**How Computers Work**

One of the first examples of the computer was the “difference engine,” an invention attributed to Charles Babbage circa 1819. This mechanical device sought to aid engineers by calculating tables of numbers by turning a crank. An example of how this might be used is in a parameter sweep — varying the inputs of arguments of a function and inspecting a result.

Up through the late 1940s, mechanical devices such as slide rules were used to aid teams of “human computers” in calculations. They were called “computers” because they were asked to “compute” numbers and data.

At the height of “human computing,” the National Advisory Committee for Aeronautics (NACA, the precursor to NASA) employed over 3,000 human computers at Langley Memorial Aeronautical Laboratory, many of whom were women!

In 1947, NACA acquired the first “Bell Electronic Computer.” Their department was headed by Sara Bullock. This machine used vacuum tubes and electromechanical relays to represent logic states in the device; energized representing “true” and de-energized representing “false.” This was the turning point for modern computing! The NACA team grew and eventually developed the world’s first mass-produced computer, the IBM 704!

The input to this machine was by punch card, a physical sheet of paper representing the lines of a computer program. The cards were fed in sequentially to process through the lines of code. The actual code was written in a language called JCL, or “job control language.”

Through the 60s, punch cards gave way to magnetic media (such as tape and eventually hard-disk drives). Vacuum tubes and electro-mechanical relays gave way to solid state transistor devices.

As materials improved, speeds and information density improved. Vendor-specific languages and instruction sets (information placed on punch cards) eventually transitioned to industry-standard instruction sets, such as ASCII (the computer-level representation of the written language).

The machine that generated punch cards (known as a teletype, or TTY for short) transitioned into what we know today as a computer, or a glass TTY (literally a keyboard with a cathode ray tube monitor mounted to it).

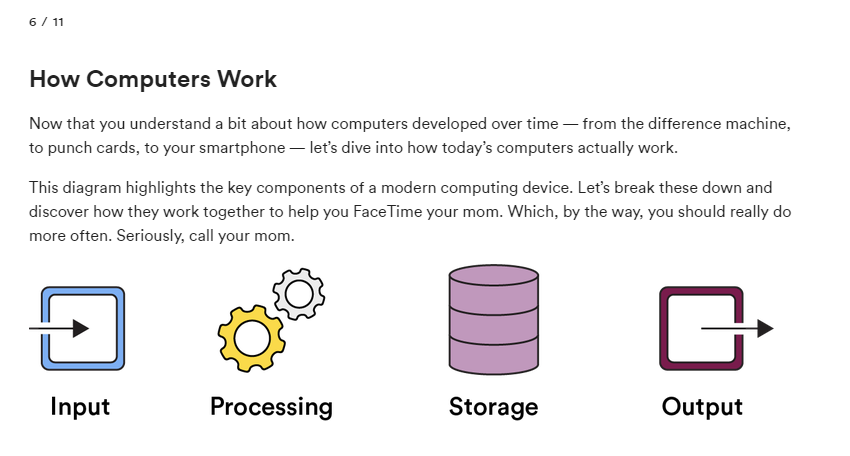
Although the first models were released in the late 60s, personal computers (PCs) didn’t achieve mainstream status until the late 70s, early 80s.

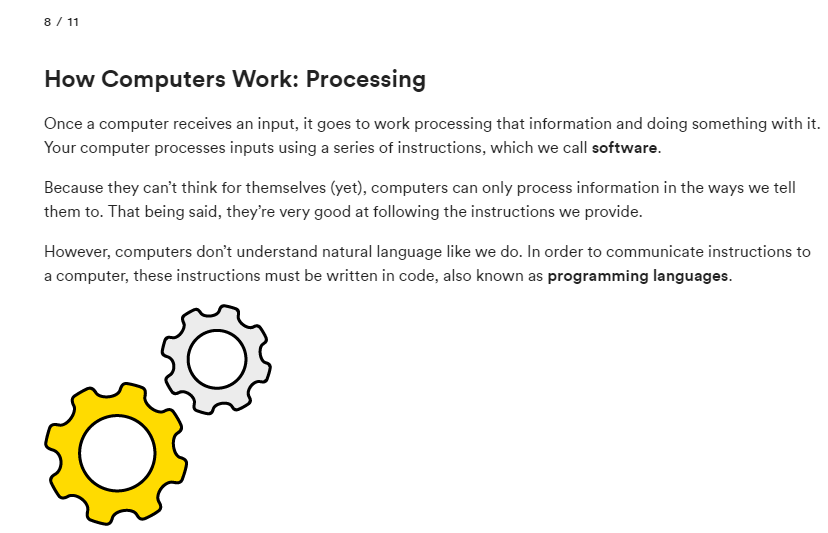
*From <*[*https://my.generalassemb.ly/activities/1033?assignmentUuid=40ccbb05-b54a-43a3-9083-9166525879fd*](https://my.generalassemb.ly/activities/1033?assignmentUuid=40ccbb05-b54a-43a3-9083-9166525879fd)*>*

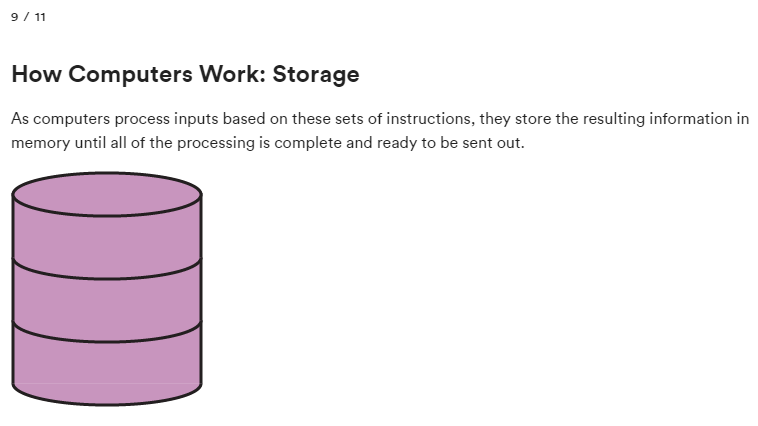
JCL = Job Control Language

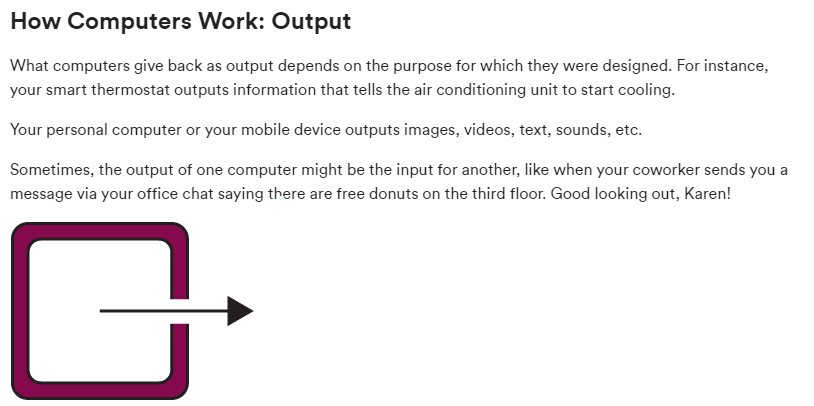
Teletype = TTY

PCs = Achieved mainstream status until late 70s and early 80s









Let’s look at the four components of a computer — input, processing, storage, and output — through a real-life example.

Oh my gosh! What’s that? A flying pig? Quick! Take a picture! ...OK, we know we said “a real-life example,” but we couldn’t resist a little fantasy.

You whip out your phone and touch the “camera” icon. This is the input. In one touch, you sent a signal to your phone that says, “Hey, open the camera app.”

To process that signal, your phone asks storage, “What do I do if I receive an input that says ‘Hey, open the camera app.’?”

Storage sends a set of instructions that, when executed or “processed”, will open the camera app — the output.

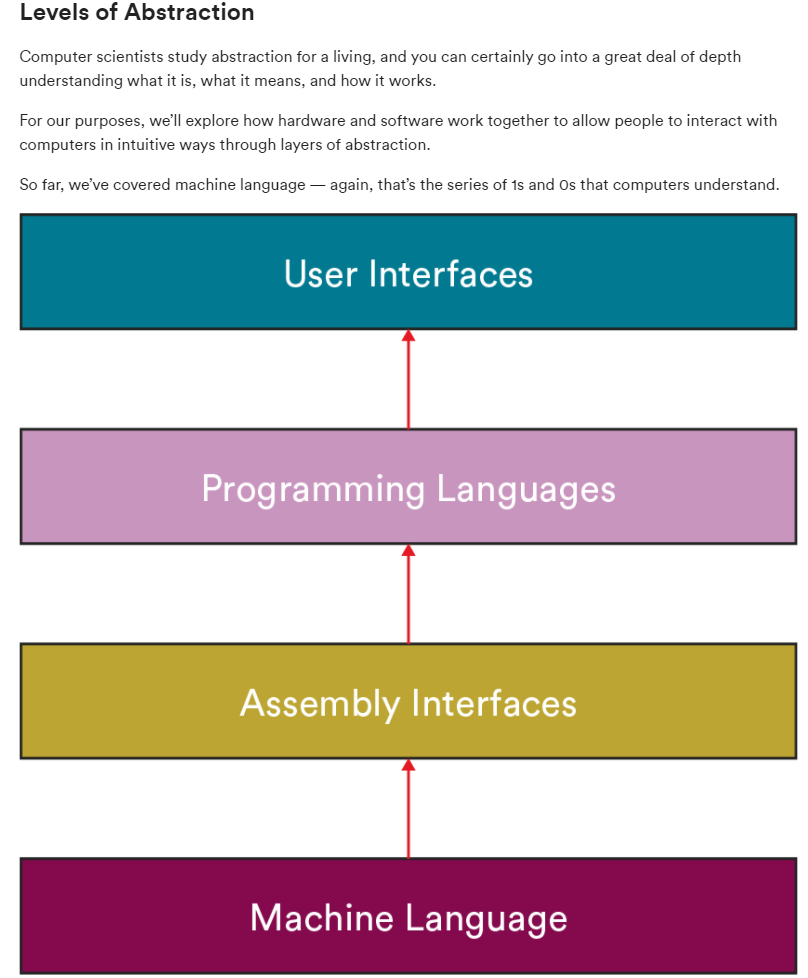
Thankfully, all of this happens so quickly it is almost instantaneous, so you open your phone camera before that pig flies away.

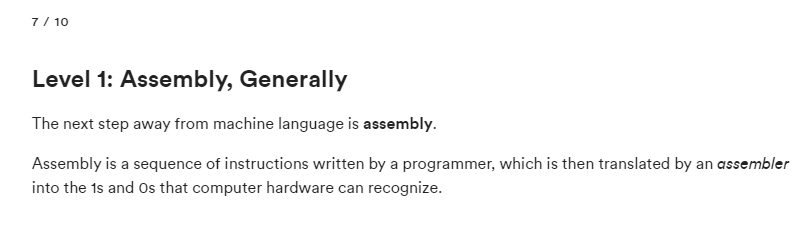
*From <*[*https://my.generalassemb.ly/activities/1033?assignmentUuid=40ccbb05-b54a-43a3-9083-9166525879fd*](https://my.generalassemb.ly/activities/1033?assignmentUuid=40ccbb05-b54a-43a3-9083-9166525879fd)*>*

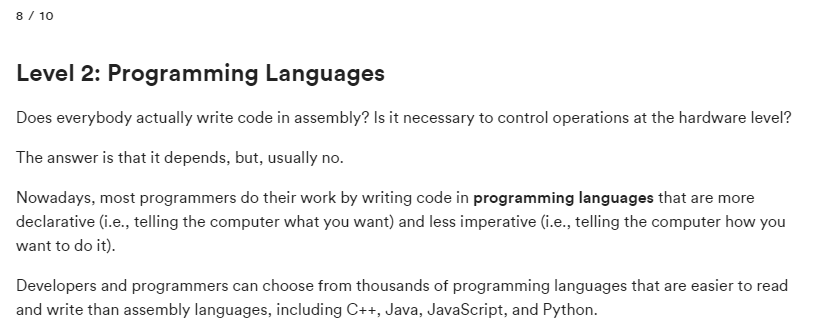
**Introducing the Layers of Abstraction**

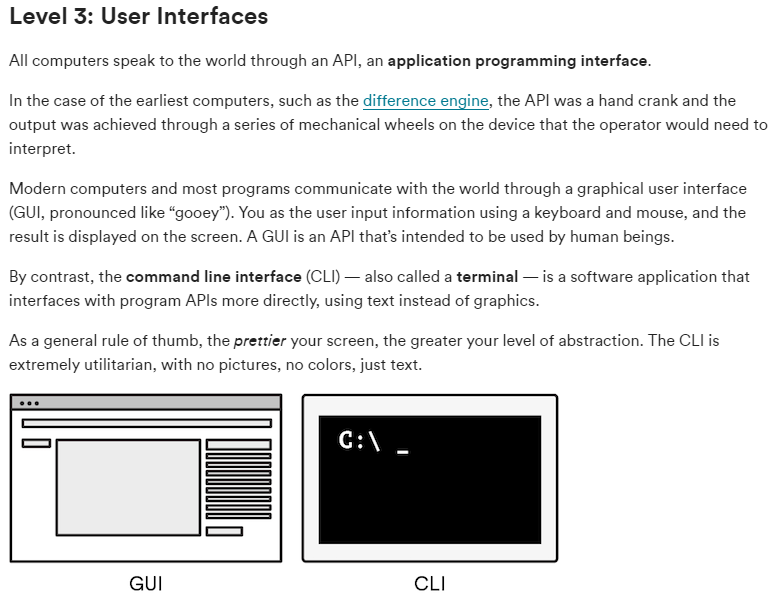
<https://money.cnn.com/interactive/technology/computing-power-timeline/>

<https://www.computerhistory.org/timeline/>

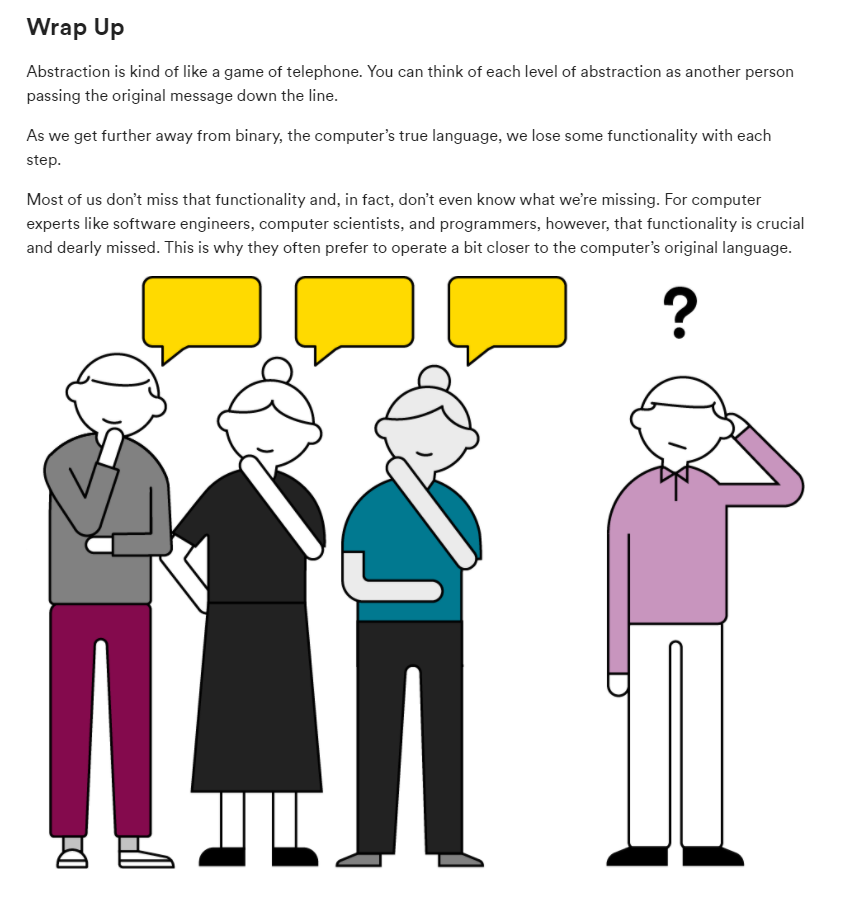




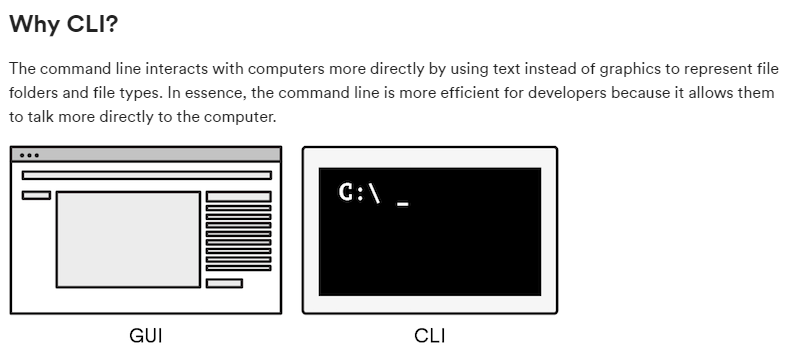


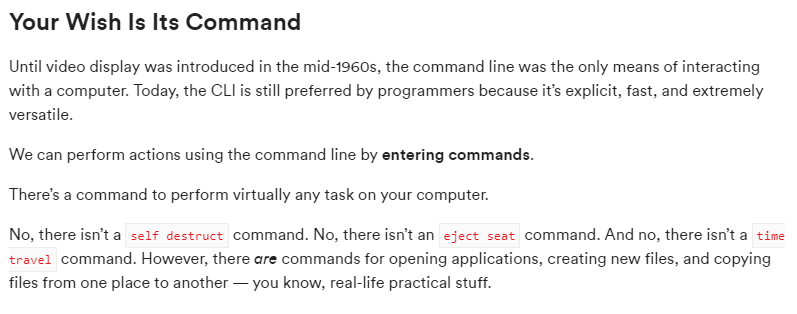


<https://www.computerhistory.org/babbage/>

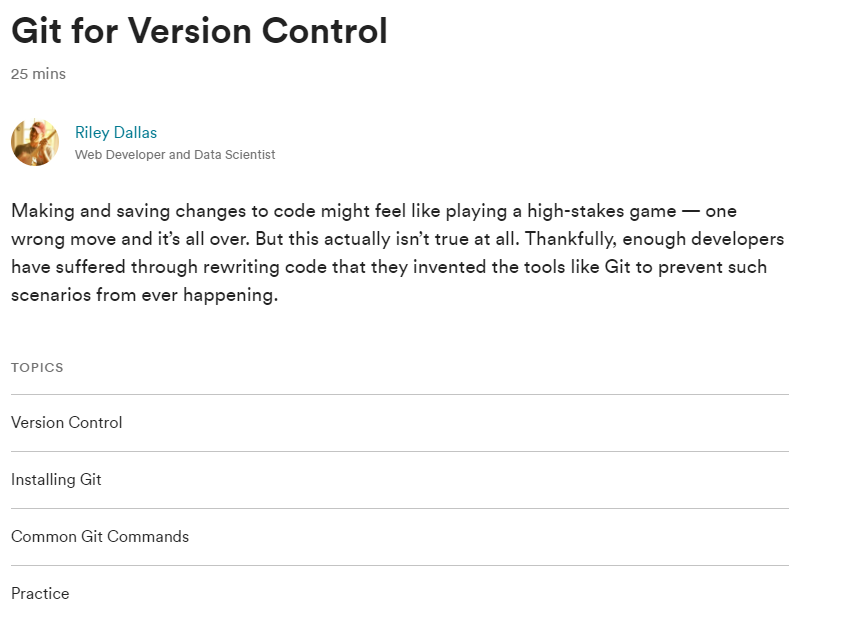


**Accessing and Navigating the Command Line Interface**





**Git for Version Control**



**Intro to Github**

**Git for Data Science Immersive Practice**