

Section 1.1: Sets

Definitions

We shall first start out with some basic definitions.

set = a collection of objects

element = a member of a set

A set can either be defined by describing its elements or by listing them. For example, the set A of natural numbers from 1 to 3 can be written as

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

where the curly brackets $\{ \}$, also known as “set braces”, are used to denote the elements. Note that the elements of a set do not have to be numbers. For example, the set of computing courses that Computer Systems Technology students take in their first term is

$$B = \{\text{“Comp 112”}, \text{“Comp 140”}, \text{“Comp 176”}\}$$

and each element is given in quotations so that the two words “Comp” and “112” are seen as joined into a single entity. (In programming, these are called character strings.)

The symbol \in means “is an element of”, so $4 \in \{2, 4, 6, 8\}$. Adding a diagonal slash /, meaning “not”, to a symbol negates it so \notin means “is not an element of” and $5 \notin \{2, 4, 6, 8\}$.

Equality

Two sets are equal if and only if they contain exactly the same elements. Elements in a set can be written in any order, and duplicates in a list do not change the set. Therefore, the following sets are all equal:

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

If a set has a large number of members in a definite pattern, we can use the following notation. If C is the set of even natural numbers from 2 to 100, then

$$C = \{2, 4, 6, 8, \dots 100\}.$$

As C has a finite number of elements, it is called a **finite** set. This notation can also be used for infinite sets, provided that the final term is omitted to show that the set goes on forever. For example, the set of natural numbers, an infinite set which is often called N , can be written as

$$N = \{1, 2, 3, 4, \dots\}.$$

The set of integers, \mathbb{Z} , is then $\mathbb{Z} = \{\dots-3, -2, -1, 0, 1, 2, 3, \dots\}$. (The symbol \mathbb{Z} is also frequently used.)

Empty Set

The empty set, sometimes called the null or void set, is the set with no elements. It is written as either \emptyset or $\{\}$.

Warning: do not write $\{\emptyset\}$. This is a set which contains the empty set. In the same way that a box with an empty box inside is not itself empty, \emptyset is an empty set, but a set containing the empty set is not itself empty!

Set Builder Notation

Another way to define sets is by using set-builder notation, such as

$$A = \{ x \mid x \text{ is an even natural number} \}.$$

The vertical bar means “such that”, so the above equation is read as “the set A equals the set of all x such that x is an even natural number”. Although this notation is cumbersome, it is useful to describe sets for which you can’t just list all of the elements. Examples of this are the sets

$$B = \{ x \mid x \text{ is an irrational number less than } 7 \} \text{ and}$$

$$C = \{ y \mid y \text{ is a positive real number} \}.$$

Example

Let $A = \{ x \mid x \text{ is a natural number between } 3 \text{ and } 8 \}$. Rewrite this set by listing the elements.

Answer: $A = \{ 4, 5, 6, 7 \}$. Note that numbers **between** 3 and 8 do not **include** 3 and 8.

Example

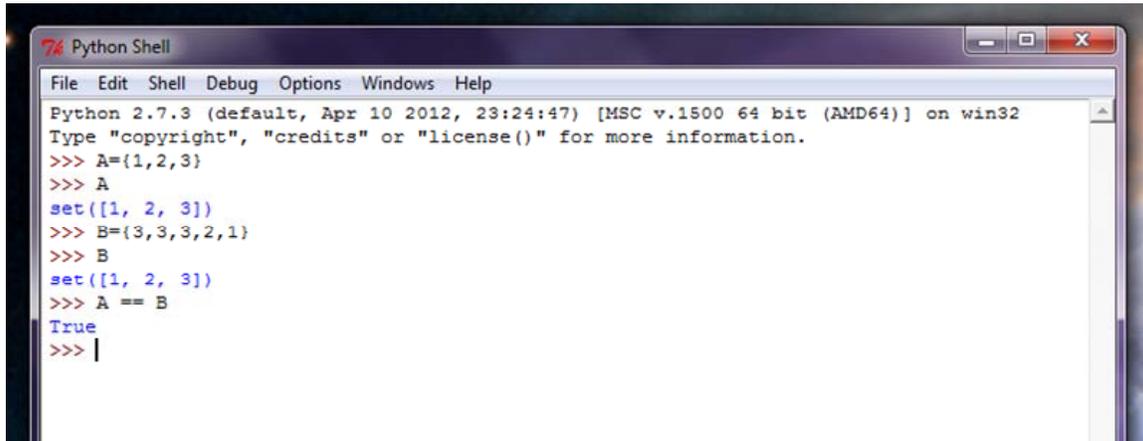
Let $A = \{ y \mid y \text{ is a negative natural number} \}$. Rewrite this set by listing the elements.

Answer: There are no negative natural numbers, so $A = \emptyset$. You could also say that $A = \{\}$.

Sets in the Computer Language “Python”

As you can see in the accompanying figure, inputting sets in Python uses exactly the same syntax as we have been using here. There are differences in what the output looks like, formally writing `set([` instead of just using parentheses, but Python does collapse

duplicates in the way you would expect. (The ordering of the elements is occasionally a little odd, though, as you can see in the later set-builder examples.)



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 2.7.3 (default, Apr 10 2012, 23:24:47) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> A={1,2,3}
>>> A
set([1, 2, 3])
>>> B={3,3,3,2,1}
>>> B
set([1, 2, 3])
>>> A == B
True
>>> |
```

Figure 1: Defining Sets in Python

Python also allows you to define sets in a way that is very similar to set-builder notation, as you can see in the following examples.

```
>>> A={x for x in range(10)}
>>> A
set([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> |

>>> powers_of_two = {2**n for n in range(6)}
>>> powers_of_two
set([32, 1, 2, 4, 8, 16])
>>>

>>> leaps = {y for y in range(1969,2013) if y % 4 == 0}
>>> leaps
set([1984, 1988, 1992, 2012, 1996, 2000, 2008, 1972, 1976, 2004, 1980])
>>>
```

Figure 2: Set-builder Notation in Python

In Python, if we wish to determine if an element belongs to a particular set, we use the following (extremely terse!) syntax.

```
>>> A={1,2,3}
>>> 2 in A
True
>>> 4 in A
False
>>> 4 not in A
True
>>>
```

Figure 3: Elements of Sets in Python