

# Python Memory Model

## Table of contents

|                                     |   |
|-------------------------------------|---|
| Today's outline . . . . .           | 1 |
| Learning objectives . . . . .       | 1 |
| How much can a byte hold? . . . . . | 2 |
| Variables and memory . . . . .      | 2 |
| Sequencing: order matters . . . . . | 2 |
| Summary . . . . .                   | 3 |
| Next lecture . . . . .              | 3 |

See [csse1001<sup>1</sup>](#) for course logistics — this note covers Lecture 2A's technical content.

## Today's outline

- Memory as a sequence of bits
- Variables as named memory locations
- Sequencing: why the order of instructions matters

## Learning objectives

1. Memory is a sequence of bits that we can toggle.
2. We can (and should) store values in named memory locations called variables.
3. The order of instructions matters.

---

<sup>1</sup>[courses/csse1001/csse1001.pdf](#)

## How much can a byte hold?

Computer memory is a very long series of on/off switches (**bits**, short for *binary digit*). Eight consecutive bits form a **byte**, and 4 or 8 bytes form a **word** (depending on machine architecture).

In general,  $n$  bits can store  $2^n$  distinct things:

- 1 bit:  $2^1 = 2$  things (off, on)
- 2 bits:  $2^2 = 4$  things (off-off, on-off, off-on, on-on)
- 3 bits:  $2^3 = 8$  things
- 4 bits:  $2^4 = 16$  things

## Variables and memory

See [python-variables-and-memory-model<sup>2</sup>](#) for how Python interprets memory as typed objects, how `id()` reveals an object's address, and the rules for assigning, retrieving, naming, and reassigning variables.

## Sequencing: order matters

The same instructions executed in a different order will (usually) produce a different outcome.

Python Tutor 1 — note we *cannot* swap these two lines: `x = x + 1` on its own throws a `NameError`, since `x` has no value yet.

```
>>> x = 1
>>> x = x + 1
```

Python Tutor 2:

```
>>> x = 1
>>> x = x + 1
>>> x = 2*x
>>> x
4
```

Python Tutor 3 — last two lines swapped:

---

<sup>2</sup>[courses/csse1001/python-variables-and-memory-model.pdf](#)

```
>>> x = 1
>>> x = 2*x
>>> x = x + 1
>>> x
3
```

Python Tutor 4:

```
>>> x = 2
>>> y = 3
>>> y = x
>>> x = y
>>> x
2
>>> y
2
```

Python Tutor 5 — middle two lines swapped:

```
>>> x = 2
>>> y = 3
>>> x = y
>>> y = x
>>> x
3
>>> y
3
```

## Summary

We can manipulate memory using Python and refer to that memory with variables. The order in which we do this manipulation matters — the same sequence of instructions done in a different order will (usually) result in a different outcome.

## Next lecture

2025-08-04-primitive-data<sup>3</sup> — immutable objects (the things we put in memory that cannot be changed).

---

<sup>3</sup>[courses/csse1001/2025-08-04-primitive-data.pdf](https://courses/csse1001/2025-08-04-primitive-data.pdf)