COMP110: Principles of Computing

2: Basic Principles for Computation

Learning outcomes

By the end of today’s session, you should be able to:

- Recall the historic significance of Alan Turing and his...
- Explain the basic concept of Turing Machines
-

Agenda

- The PyCharm IDE
- Basic Python programs
  - Variable assignment
  - Conditionals
  - Loops
- Coffee break
- SpaceChem worksheet review

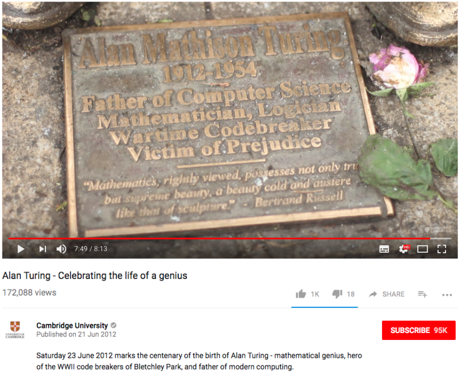
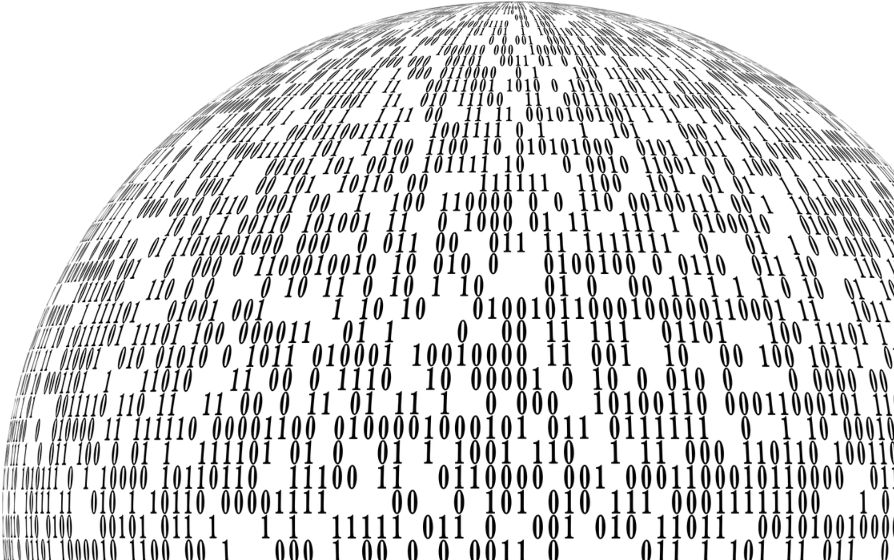


Figure: https://www.youtube.com/watch?v=gtRLmL70TH0

Activity - Groups of Six



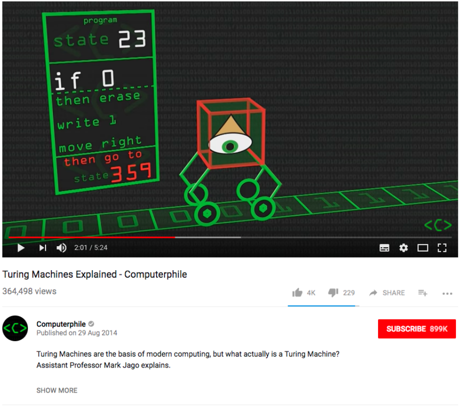


Figure: https://www.youtube.com/watch?v=dNRDvLACg5Q

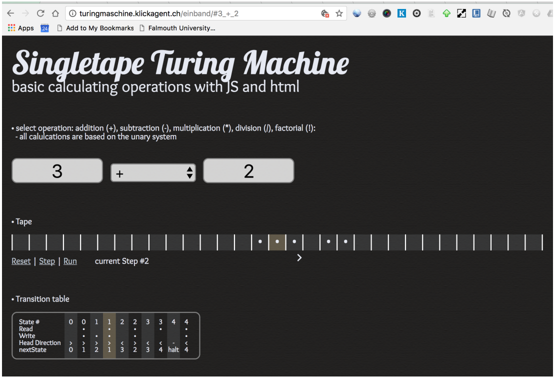


Figure: http://turingmaschine.klickagent.ch/einband/

Turing Completeness

To show that something is Turing complete, it is enough to show that it can be used to simulate some Turing complete system.

For an imperative language to be classed as Turing Complete it must have:

- Conditional branching (e.g., "if" and "goto" statements, or a "branch if zero" instruction)
- Ability to change an arbitrary amount of memory (e.g., the ability to maintain an arbitrary number of variables).

!!! Since this is almost always the case, most if not all imperative languages are Turing complete if the limitations of finite memory are ignored.