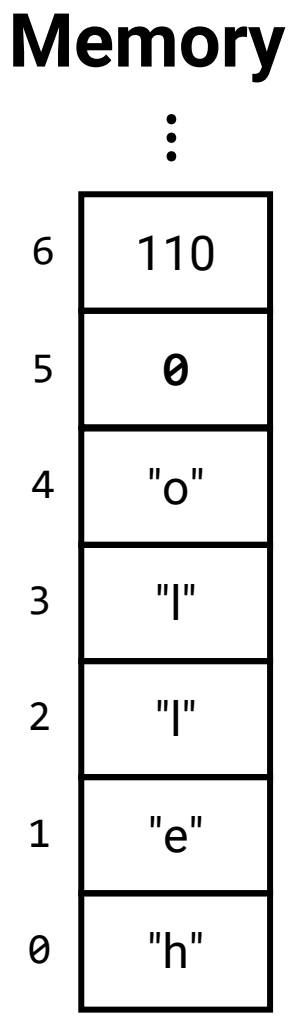
A Basic Model of a Computer

1. **Memory**
   - Addressable storage space for data.
   - Billions of tiny cells to store numbers and letters.
   - Complex data like words, sentences, and pictures are stored in memory by treating many cells grouped as a single *data structure*.

2. **Processor** *(Central Processing Unit)*
   - Follow instructions like computations and memory storage/retrieval
   - Far fewer and much simpler kinds of instructions than you'd think!
     - Add, subtract, load, store, compare two numbers, jump, etc.
   - A processor's magic is that it can run *billions* of instructions *per second*.

3. **Programs**
   - A sequence of simple instructions the processor follows one-by-one.
   - As a programmers, *you* are authoring these instructions!



What is a program?

- A program is a series of ***instructions*** that *load*, *compute changes to*, and *store* data in ***memory***

- Computer chips understand ***machine code*** instructions in a ***binary*** code format
  - It's *very* painful for humans to author directly

- Humans can write machine-level code in a *slightly* nicer format called ***assembly code*** that is ***assembled***, or translated, into machine code.

```
add $t1, $s3, $s3    # Temp reg $t1 = 2 * i
add $t1, $t1, $t1    # Temp reg $t1 = 4 * i
add $t1, $t1, $s6    # $t1 = address of save[i]
lw $t0, 0($t1)       # Temp reg $t0 = save[i]
bne $t0, $s5, Exit   # go to Exit if save[i] ≠ k
 add $s3, $s3, $s4   # i = i + j
add $t1, $s3, $s3    # Temp reg $t1 = 2 * i
add $t1, $t1, $t1    # Temp reg $t1 = 4 * i
add $t1, $t1, $s6    # $t1 = address of save[i]
lw $t0, 0($t1)       # Temp reg $t0 = save[i]
beq $t0, $s5, Loop # go to Loop if save[i] = k
```

What is a programming language?

- Programming languages enable you to write programs in a more humane way than writing assembly code.
  - They're designed by humans to be human-readable
  - A concrete medium for expressing processes
- The first English-like programming language was FLOW-MATIC, invented in 1955 by Grace Hopper
- Languages, like written ones, have **syntax rules** and **semantics**
- High-level programs are ***interpreted*** *or* ***compiled*** *into* the much more rudimentary, binary machine code which the processor can then follow

Grace Hopper

Programming Language: Python

- In this course you will use a modern version of the Python language
- The concepts you learn will apply to *any* programming language
- Why Python?
  - "Batteries included" - Python's standard libraries have *a lot* of powerful capabilities
  - Mature ecosystem - free 3rd-party libraries for data science, machine learning, & more
  - Minimal syntax - fewer curly braces and special characters than C, Java, and JavaScript
  - It's one of the most valuable languages to know in 2020 whether you're studying computer science or *any other field that involves data processing and automation*.