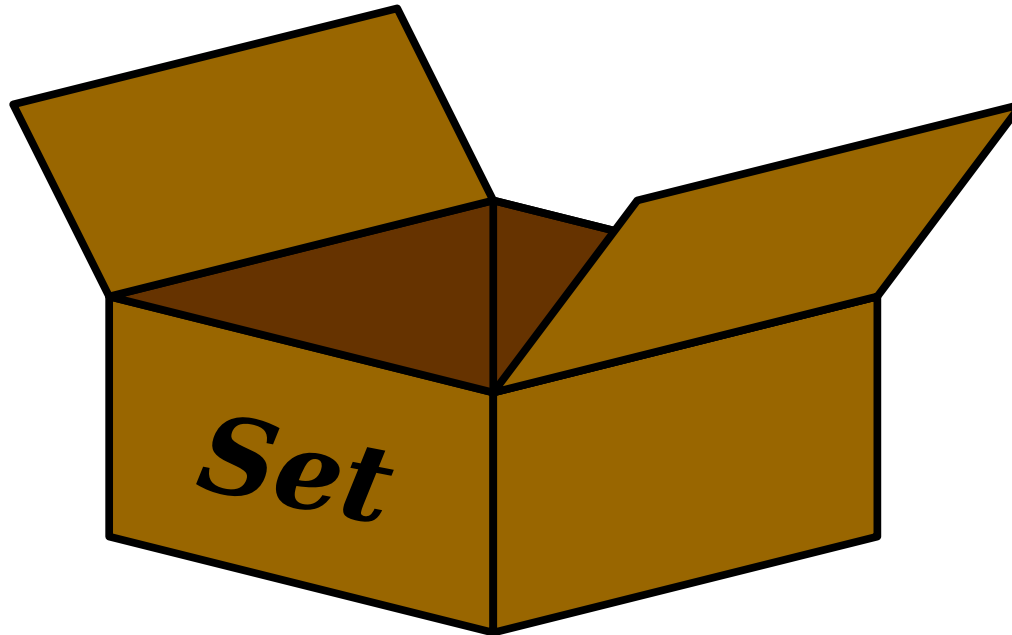
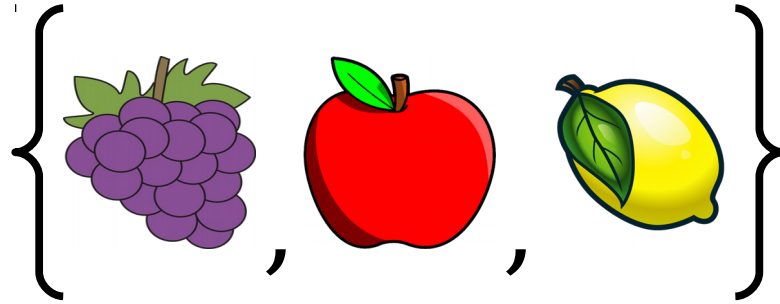
---

Two sets are equal when they have the same contents, ignoring order.



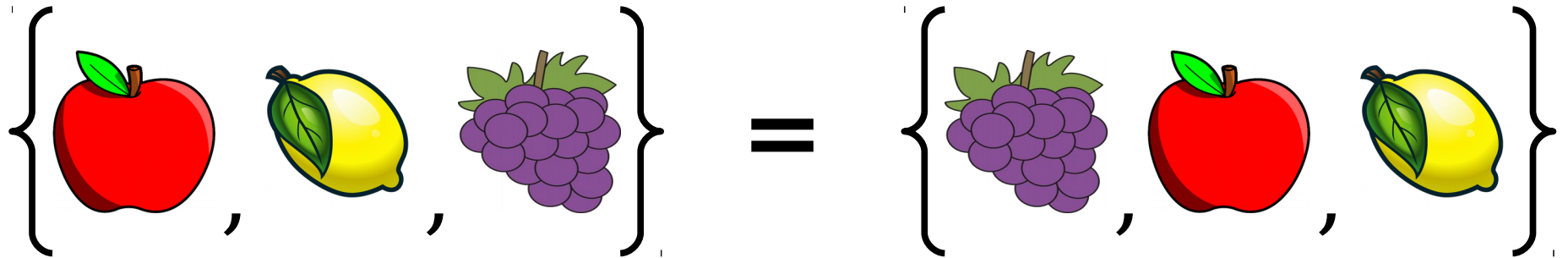
---

Two sets are equal when they have the same contents, ignoring order.



---

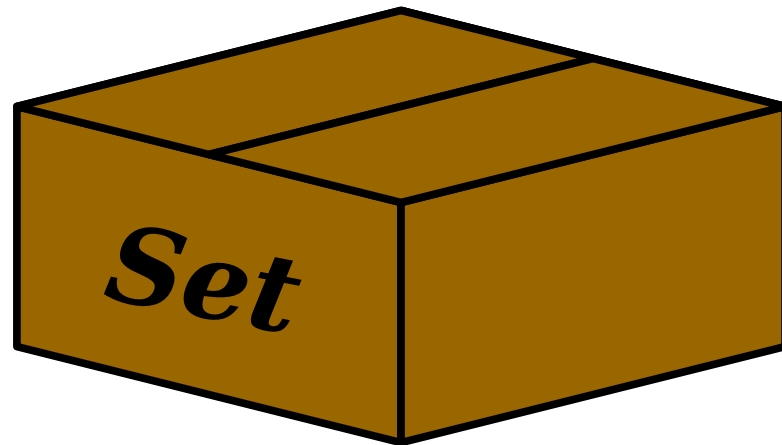
Two sets are equal when they have the same contents, ignoring order.



These are two different descriptions of exactly the same set.

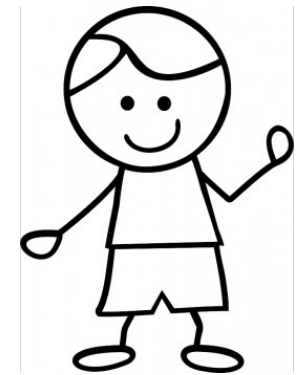
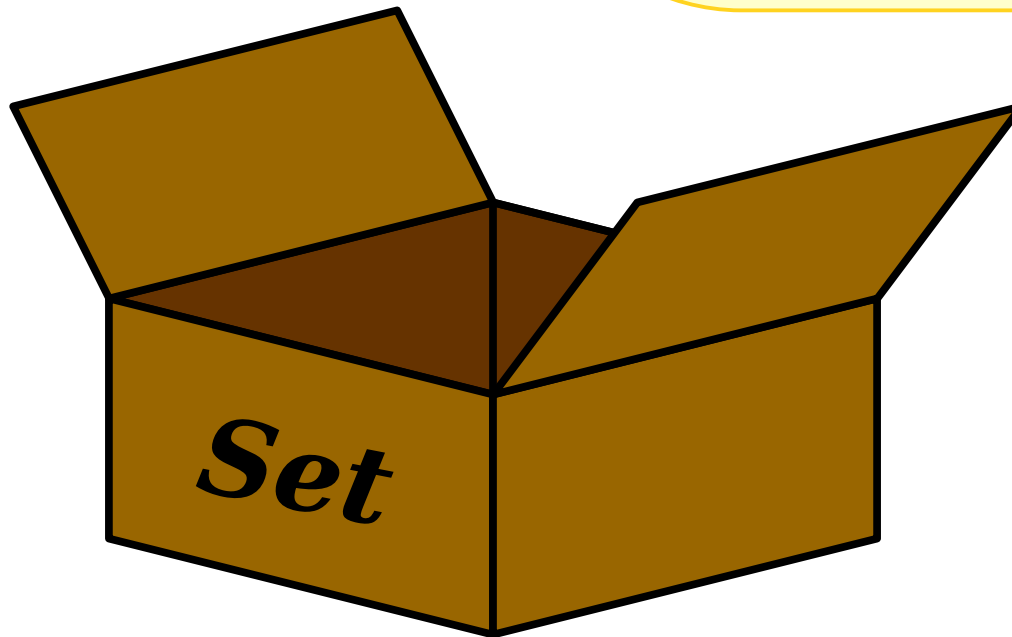
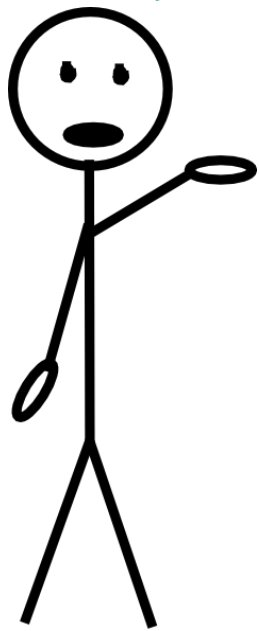
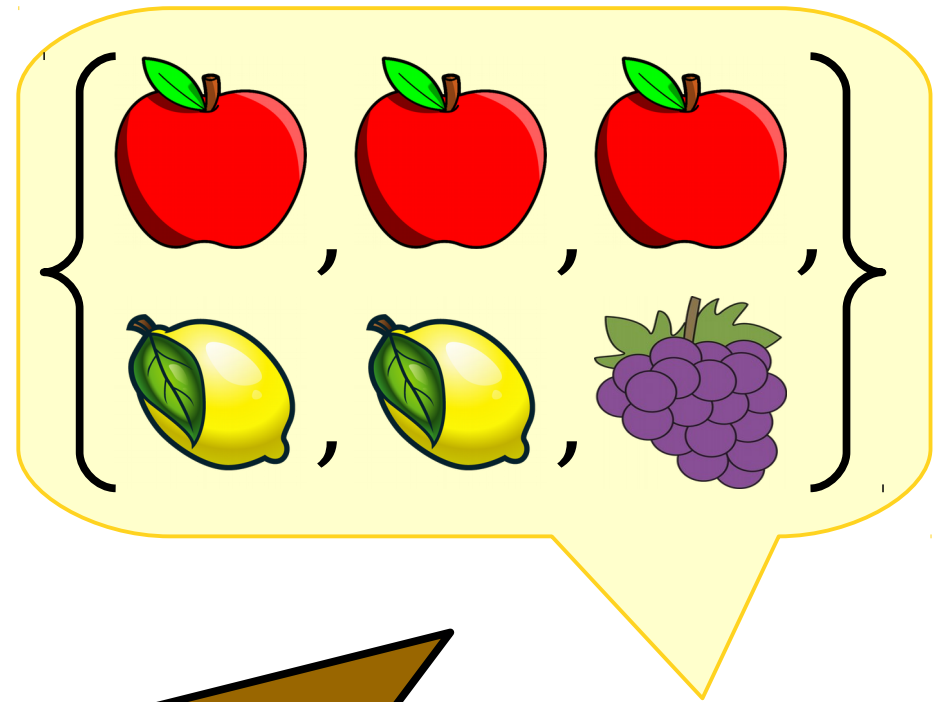
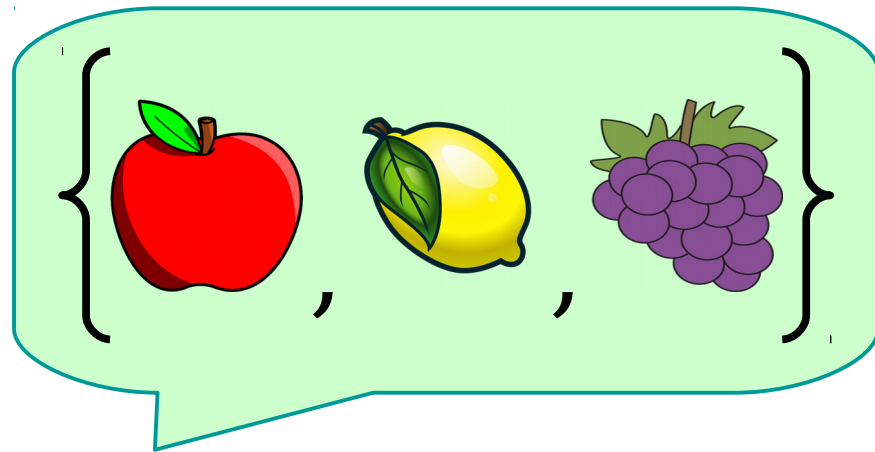
---

Two sets are equal when they have the same contents, ignoring order.

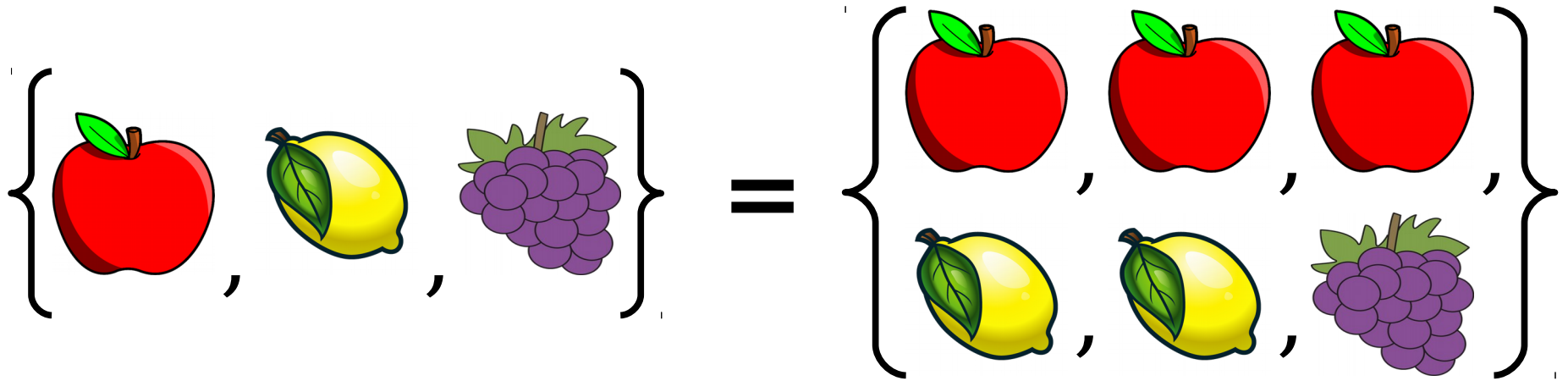


---

Sets cannot contain duplicate elements.  
Any repeated elements are ignored.



Sets cannot contain duplicate elements.  
Any repeated elements are ignored.



These are two different descriptions of exactly the same set.

Sets cannot contain duplicate elements.  
Any repeated elements are ignored.

---

The objects that make up a set are called the ***elements*** of that set.



---

The objects that make up a set are called the ***elements*** of that set.



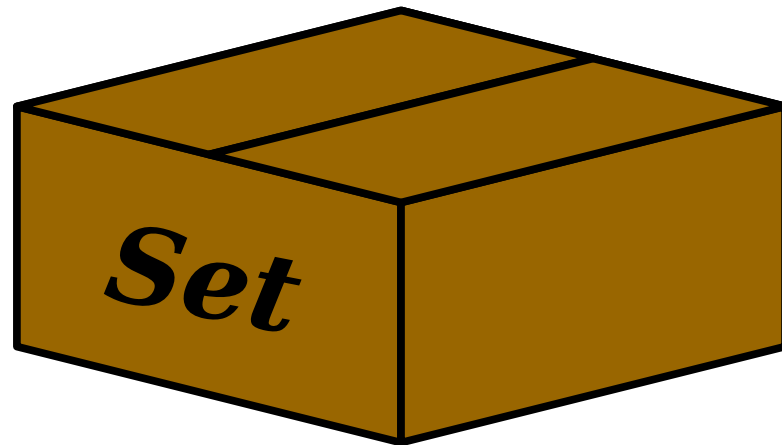
This symbol means  
"is an element of."

The objects that make up a set are called the  
***elements*** of that set.



This symbol means  
"is not an element  
of."

The objects that make up a set are called the **elements** of that set.



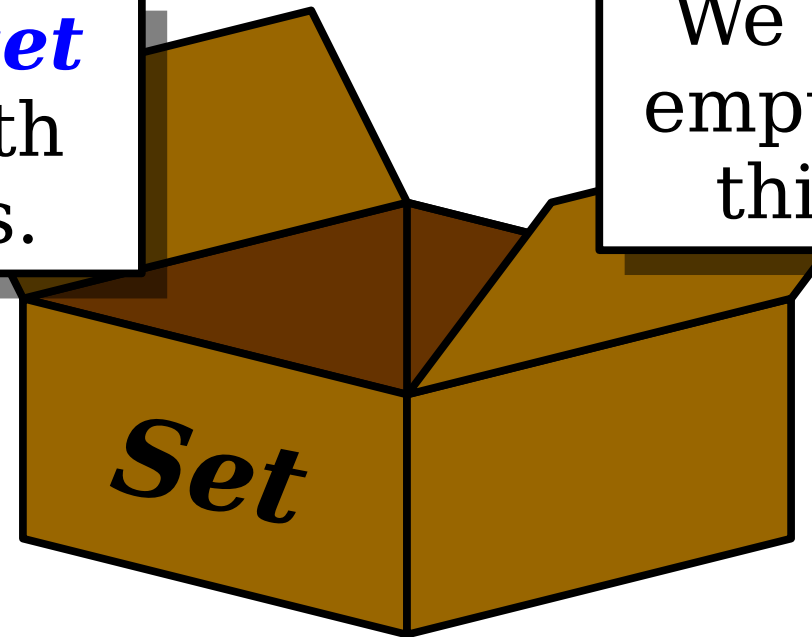
---

Sets can contain any number of elements.

$$\{\} = \emptyset$$

The *empty set* is the set with no elements.

We denote the empty set using this symbol.



---

Sets can contain any number of elements.

**1**  $\stackrel{?}{=}$  **{ 1 }**

***Question:*** Are these two objects equal?

***Respond at [pollev.com/cs103](https://pollev.com/cs103)***

*(Enter your name as it appears in Axxess in order to count towards your participation)*

$$1 \stackrel{?}{=} \{1\}$$

---

**Question:** Are these objects equal?

1

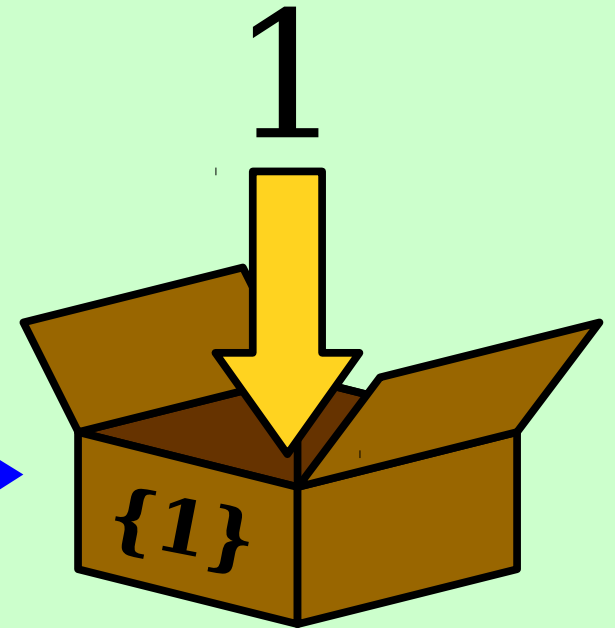
?

{ 1 }

1

This is a number.

This is a set.  
It contains a number.



**Question:** Are these objects equal?

1

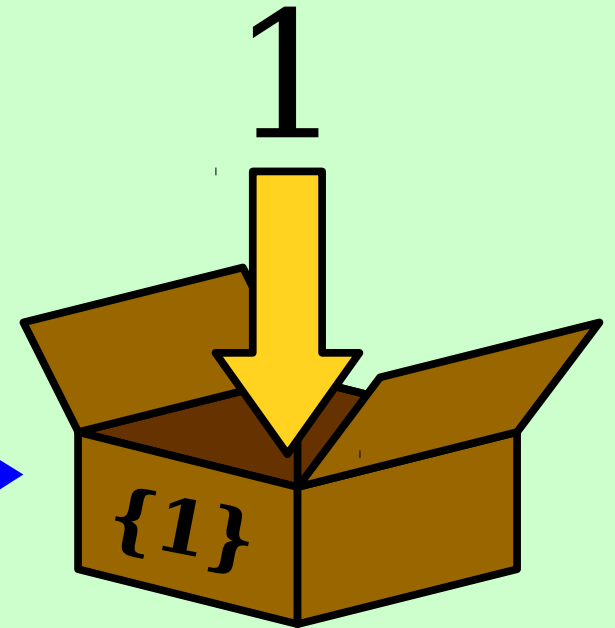
$\neq$

{ 1 }

1

This is a number.

This is a set.  
It contains a number.



**Question:** Are these objects equal?

$$\emptyset \stackrel{?}{=} \{\emptyset\}$$

**Question:** Are these two objects equal?

**Respond at [pollev.com/cs103](https://pollev.com/cs103)**

*(Enter your name as it appears in Axess in order to count towards your participation)*

$\emptyset$  $\stackrel{?}{=}$  $\{\emptyset\}$ 

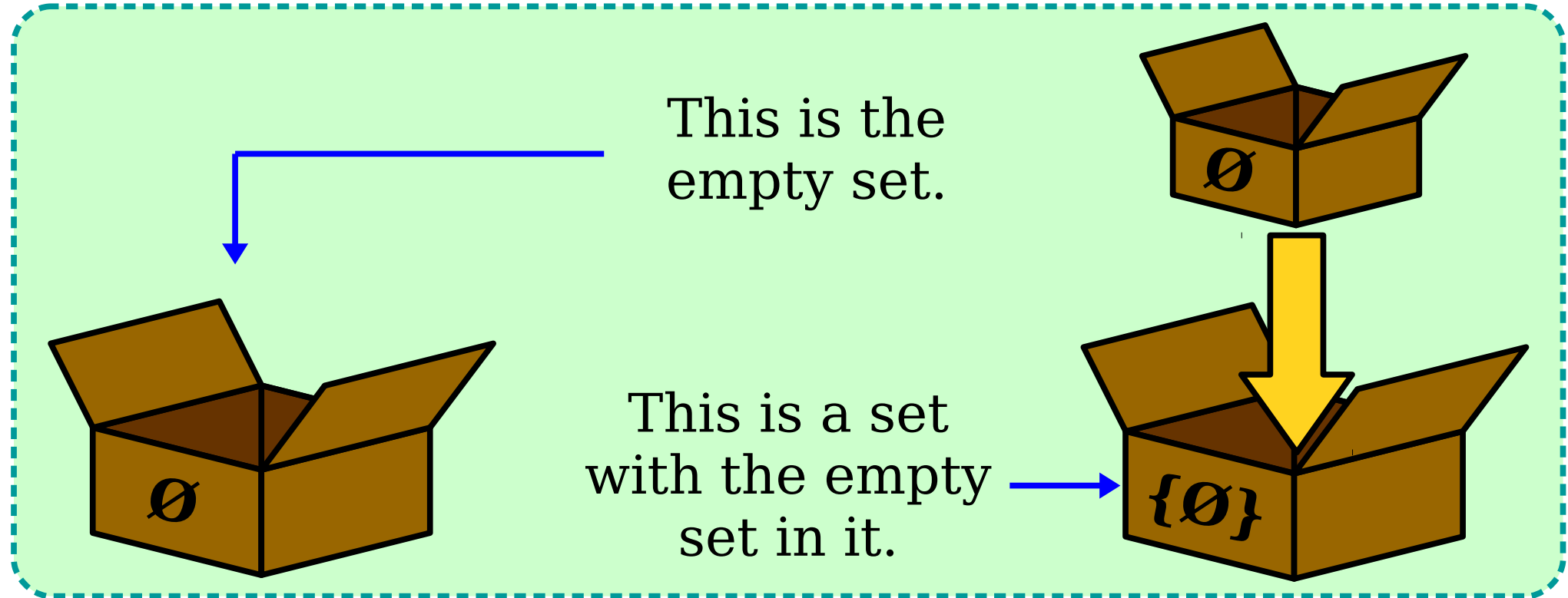
---

**Question:** Are these objects equal?

$\emptyset$

?

$\{\emptyset\}$

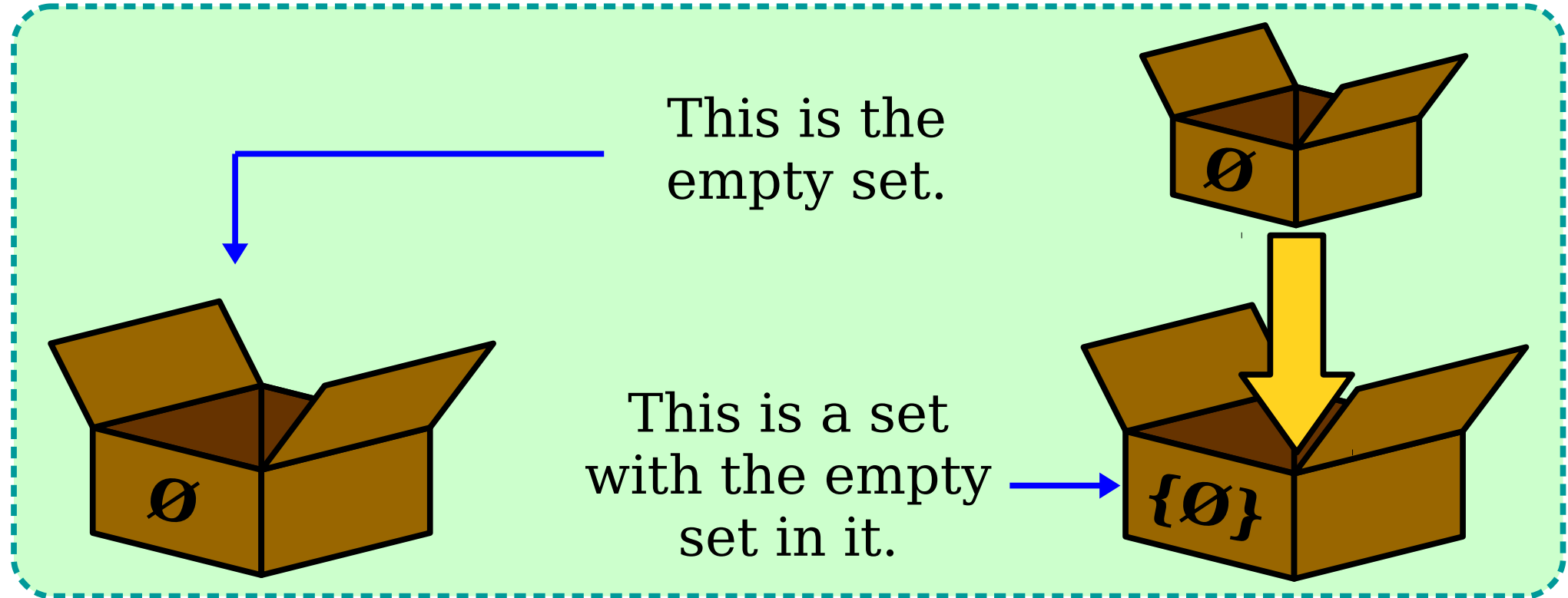


**Question:** Are these objects equal?

$\emptyset$

$\neq$

$\{\emptyset\}$



**Question:** Are these objects equal?

$x$

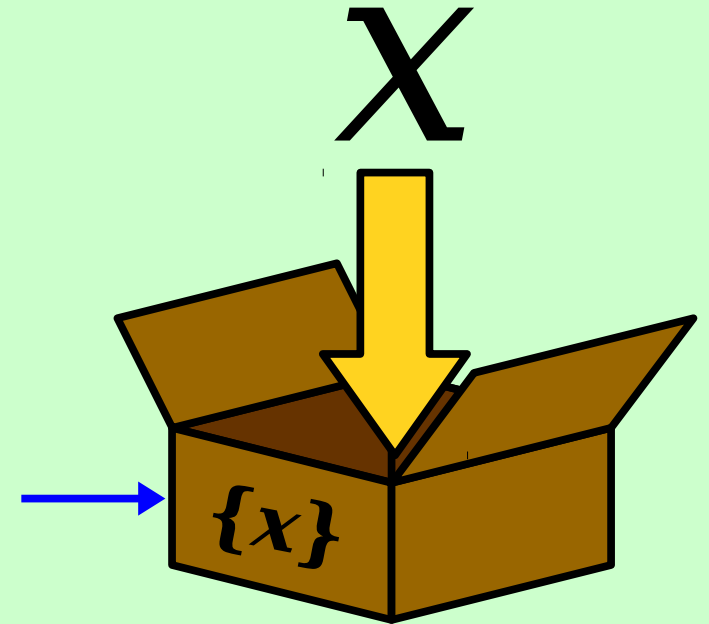
$\neq$

$\{x\}$

$x$

This is  $x$   
itself.

This is a box  
that has  $x$   
inside it.



No object  $x$  is equal to the set containing  $x$ .

# Infinite Sets

- Some sets contain *infinitely many* elements!
- The set  $\mathbb{N} = \{ 0, 1, 2, 3, \dots \}$  is the set of all the ***natural numbers***.
  - Some mathematicians don't include zero; in this class, assume that 0 is a natural number.
- The set  $\mathbb{Z} = \{ \dots, -2, -1, 0, 1, 2, \dots \}$  is the set of all the ***integers***.
  - Z is from German “Zahlen.”
- The set  $\mathbb{R}$  is the set of all ***real numbers***.
  - $e \in \mathbb{R}$ ,  $\pi \in \mathbb{R}$ ,  $4 \in \mathbb{R}$ , etc.

# Describing Complex Sets

- Here are some English descriptions of infinite sets:
  - “The set of all even natural numbers.”
  - “The set of all real numbers less than 137.”
  - “The set of all negative integers.”
- To describe complex sets like these mathematically, we'll use ***set-builder notation***.

# Even Natural Numbers

$\{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$

# Even Natural Numbers

$\{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$

# Even Natural Numbers

$\{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$

The set of all  $n$



# Even Natural Numbers

$$\{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$$

The set of all  $n$

where

# Even Natural Numbers

$\{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$

The set of all  $n$

where

$n$  is a natural  
number

# Even Natural Numbers

$\{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$

The set of all  $n$

where

$n$  is a natural  
number

and  $n$  is even

# Even Natural Numbers

$\{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$

The set of all  $n$

where

$n$  is a natural  
number

and  $n$  is even

$\{ 0, 2, 4, 6, 8, 10, 12, 14, 16, \dots \}$

# Set Builder Notation

- A set may be specified in ***set-builder notation***:

$\{ x \mid \text{some property } x \text{ satisfies} \}$

$\{ x \in S \mid \text{some property } x \text{ satisfies} \}$

- For example:

$\{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$

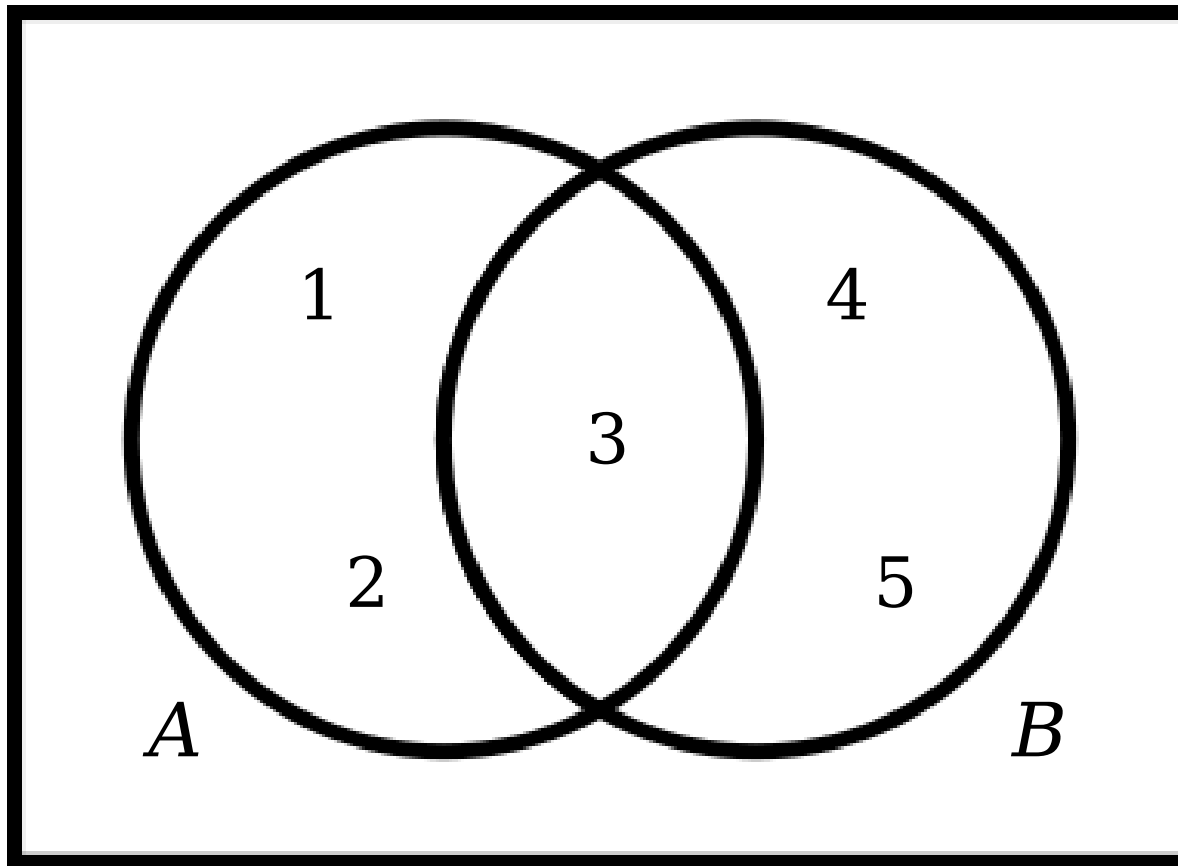
$\{ C \mid C \text{ is a set of US currency} \}$

$\{ r \in \mathbb{R} \mid r < 3 \}$

$\{ n \in \mathbb{N} \mid n < 3 \}$  **(the set  $\{0, 1, 2\}$ )**

# Combining Sets

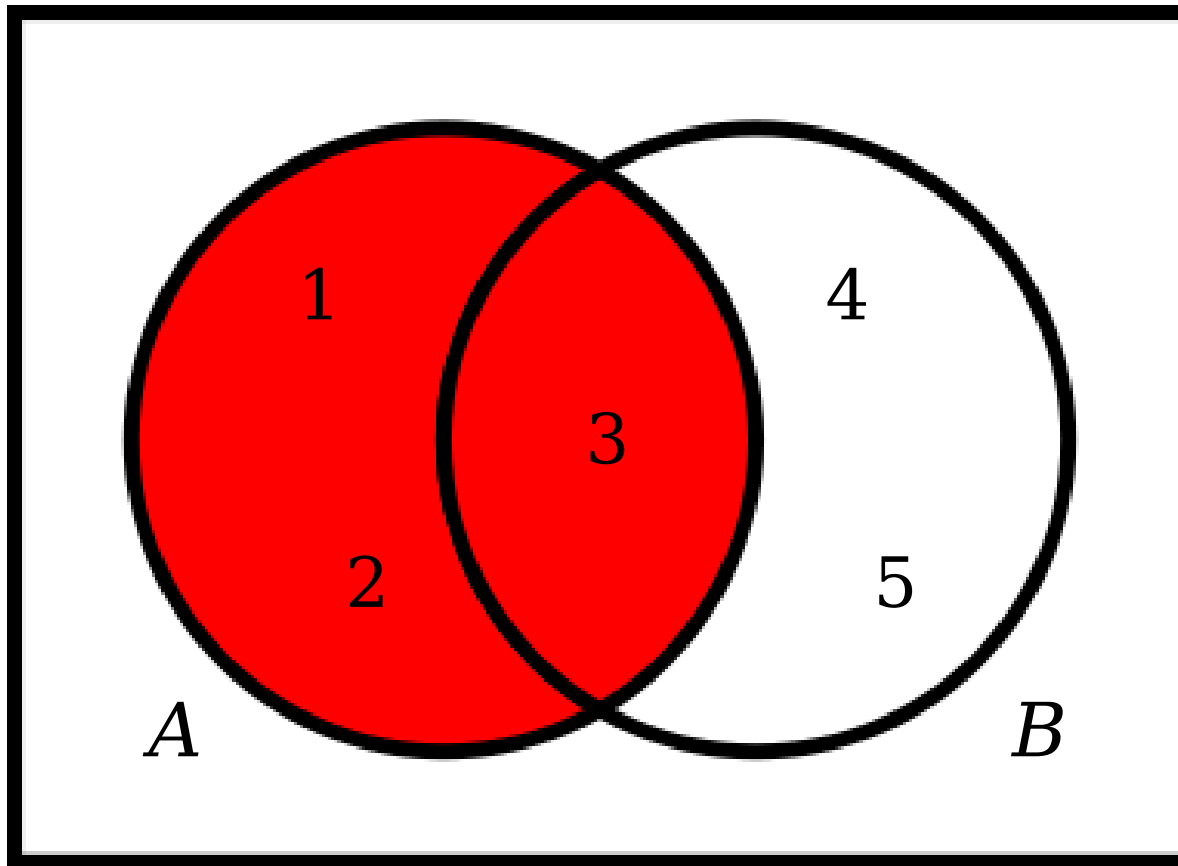
# Venn Diagrams



$$A = \{ 1, 2, 3 \}$$

$$B = \{ 3, 4, 5 \}$$

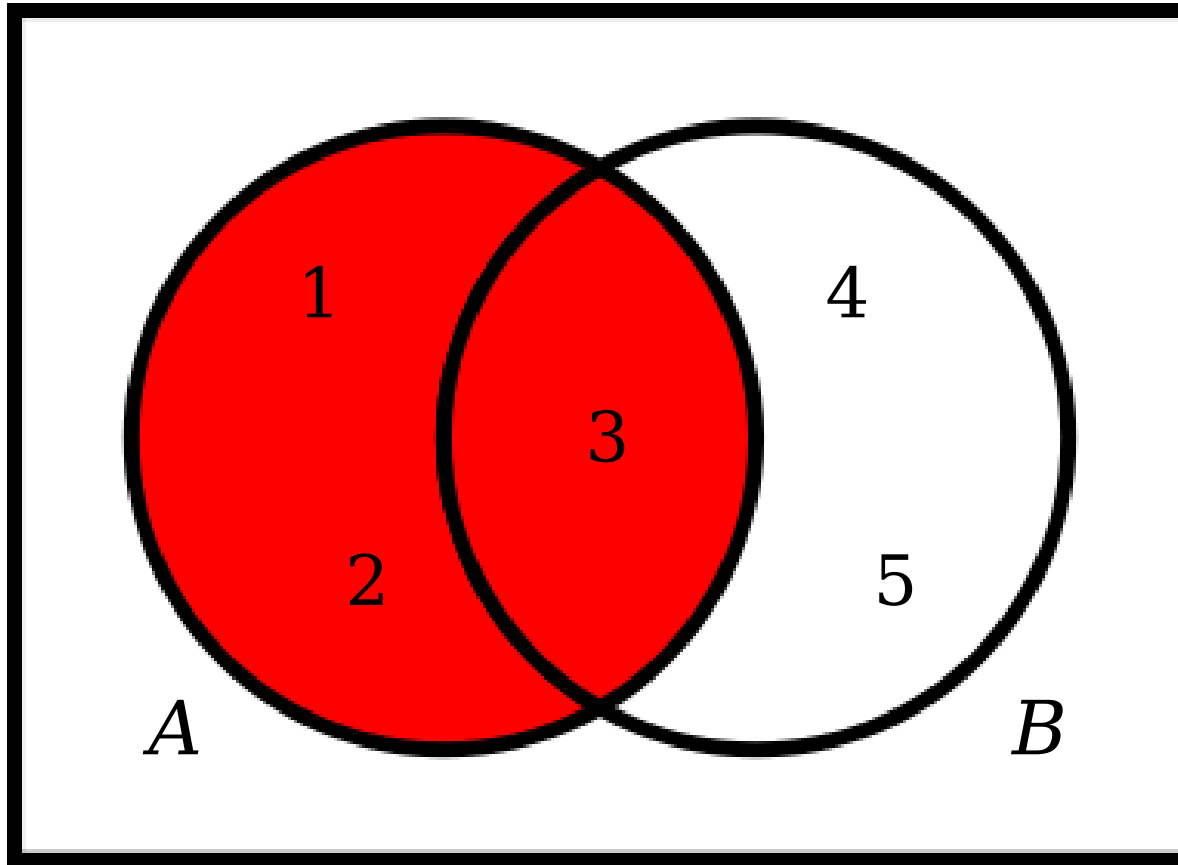
# Venn Diagrams



$$A = \{ 1, 2, 3 \}$$

$$B = \{ 3, 4, 5 \}$$

# Venn Diagrams

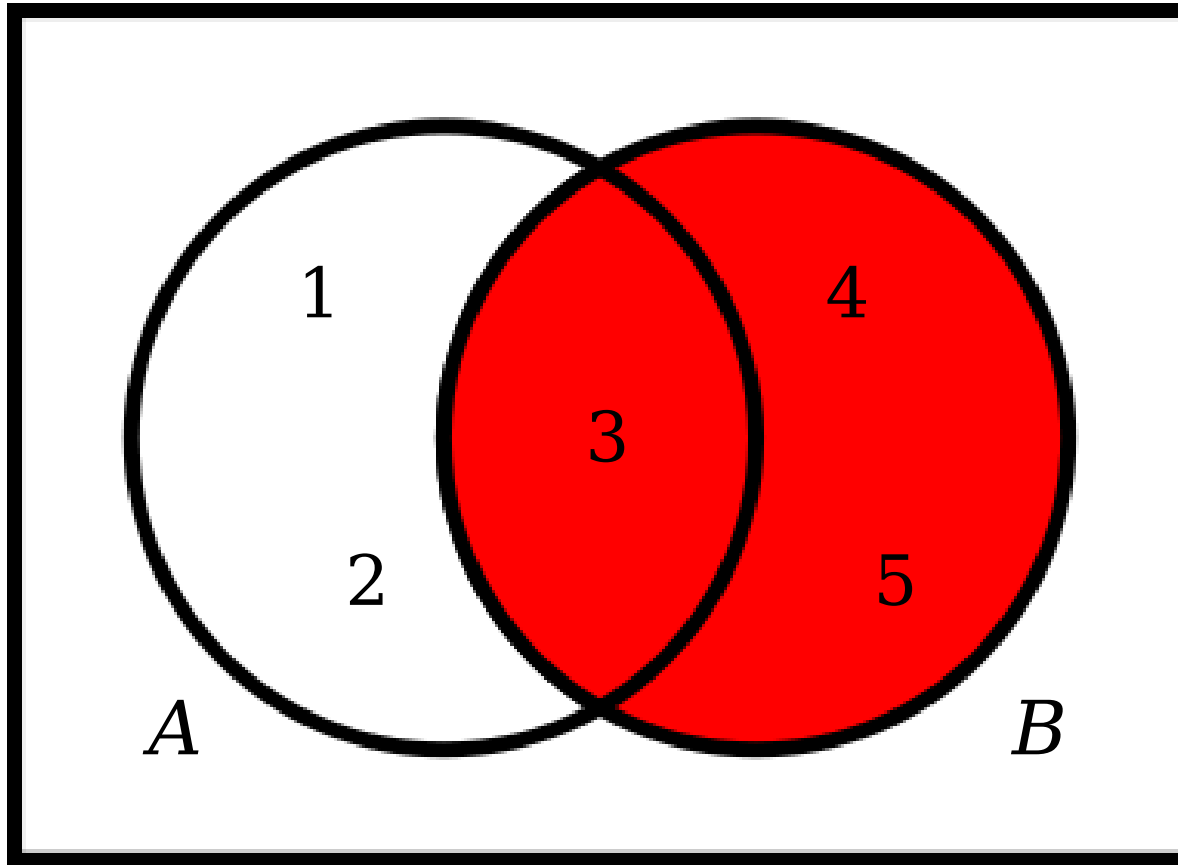


*A*

$$A = \{ 1, 2, 3 \}$$

$$B = \{ 3, 4, 5 \}$$

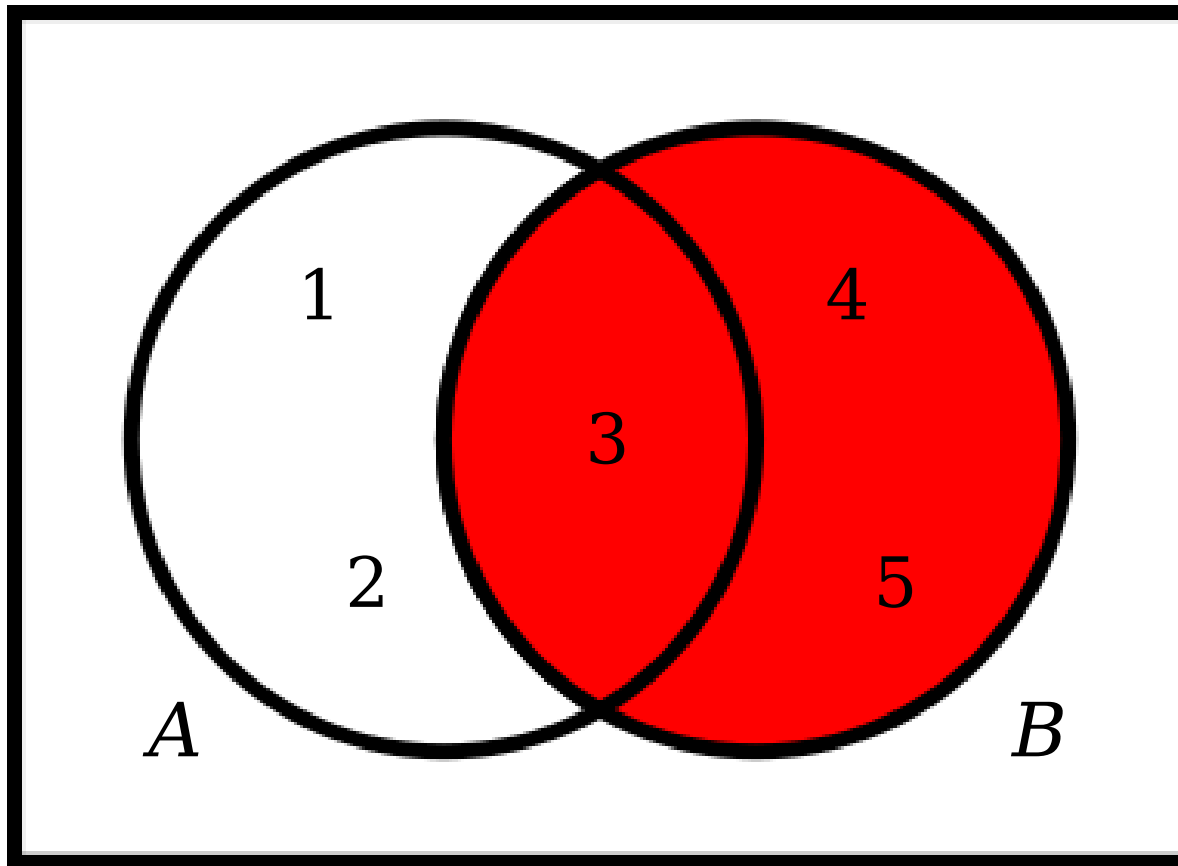
# Venn Diagrams



$$A = \{ 1, 2, 3 \}$$

$$B = \{ 3, 4, 5 \}$$

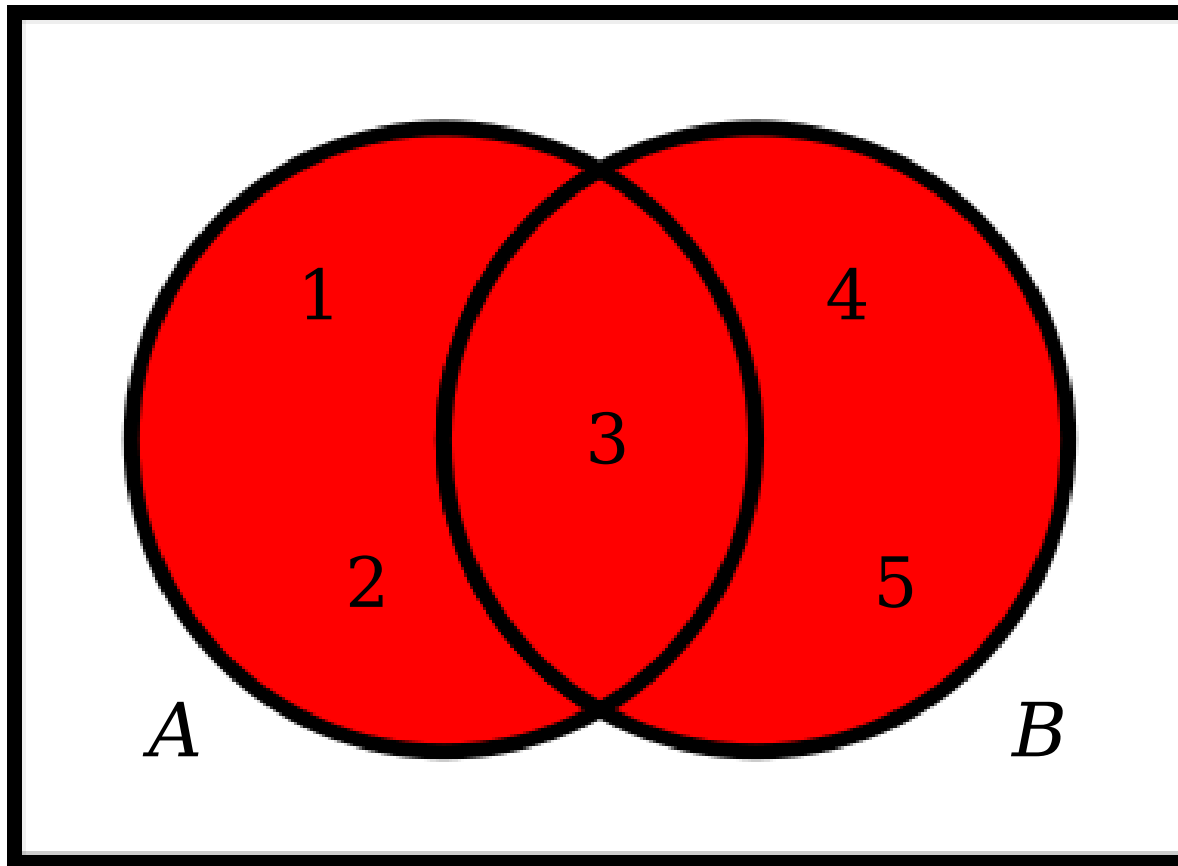
# Venn Diagrams



$$A = \{ 1, 2, 3 \}$$

$$B = \{ 3, 4, 5 \}$$

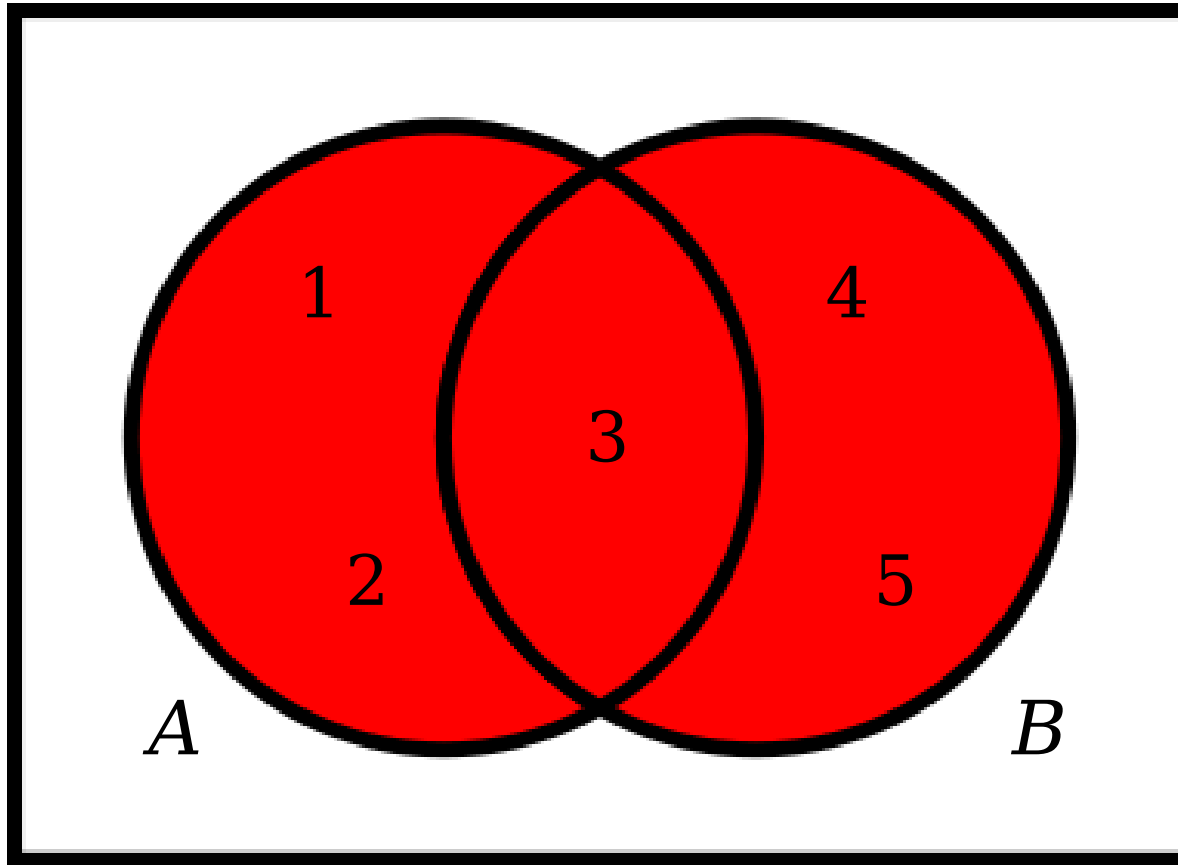
# Venn Diagrams



$$A = \{ 1, 2, 3 \}$$

$$B = \{ 3, 4, 5 \}$$

# Venn Diagrams

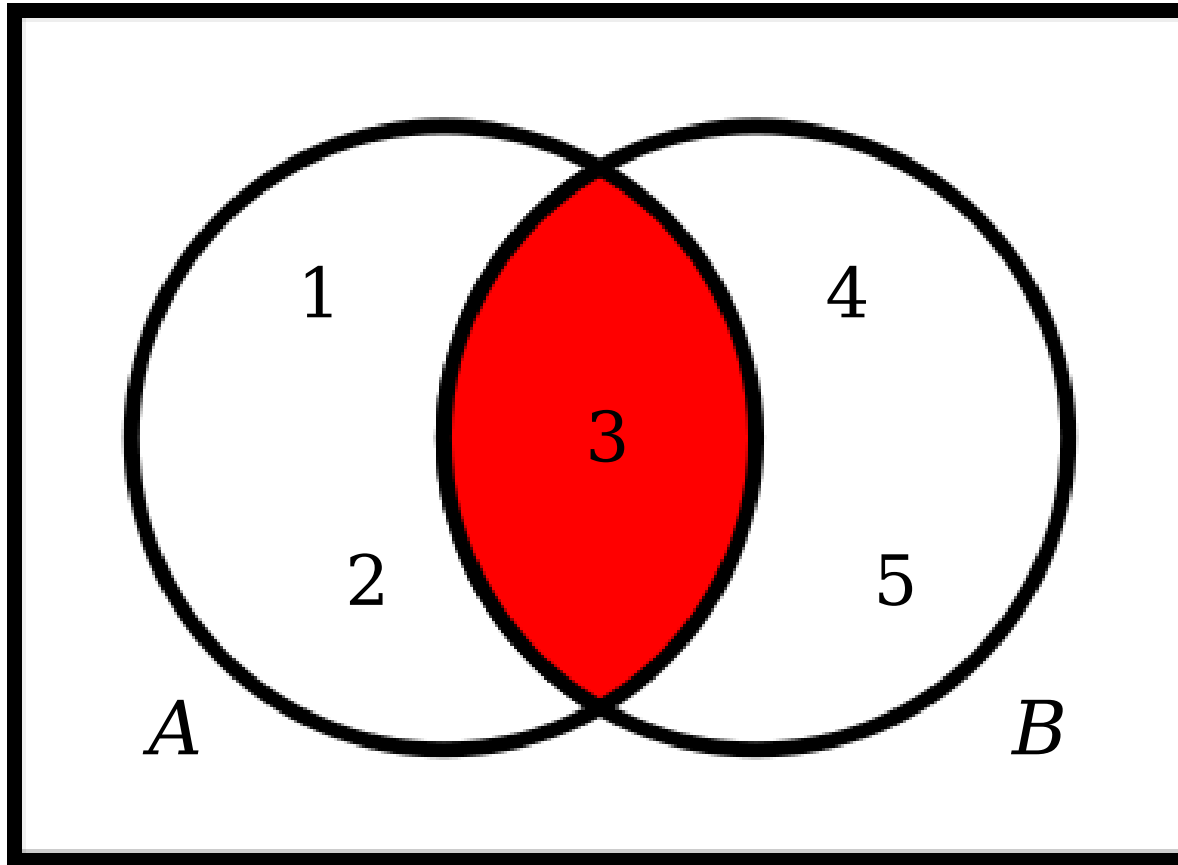


Union  
 $A \cup B$   
 $\{ 1, 2, 3, 4, 5 \}$

$$A = \{ 1, 2, 3 \}$$

$$B = \{ 3, 4, 5 \}$$

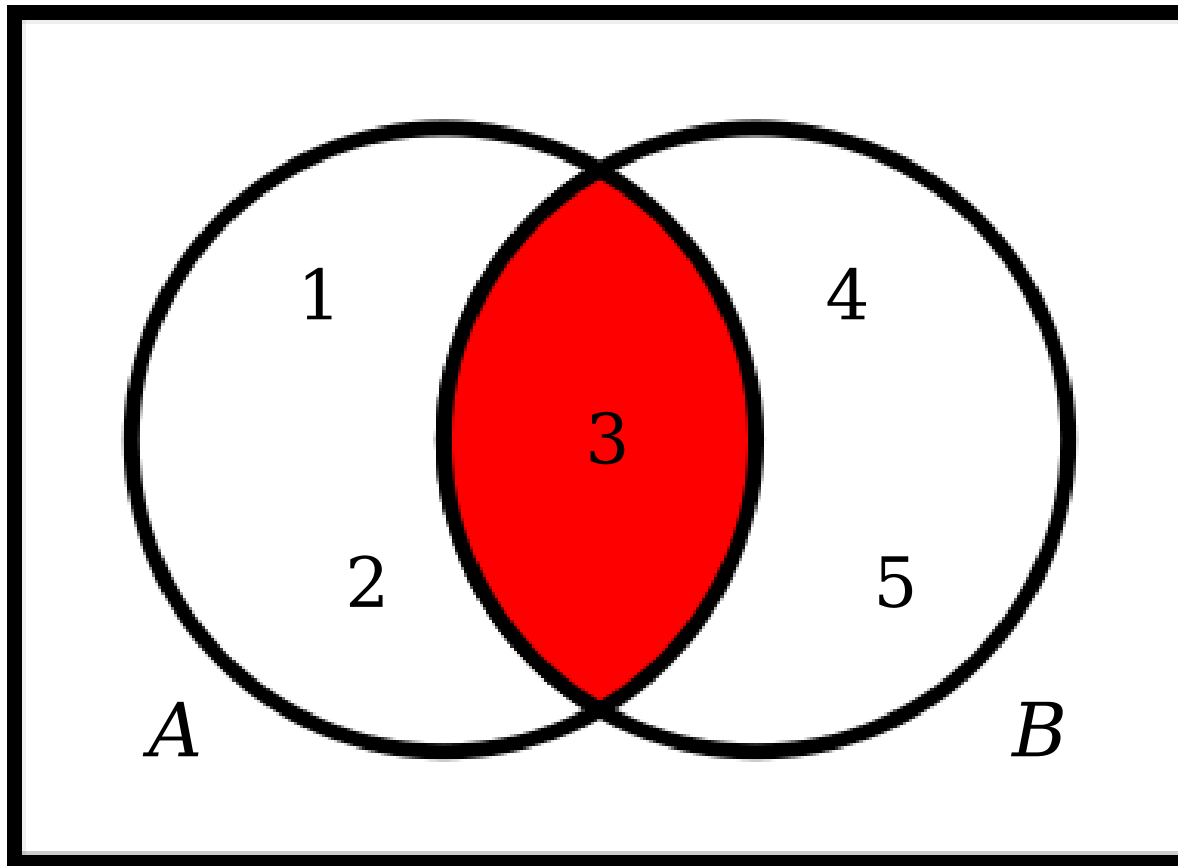
# Venn Diagrams



$$A = \{ 1, 2, 3 \}$$

$$B = \{ 3, 4, 5 \}$$

# Venn Diagrams



Intersection

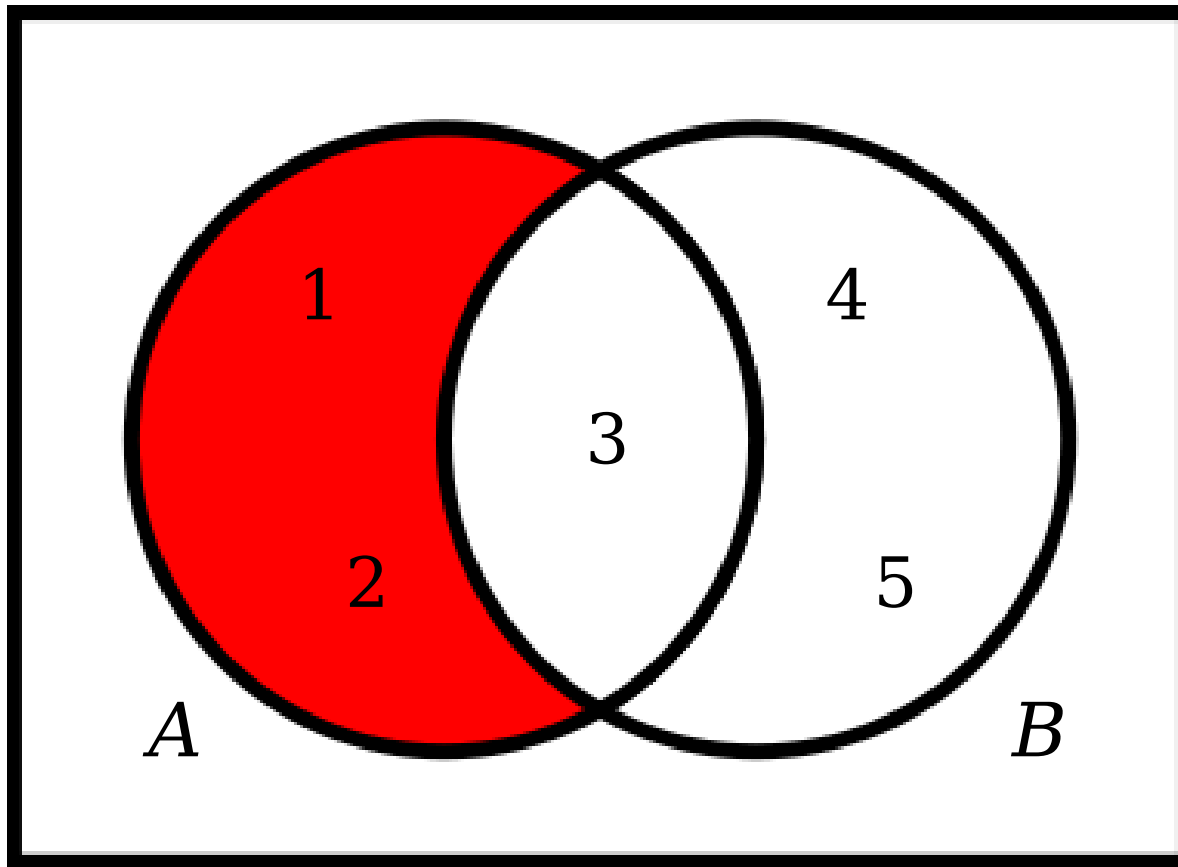
$$A \cap B$$

$$\{ 3 \}$$

$$A = \{ 1, 2, 3 \}$$

$$B = \{ 3, 4, 5 \}$$

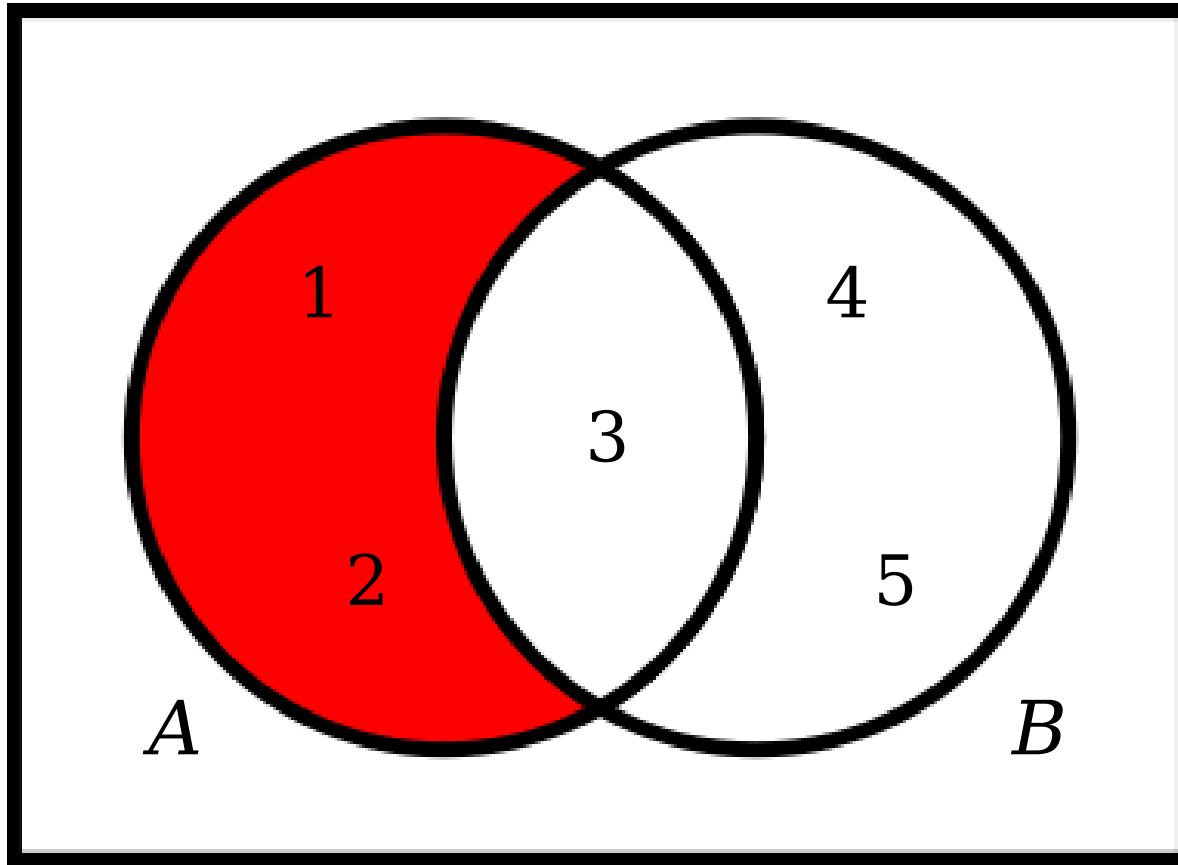
# Venn Diagrams



$$A = \{ 1, 2, 3 \}$$

$$B = \{ 3, 4, 5 \}$$

# Venn Diagrams



Difference

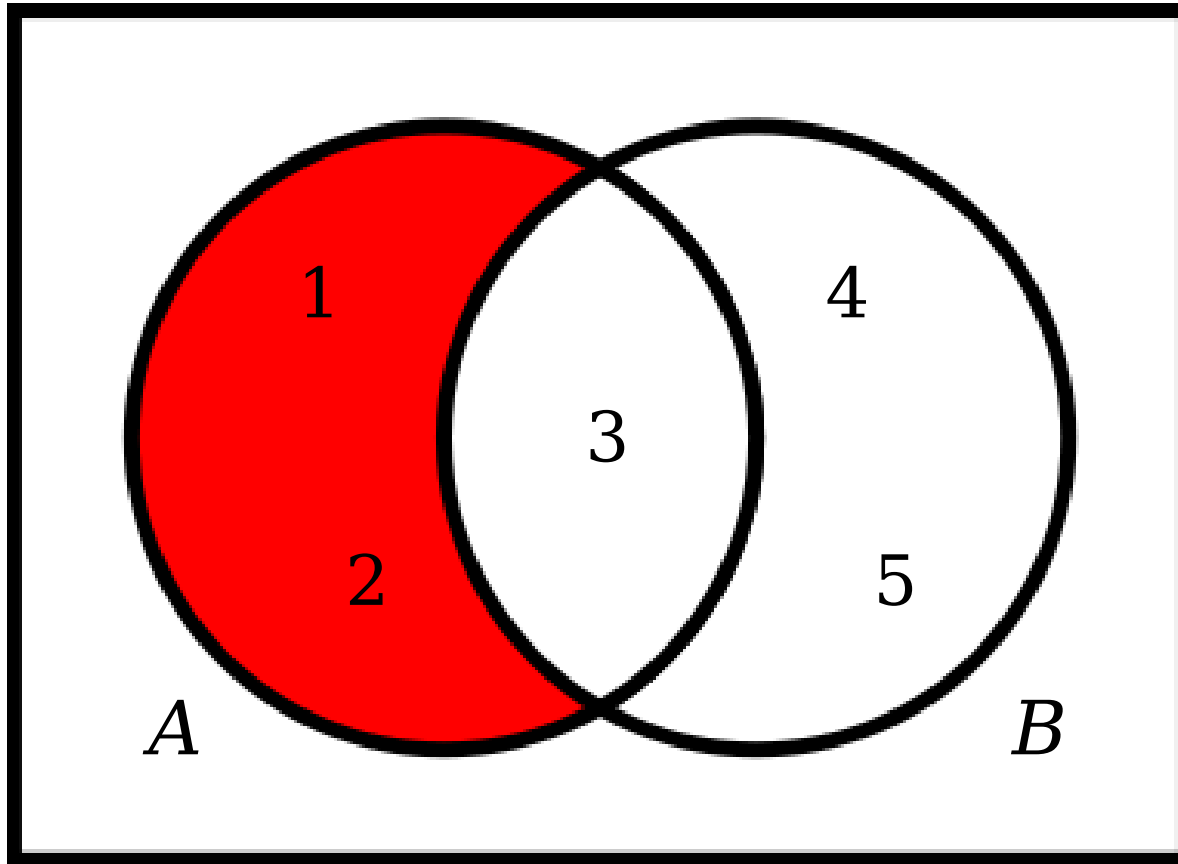
$$A - B$$

$$\{ 1, 2 \}$$

$$A = \{ 1, 2, 3 \}$$

$$B = \{ 3, 4, 5 \}$$

# Venn Diagrams



Difference

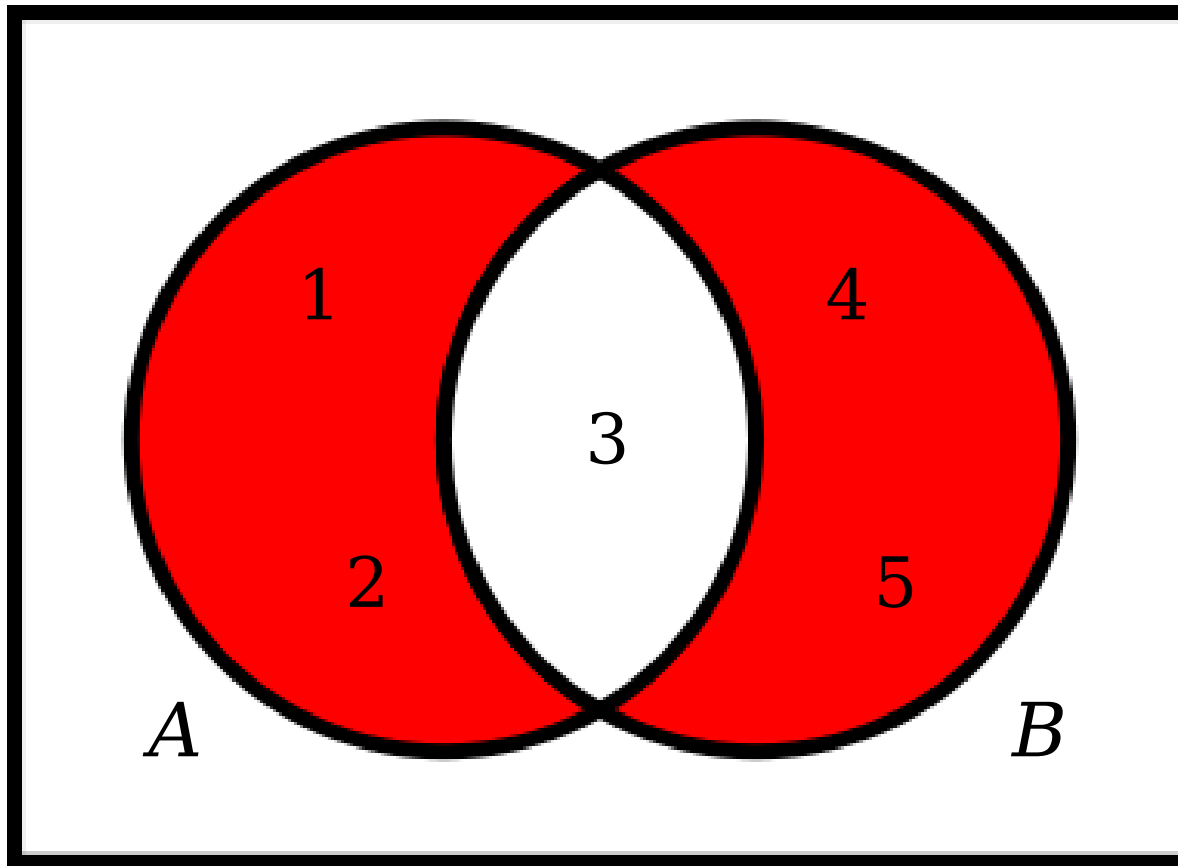
$$A \setminus B$$

$$\{ 1, 2 \}$$

$$A = \{ 1, 2, 3 \}$$

$$B = \{ 3, 4, 5 \}$$

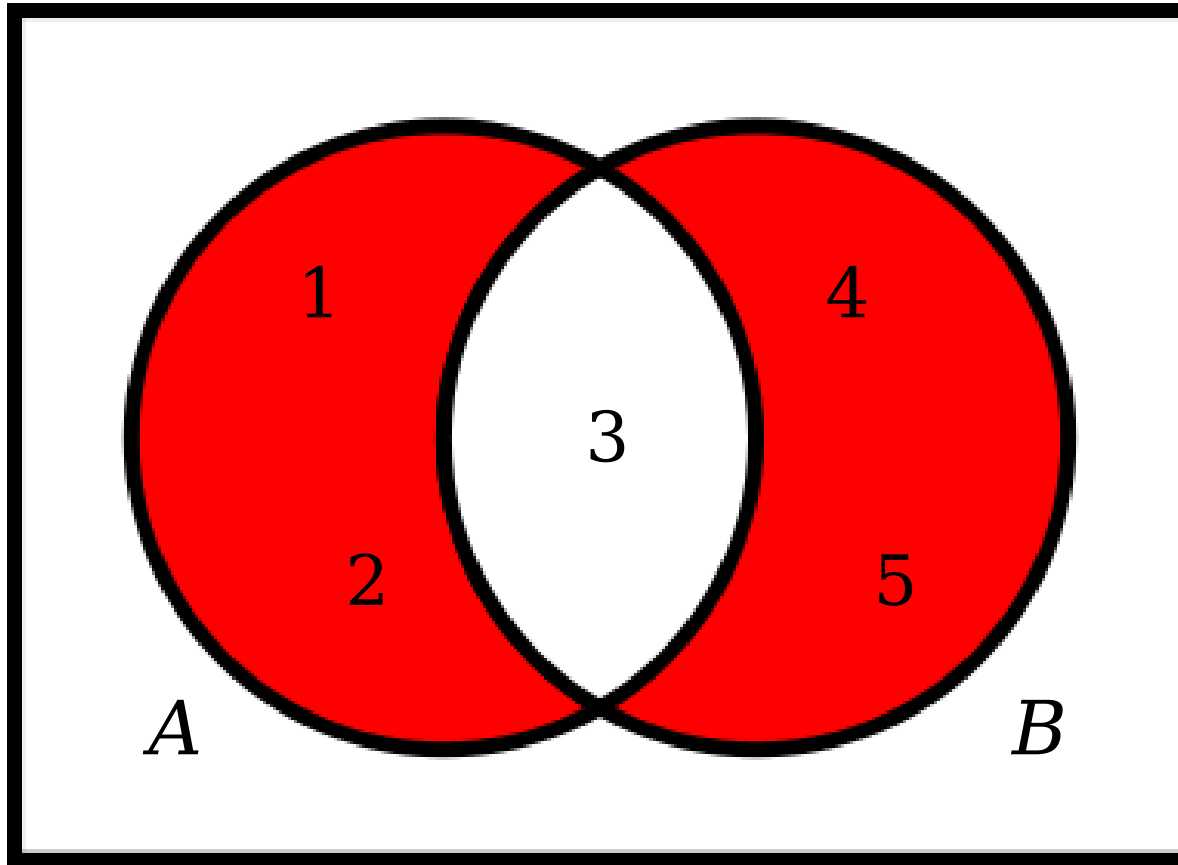
# Venn Diagrams



$$A = \{ 1, 2, 3 \}$$

$$B = \{ 3, 4, 5 \}$$

# Venn Diagrams

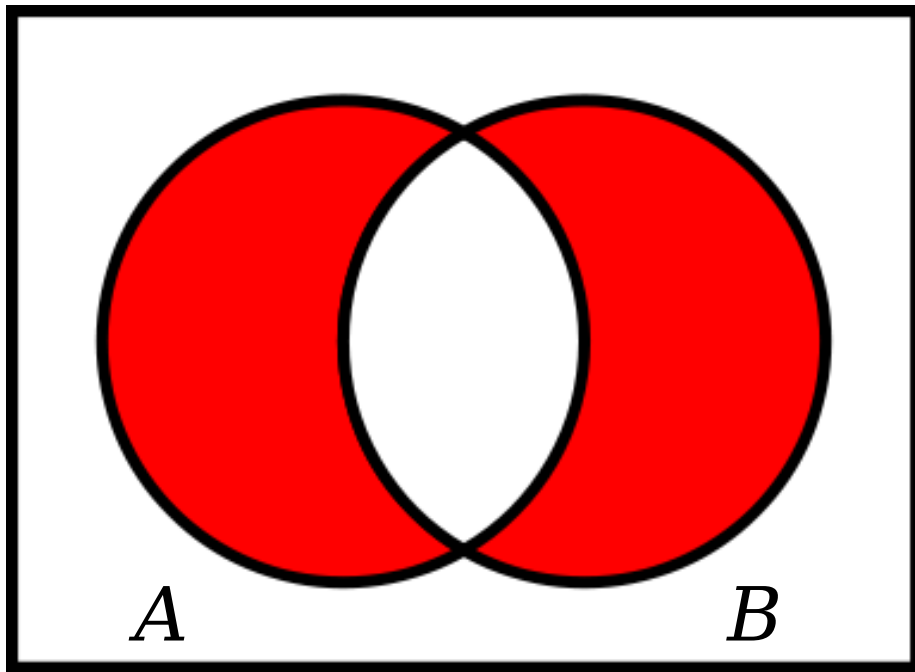


Symmetric  
Difference  
 $A \Delta B$   
 $\{ 1, 2, 4, 5 \}$

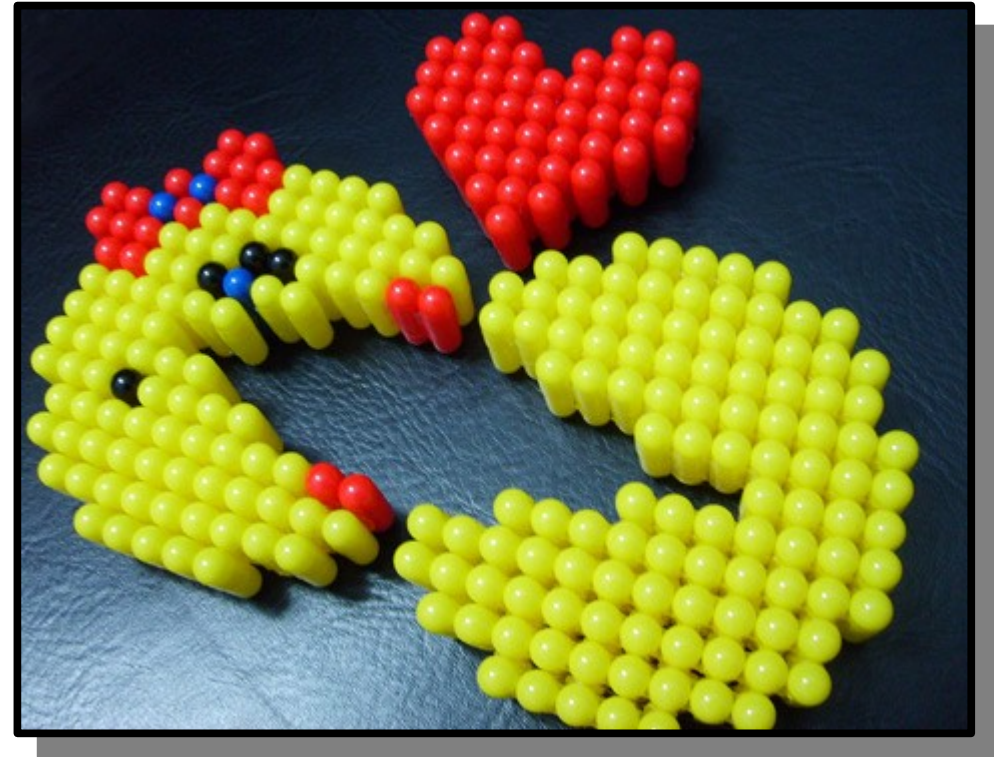
$$A = \{ 1, 2, 3 \}$$

$$B = \{ 3, 4, 5 \}$$

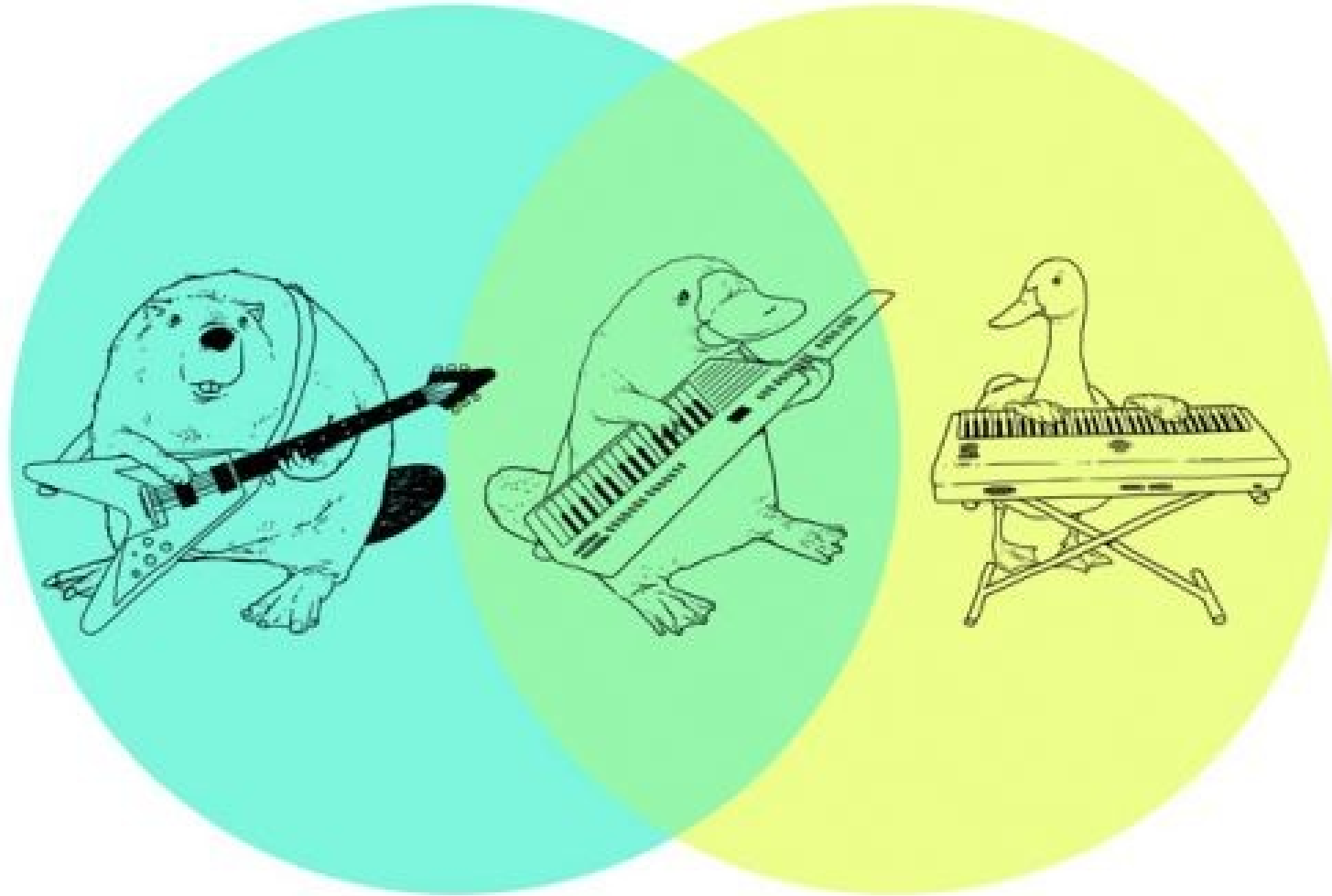
# Venn Diagrams



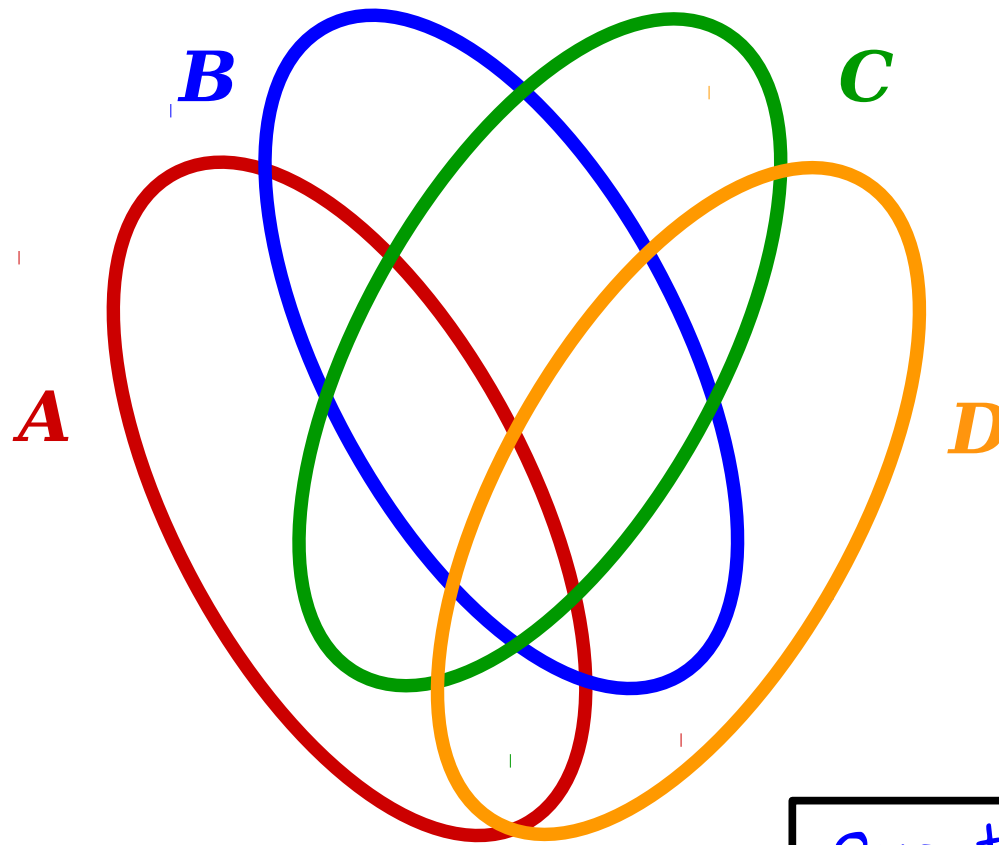
$$A \Delta B$$



# Venn Diagrams

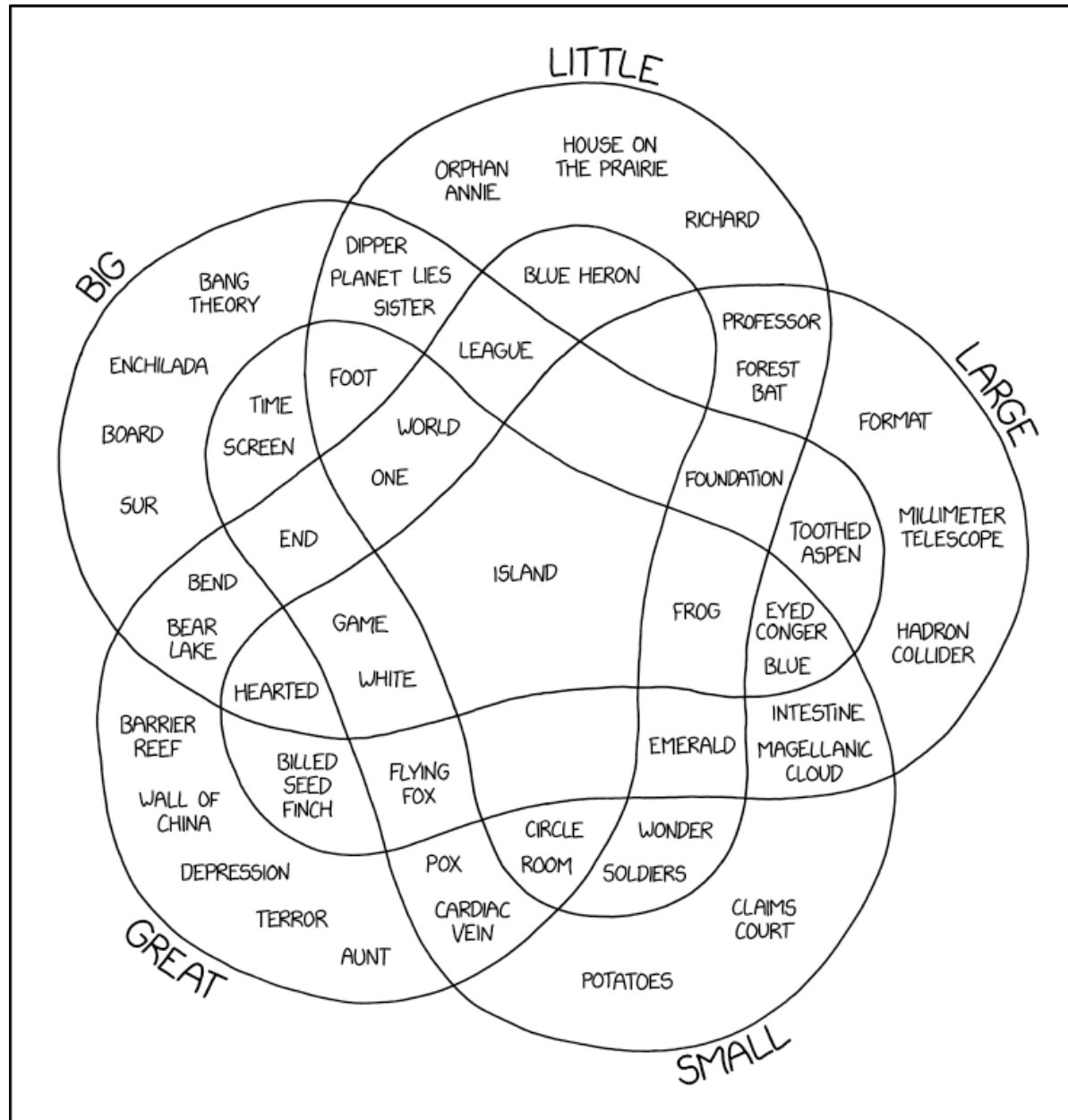


# Venn Diagrams for Four Sets



Question to ponder:  
why don't we just  
draw four circles?

# Venn Diagrams for Five Sets



# Venn Diagrams for Seven Sets

**<http://moebio.com/research/sevensets/>**

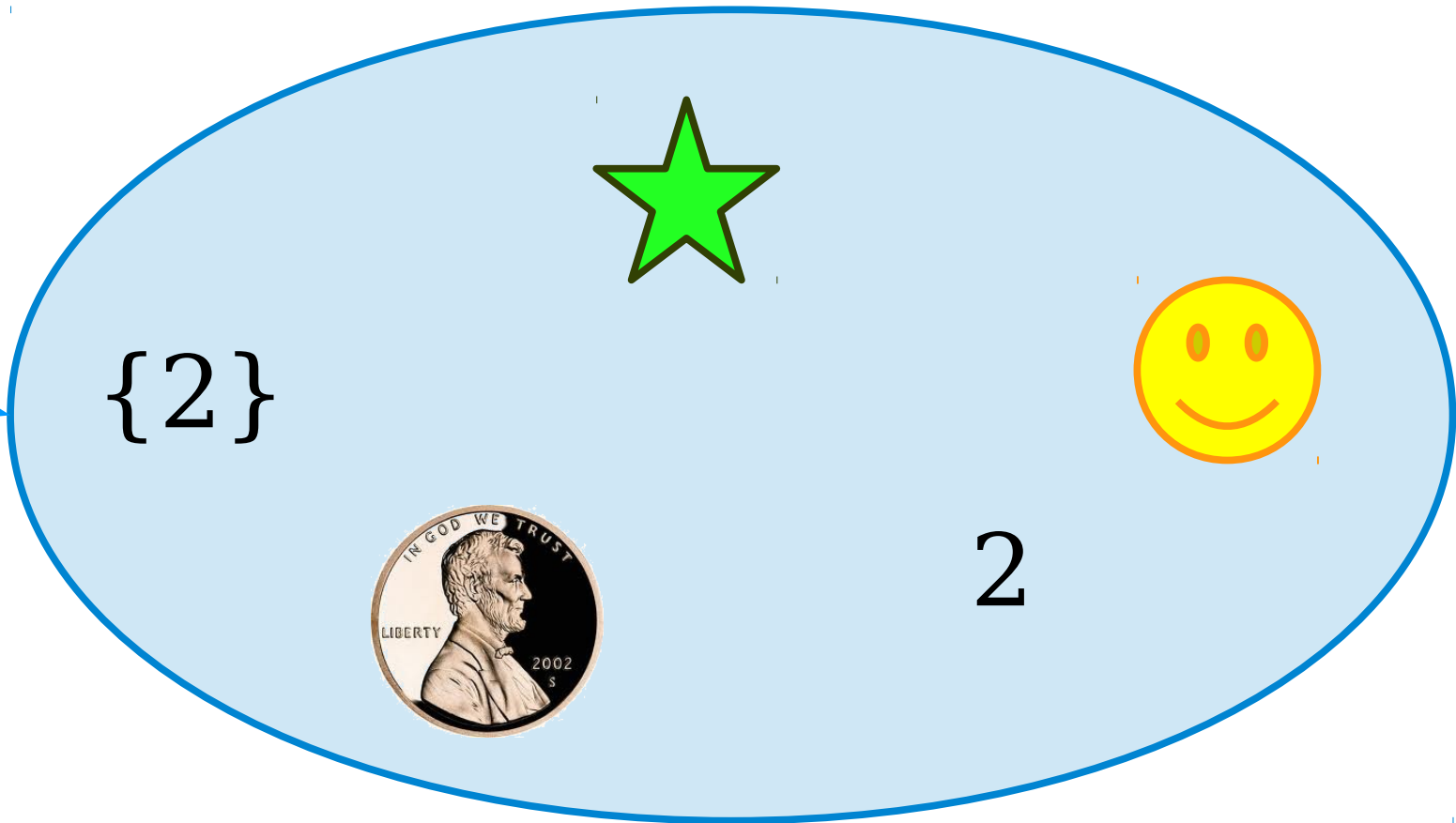
# Subsets and Power Sets

# Subsets

- A set  $S$  is called a **subset** of a set  $T$  (denoted  $S \subseteq T$ ) if all elements of  $S$  are also elements of  $T$ .
- Examples:
  - $\{ 1, 2, 3 \} \subseteq \{ 1, 2, 3, 4 \}$
  - $\{ b, c \} \subseteq \{ a, b, c, d \}$
  - $\{ \text{H}, \text{He}, \text{Li} \} \subseteq \{ \text{H}, \text{He}, \text{Li} \}$
  - $\mathbb{N} \subseteq \mathbb{Z}$  (*every natural number is an integer*)
  - $\mathbb{Z} \subseteq \mathbb{R}$  (*every integer is a real number*)

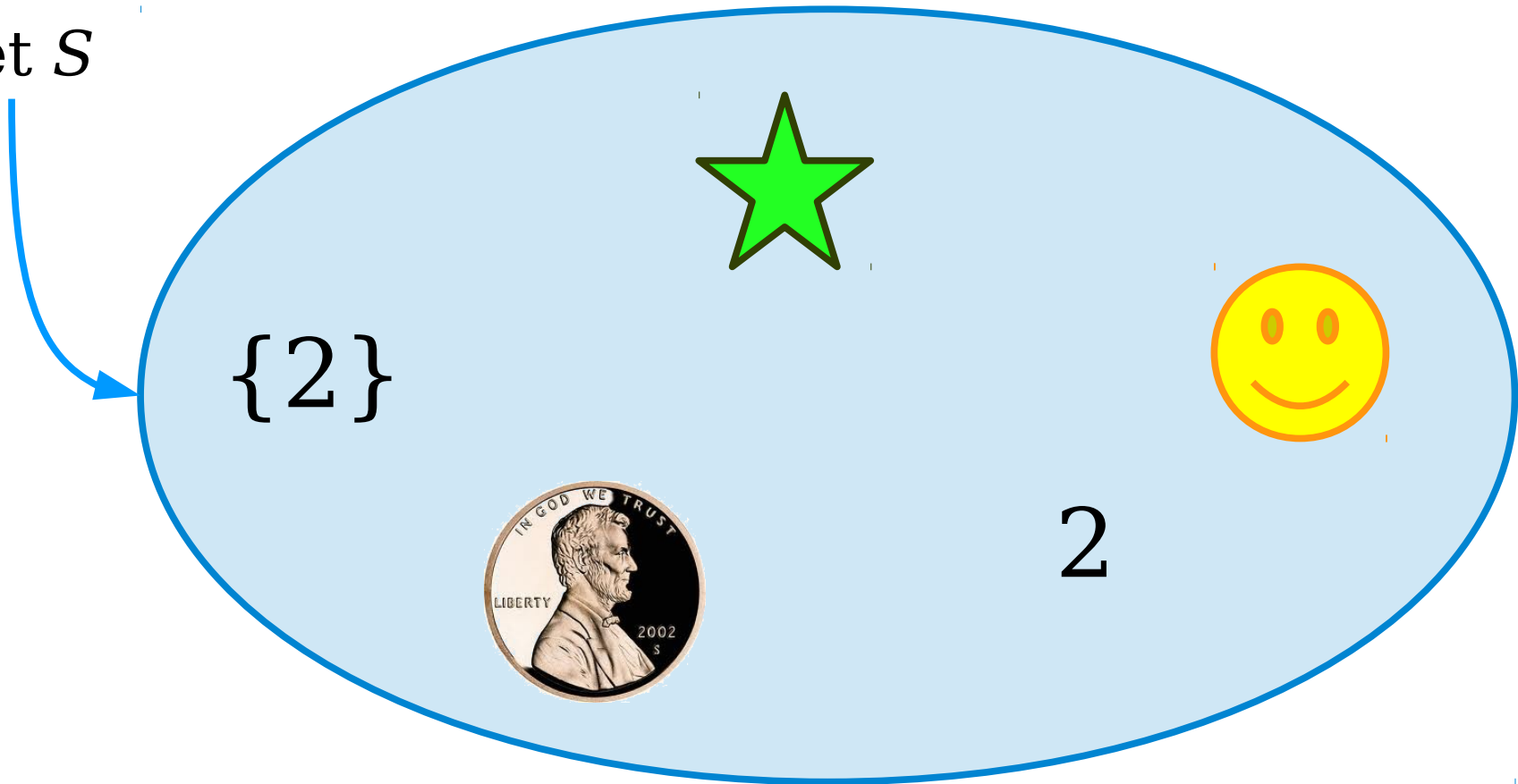
# Subsets and Elements

Set  $S$



# Subsets and Elements

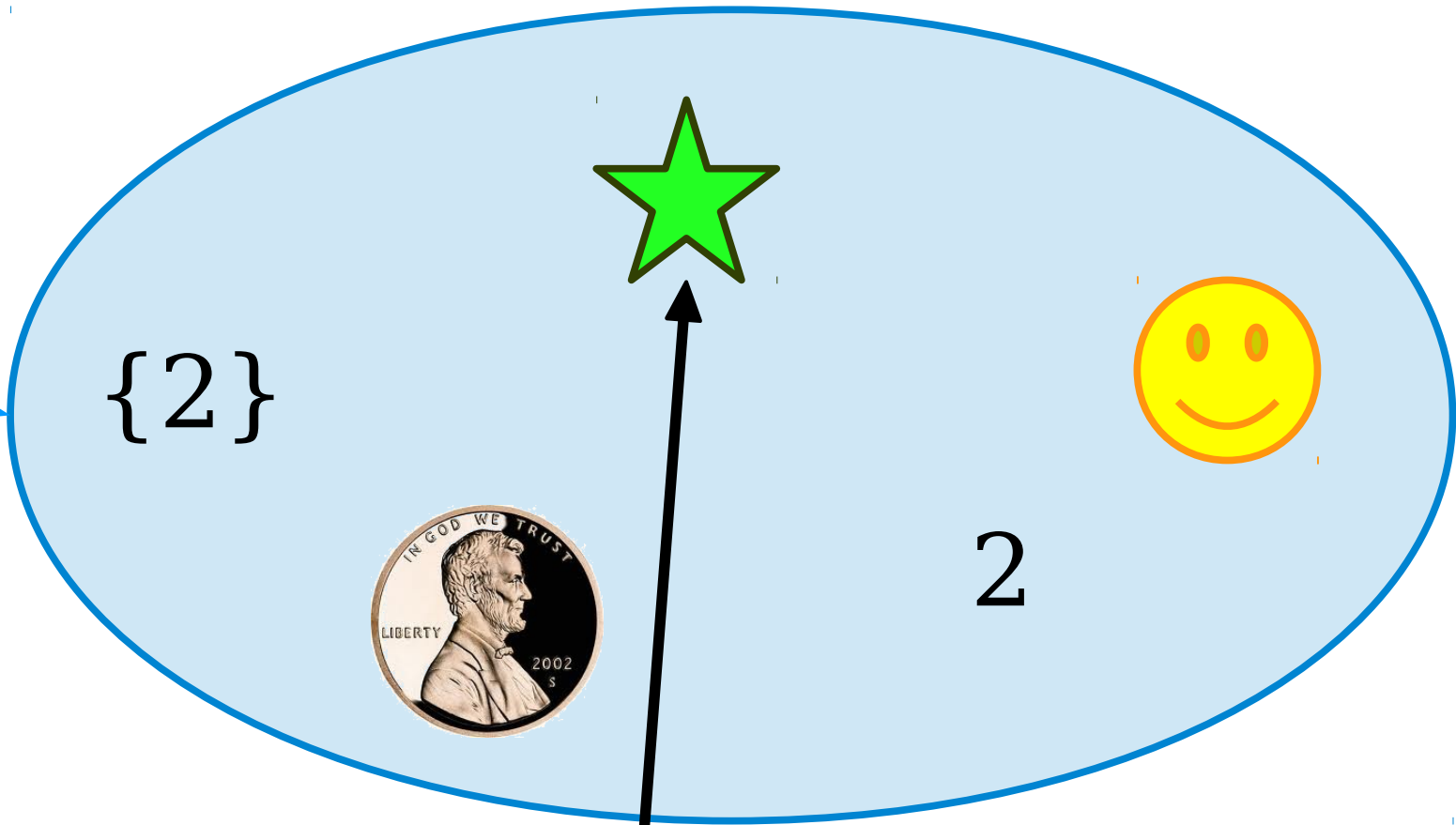
Set  $S$



$$\star \in S$$

# Subsets and Elements

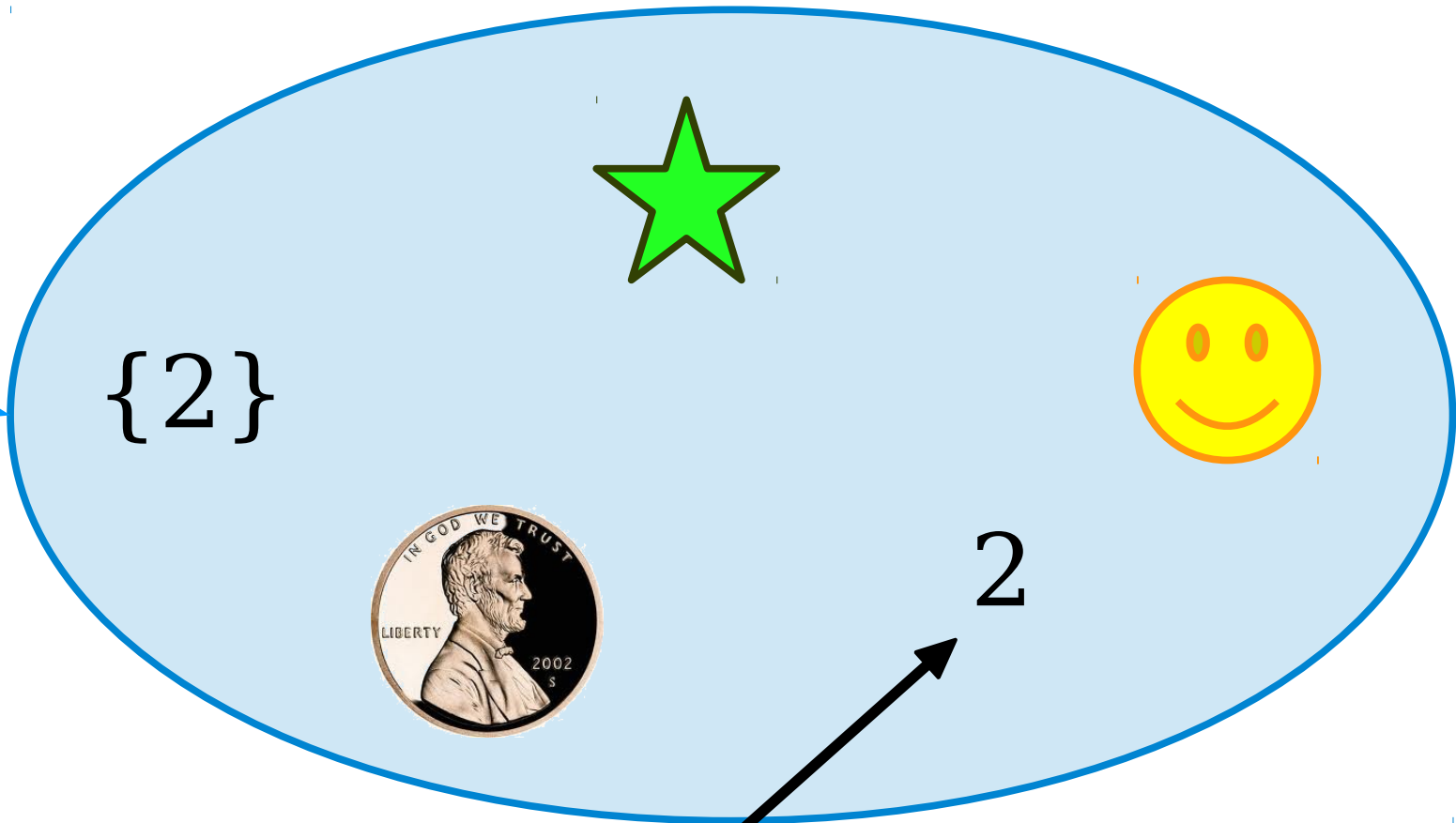
Set  $S$



  $\in S$

# Subsets and Elements

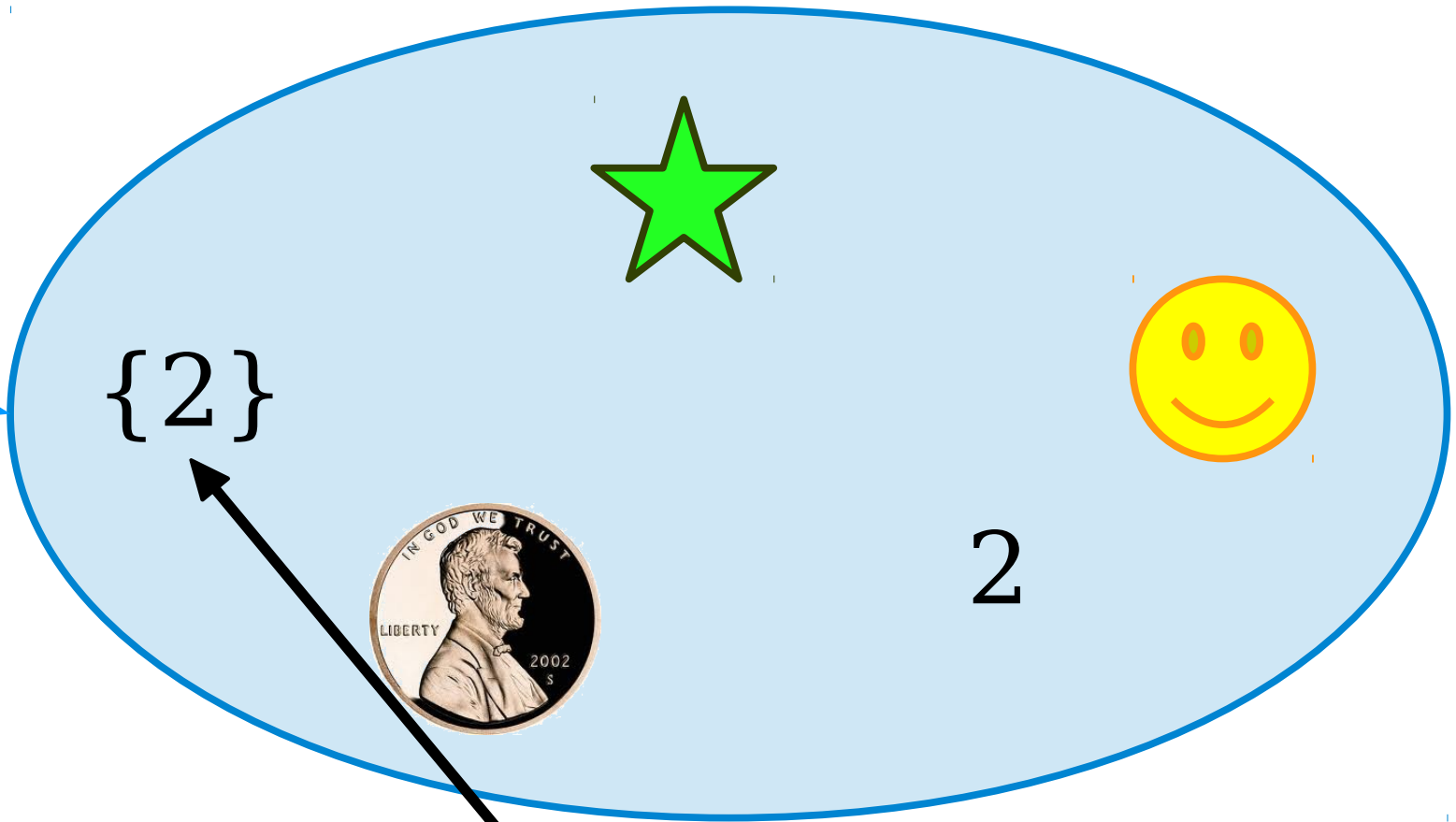
Set  $S$



$$2 \in S$$

# Subsets and Elements

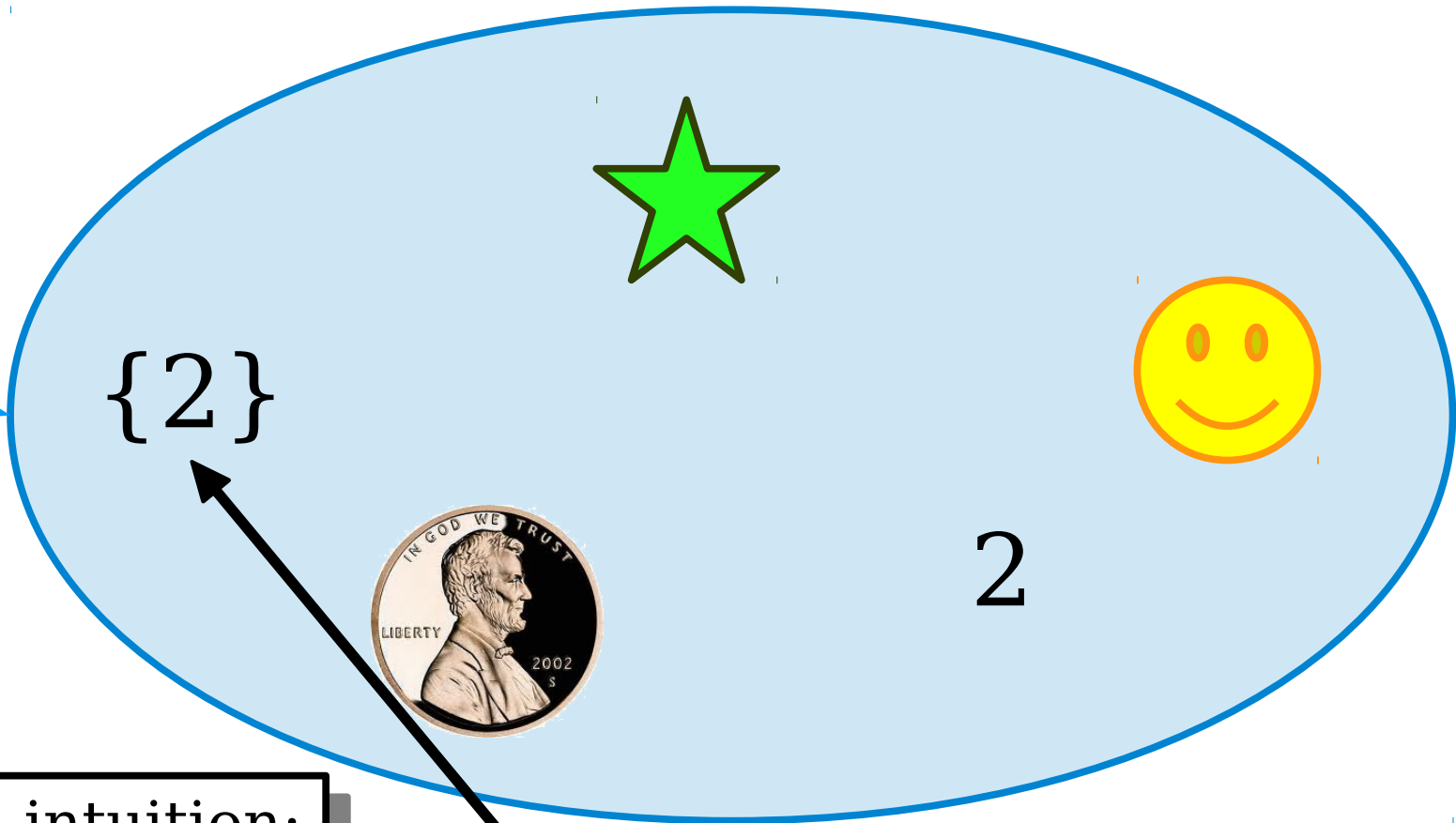
Set  $S$



$$\{2\} \in S$$

# Subsets and Elements

Set  $S$

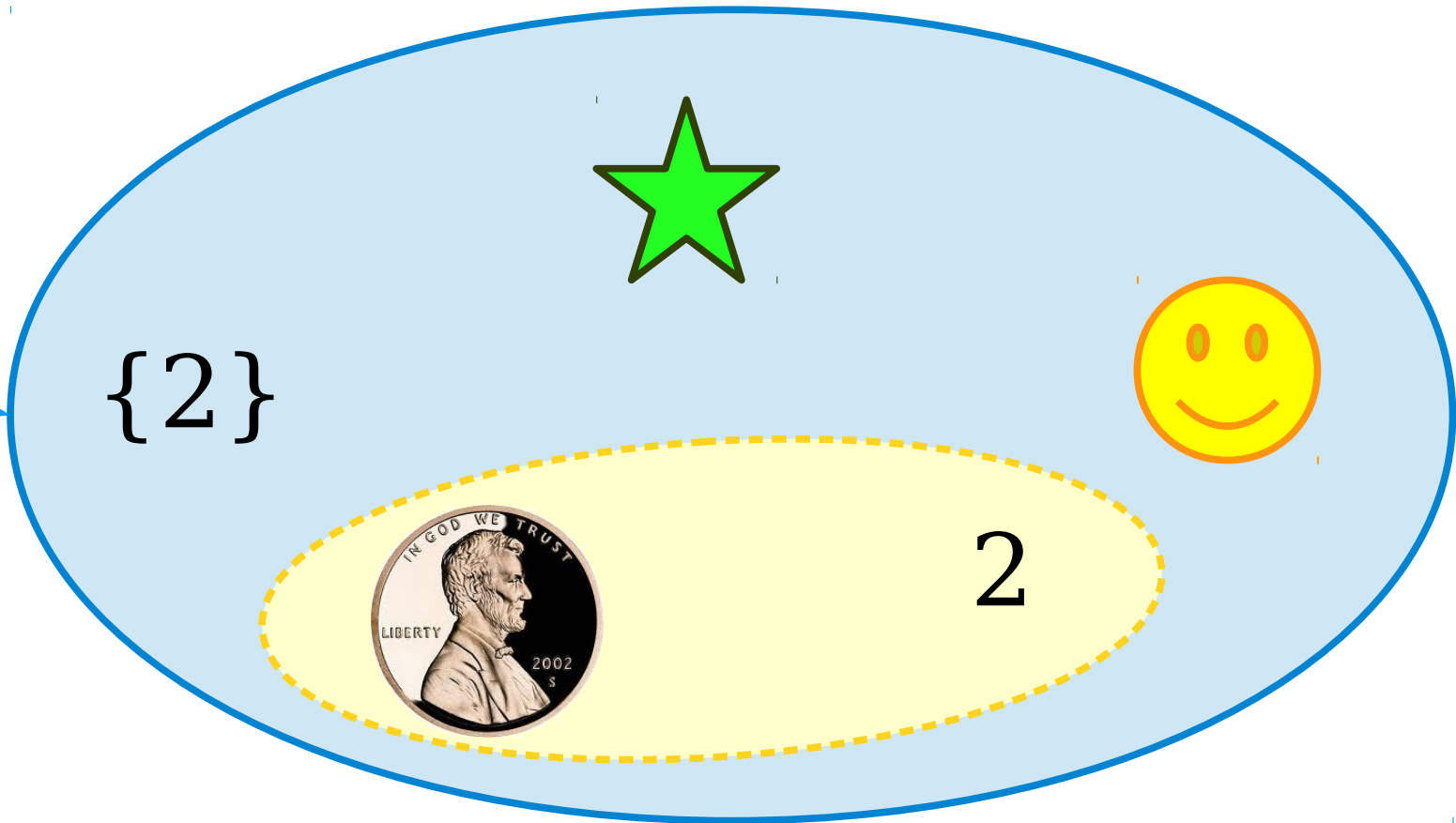


General intuition:  
 $x \in S$  means you  
can ***point at  $x$***   
***inside of  $S$*** .

$$\{2\} \in S$$

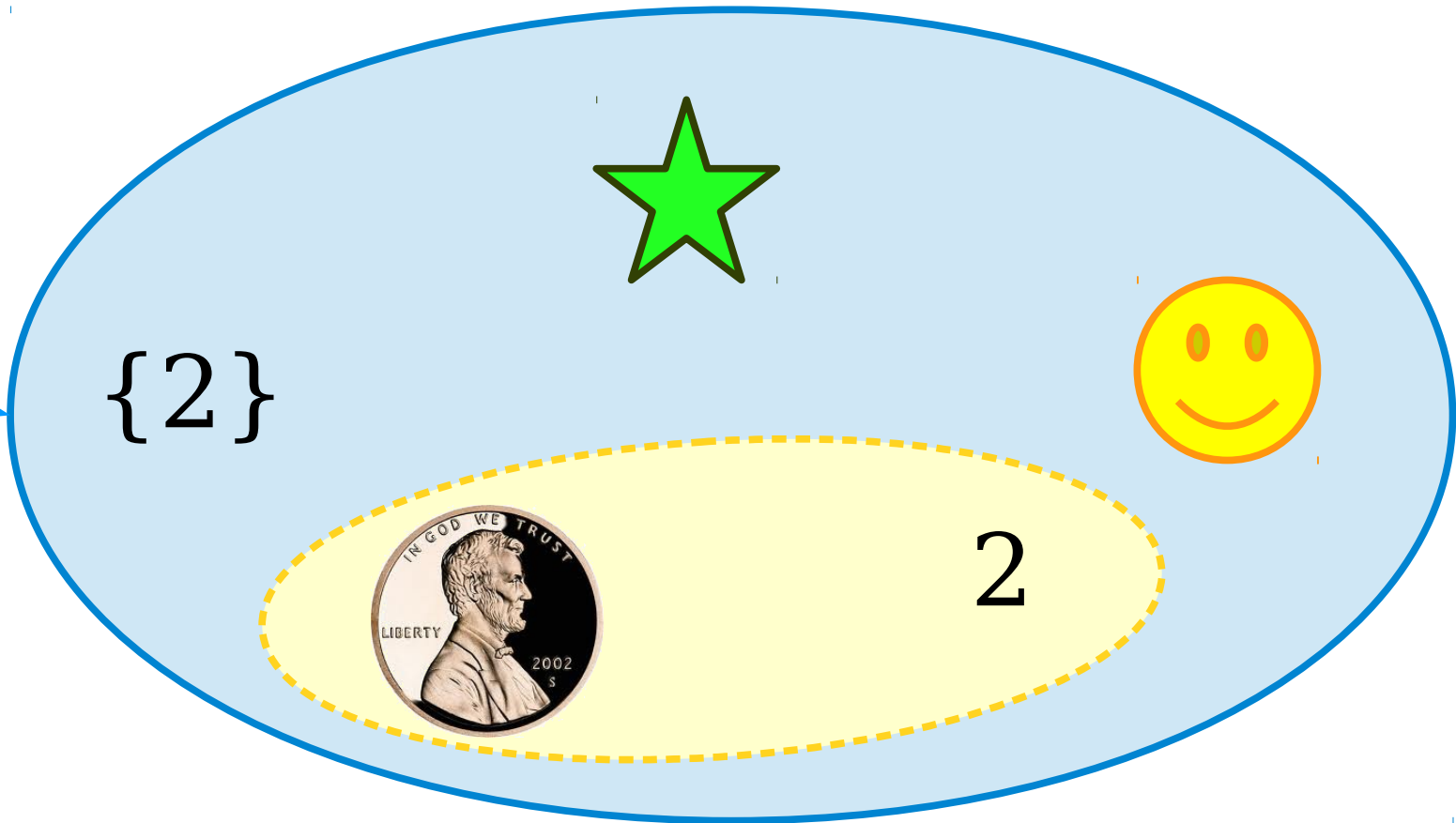
# Subsets and Elements

Set  $S$



# Subsets and Elements

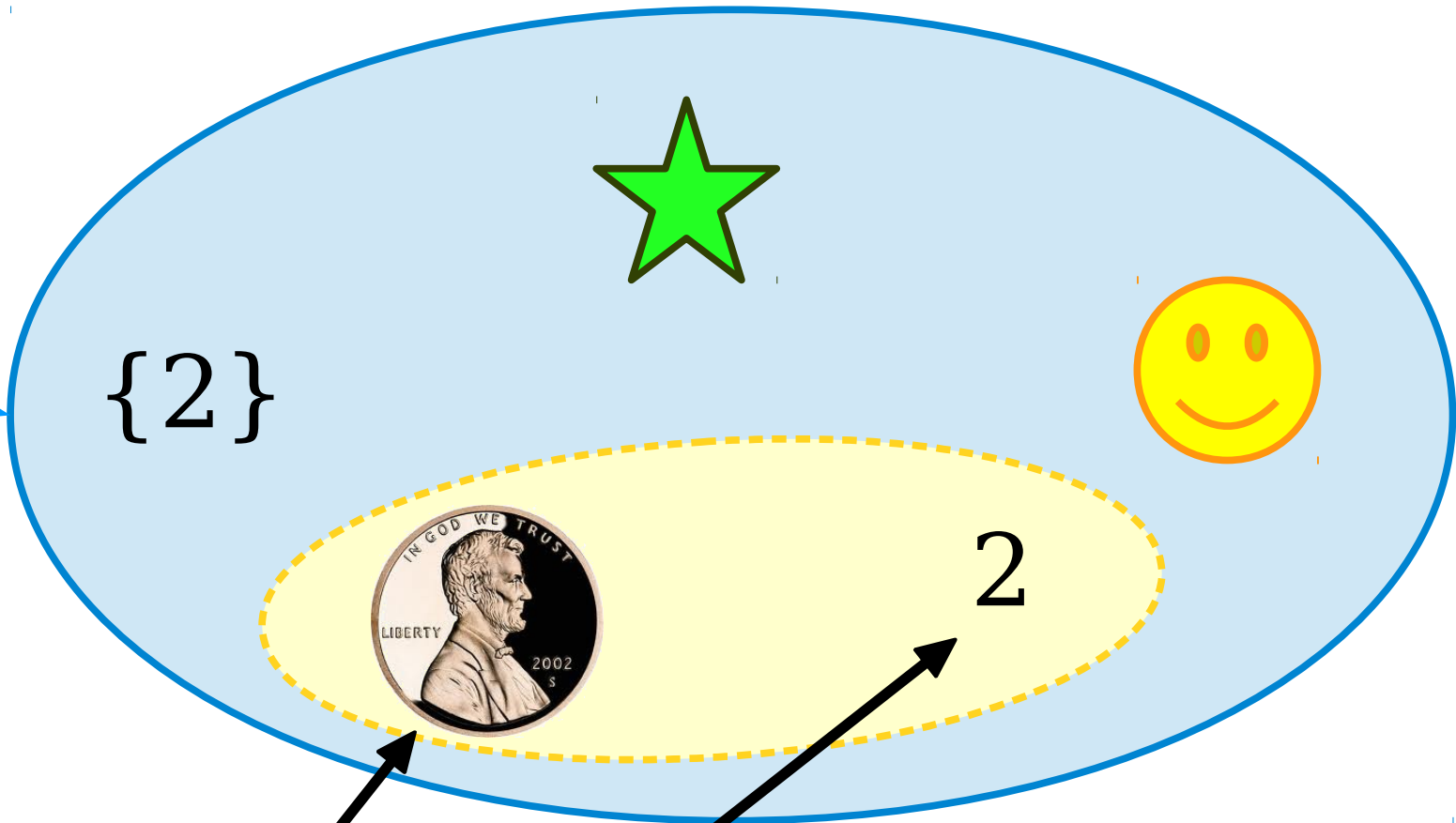
Set  $S$



$$\left\{ \text{penny}, 2 \right\} \subseteq S$$

# Subsets and Elements

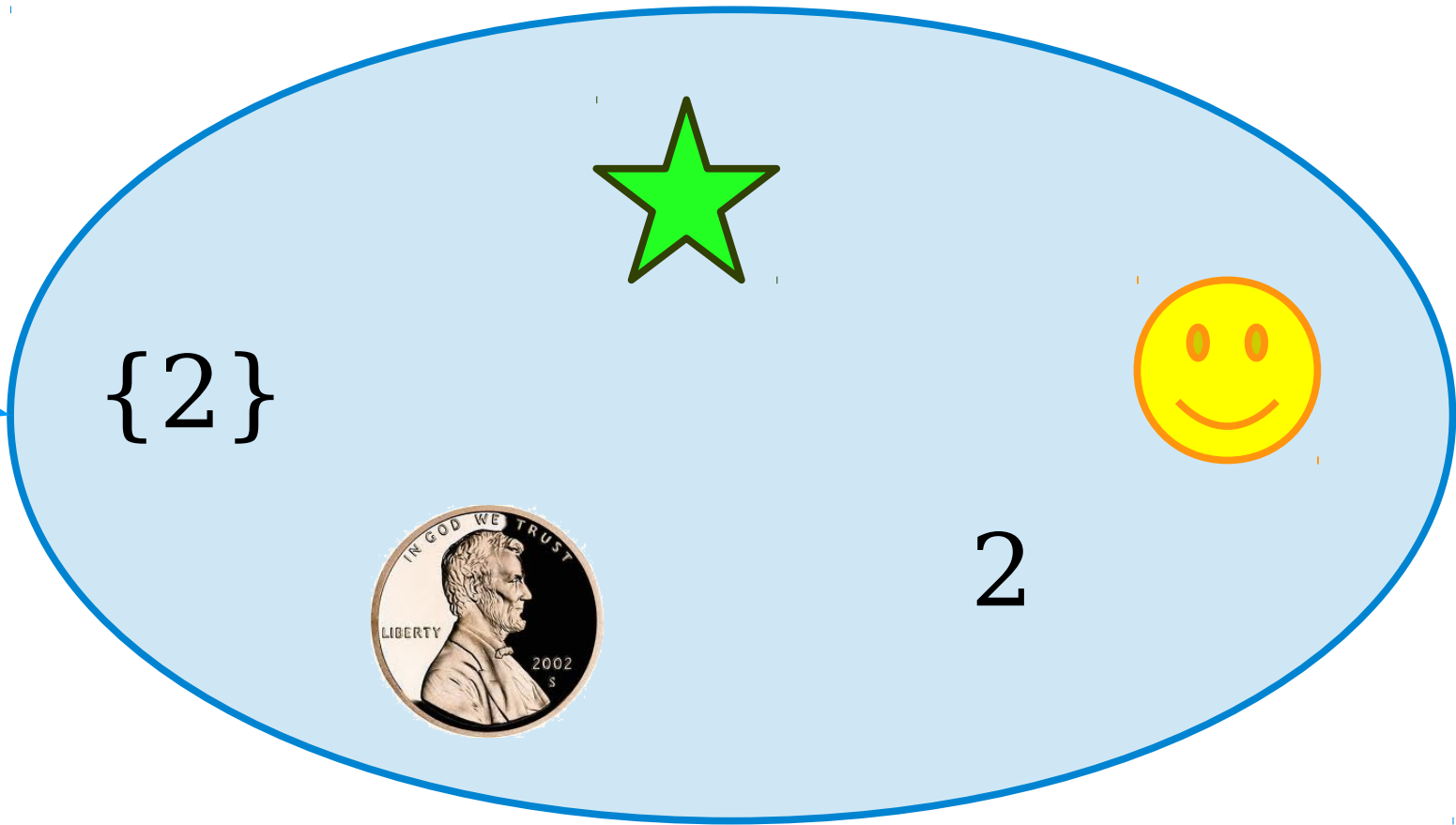
Set  $S$



$$\left\{ \text{penny}, 2 \right\} \subseteq S$$

# Subsets and Elements

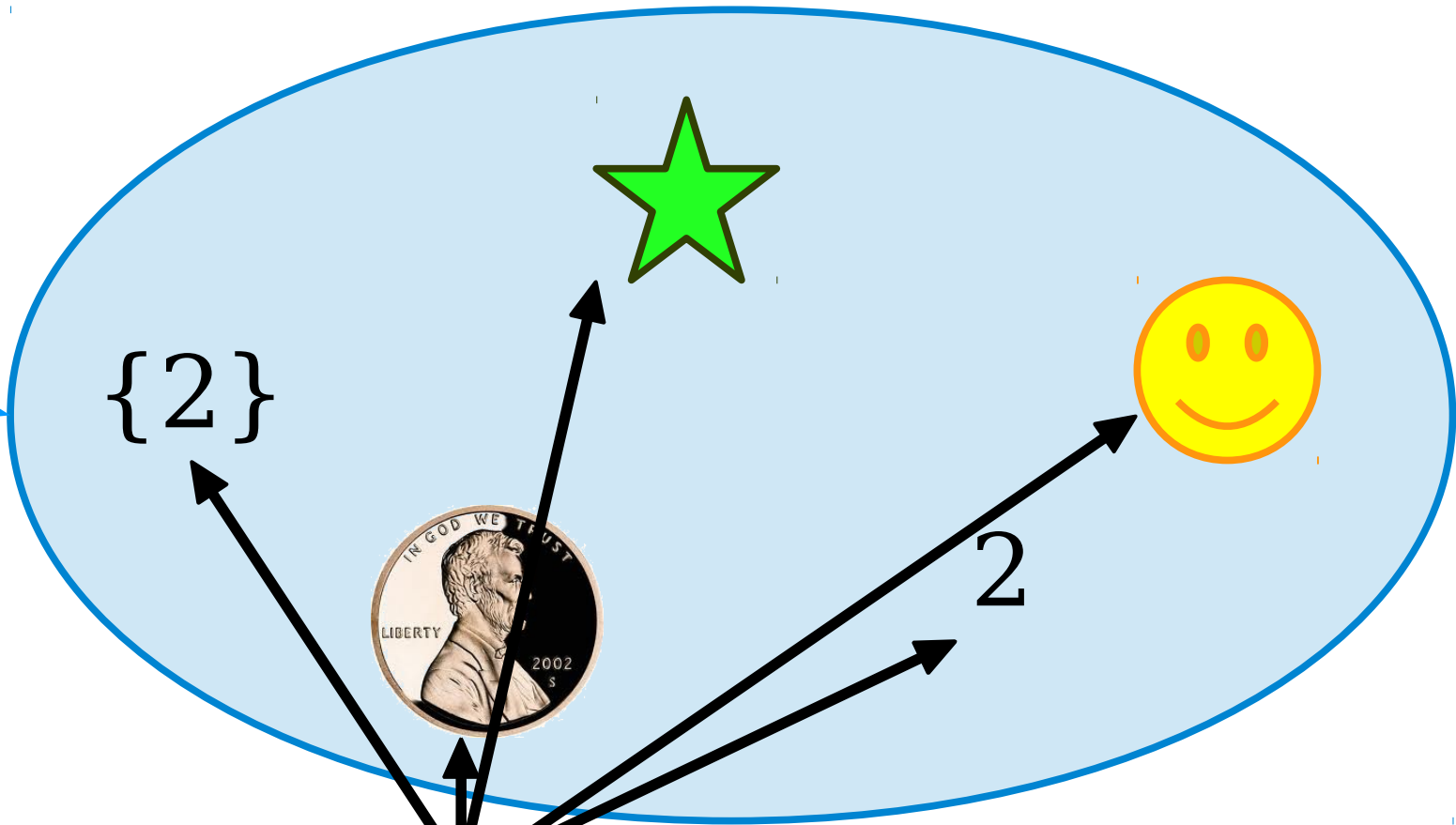
Set  $S$



$$\left\{ \text{penny}, 2 \right\} \notin S$$

# Subsets and Elements

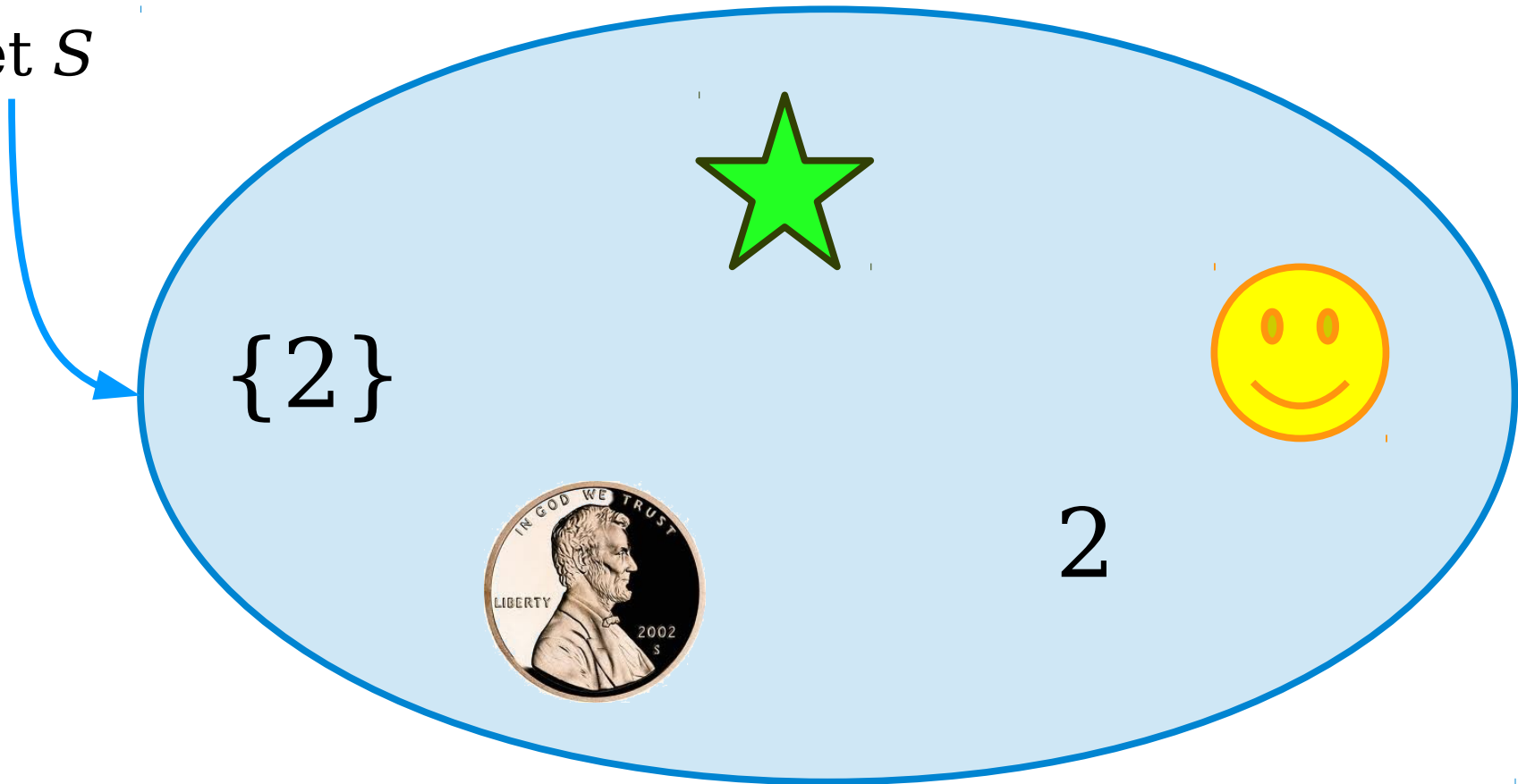
Set  $S$



$$\left\{ \text{penny}, 2 \right\} \notin S$$

# Subsets and Elements

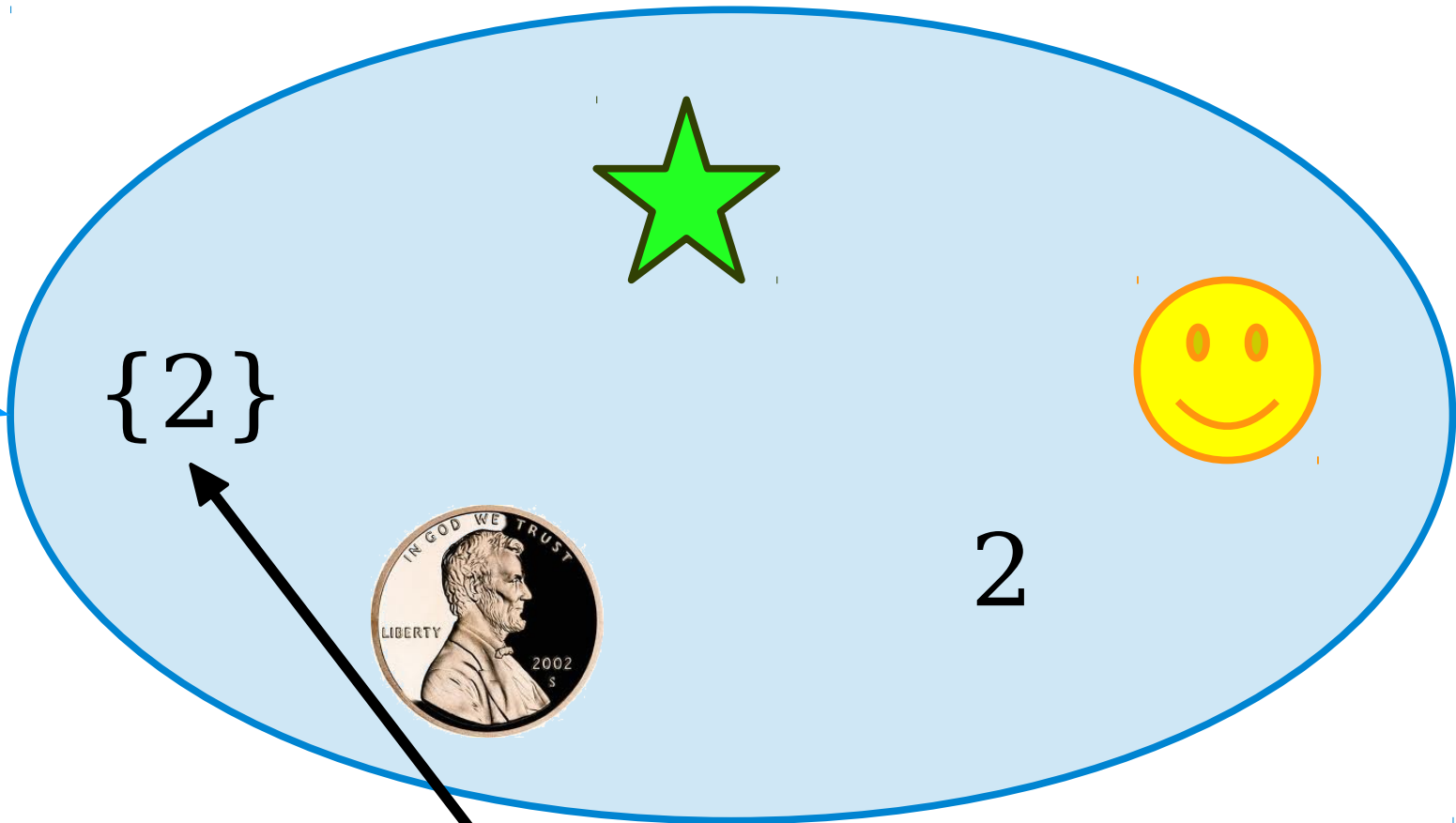
Set  $S$



$$\{2\} \in S$$

# Subsets and Elements

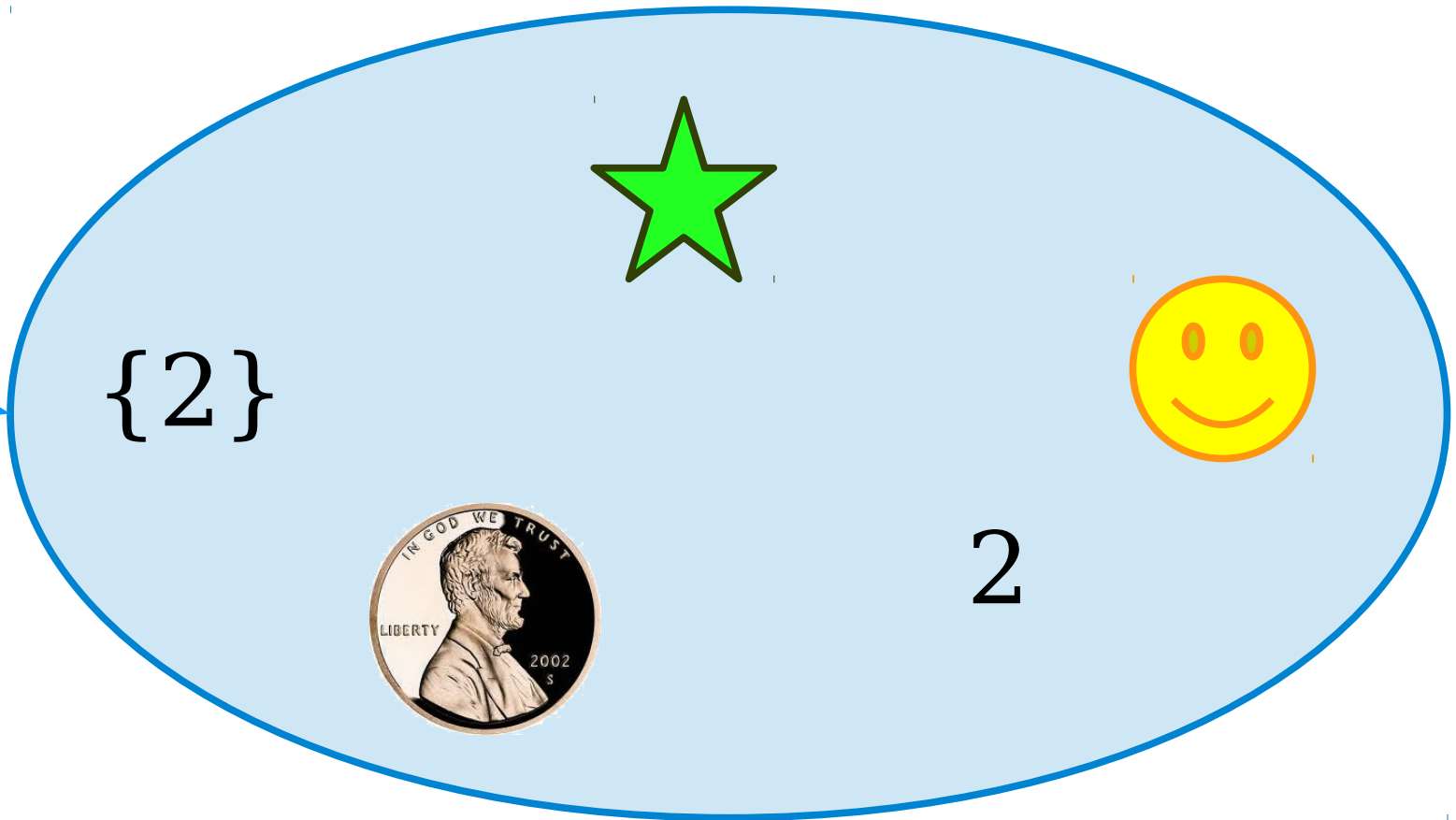
Set  $S$



$$\{2\} \in S$$

# Subsets and Elements

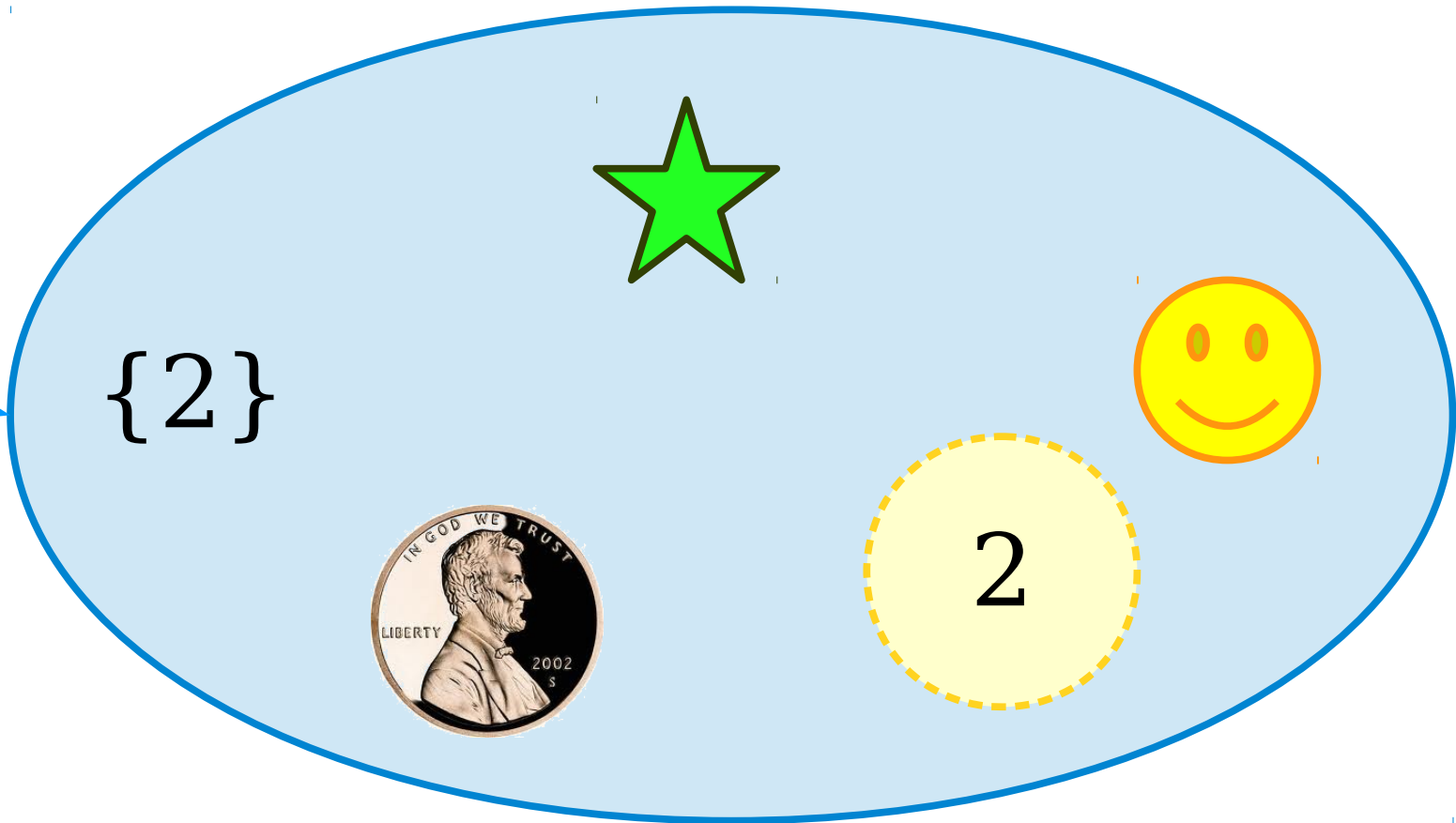
Set  $S$



$$\{2\} \subseteq S$$

# Subsets and Elements

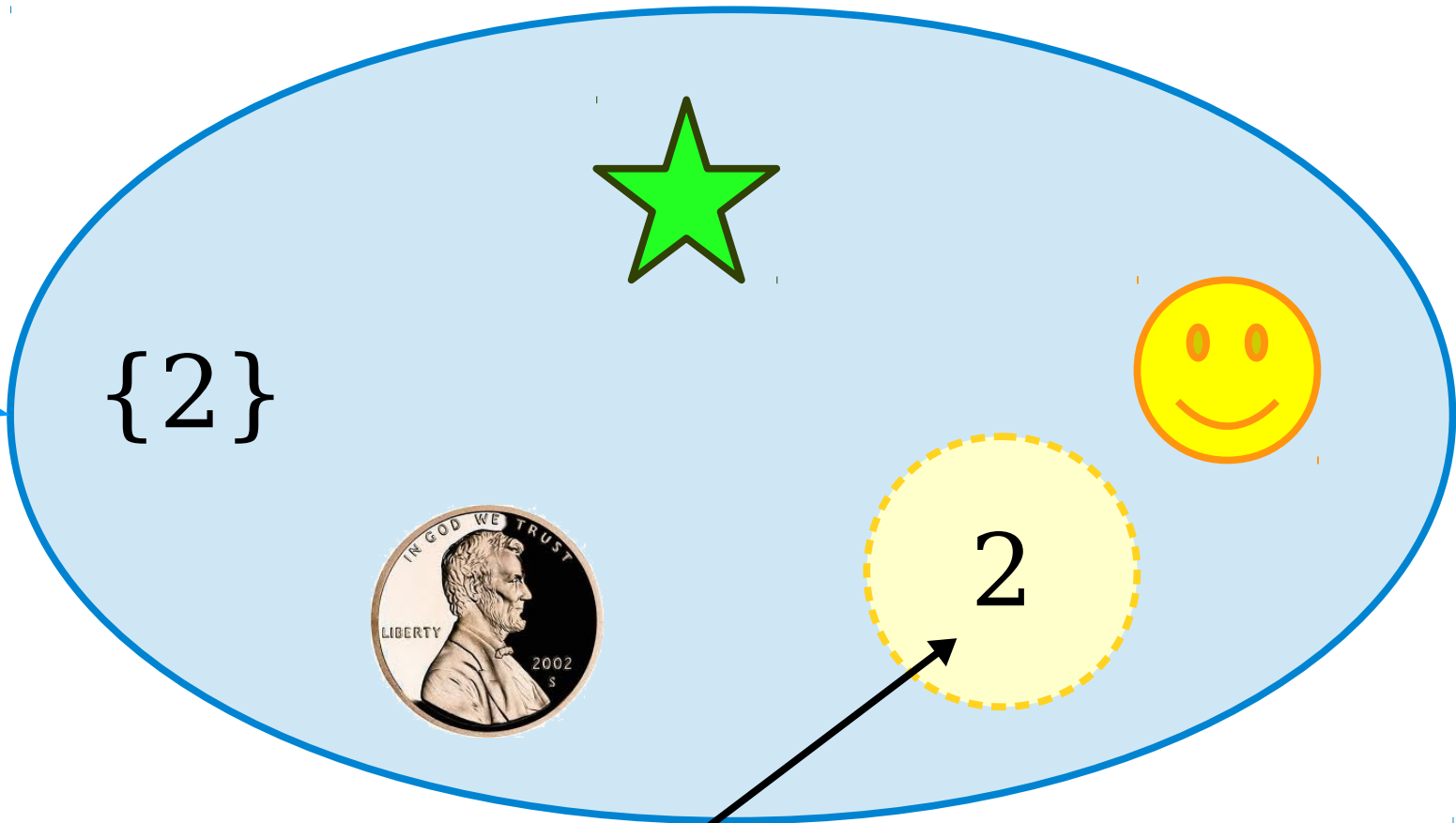
Set  $S$



$$\{2\} \subseteq S$$

# Subsets and Elements

Set  $S$



$\{2\}$

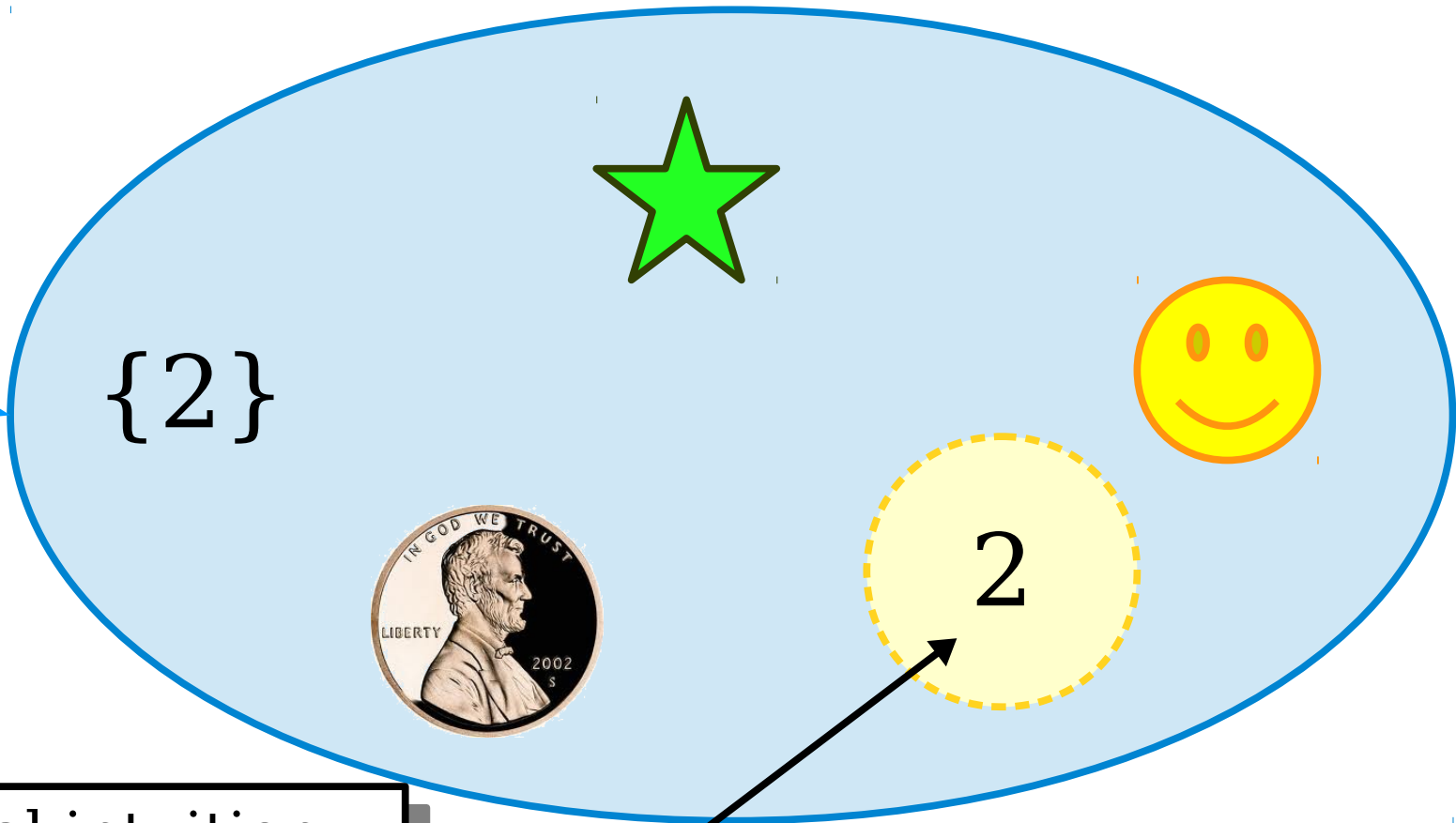


2

$$\{2\} \subseteq S$$

# Subsets and Elements

Set  $S$

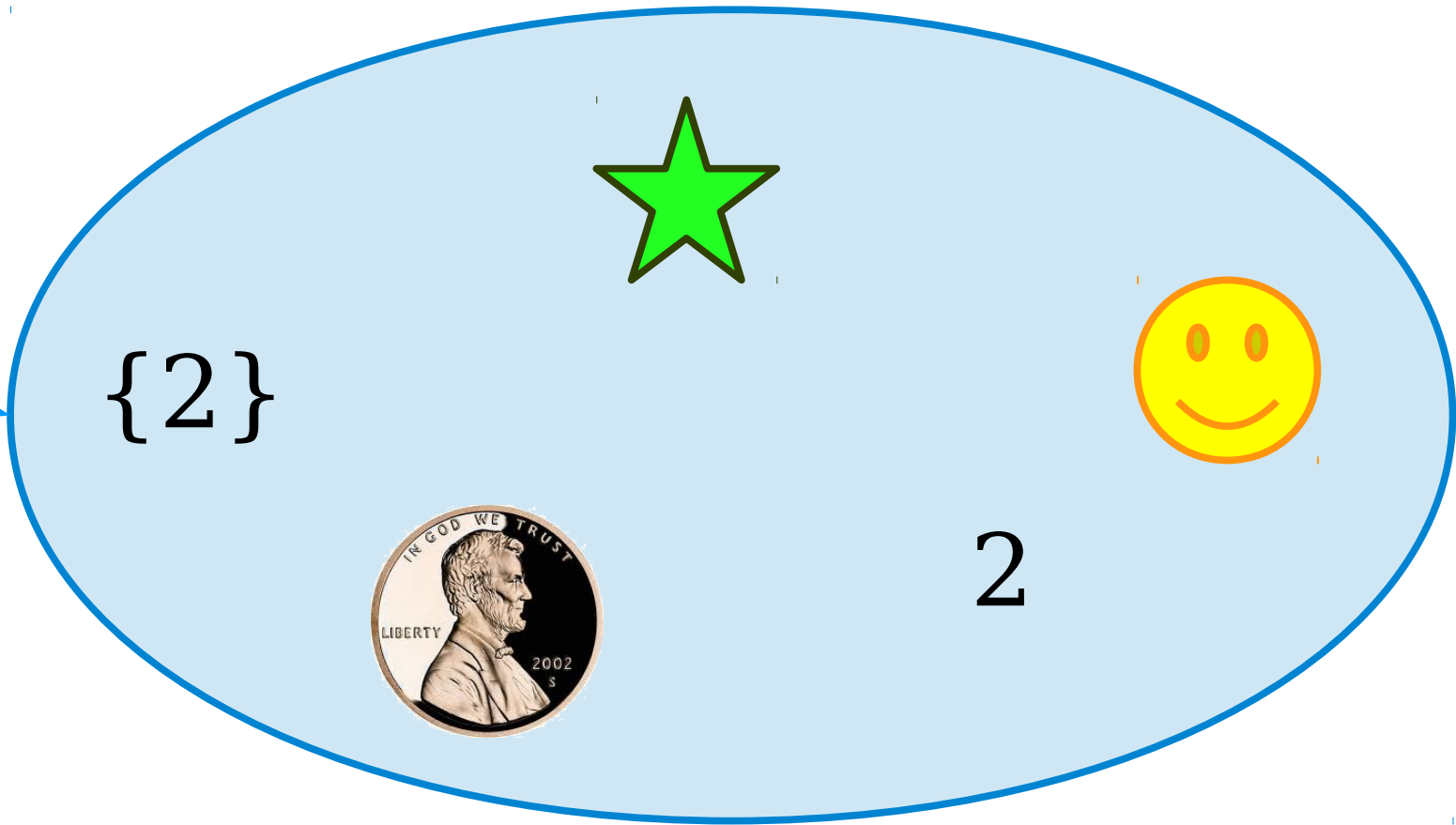


General intuition:  
 $A \subseteq B$  if you can  
form  $A$  by ***circling***  
***elements of B.***

$$\{2\} \subseteq S$$

# Subsets and Elements

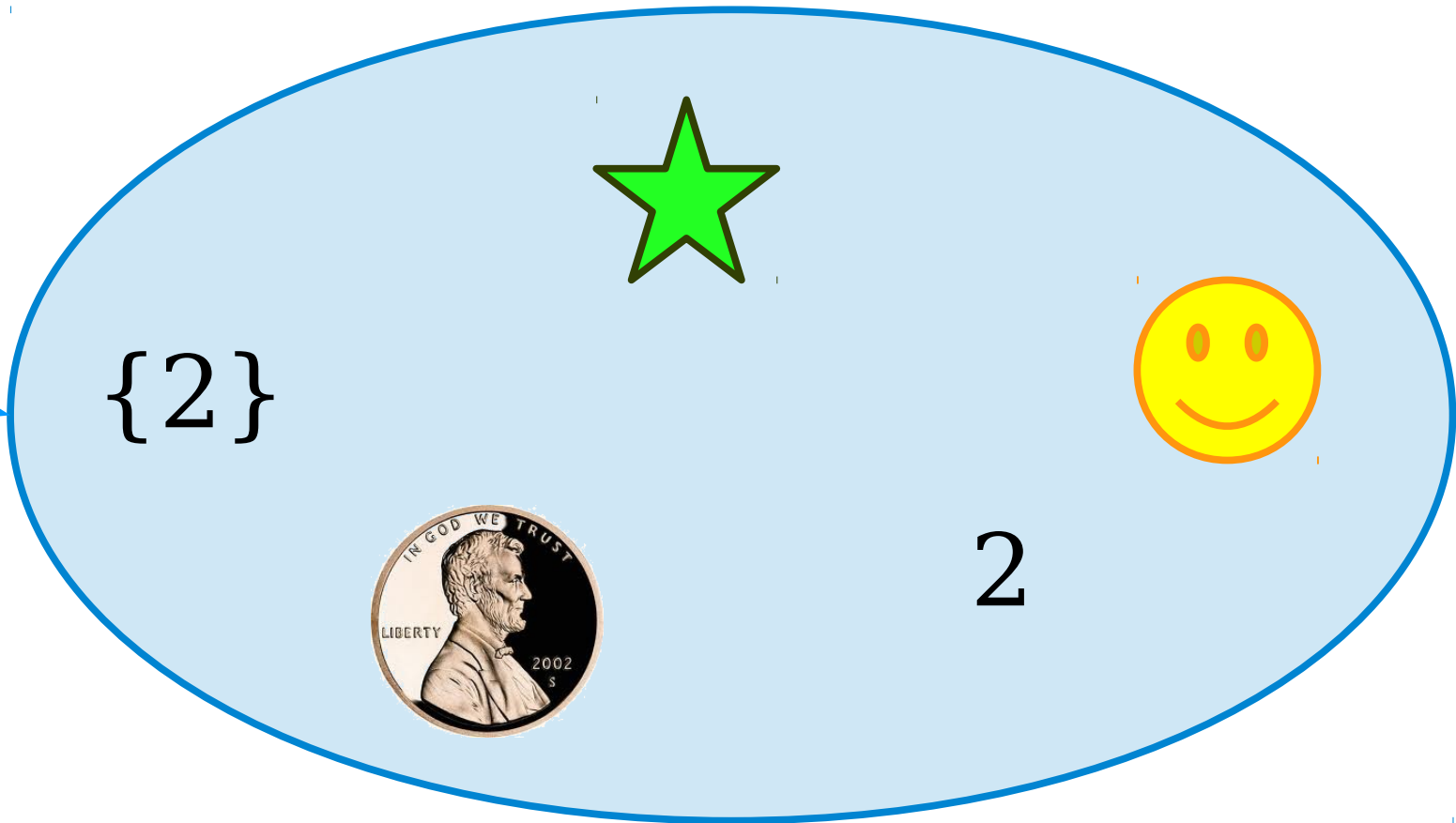
Set  $S$



$2 \notin S$

# Subsets and Elements

Set  $S$

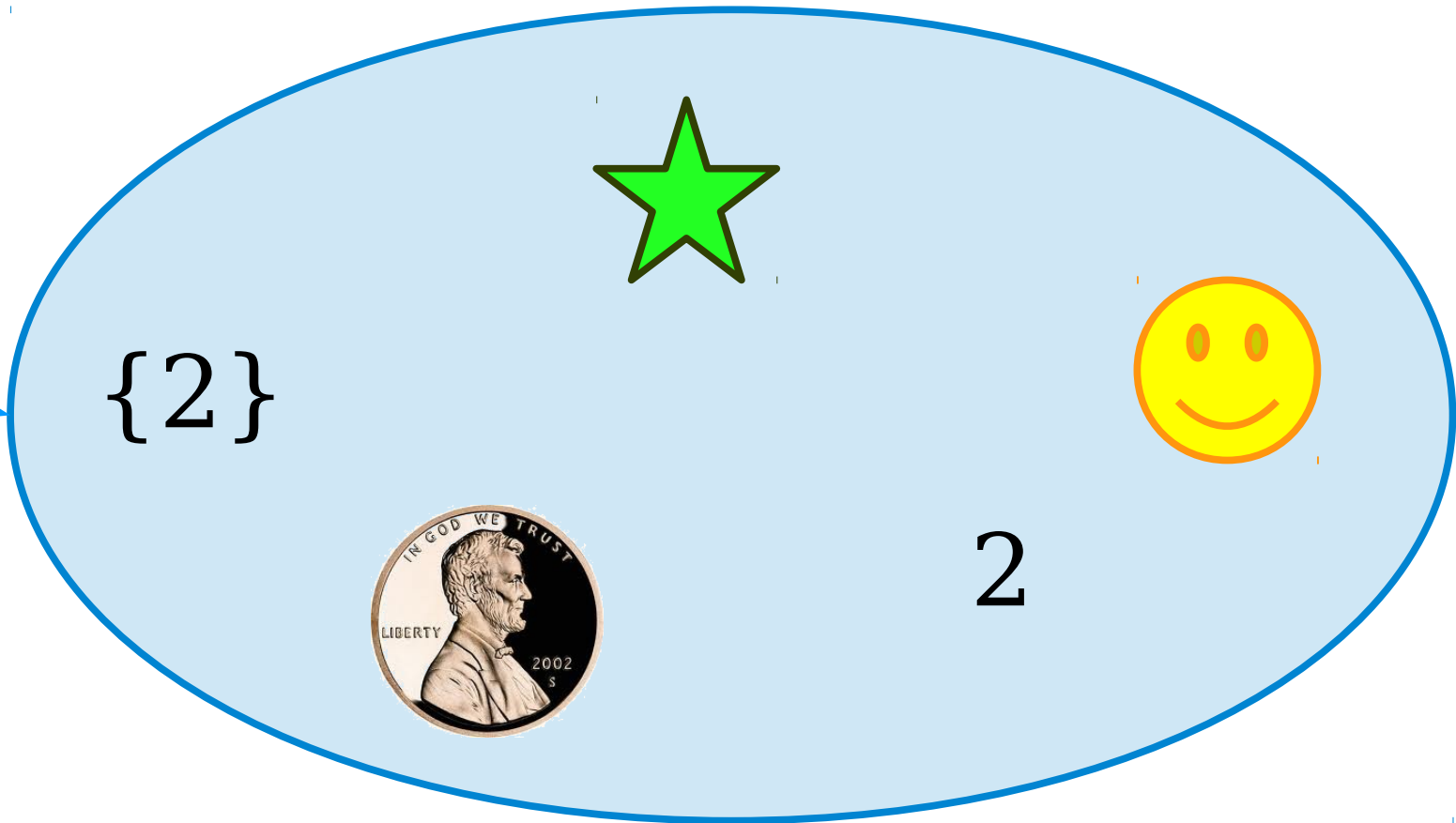


$$2 \notin S$$

(Since 2  
isn't a  
set.)

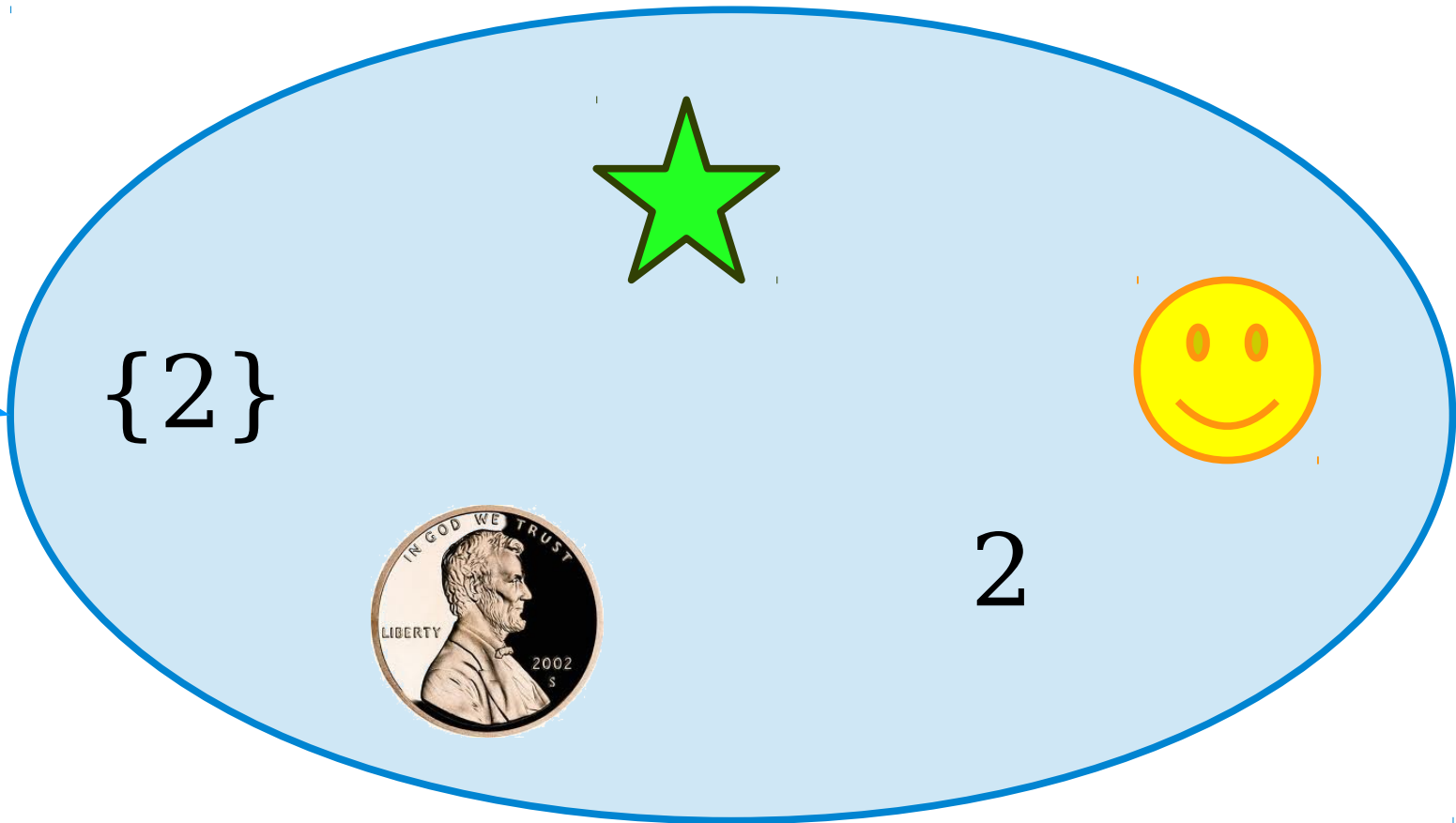
# Subsets and Elements

Set  $S$



# Subsets and Elements

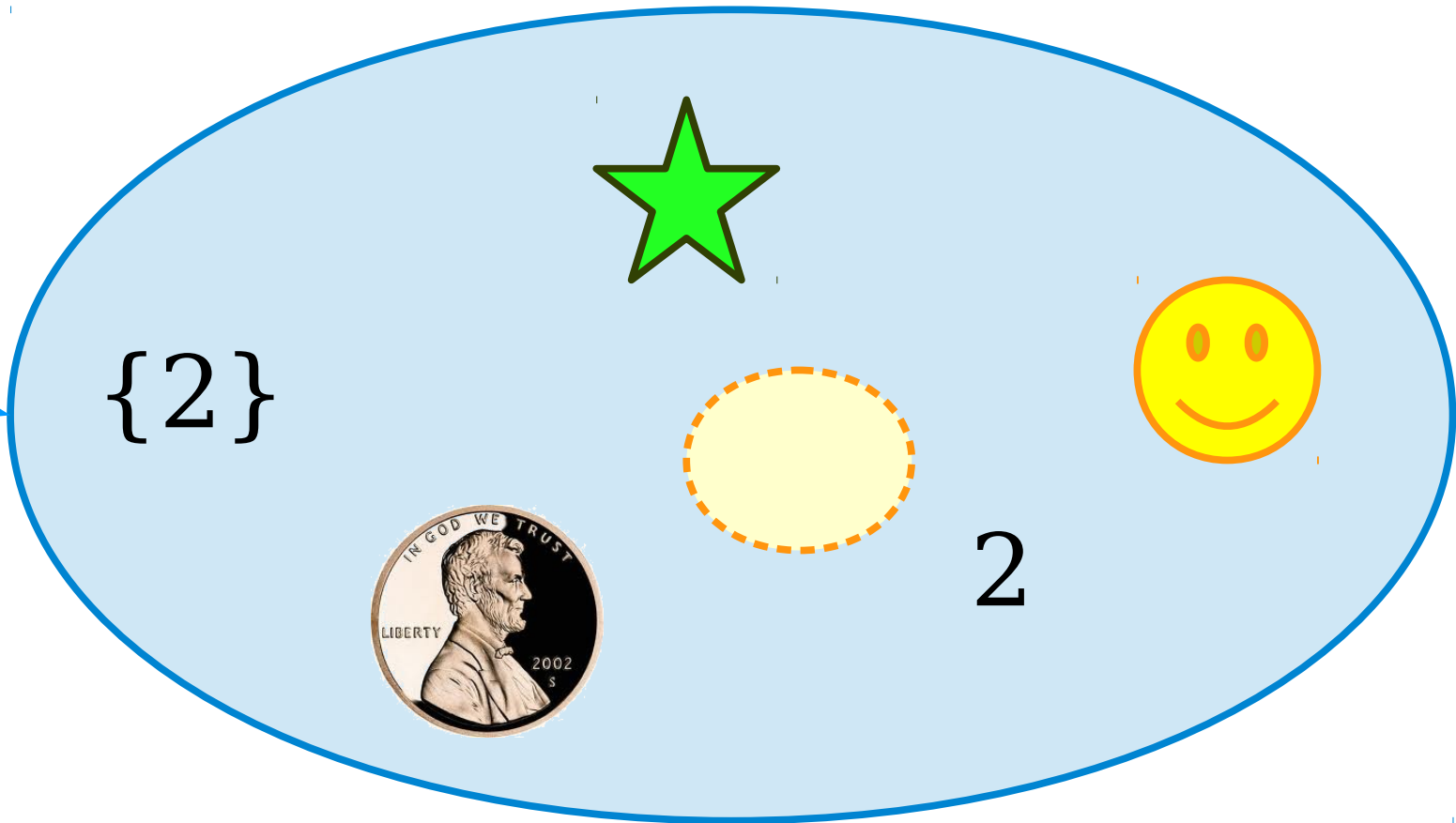
Set  $S$



$$\emptyset \subseteq S$$

# Subsets and Elements

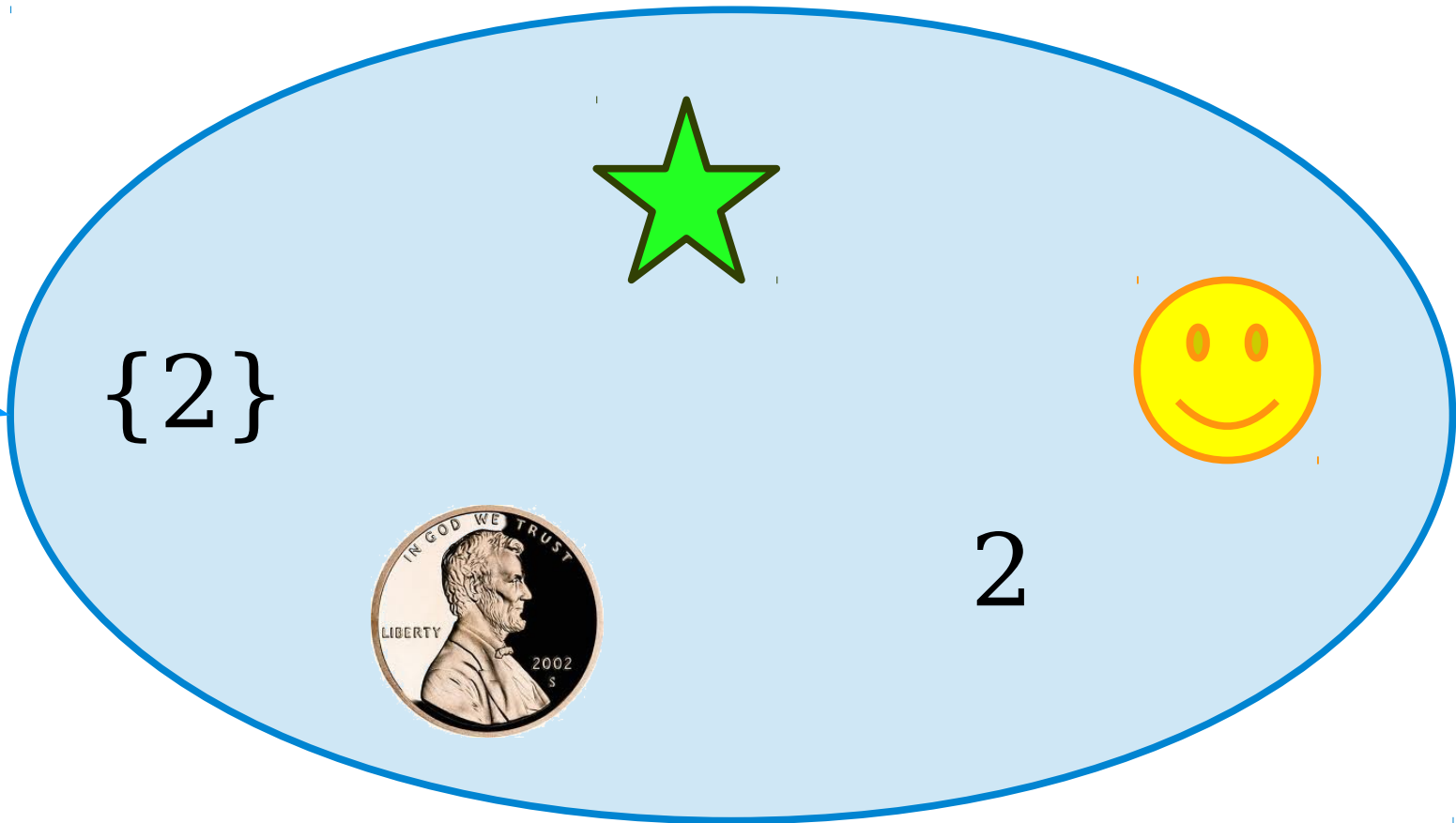
Set  $S$



$$\emptyset \subseteq S$$

# Subsets and Elements

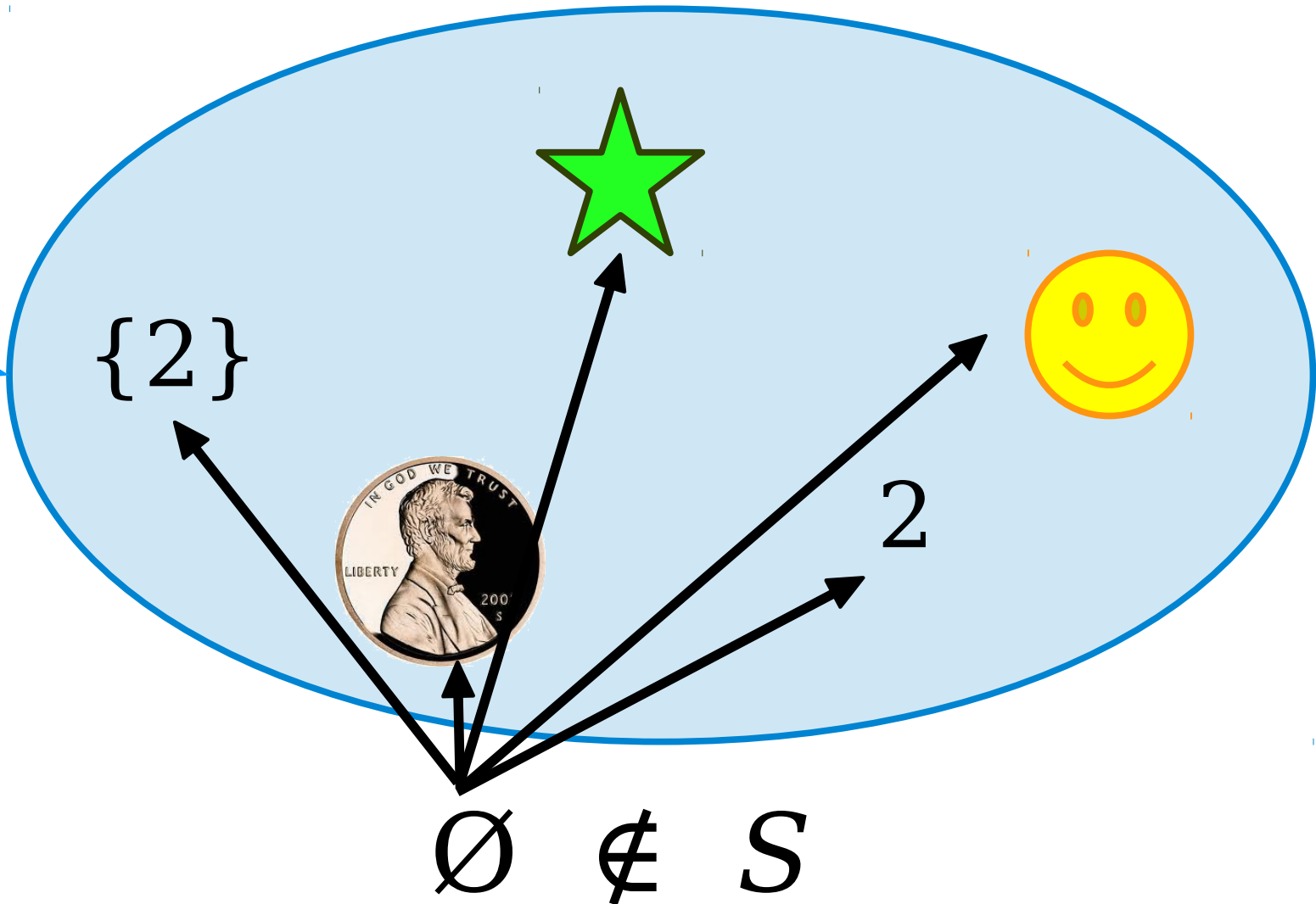
Set  $S$



$\emptyset \notin S$

# Subsets and Elements

Set  $S$



# Subsets and Elements

- We say that  $S \in T$  if, among the elements of  $T$ , one of them is *exactly* the object  $S$ .
- We say that  $S \subseteq T$  if  $S$  is a set and every element of  $S$  is also an element of  $T$ . ( $S$  has to be a set for the statement  $S \subseteq T$  to be true.)
- Although these concepts are similar, ***they are not the same!*** Not all elements of a set are subsets of that set and vice-versa.
- We have a resource on the course website, the Guide to Elements and Subsets, that explores this in more depth.

$$S = \left\{ \text{Lincoln Penny}, \text{Lincoln Dime} \right\}$$

$$\wp(S) = \left\{ \emptyset, \left\{ \text{Lincoln Dime} \right\}, \left\{ \text{Lincoln Penny} \right\}, \left\{ \text{Lincoln Penny}, \text{Lincoln Dime} \right\} \right\}$$

This is the **power set** of  $S$ , the set of all subsets of  $S$ . We write the power set of  $S$  as  $\wp(S)$ .

Formally,  $\wp(S) = \{ T \mid T \subseteq S \}$ .  
*(Do you see why?)*

**Question:** What is  $\wp(\emptyset)$ ?

**Respond at [pollev.com/cs103](https://pollev.com/cs103)**

*(Enter your name as it appears in Axxess in order to count towards your participation)*

What is  $\wp(\emptyset)$ ?

**Answer:**  $\{\emptyset\}$

*Remember that  $\emptyset \neq \{\emptyset\}$ !*

# Cardinality

# Cardinality

- The ***cardinality*** of a set is the number of elements it contains.
- If  $S$  is a set, we denote its cardinality as  $|S|$ .
- Examples:
  - $|\{\textit{whimsy}, \textit{mirth}\}| = 2$
  - $|\{\{a, b\}, \{c, d, e, f, g\}, \{h\}\}| = 3$
  - $|\{1, 2, 3, 3, 3, 3, 3\}| = 3$
  - $|\{n \in \mathbb{N} \mid n < 4\}| = |\{0, 1, 2, 3\}| = 4$
  - $|\emptyset| = 0$
  - $|\{\emptyset\}| = 1$

# The Cardinality of $\mathbb{N}$

- What is  $|\mathbb{N}|$ ?
  - There are infinitely many natural numbers.
  - $|\mathbb{N}|$  can't be a natural number, since it's infinitely large.

# The Cardinality of $\mathbb{N}$

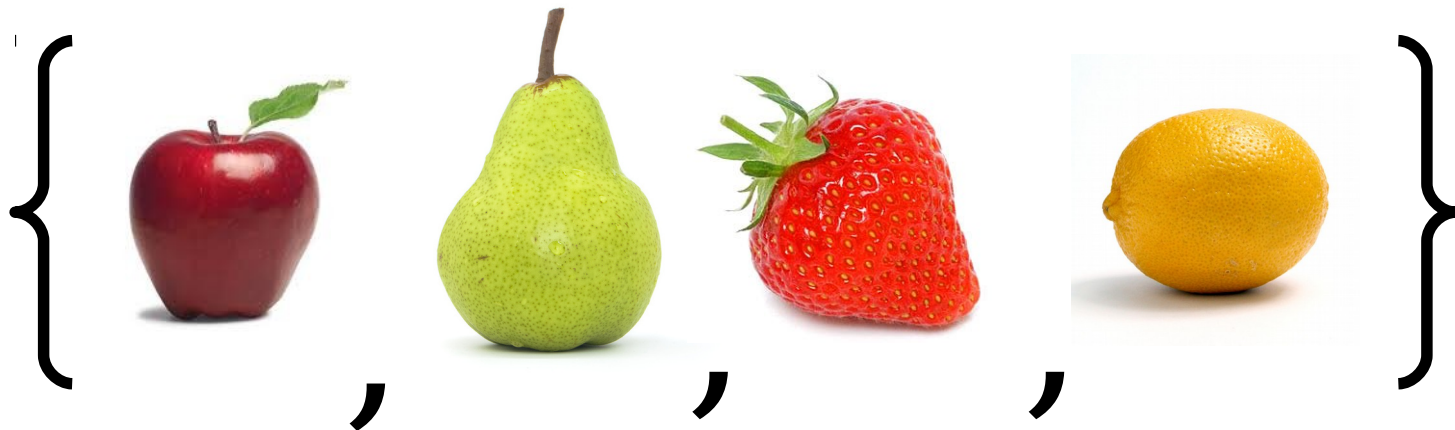
- What is  $|\mathbb{N}|$ ?
  - There are infinitely many natural numbers.
  - $|\mathbb{N}|$  can't be a natural number, since it's infinitely large.
- We need to introduce a new term.
- Let's define  $\aleph_0 = |\mathbb{N}|$ .
  - $\aleph_0$  is pronounced “aleph-zero,” “aleph-nought,” or “aleph-null.”

Consider the set

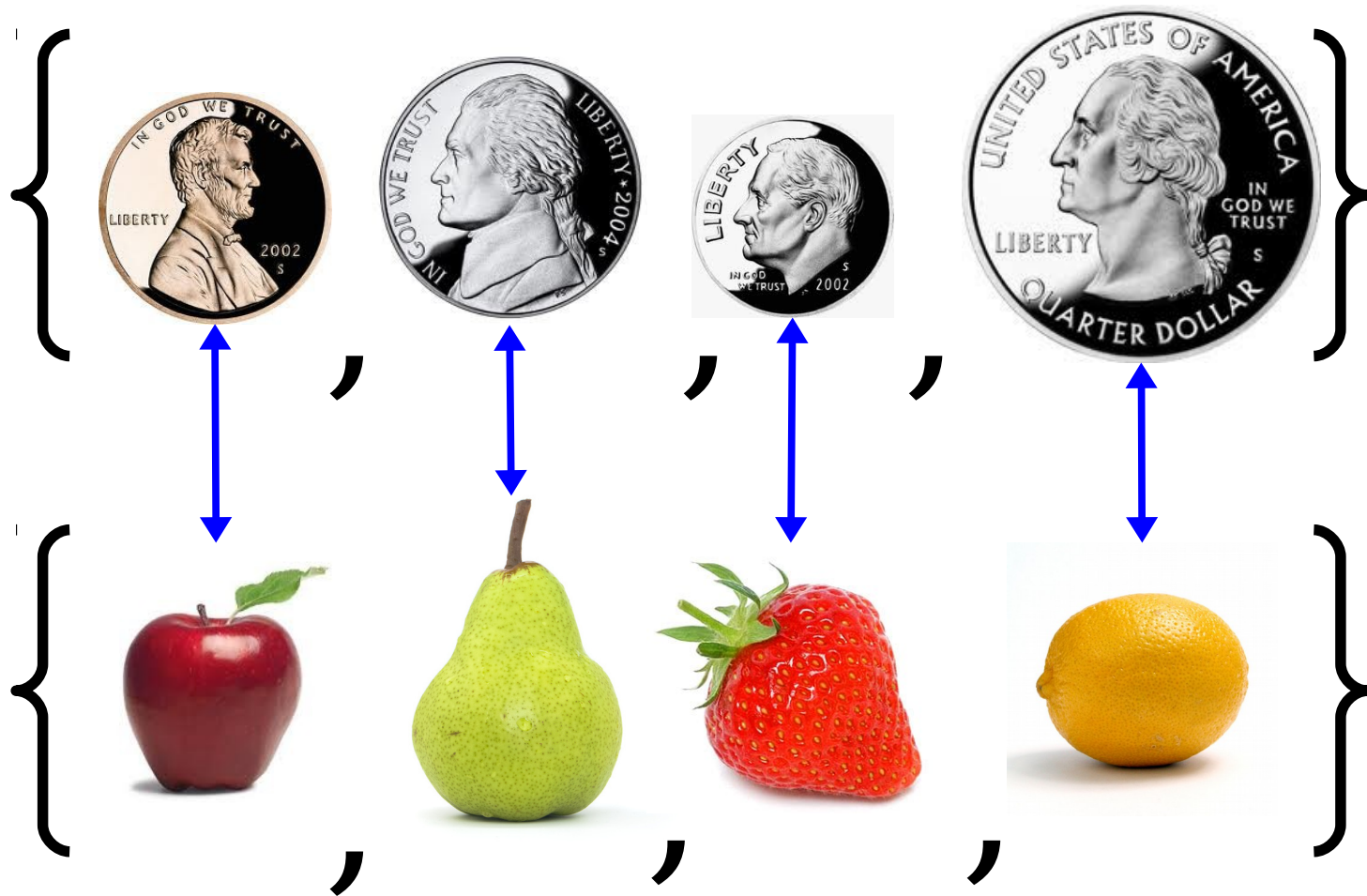
$$S = \{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}.$$

What is  $|S|$ ?

# How Big Are These Sets?

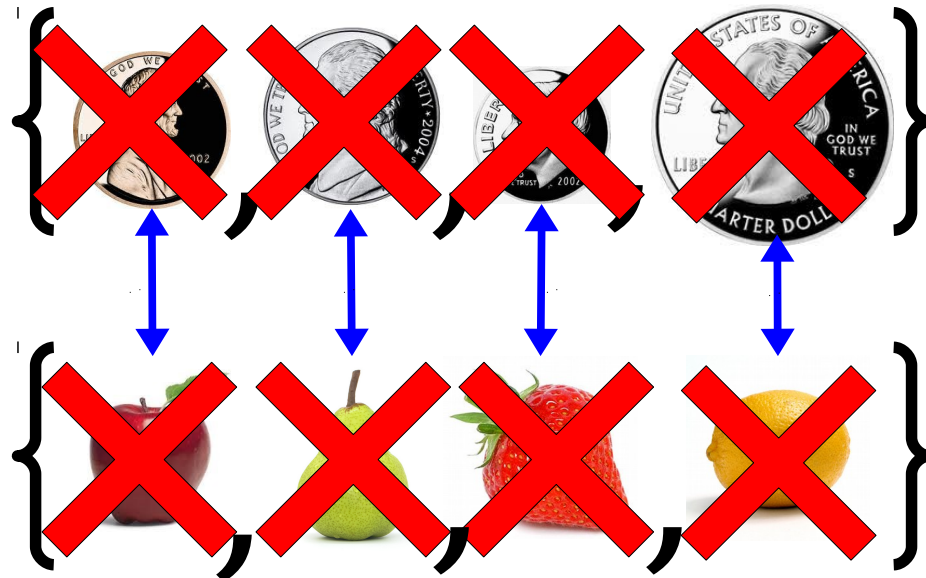


# How Big Are These Sets?



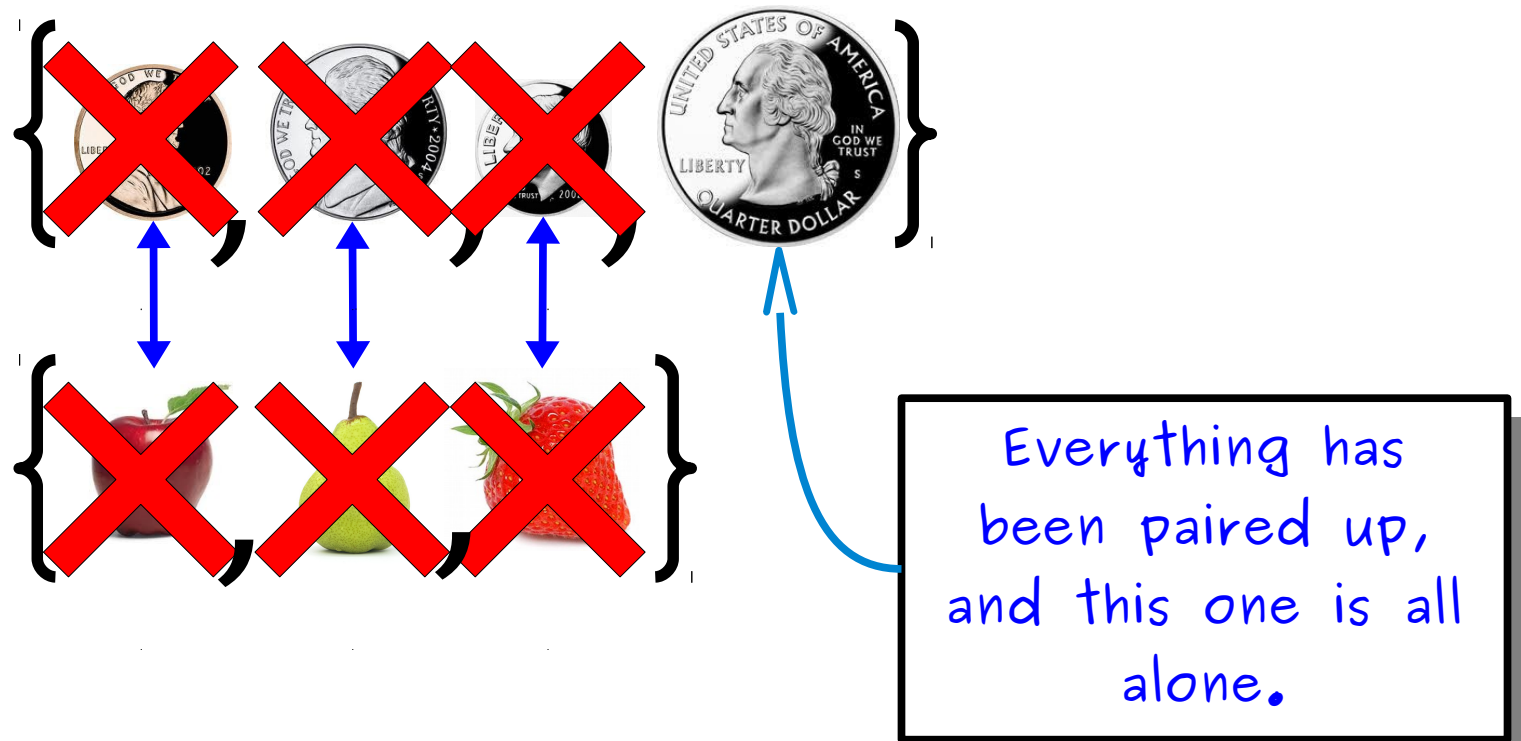
# Comparing Cardinalities

- *By definition*, two sets have the same size if there is a way to pair their elements off without leaving any elements uncovered.
- The intuition:

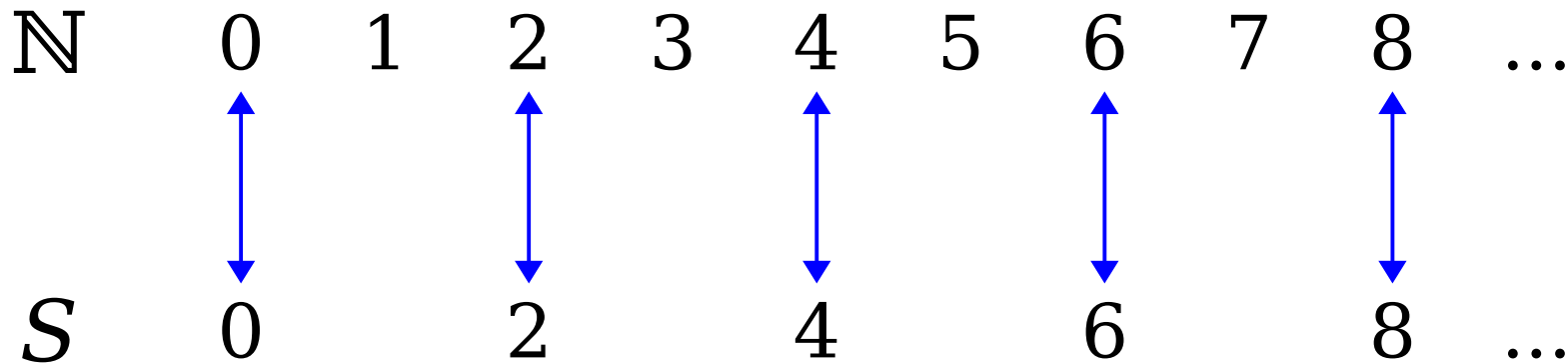


# Comparing Cardinalities

- *By definition*, two sets have the same size if there is a way to pair their elements off without leaving any elements uncovered.
- The intuition:



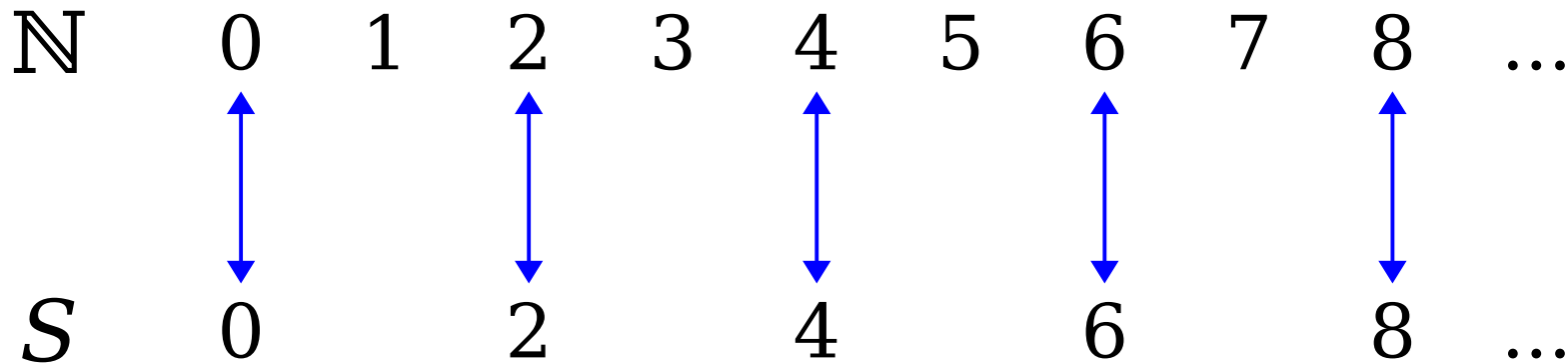
# Infinite Cardinalities



$$S = \{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$$

Two sets have the same size if there is a way to pair their elements off without leaving any elements uncovered

# Infinite Cardinalities



$$S = \{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$$

Two sets have the same size if *there is a way* to pair their elements off without leaving any elements uncovered

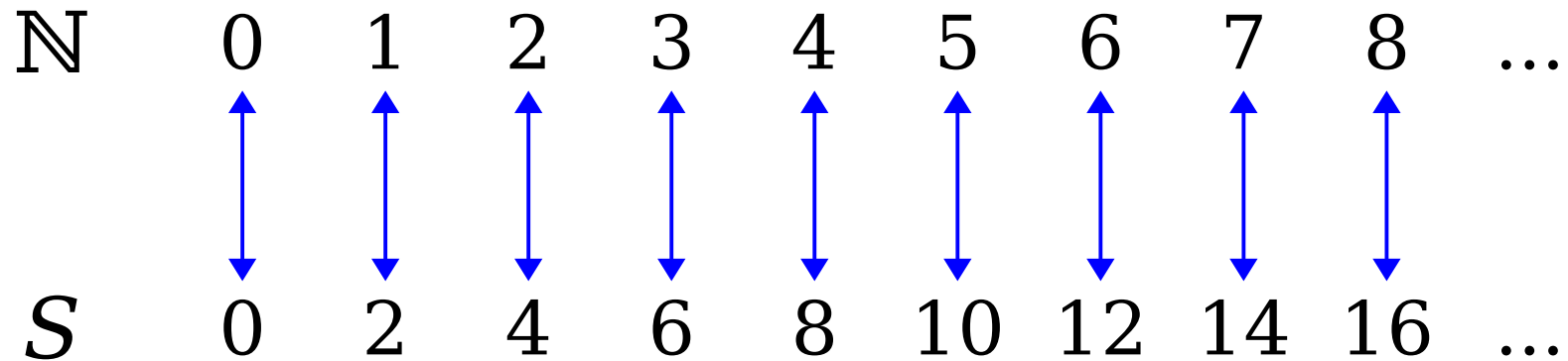
# Infinite Cardinalities

$\mathbb{N}$     0    1    2    3    4    5    6    7    8    ...

$S$     0            2            4            6            8    ...

$$S = \{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$$

# Infinite Cardinalities



$$n \leftrightarrow 2n$$

$$S = \{ n \mid n \in \mathbb{N} \text{ and } n \text{ is even} \}$$

$$|S| = |\mathbb{N}| = \aleph_0$$

# Infinite Cardinalities

$\mathbb{N}$     0    1    2    3    4    5    6    7    8    ...

$\mathbb{Z}$     ...    -3    -2    -1    0    1    2    3    4    ...

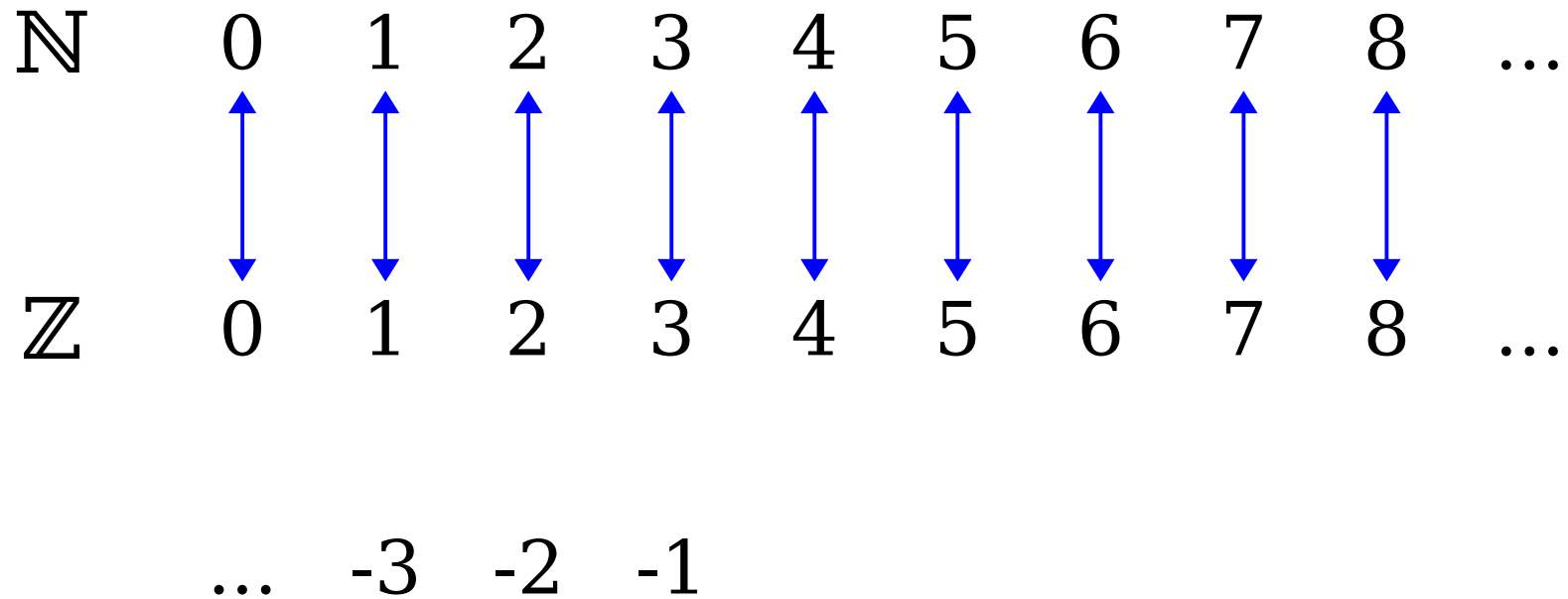
# Infinite Cardinalities

$\mathbb{N}$     0    1    2    3    4    5    6    7    8    ...

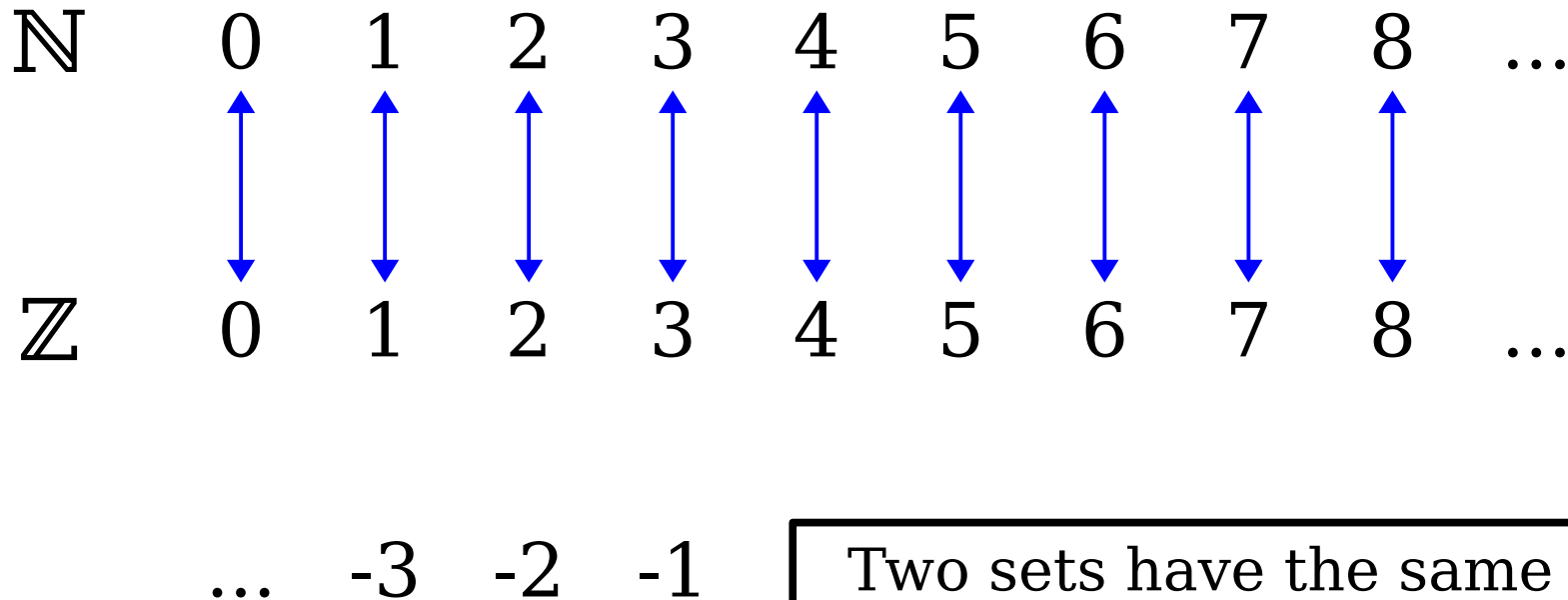
$\mathbb{Z}$                     0    1    2    3    4    ...

...    -3    -2    -1

# Infinite Cardinalities



# Infinite Cardinalities



Two sets have the same size if *there is a way* to pair their elements off without leaving any elements uncovered

# Infinite Cardinalities

$\mathbb{N}$     0    1    2    3    4    5    6    7    8    ...

$\mathbb{Z}$     ...    -3    -2    -1    0    1    2    3    4    ...

# Infinite Cardinalities

$\mathbb{N}$     0    1    2    3    4    5    6    7    8    ...

$\mathbb{Z}$

...   -3   -2   -1   0   1   2   3   4   ...

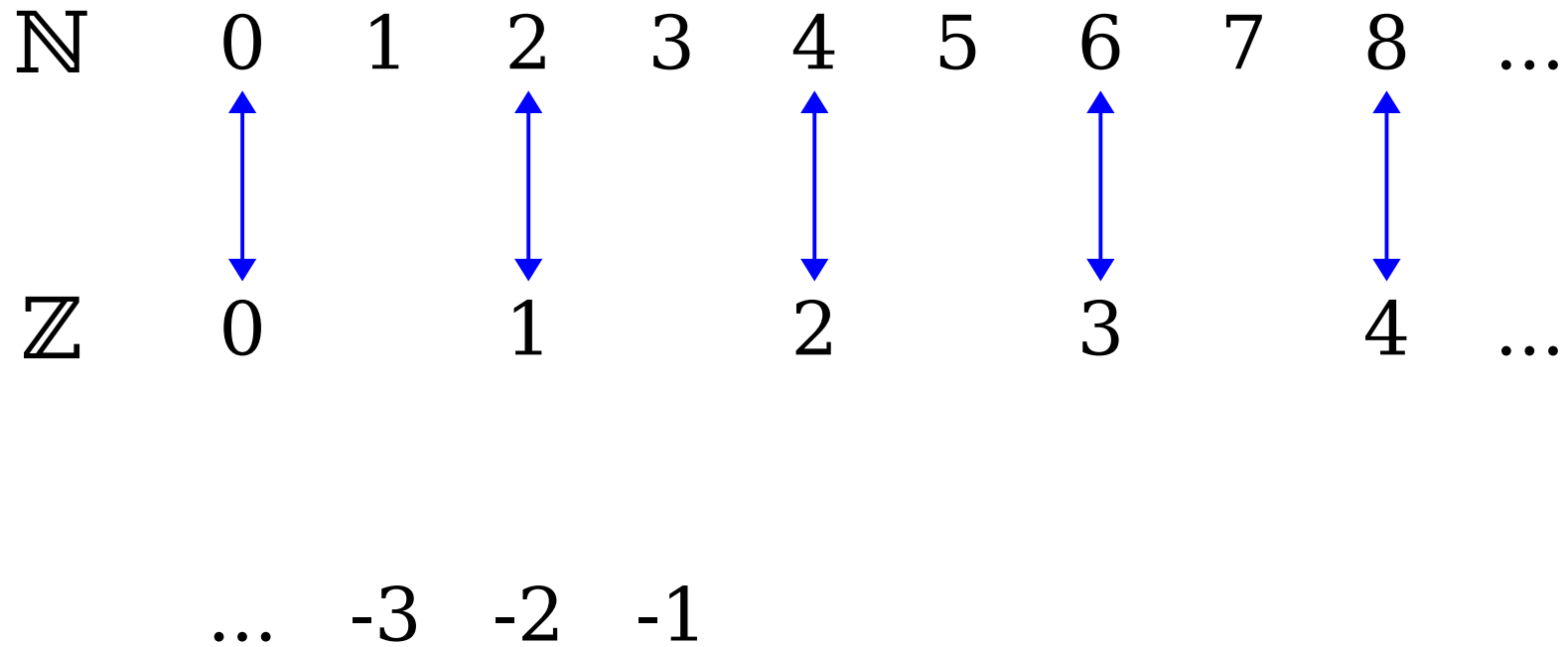
# Infinite Cardinalities

$\mathbb{N}$     0    1    2    3    4    5    6    7    8    ...

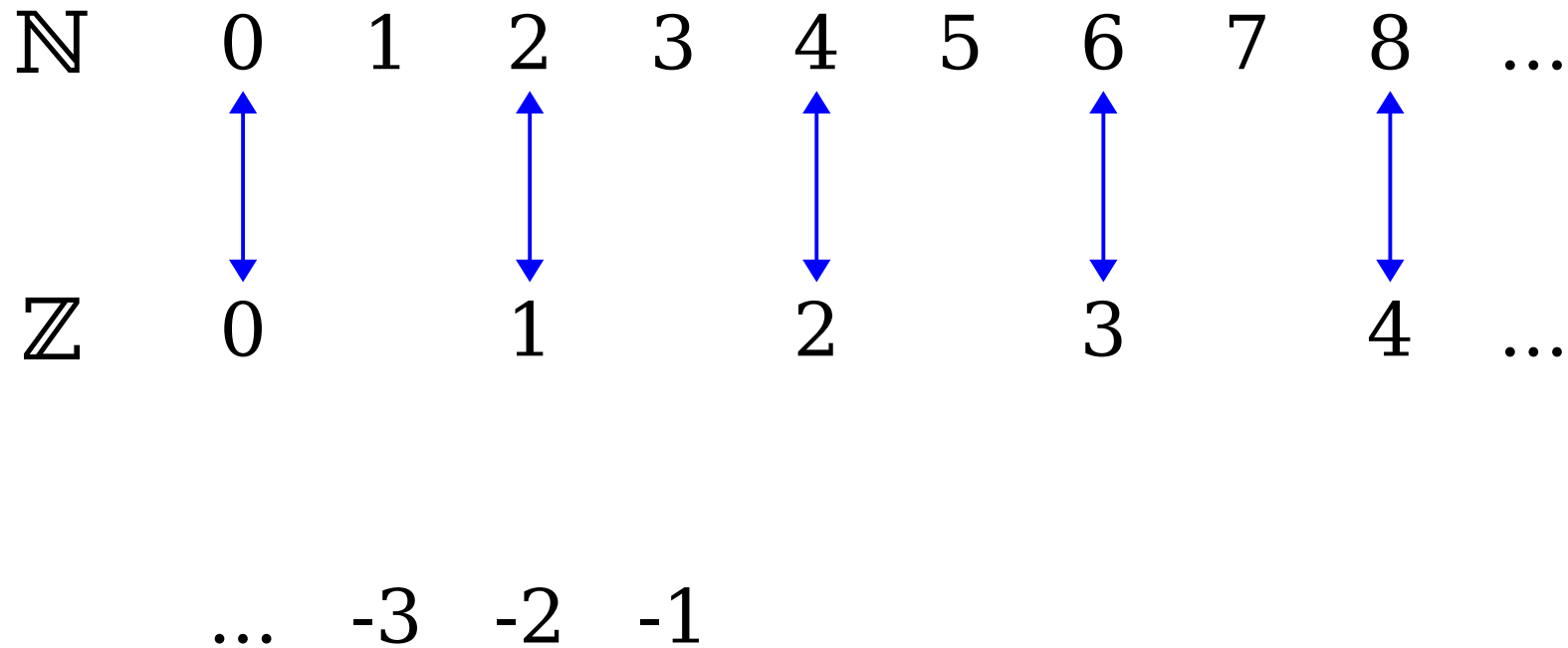
$\mathbb{Z}$     0            1            2            3            4    ...

...   -3   -2   -1

# Infinite Cardinalities

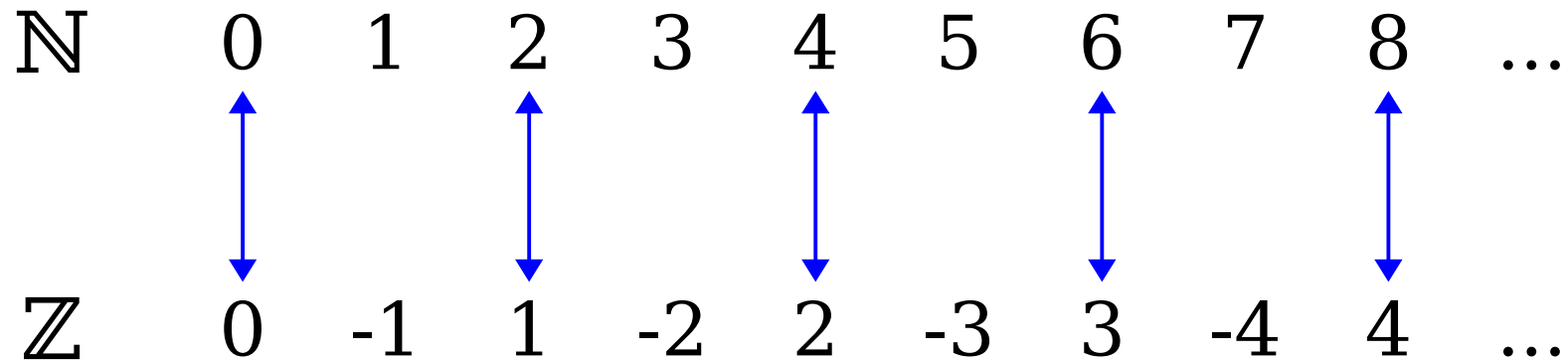


# Infinite Cardinalities



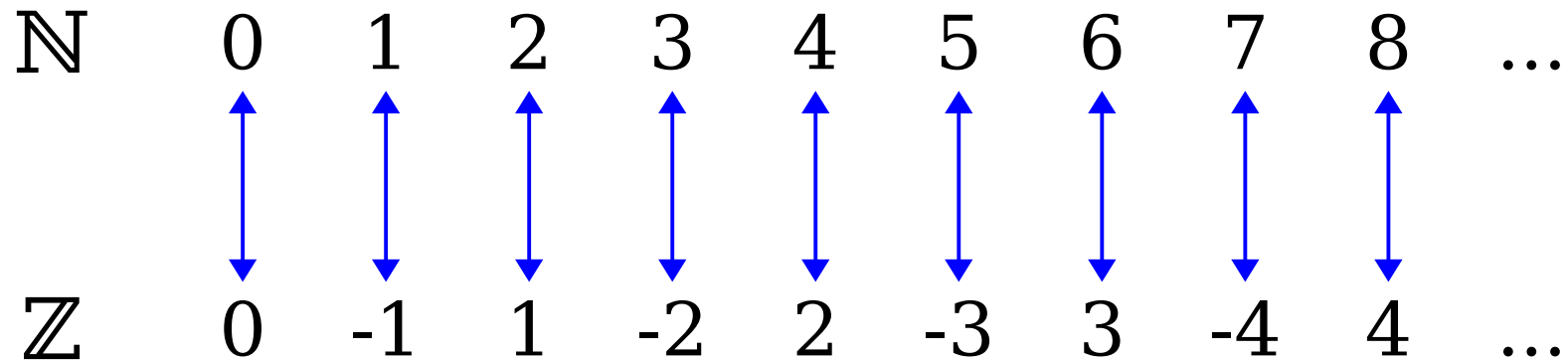
Pair nonnegative integers with even natural numbers.

# Infinite Cardinalities



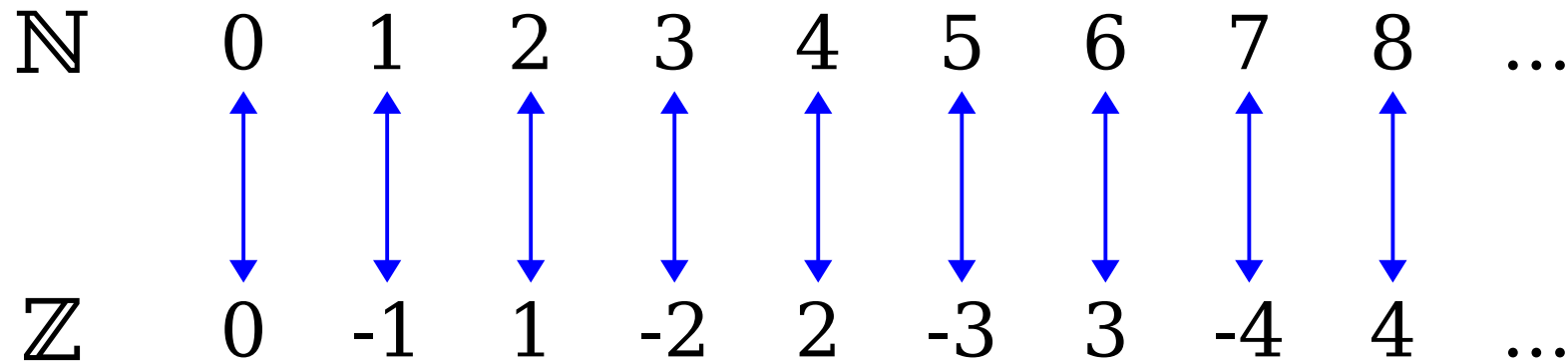
Pair nonnegative integers with even natural numbers.

# Infinite Cardinalities



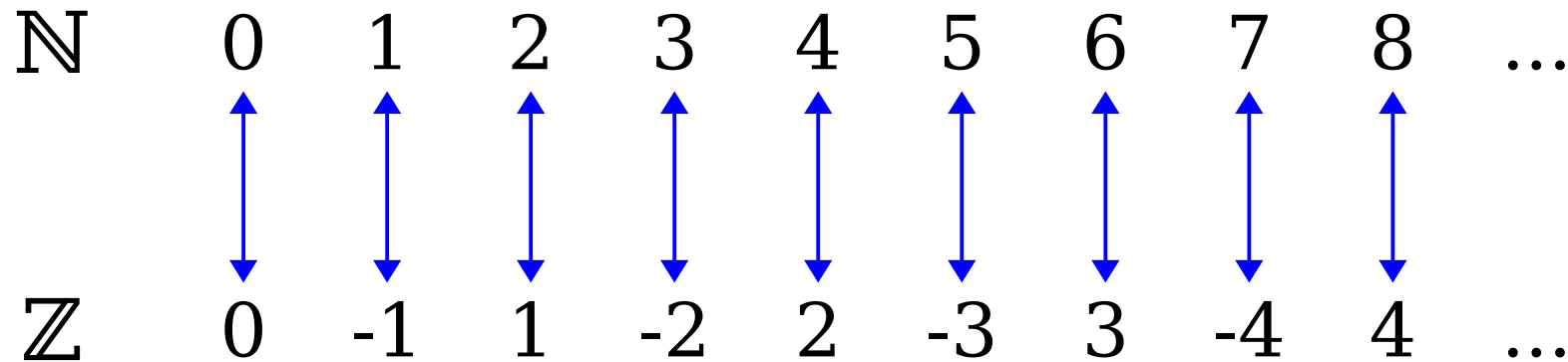
Pair nonnegative integers with even natural numbers.

# Infinite Cardinalities



Pair nonnegative integers with even natural numbers.  
Pair negative integers with odd natural numbers.

# Infinite Cardinalities



$$|\mathbb{N}| = |\mathbb{Z}| = \aleph_0$$

Pair nonnegative integers with even natural numbers.  
Pair negative integers with odd natural numbers.

***Important Question:***

Do all infinite sets have  
the same cardinality?

$$S = \left\{ \text{Lincoln Penny}, \text{Lincoln Nickel} \right\}$$

$$\wp(S) = \left\{ \emptyset, \left\{ \text{Lincoln Nickel} \right\}, \left\{ \text{Lincoln Penny} \right\}, \left\{ \text{Lincoln Penny}, \text{Lincoln Nickel} \right\} \right\}$$

$$|S| < |\wp(S)|$$

$$S = \{ \text{Lincoln Penny}, \text{Jefferson Nickel}, \text{Button} \}$$

$$\wp(S) = \{ \emptyset, \{ \text{Lincoln Penny} \}, \{ \text{Jefferson Nickel} \}, \{ \text{Button} \}, \{ \text{Lincoln Penny}, \text{Jefferson Nickel} \}, \{ \text{Lincoln Penny}, \text{Button} \}, \{ \text{Jefferson Nickel}, \text{Button} \}, \{ \text{Lincoln Penny}, \text{Jefferson Nickel}, \text{Button} \} \}$$

$$|S| < |\wp(S)|$$

$$S = \{a, b, c, d\}$$

$$\begin{aligned} \wp(S) = \{ & \\ & \emptyset, \\ & \{a\}, \{b\}, \{c\}, \{d\}, \\ \{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\} & \\ \{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}, & \\ \{a, b, c, d\} & \\ & \} \end{aligned}$$

$$|S| < |\wp(S)|$$

If  $|S|$  is infinite, what is the relation between  $|S|$  and  $|\wp(S)|$ ?

Does  $|S| = |\wp(S)|$ ?

If  $|S| = |\wp(S)|$ , we can pair up the elements of  $S$  and the elements of  $\wp(S)$  without leaving anything out.

If  $|S| = |\wp(S)|$ , we can pair up the elements of  $S$  and **the elements of  $\wp(S)$**  without leaving anything out.

If  $|S| = |\wp(S)|$ , we can pair up the elements of  $S$  and **the subsets of  $S$**  without leaving anything out.

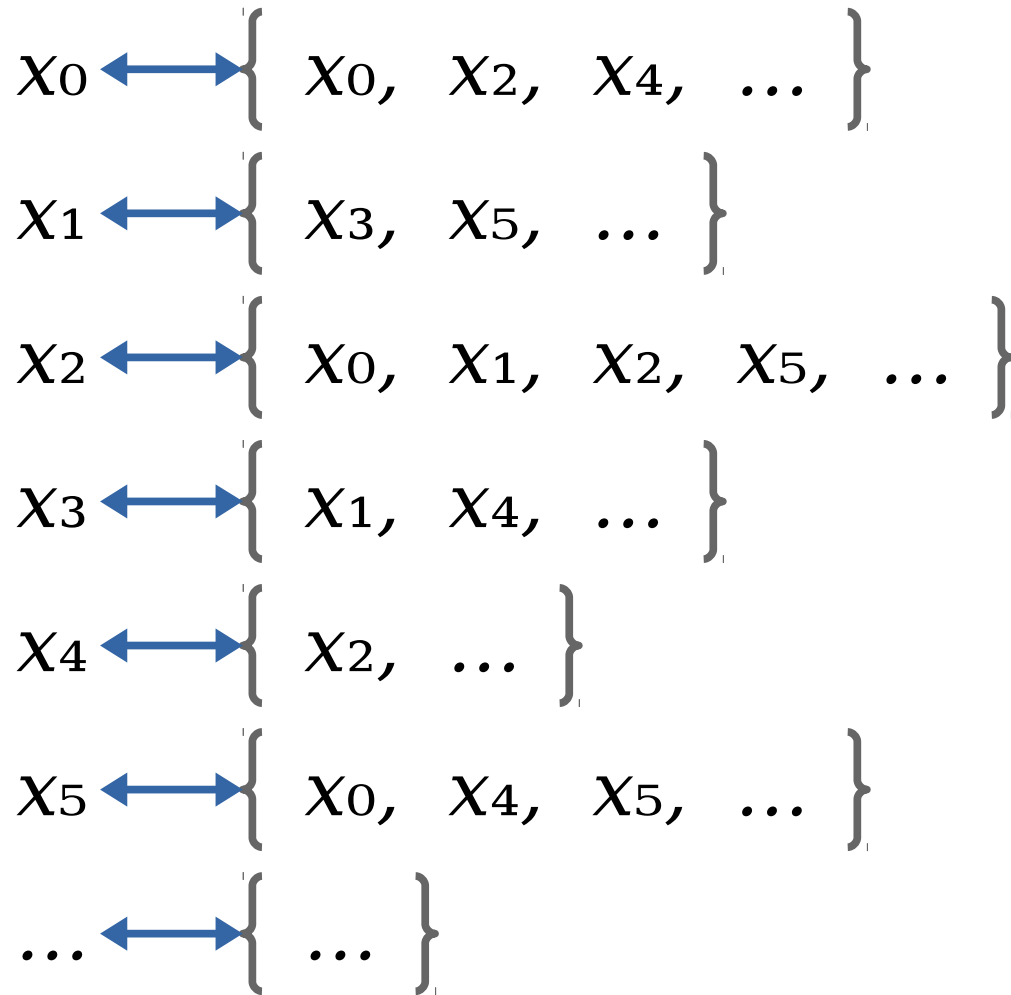
If  $|S| = |\wp(S)|$ , we can pair up the elements of  $S$  and the subsets of  $S$  without leaving anything out.

If  $|S| = |\wp(S)|$ , we can pair up the elements of  $S$  and the subsets of  $S$  without leaving anything out.

What would that look like?

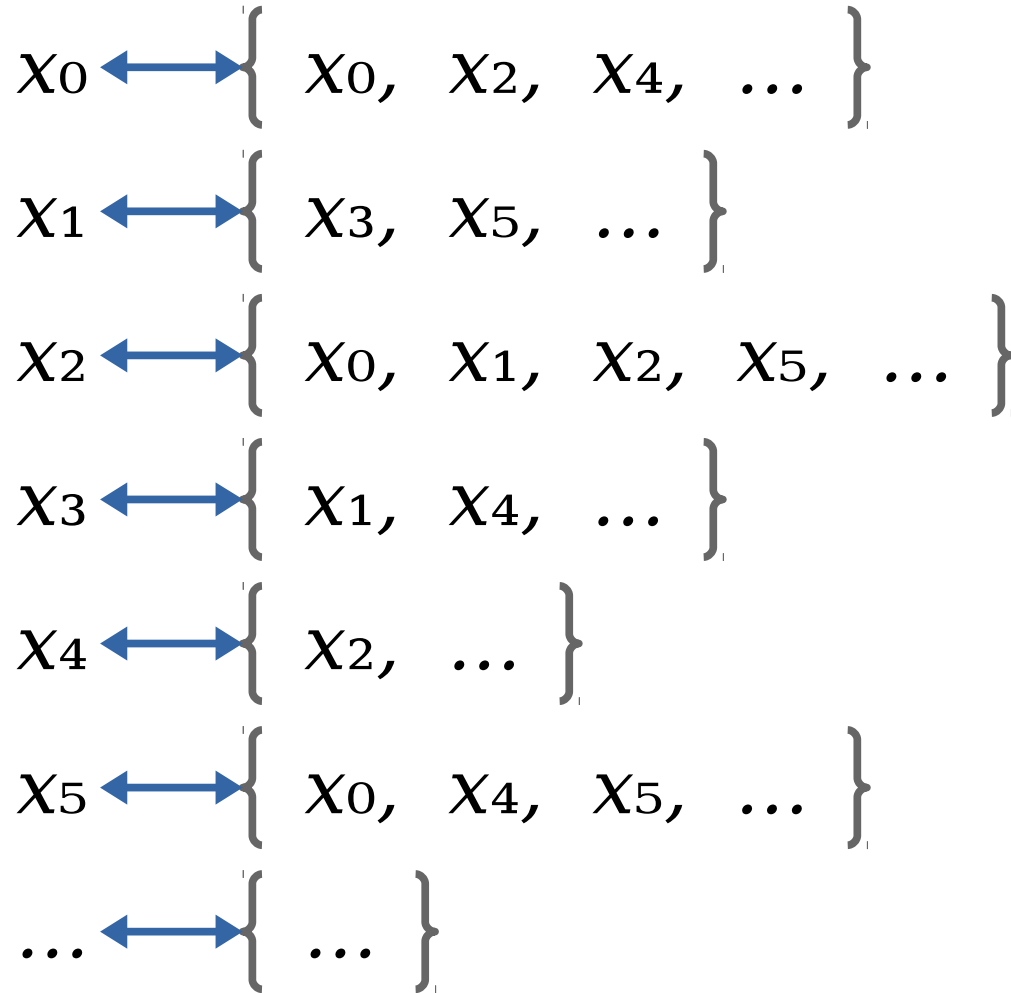
Elements  
of  $S$

Elements  
of  $\wp(S)$

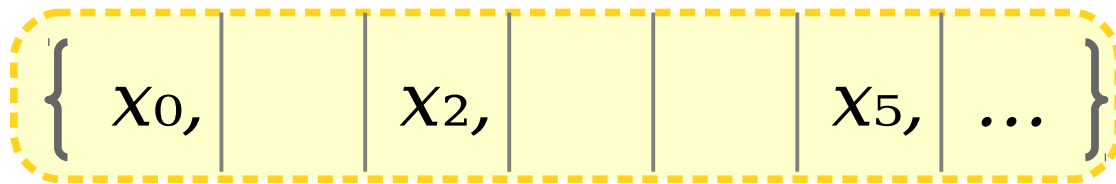
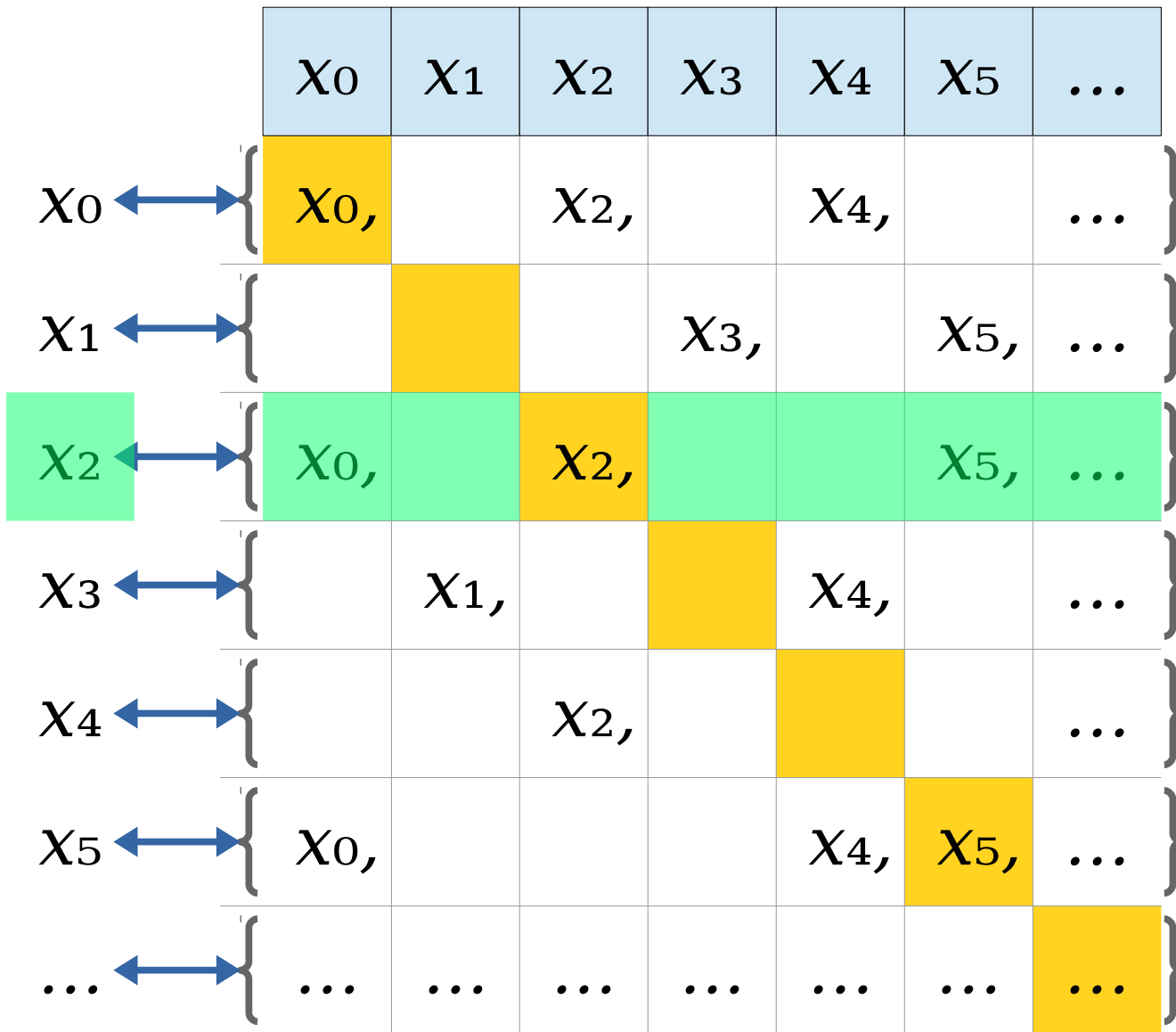


An arbitrary mapping  
between elements of  $S$   
and elements of  $\wp(S)$

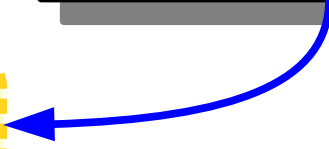
$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$\dots$
-------	-------	-------	-------	-------	-------	---------







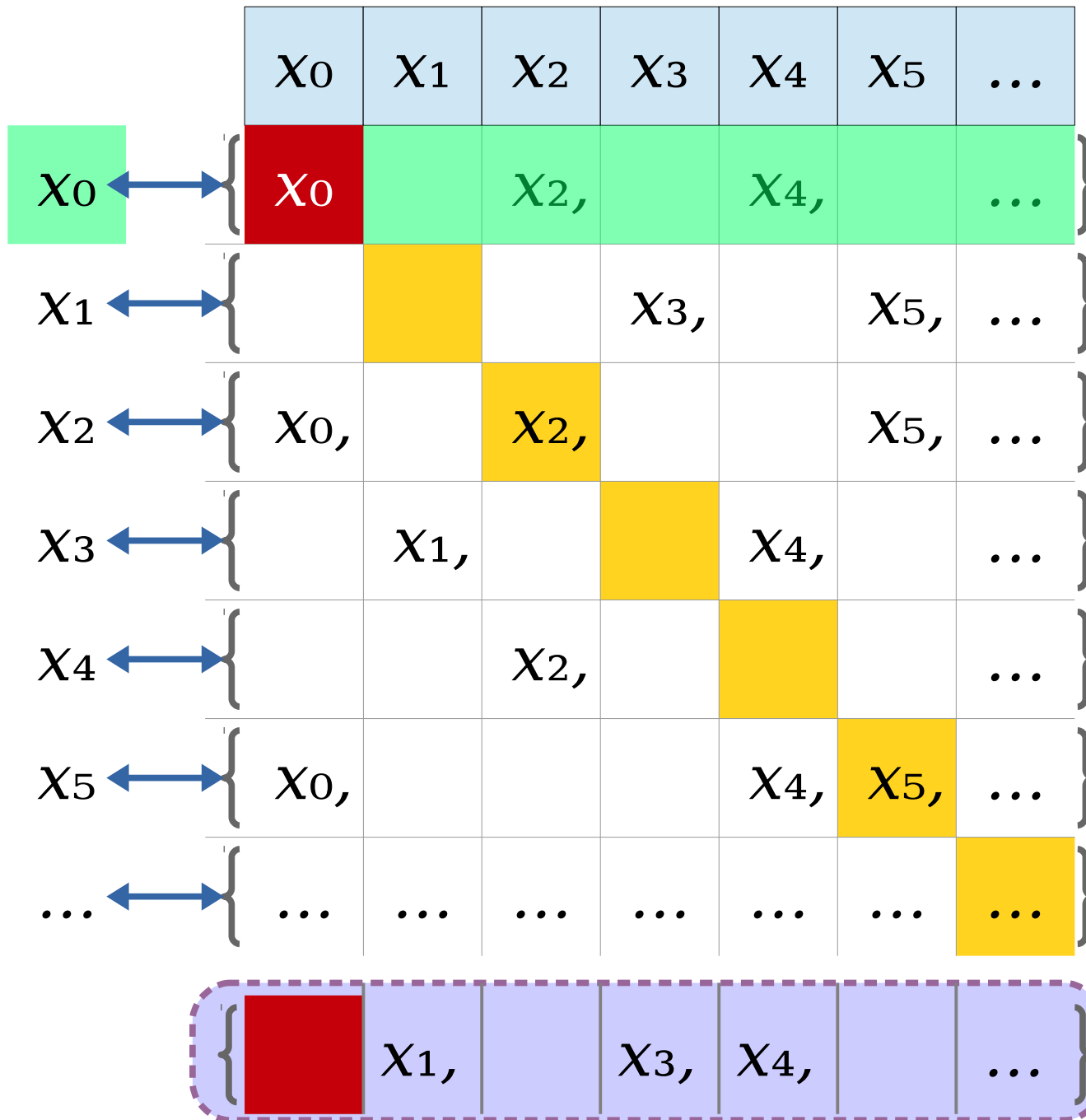
Which element is paired with this set?



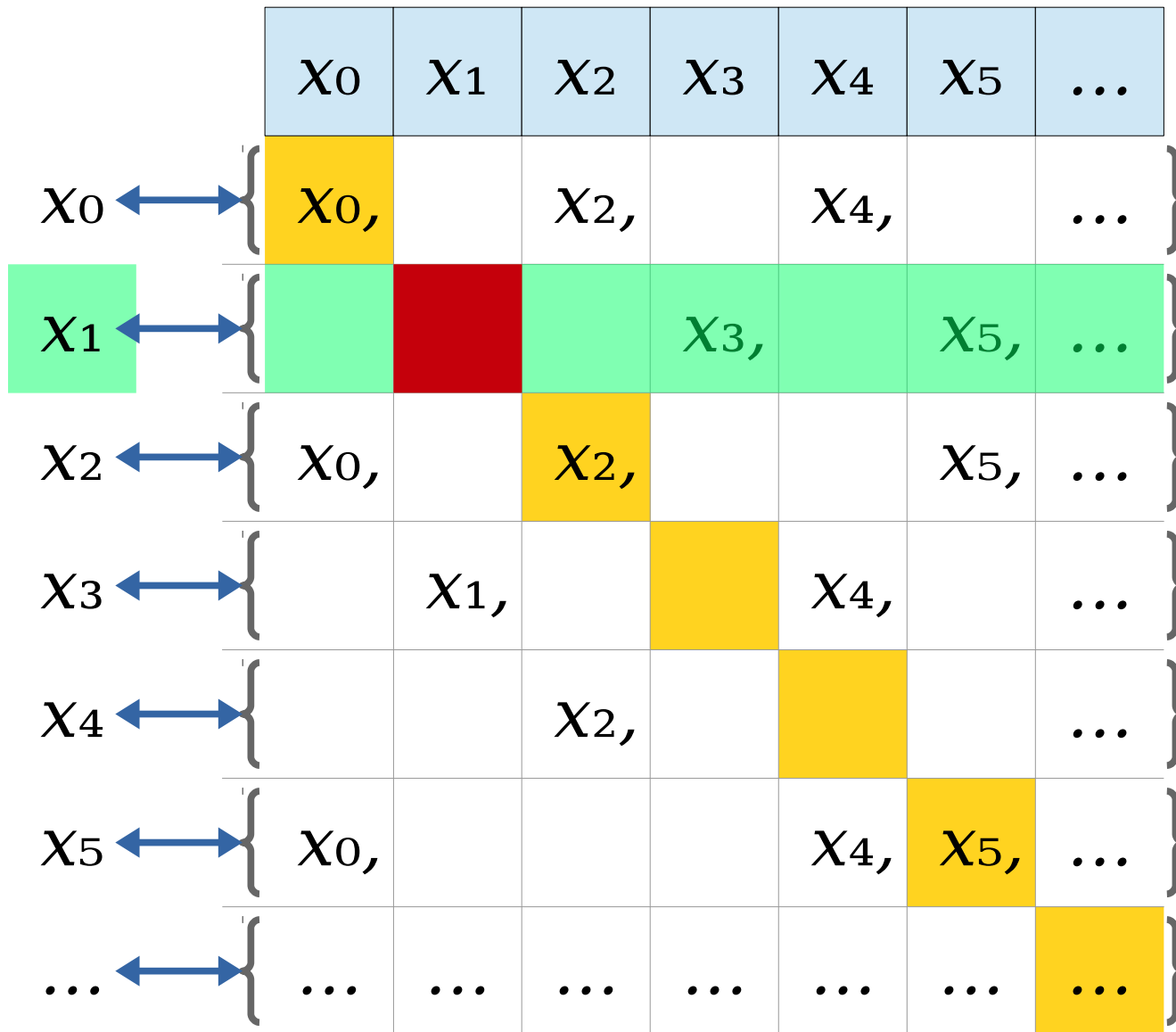
	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	...
$x_0$ ↔	$x_0,$		$x_2,$		$x_4,$		...
$x_1$ ↔				$x_3,$		$x_5,$	...
$x_2$ ↔	$x_0,$		$x_2,$			$x_5,$	...
$x_3$ ↔		$x_1,$			$x_4,$		...
$x_4$ ↔			$x_2,$				...
$x_5$ ↔	$x_0,$				$x_4,$	$x_5,$	...
...	...	...	...	...	...	...	...

"Flip" this set.  
 Swap what's  
 included and  
 what's  
 excluded.

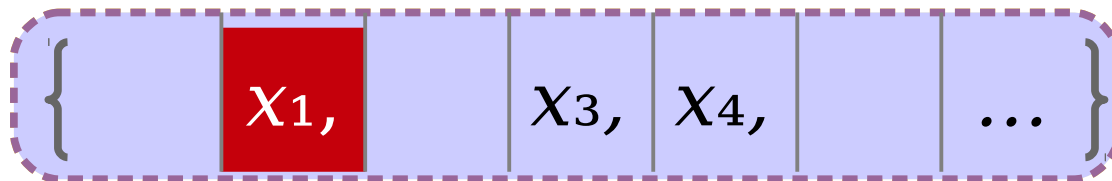
{  $x_1,$   $x_3,$   $x_4,$  ... }

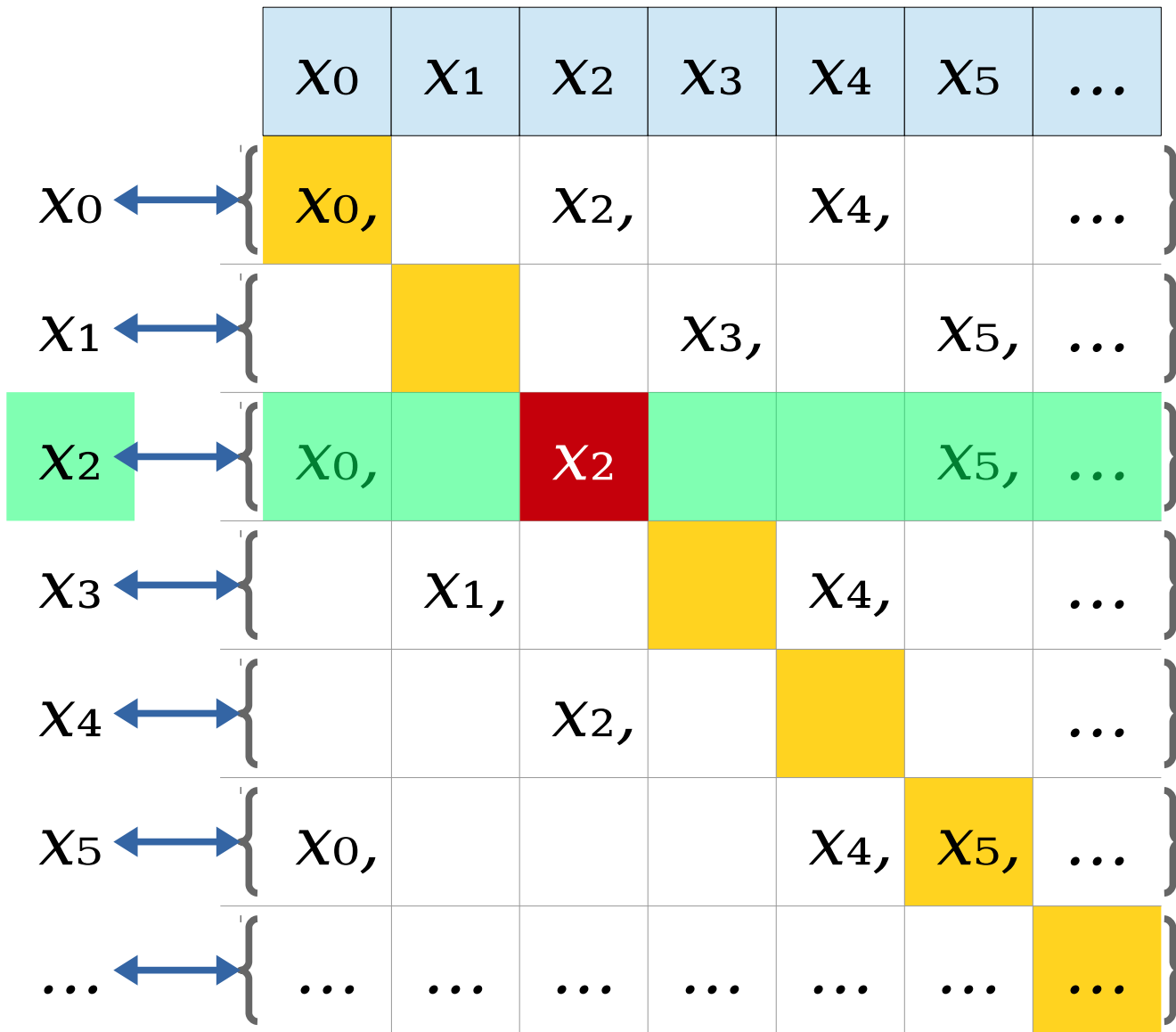


Which element is paired with this set?

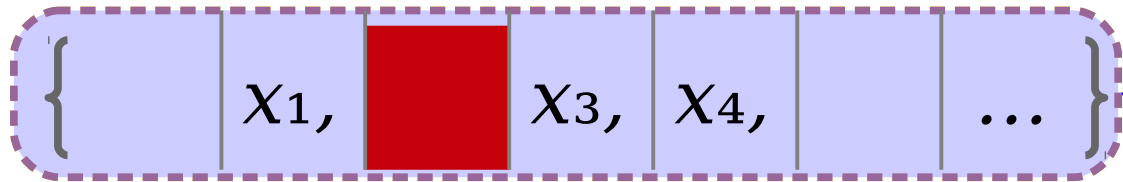


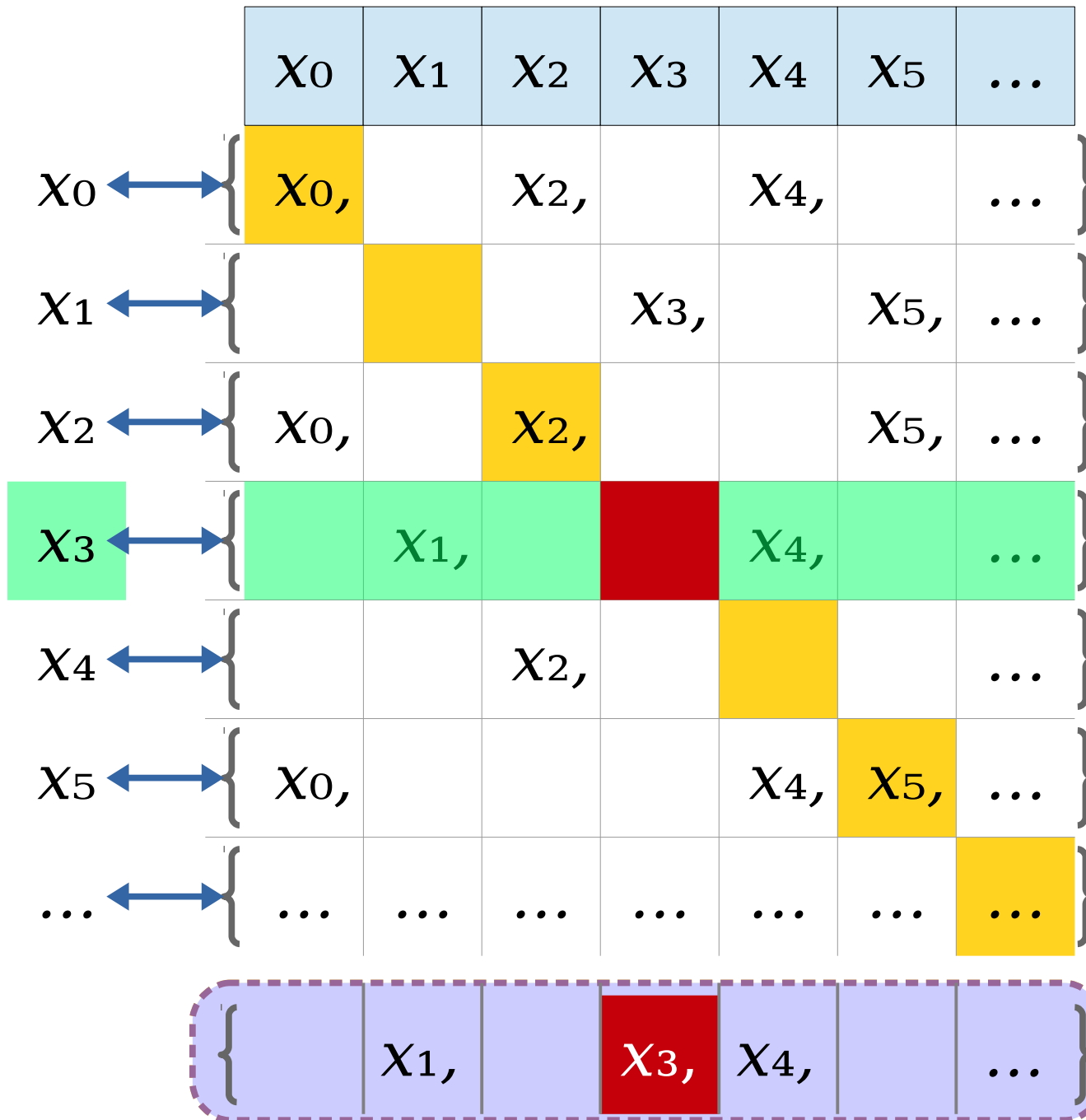
Which element is paired with this set?



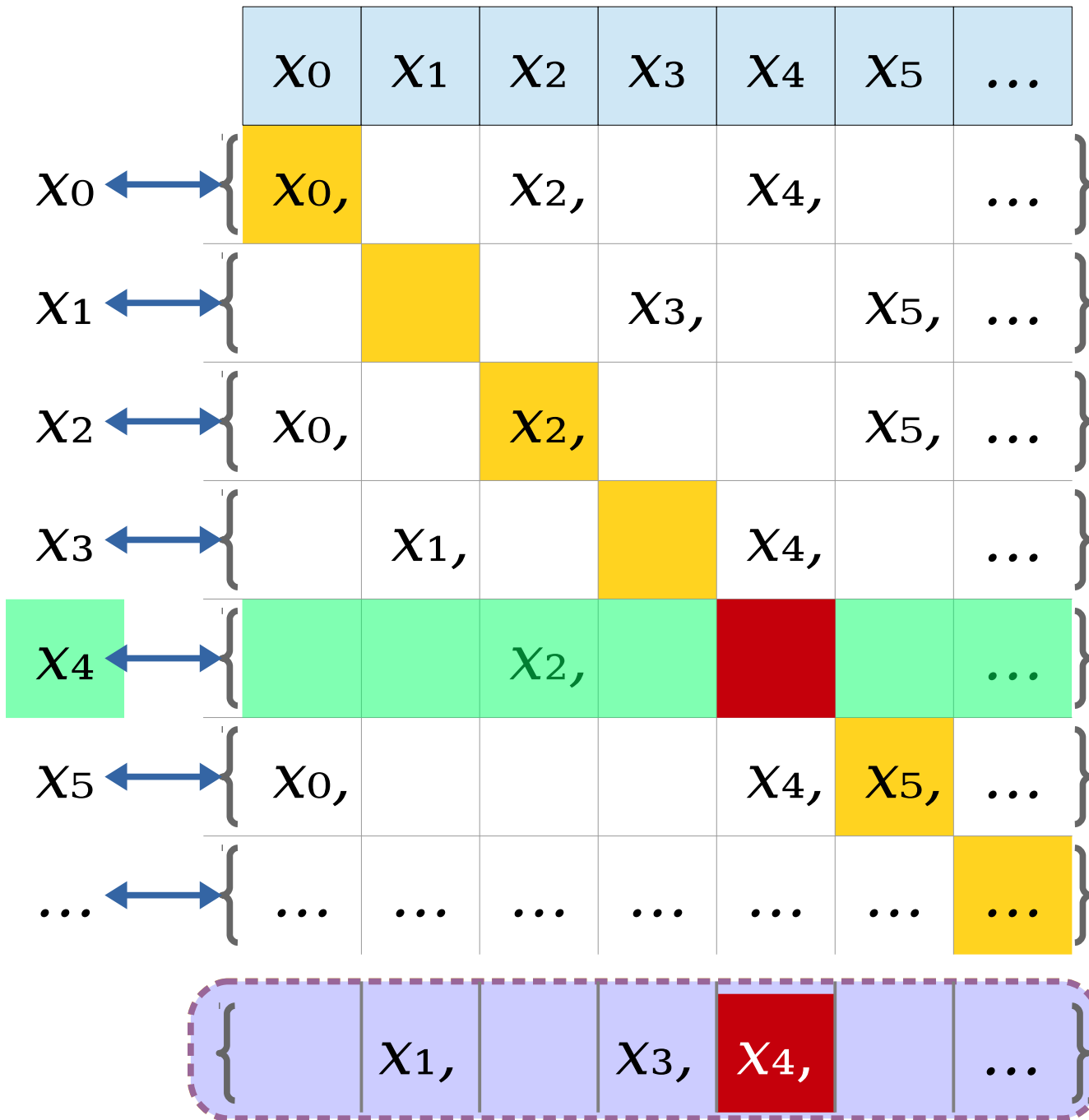


Which element is paired with this set?

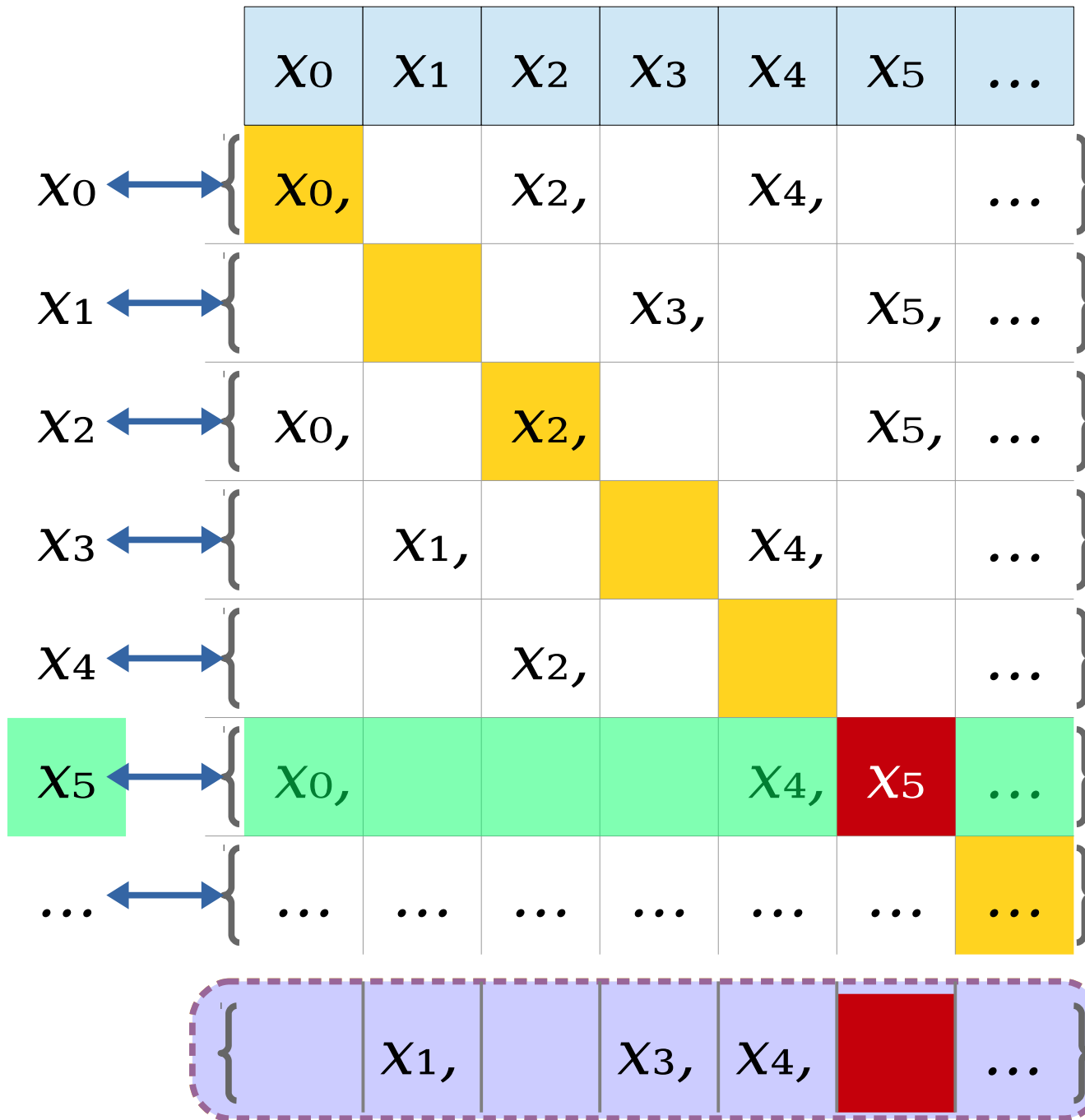




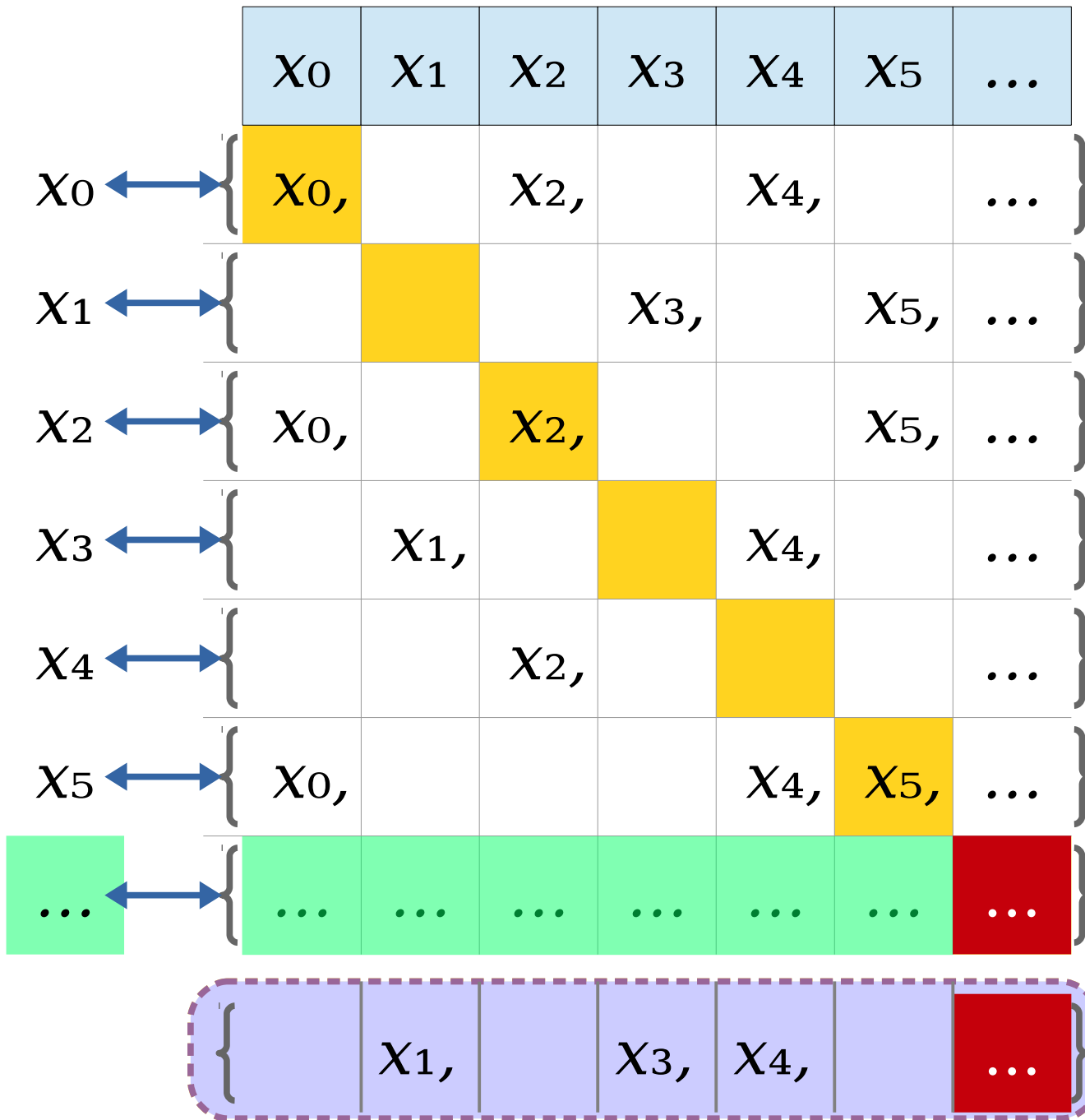
Which element is paired with this set?



Which element is paired with this set?



Which element is paired with this set?



Which element is paired with this set?

# The Diagonalization Proof

- No matter how we pair up elements of  $S$  and subsets of  $S$ , the complemented diagonal won't appear in the table.
  - In row  $n$ , the  $n$ th element must be wrong.
- No matter how we pair up elements of  $S$  and subsets of  $S$ , there is *always* at least one subset left over.
- This result is ***Cantor's theorem***: Every set is strictly smaller than its power set:

**If  $S$  is a set, then  $|S| < |\wp(S)|$ .**

# Two Infinities...

- By Cantor's Theorem:

$$|\mathbb{N}| < |\wp(\mathbb{N})|$$

# ...And Beyond!

- By Cantor's Theorem:

$$|\mathbb{N}| < |\wp(\mathbb{N})|$$

$$|\wp(\mathbb{N})| < |\wp(\wp(\mathbb{N}))|$$

$$|\wp(\wp(\mathbb{N}))| < |\wp(\wp(\wp(\mathbb{N})))|$$

$$|\wp(\wp(\wp(\mathbb{N})))| < |\wp(\wp(\wp(\wp(\mathbb{N}))))|$$

...

- ***Not all infinite sets have the same size!***
- ***There is no biggest infinity!***
- ***There are infinitely many infinities!***

What does this have to do  
with computation?

***“The set of all computer programs”***

***“The set of all problems to solve”***

# Where We're Going

- A ***string*** is a sequence of characters.
- We're going to prove the following results:
  - There are ***at most*** as many programs as there are strings.
  - There are ***at least*** as many problems as there are sets of strings.
- This leads to some *incredible* results – we'll see why in a minute!

# Where We're Going

A *string* is a sequence of characters.

We're going to prove the following results:

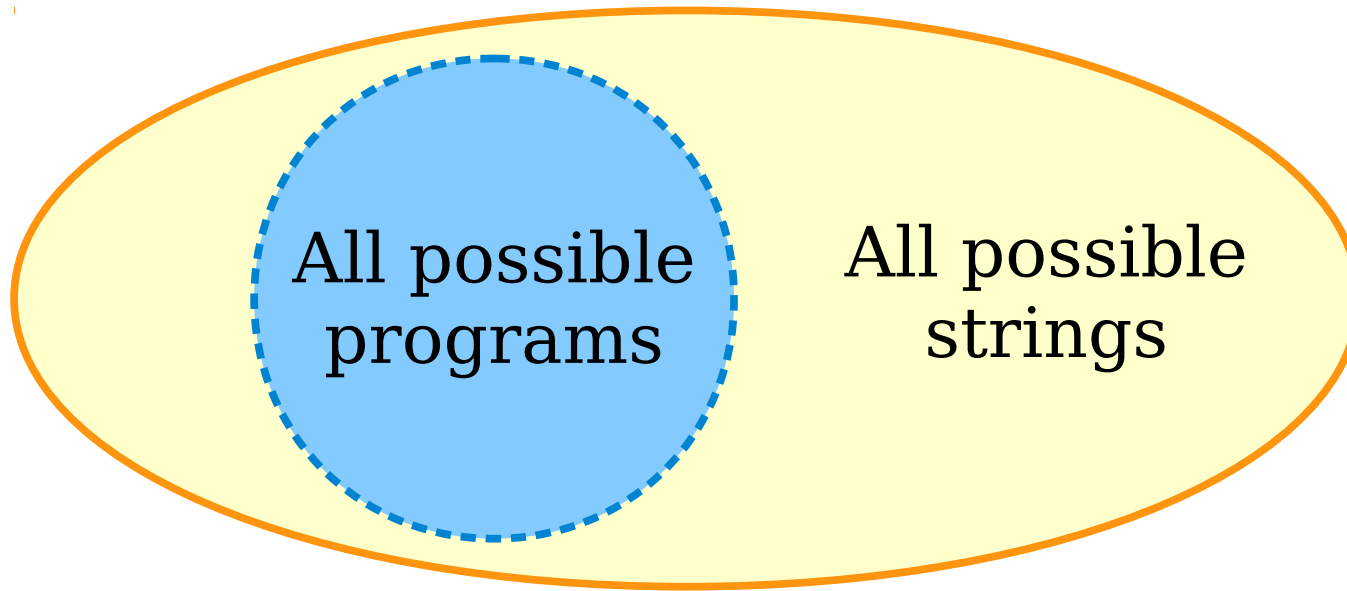
- There are ***at most*** as many programs as there are strings.

There are ***at least*** as many problems as there are sets of strings.

This leads to some *incredible* results – we'll see why in a minute!

# Strings and Programs

- The source code of a computer program is just a (long, structured, well-commented) string of text.
- All programs are strings, but not all strings are necessarily programs.



$$|\mathbf{Programs}| \leq |\mathbf{Strings}|$$

# Where We're Going

- A ***string*** is a sequence of characters.
- We're going to prove the following results:
  - There are ***at most*** as many programs as there are strings.
  - There are ***at least*** as many problems as there are sets of strings.
- This leads to some *incredible* results – we'll see why in a minute!

# Where We're Going

- A ***string*** is a sequence of characters.
- We're going to prove the following results:
  - There are ***at most*** as many programs as there are strings. ✓
  - There are ***at least*** as many problems as there are sets of strings.
- This leads to some *incredible* results – we'll see why in a minute!

# Where We're Going

A *string* is a sequence of characters.

We're going to prove the following results:

There are *at most* as many programs as there are strings. ✓

- There are *at least* as many problems as there are sets of strings.

This leads to some *incredible* results – we'll see why in a minute!

# Strings and Problems

- There is a connection between the number of sets of strings and the number of problems to solve.
- Let  $S$  be any set of strings. This set  $S$  gives rise to a problem to solve:

**Given a string  $w$ , determine whether  $w \in S$ .**

# Strings and Problems

**Given a string  $w$ , determine whether  $w \in S$ .**

- Suppose that  $S$  is the set

$$S = \{ "a", "b", "c", \dots, "z" \}$$

- From this set  $S$ , we get this problem:

**Given a string  $w$ , determine whether  $w$  is a single lower-case English letter.**

# Strings and Problems

**Given a string  $w$ , determine whether  $w \in S$ .**

- Suppose that  $S$  is the set

$$S = \{ "0", "1", "2", \dots, "9", "10", "11", \dots \}$$

- From this set  $S$ , we get this problem:

**Given a string  $w$ , determine whether  $w$  represents a natural number.**

# Strings and Problems

**Given a string  $w$ , determine whether  $w \in S$ .**

- Suppose that  $S$  is the set

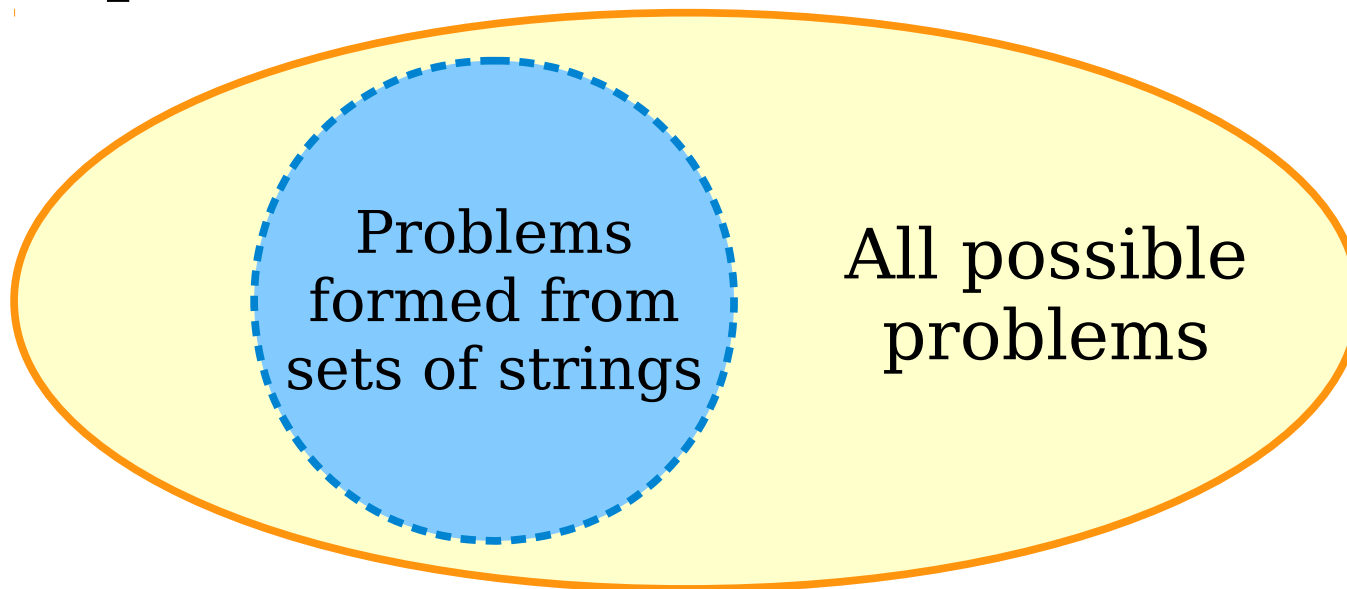
$$S = \{ p \mid p \text{ is a legal C++ program} \}$$

- From this set  $S$ , we get this problem:

**Given a string  $w$ , determine whether  $w$  is a legal C++ program.**

# Strings and Problems

- Every set of strings gives rise to a unique problem to solve.
- Other problems exist as well.



$$|\mathbf{Sets\ of\ Strings}| \leq |\mathbf{Problems}|$$

# Where We're Going

- A ***string*** is a sequence of characters.
- We're going to prove the following results:
  - There are ***at most*** as many programs as there are strings. ✓
  - There are ***at least*** as many problems as there are sets of strings.
- This leads to some *incredible* results – we'll see why in a minute!

# Where We're Going

- A ***string*** is a sequence of characters.
- We're going to prove the following results:
  - There are ***at most*** as many programs as there are strings. ✓
  - There are ***at least*** as many problems as there are sets of strings. ✓
- This leads to some *incredible* results – we'll see why in a minute!

# Where We're Going

A *string* is a sequence of characters.

We're going to prove the following results:

There are *at most* as many programs as there are strings. ✓

There are *at least* as many problems as there are sets of strings. ✓

- This leads to some *incredible* results – we'll see why in a minute!

# Where We're Going

A *string* is a sequence of characters.

We're going to prove the following results:

There are *at most* as many programs as there are strings. ✓

There are *at least* as many problems as there are sets of strings. ✓

- This leads to some *incredible* results – we'll see why ~~in a minute!~~ *right now!*

Every computer program is a string.

So, the number of programs is at most the number of strings.

From Cantor's Theorem, we know that there are more sets of strings than strings.

There are at least as many problems as there are sets of strings.

$$|\mathbf{Programs}| \leq |\mathbf{Strings}| < |\wp(\mathbf{Strings})| \leq |\mathbf{Problems}|$$

Every computer program is a string.

So, the number of programs is at most the number of strings.

From Cantor's Theorem, we know that there are more sets of strings than strings.

There are at least as many problems as there are sets of strings.

**|Programs| < |Problems|**

*There are more problems to solve than there are programs to solve them.*

**|Programs| < |Problems|**

# It Gets Worse

- Using more advanced set theory, we can show that there are *infinitely more* problems than solutions.
- In fact, if you pick a totally random problem, the probability that you can solve it is *zero*.
- ***More troubling fact:*** We've just shown that *some* problems are impossible to solve with computers, but we don't know *which* problems those are!

We need to develop a more nuanced understanding of computation.

# Where We're Going

- ***What makes a problem impossible to solve with computers?***
  - Is there a deep reason why certain problems can't be solved with computers, or is it completely arbitrary?
  - How do you know when you're looking at an impossible problem?
  - Are these real-world problems, or are they highly contrived?
- ***How do we know that we're right?***
  - How can we back up our pictures with rigorous proofs?
  - How do we build a mathematical framework for studying computation?

# Next Time

- ***Mathematical Proof***
  - What is a mathematical proof?
  - How can we prove things with certainty?