Maybe some context would help

Let's think in terms of nouns and verbs

- Opcodes are the verbs: "Do something"
- There are 3 kinds of nouns:
  - The registers –fastest and most opcodes can get to them
  - Immediate data – constants that get packed in with the instructions
  - Memory itself – we need an address to know where in memory

We don’t have variables as we know them

The compiler assigns C variables to registers or to memory locations

- Data in the heap will go at an address the compiler can determine
  - It doesn’t move or go away
  - The compiler generates a fixed constant as the address
- Data in the stack will wind up at an unknown location, and recursion means that it could exist multiple times
  - Its location will always be the same relative to the current stack pointer
  - The compiler generates an offset to be added to %rsp to give the address
- Data in registers can be directly accessed by the register name
  - There are only 16 of them and some are spoken for
  - Functions of few variables run faster

All memory accesses are effectively via a void pointer

- Any 64 bit register could hold a memory address
- We can use the registers as pointers
  - We do not know “type”
  - We **do** specify *length*
  - Any “variable” that is not in a register is in memory and the previous slide showed how we know where in memory

There's an op for that

C operators correspond to 1 (or sometimes a short sequence of) opcodes built into the processor:

- Add
- Bitwise and
- Arithmetic right shift
- Integer multiply
- Equals (single = sign)
- … and many more!

The suffix tells how many bytes to use