ISM
Rik Goldman

Scripting, IDEs, Revision Control

Prior Learning

- ISM II/B created bitbucket accounts (chelsea credentials)
- ISM I collaborative script model initiated

Agenda

1. Agenda books
2. Anticipatory Set
3. Teach and Model
4. Check for Understanding
5. Closure

Standards

- See daily plans

Objectives

- Create an executable file with an authentic purpose
- Prompt user for input (bash)
- Prompt user for hidden input (password)
- Hold user input as variable
- Write variable to file
- Change file mode (access rights)
- Create rev control repository
- Add file to repository
- Commit and comment on changes to repository
- Sync repository
- Work with team repository
- Craft a *for* or *while* loop
- Differentiate open source licenses

Anticipatory Sets

- Review: shell script + cron = automation
- Research contract (~~Alex~~ will differ)
- Input
- Output
- Control flow
- Pseudo-code: coding regardless of language
- Why Revision Control: Git and Subversion
- Discussion in the round: why not use rev control system for composition and research?

Model and Teach

- Flow Chart Conventions
- GIT and either github or bitbucket repositories
- Install, create, commit, push with GIT
- git clone, git pull
- markdown (nominate a teacher)
- Choosing among licenses for your creation; git add LICENSE
- Crafting helpful documentation: git add README.md
- Collaborate on purposeful script
- Contract: Forked and Modified Code

Check for Understanding

- Collaboratively plan technology development using flowchart conventions (white board and dia)
- Complete and submit research (contrast model): Shell script languages (ISM II)
- Initiate, commit, add forked code to git repository
- Purposefully modify existing open code from git repo (provided)
- Choose purposefully from among open source licenses (include in repo)
- Demonstrate working code; present rationale and control flow to class