Variables

An introduction to Variables, which act as label for our data.

We'll cover the following

- Variable names

Like in other high-level programming languages, in C, we can assign symbolic names, known as **variables**, for storing information in memory. Then we can refer to those pieces of information in memory by using the symbolic variable name, instead of having to use the raw address in memory. Variables can be used to store floating-point numbers, characters, and even pointers to other locations in memory.

Variable names

There are some restrictions on the names of variables in C. Names are made up of letters and digits, but the first character must be a letter (not a digit). The underscore "_" counts as a letter. Also remember in unix, uppercase and lowercase letters are distinct and so `Age` is distinct from `age`.

Here is a list of **reserved keywords** in C that cannot be used as variable names:

- `_Bool`
- `default`
- `if`
- `sizeof`
- `while`
- `_Complex`
- `do`
- `inline`

- `inline`
- `static`
- `_Imaginary`
- `double`
- `int`
- `struct`
- `auto`
- `else`
- `long`
- `switch`
- `break`
- `enum`
- `register`
- `typedef`
- `case`
- `extern`
- `restrict`
- `union`
- `char`
- `float`
- `return`
- `unsigned`
- `const`
- `for`

- `short`

- `void`
- `continue`
- `goto`
- `signed`
- `volatile`

A variable itself is just a name. Now we’ll learn about the different type of things we can assign a variable to.