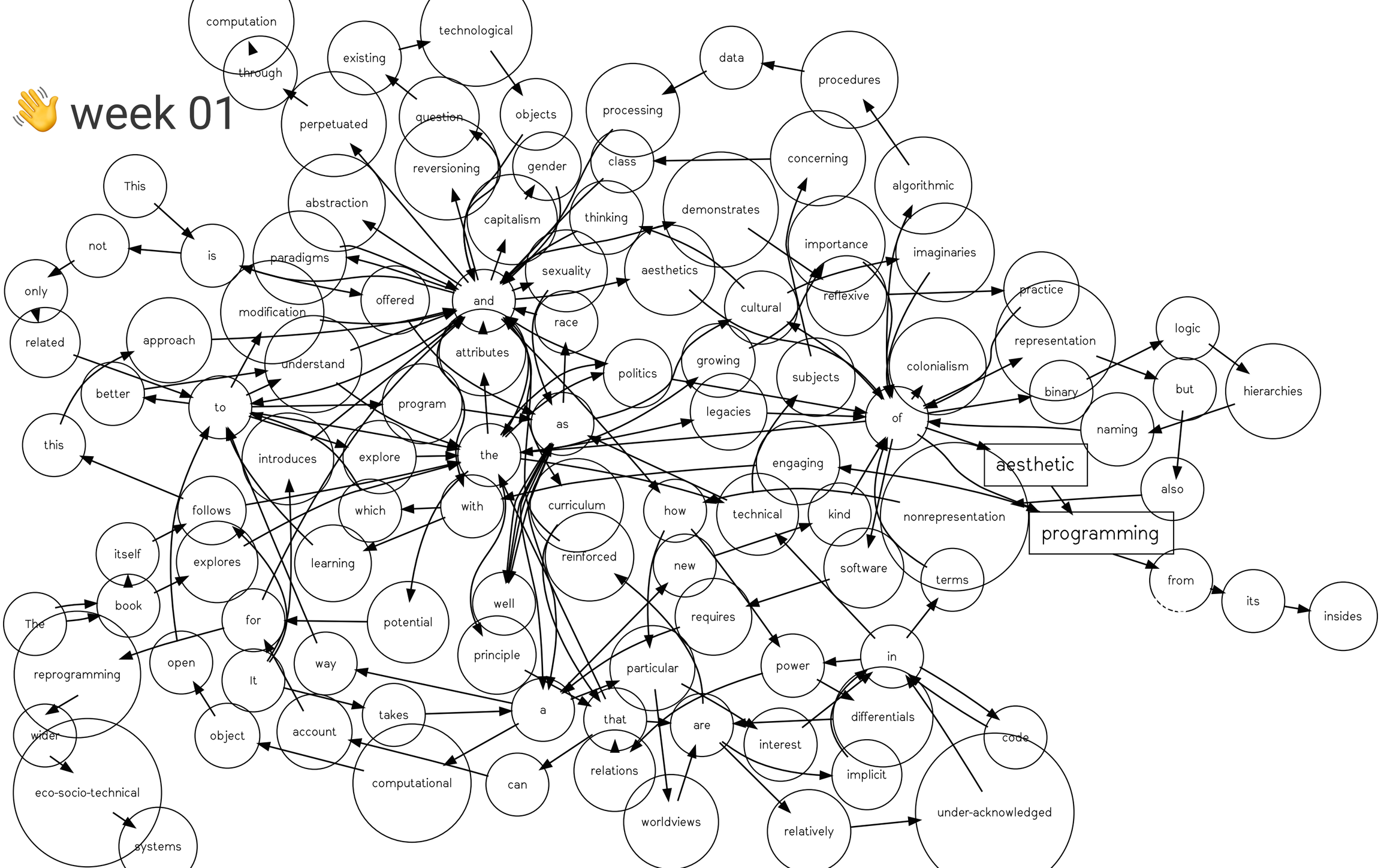
MS*059*F21 | Python and Viz

Professor Douglas Goodwin

MW 2:55-4:10PM

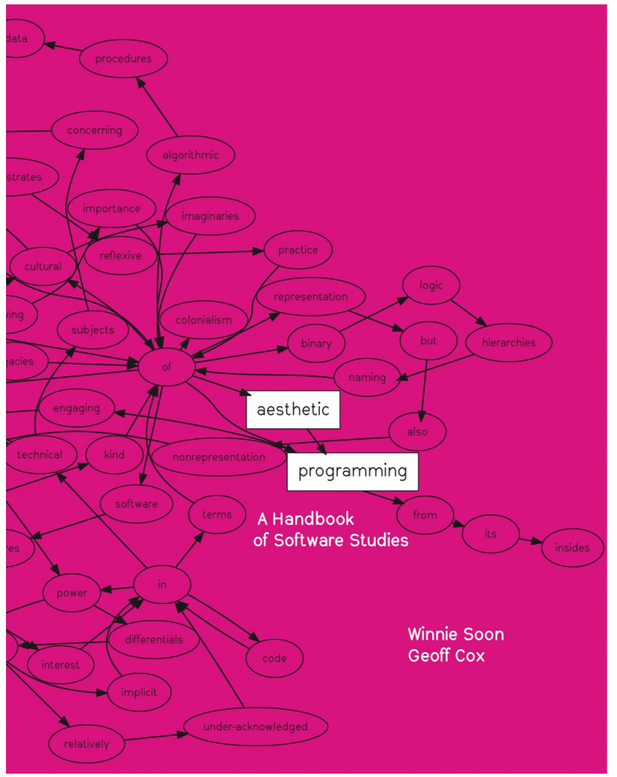
SC Campus, Steele Hall, 229

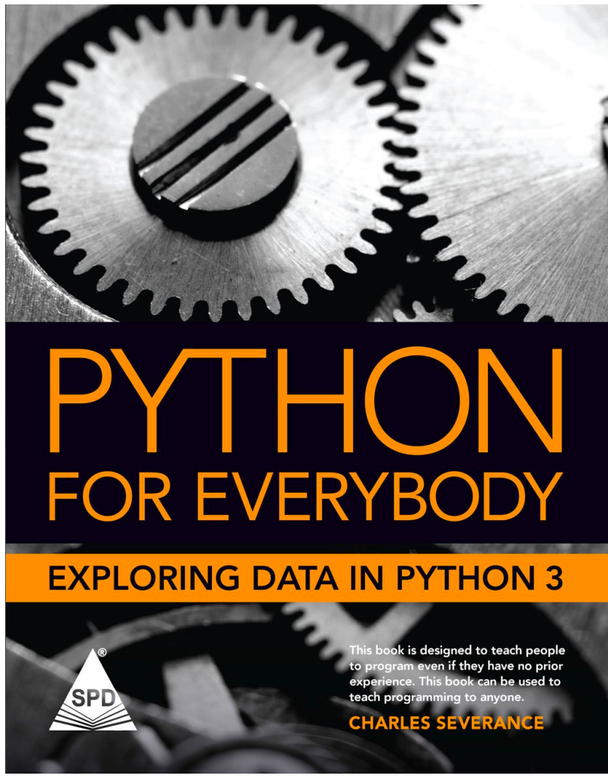
Syllabus | Discord


👋 week 01
computation
through
technological
existing
perpetuated
question
objects
data
procedures
processing
reversioning
gender
class
concerning
algorithmic
This
abstraction
capitalism
thinking
demonstrates
importance
imaginaries
not
is
paradigms
sexuality
aesthetics
only
offered
and
cultural
reflexive
practice
modification
race
related
approach
understand
attributes
politics
growing
subjects
colonialism
representation
logic
better
to
program
legacies
binary
but
hierarchies
this
as
of
naming
introduces
explore
the
engaging
aesthetic
follows
which
with
curriculum
how
technical
kind
nonrepresentation
also
programming
itself
explores
learning
reinforced
new
software
terms
from
book
for
potential
well
requires
its
insides
The
reprogramming
open
way
principle
particular
power
in
wider
It
takes
a
that
are
differentials
code
object
account
interest
eco-socio-technical
computational
can
relations
implicit
worldviews
relatively
under-acknowledged
systems

1.0 What?

This class will blend materials from two open-source books into a culturally inflected course in programming. Both courses start with utopian goals as the name suggests. The authors of these books are proponents of open source software and publishing, and this remix of this kind would be impossible without their generosity.





who?

“high culture” has traditionally been described as the domain of university-educated (wealthy, white) people, whilst “low culture” the domain of non-university-educated (working class) ordinary people.

Programming *cuts across this class divide* as both an exclusive and specialized practice that is also one rooted in the acquisition of skills with applied real-world use in both work and play.

Yet access to the means of production at the level of programming remains an issue all the same.

who?

We might usefully characterize this in terms of *literacy* and to further include the reading and writing of code. Knowing basic coding skills will not only enhance future employability, but will also enable the improved understanding of how media is “encoded” and “decoded.” Programming is a way of thinking and understanding other codes.

why program?

Computers Want to be Helpful...

- Computers are design *to do things for us*
- But we need to speak their language to describe what we want done
- Users have it easy - someone already put many different programs (instructions) into the computer and users just pick the ones they want to use

Users vs. Programmers

- Users see computers as a set of tools - word processor, spreadsheet, map, to-do list, etc.
- Programmers learn the computer “ways” and the computer language
- Programmers have some tools that allow them to build new tools
- Programmers sometimes write tools for lots of users and sometimes programmers write little “helpers” for themselves to automate a task

Why be a Programmer?

- To complete a task, e.g. Clean up survey data
- To produce something for others to use, e.g. Fix a performance problem in the Sakai software

What is Code? Software? A Program?

- A sequence of stored instructions
  - It is a little piece of our intelligence in the computer

- We figure something out and then we encode it and then give it to someone else to save them the time and energy of figuring it out

- A piece of creative art - particularly when we do a good job on user experience

how humans make cake

how computers make cake

Hardware Architecture

http://upload.wikimedia.org/wikipedia/commons/3/3d/RaspberryPi.jpg

Software

Central Processing Unit

Main Memory

What Next?

Generic Computer

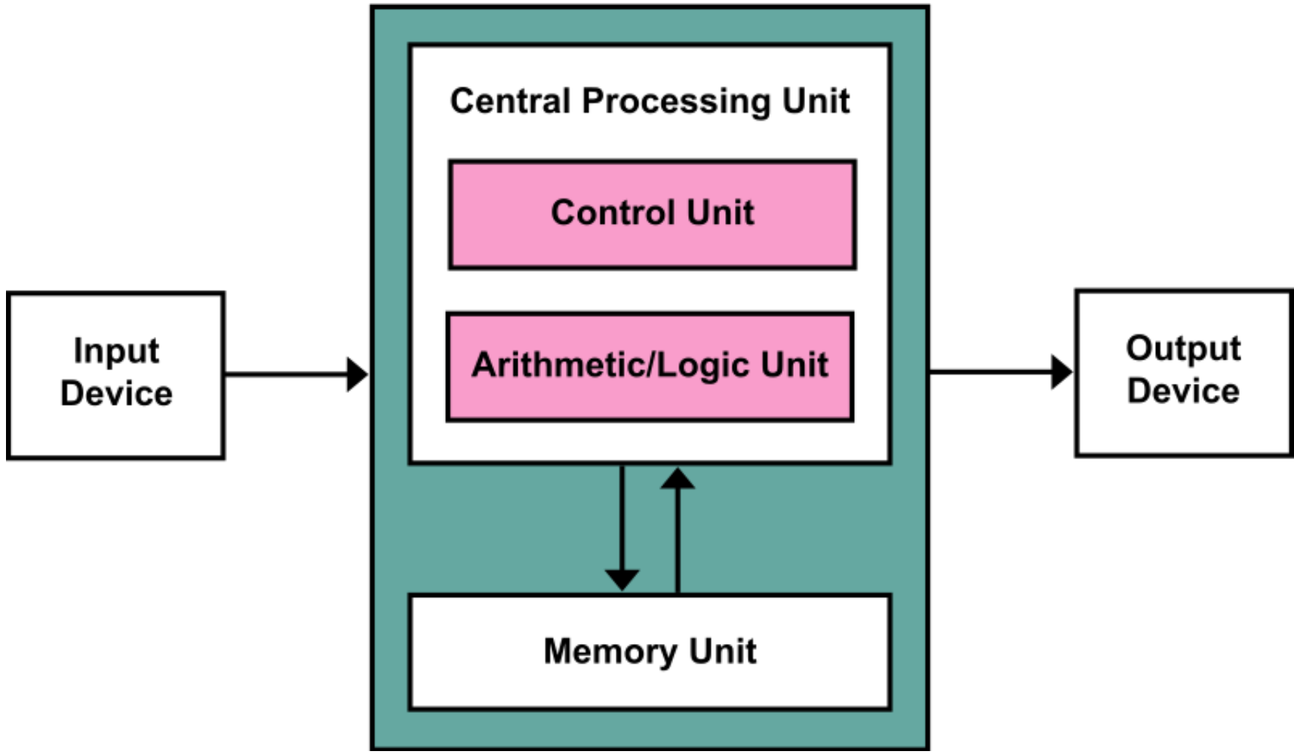
Input and Output Devices

Secondary Memory

Definitions

the von Neumann architecture

- **Central Processing Unit**: Runs the Program - The CPU is always wondering “what to do next”. Dumb, but very very fast
- **Input Devices**: Keyboard, Mouse, Touch Screen
- **Output Devices**: Screen, Speakers, Printer, DVD Burner
- **Main Memory**: Fast small temporary storage - lost on reboot - aka RAM



alternative computer architectures? the non von Neumann machine

A non von Neumann machine may thus be without the concept control flow (i.e. without registers that indicate the current execution point that has been reached in a program) and/or without the concept of a variable (i.e. without “named” storage locations in which a value may be stored and subsequently referenced or changed).

dataflow machine

A computer in which the primitive operations are triggered by the availability of inputs or operands.

Doesn't keep track of the program flow. where are we in the execution?

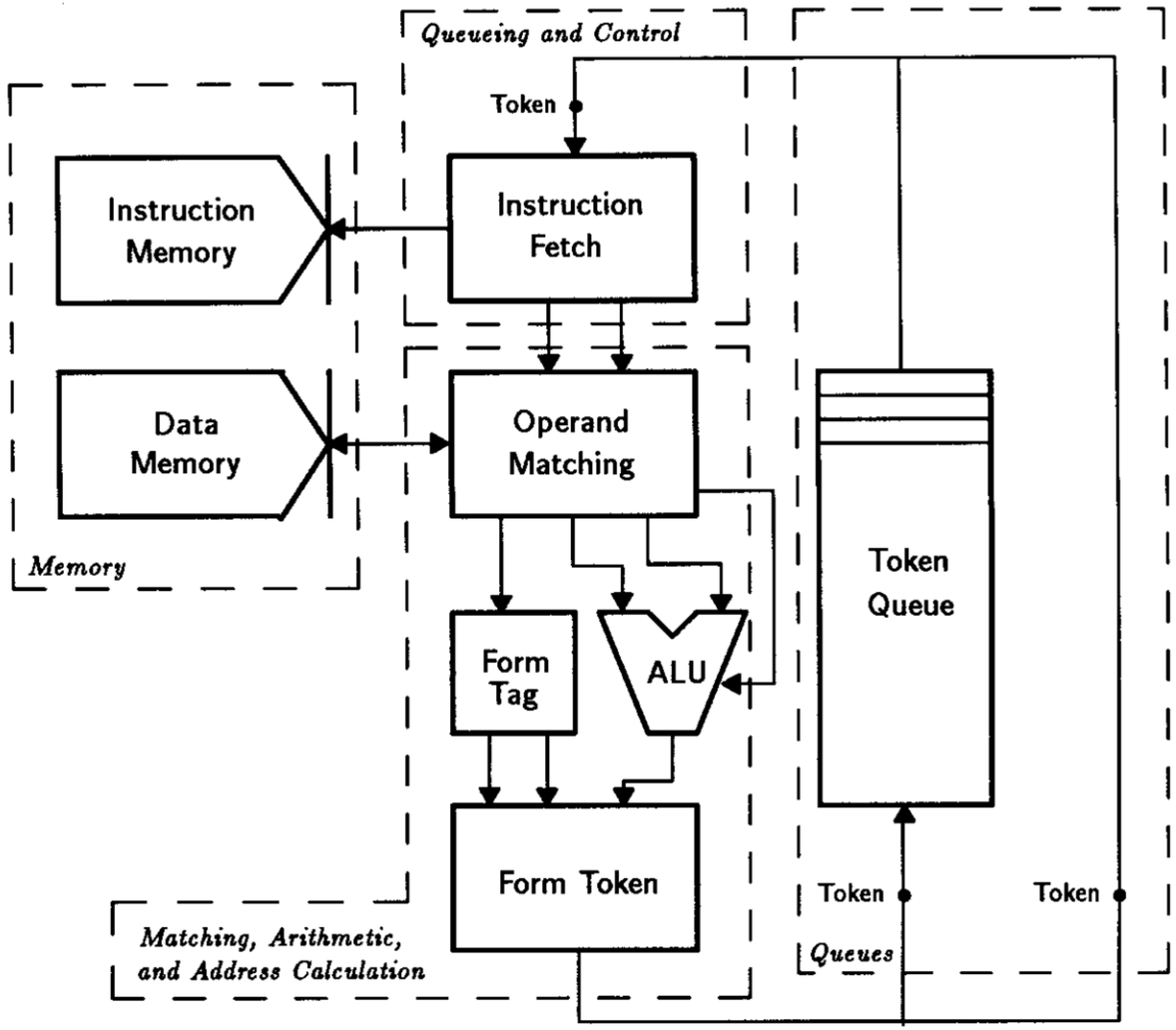


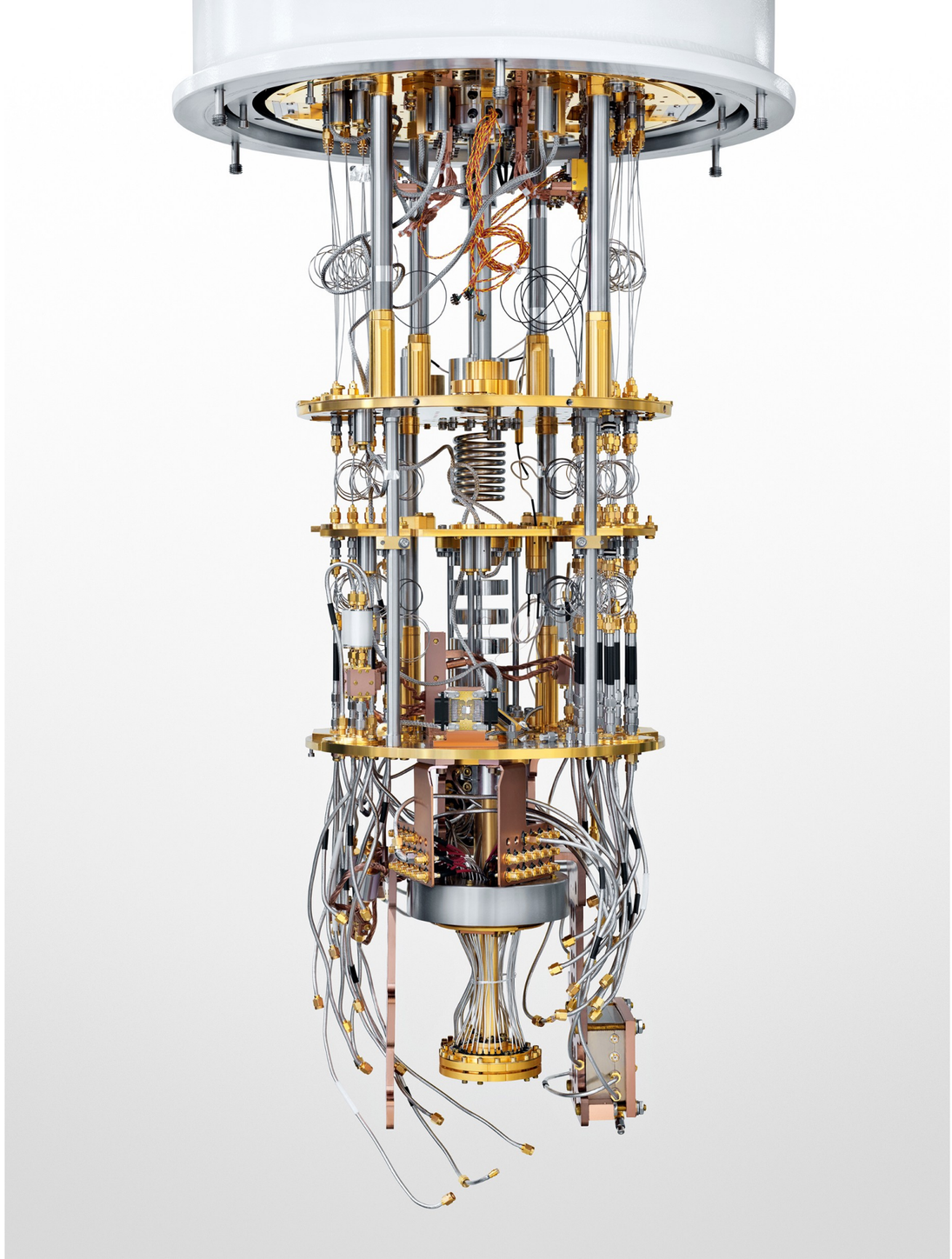
Figure 4.2: MINT Module Distribution

reduction machine

A machine that evaluates expressions by successively reducing all component subexpressions until only simple terms representing data values remain.

Quantum computing

Quantum Computing is the field that integrates quantum mechanical phenomena into a computing device.



Hard Disk in Action

http://www.youtube.com/watch?v=9eMWG3fwiEU

Python as a Language

Python is the language of the Python Interpreter and those who can converse with it. An individual who can speak Python is known as a Pythonista. Nearly all known Pythonistas use software initially developed by Guido van Rossum.

Early Learner: Syntax Errors

• We need to learn the Python language so we can communicate our instructions to Python. In the beginning we will make lots of mistakes and speak gibberish like small children.

• When you make a mistake, the computer does not think you are “cute”. It says “syntax error” - given that it knows the language and you are just learning it. It seems like Python is cruel and unfeeling.

• You must remember that you are intelligent and can learn. The computer is simple and very fast, but cannot learn. So it is easier for you to learn Python than for the computer to learn English...

Talking to Python

```
csev$ python3
 Python 3.5.1 (v3.5.1:37a07cee5969, Dec 5 2015, 21:12:44)
 [GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwinType
 "help", "copyright", "credits" or "license" for more information.
 >>>
```

What next?

```
csev$ python3
 Python 3.5.1 (v3.5.1:37a07cee5969, Dec 5 2015, 21:12:44)
 [GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwinType
 "help", "copyright", "credits" or "license" for more information.
 >>> x = 1
 >>> print(x)
 1
 >>> x = x + 1
 >>> print(x)
 2
 >>> exit()
```

This is a good test to make sure that you have Python correctly installed. Note that quit() also works to end the interactive session.

What Do We Say?

Elements of Python

- **Vocabulary / Words** Variables and Reserved words (Chapter 2)
- **Sentence structure** valid syntax patterns (Chapters 3-5)
- **Story structure** constructing a program for a purpose

A short “story” to count words in a file

```
name = input('Enter file:')
handle = open(name)
counts = dict()
for line in handle:
    words = line.split()
    for word in words:
        counts[word] = counts.get(word,0) + 1
bigcount = None
bigword = None
for word,count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count
print(bigword, bigcount)
```

```
python words.py Enter file: words.txt
```

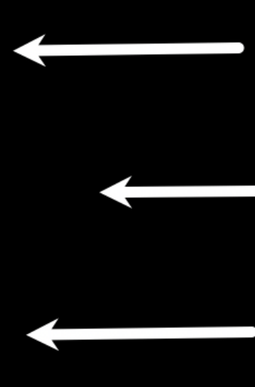
Reserved Words

You cannot use reserved words as variable names / identifiers

```
False   class   return  is      finally
None    if      for     lambda  continue
True    def     from    while   nonlocal
and     del     global  not     with
as      elif    try     or      yield
assert  else    import  pass
break   except  in      raise
```

Sentences or Lines

```
x = 2
x = x + 2
print(x)
```



Assignment statement

Assignment with expression

Print statement

Variable Operator Constant Function

Programming Paragraphs

Python Scripts

- Interactive Python is good for experiments and programs of 3-4 lines long.
- Most programs are much longer, so we type them into a file and tell Python to run the commands in the file.
- In a sense, we are *giving Python a script* to perform.
- As a convention, we add “.py” as the suffix on the end of these files to indicate they contain Python.

Interactive versus Script

- Interactive
  - You type directly to Python one line at a time and it responds

- Script
  - You enter a sequence of statements (lines) into a file using a text editor and tell Python to execute the statements in the file

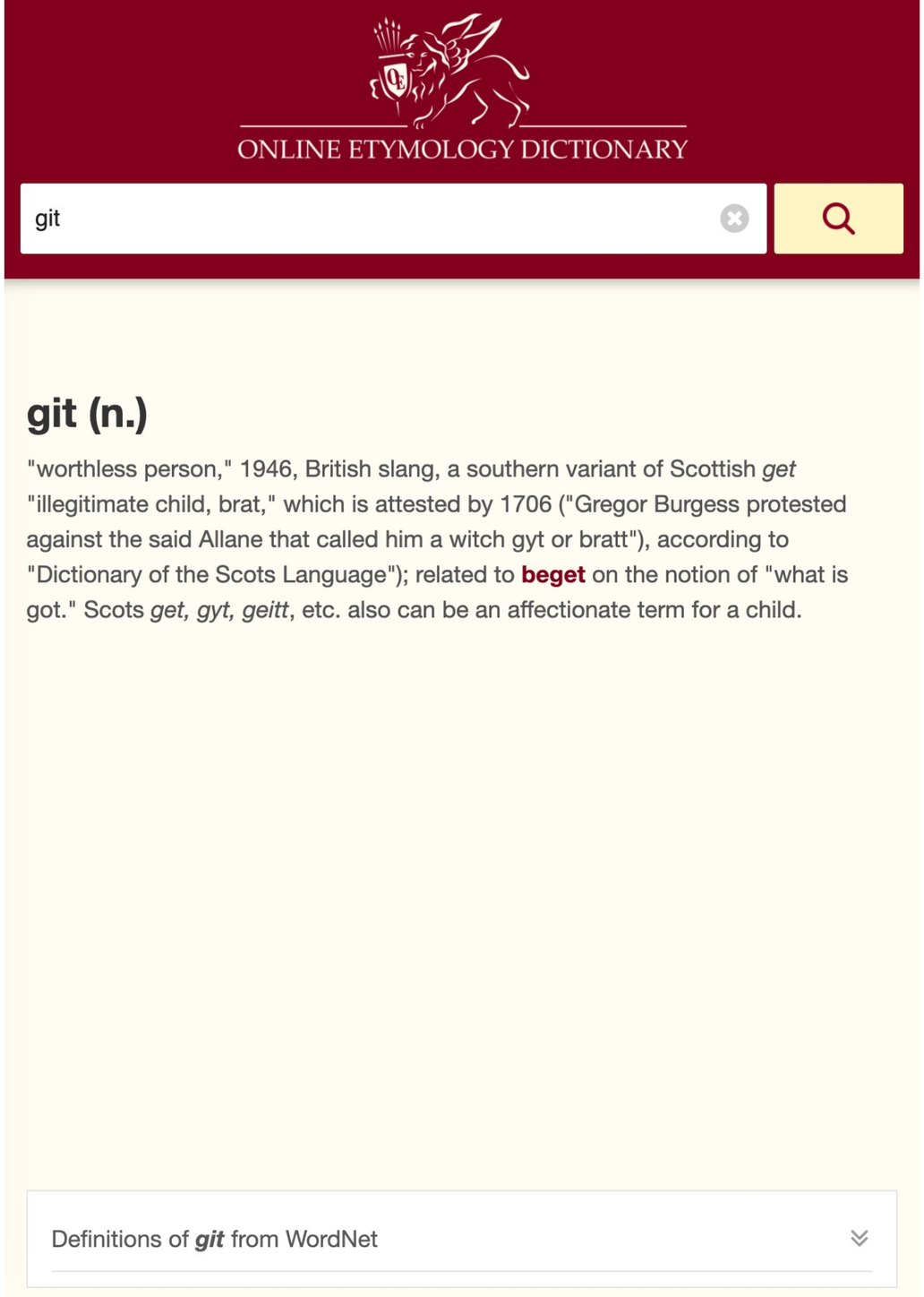
Program Steps or Program Flow

- Like a recipe or installation instructions, a program is a **sequence** of steps to be done in order.
- Some steps are **conditional** and may be skipped.
- Sometimes a step or group of steps is to be **repeated**.
- Sometimes we store a set of steps to be used over and over as needed several places throughout the program (week 4).

git

Git is an open source software management system developed by Linus Torvalds in 2005, the creator of Linux Kernel architecture that is used in the Linux operating system. It is used to track changes in any files, facilitating versioning control of variations in a distributed network. It is particularly useful for large-scale collaborative programming in which individuals work on different parts of the software with their own machine by copying (forking), splitting (branching), and combining (merging).

You will use git in this class to



Acknowledgements / Contributions

Initial Development: Charles Severance, University of Michigan School of Information

Blending with aesthetic and cultural materials: Douglas Goodwin, Scripps College

Continue...